

# FUJITSU Software

## Symfoware Server V12.1.0

A decorative horizontal band with a red-to-dark-red gradient, featuring abstract, glowing white and red lines that swirl and intersect, creating a sense of motion and technology.

# Cluster Operation Guide

Linux

J2UL-1737-05ENZ0(00)  
November 2014

# Preface

---

## Purpose of This Document

This document explains the items required to operate Symfoware Server on a cluster system.

## Intended Readers

This document is aimed at people who install and operate Symfoware Server on a cluster system. Readers of this document are also assumed to have general knowledge of:

- PRIMECLUSTER
- PostgreSQL
- SQL
- Linux

## Structure of This Document

This document is structured as follows:

### Part 1 Overview of Cluster Systems

#### Chapter 1 Definition of Cluster Operation

Provides an overview of cluster operations and their types.

### Part 2 Shared Disk Type Failover Mode

#### Chapter 2 Overview of Failover Operation

Provides an overview of failover operation.

#### Chapter 3 Failover Operation

Explains the procedures involved in failover operation.

#### Chapter 4 Procedures Required after a Failover Error

Explains the procedures required after an abnormality occurs during failover operation.

### Part 3 Database Multiplexing Mode

#### Chapter 5 Overview of Database Multiplexing Mode

Provides an overview of database multiplexing mode.

#### Chapter 6 Setting up Database Multiplexing Mode

Describes how to set up database multiplexing mode.

#### Chapter 7 Operations in Database Multiplexing Mode

Explains periodic database multiplexing mode.

#### Chapter 8 Action Required when an Error Occurs in Database Multiplexing Mode

Explains the action required when an error occurs during a database multiplexing mode.

### Appendix A Creating Resources and Creating/Modifying Cluster Applications

Explains how to create and modify cluster applications that include Symfoware resources.

### Appendix B Parameters

Explains the configuration files and parameters required for database multiplexing mode.

### Appendix C Supplementary Information on Building the Primary Server and Standby Server on the Same Server

Explains supplementary information on building the primary server and standby server on the same server.

## **Export Restrictions**

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.

## **Issue Date and Version**

Third Edition : November 2014
Second Edition: January 2014
First Edition : November 2013

## **Copyright**

Copyright 2013-2014 FUJITSU LIMITED

# Contents

---

Part 1 Overview of Cluster Systems.....	1
Chapter 1 Definition of Cluster Operation.....	2
1.1 What Is a Cluster System?.....	2
1.2 Cluster Operation Provided by Symfoware Server .....	2
Part 2 Shared Disk Type Failover Mode.....	4
Chapter 2 Overview of Failover Operation.....	5
2.1 Definition of Failover Operation.....	5
2.2 System Configuration of Failover Operation.....	6
2.3 Setup of Failover Operation.....	8
2.3.1 Installing PRIMECLUSTER and Symfoware Server.....	9
2.3.2 Configuring PRIMECLUSTER.....	9
2.3.3 Creating a GDS Volume.....	10
2.3.4 Creating a File System.....	10
2.3.4.1 Creating a File System.....	10
2.3.4.2 Editing Configuration Files.....	10
2.3.4.3 Checking.....	11
2.3.5 Creating an Operating System User to Start Symfoware Server.....	11
2.3.6 Mounting the File System.....	11
2.3.7 Creating a Symfoware Database Cluster.....	12
2.3.8 Registering Resource Information for the Symfoware Database Cluster.....	13
2.3.9 Configuring Storage Data Protection Using Transparent Data Encryption.....	14
2.3.10 Configuring the PL/extJava Environment Settings.....	15
2.3.11 Unmounting the File System.....	15
2.3.12 Creating a Cluster Application.....	15
2.3.13 Checking Operation.....	15
Chapter 3 Failover Operation.....	16
3.1 Adding Tablespaces.....	16
3.2 Modifying Database Cluster Resources.....	16
3.3 Changing the PL/extJava Environment.....	17
3.4 Operations under Normal Circumstances.....	17
3.5 Operation at State Transition.....	18
3.6 Maintenance Tasks.....	18
3.6.1 Simultaneous Stopped Node Maintenance.....	19
3.6.1.1 Stopping the Cluster Applications on Both Nodes for Node Maintenance.....	19
3.6.1.2 Maintenance Tasks on Both Nodes.....	20
3.6.1.3 Restarting the Cluster Applications Stopped on Both Nodes for Node Maintenance.....	20
3.6.2 Mutual Switch Maintenance.....	20
3.6.2.1 Stopping the Cluster Applications on the Standby Node for Node Maintenance.....	20
3.6.2.2 Standby Node Maintenance.....	21
3.6.2.3 Restarting the Cluster Applications Stopped on the Standby Node for Node Maintenance.....	21
3.6.2.4 Mutual Switch.....	21
3.6.2.5 Standby Node Maintenance after the Mutual Switch.....	21
Chapter 4 Procedures Required after a Failover Error.....	22
4.1 Identifying the Cause of an Error.....	22
4.2 Stopping the Cluster Applications for Node Maintenance.....	23
4.3 Starting the Database Cluster.....	23
4.4 Recovery Tasks.....	23
4.5 Stopping the Database Cluster.....	23
4.6 Restarting the Cluster Applications Stopped for Node Maintenance.....	23
Part 3 Database Multiplexing Mode.....	24

Chapter 5 Overview of Database Multiplexing Mode.....	25
5.1 What is Database Multiplexing Mode.....	25
5.2 System Configuration for Database Multiplexing Mode.....	27
5.2.1 Mirroring Controller Resources.....	28
5.2.2 Redundancy of the Admin and Log Transfer Networks.....	29
5.2.3 Notes on CPU Architecture and Products.....	29
5.2.4 Notes on Executing Reference Jobs on the Standby Server (when the Mirroring Controller Option is Used) .....	29
5.3 Security in Database Multiplexing.....	30
5.3.1 Authentication of the Standby Server.....	31
5.3.2 Encryption of Transaction Logs Transferred to the Standby Server.....	32
Chapter 6 Setting up Database Multiplexing Mode.....	33
6.1 Installation.....	34
6.2 Preparing the Backup Disk.....	34
6.3 Setting up the Primary Server.....	34
6.3.1 Setting up Database Multiplexing Mode on the Primary Server.....	34
6.3.2 Creating, Setting, and Registering the Primary Server Instance.....	36
6.4 Setting up the Standby Server.....	40
6.4.1 Setting up Database Multiplexing Mode on the Standby Server.....	40
6.4.2 Creating, Setting, and Registering the Standby Server Instance.....	41
6.5 Confirming the Streaming Replication Status.....	42
6.6 Setting up the PL/extJava Environment.....	43
6.7 Creating Applications.....	43
6.7.1 Application Connection Server Settings.....	44
6.7.2 Notes on Indexes.....	44
6.8 Checking the Behavior.....	44
6.9 Tuning.....	44
6.9.1 Tuning to Stabilize the Database Multiplexing Mode.....	44
6.9.2 Tuning to Stabilize Queries on the Standby Server.....	44
6.9.3 Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server).....	45
6.10 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances.....	45
6.11 Backup Operation.....	47
6.11.1 Backing up Database Multiplexing Mode Information.....	47
6.11.2 Database Backup Operation.....	47
Chapter 7 Operations in Database Multiplexing Mode.....	49
7.1 Starting and Stopping Instances.....	49
7.2 Checking the Database Multiplexing Mode Status.....	49
7.3 Manually Switching the Primary Server.....	50
7.4 Manually Disconnecting the Standby Server.....	50
7.5 Monitoring Mirroring Controller Messages.....	51
7.6 Server Maintenance.....	52
7.6.1 Rolling Updates.....	52
7.6.2 Stopping for Maintenance .....	57
7.7 Changes in Operation .....	57
7.7.1 Changes Required when the Standby Server is Stopped.....	58
7.7.2 Changing from Single Server Mode to Database Multiplexing Mode.....	58
7.7.3 Changing from Database Multiplexing Mode to Single Server Mode.....	59
Chapter 8 Action Required when an Error Occurs in Database Multiplexing Mode.....	62
8.1 Action Required when Server Degradation Occurs.....	62
8.1.1 Operations when the Server has Started Degrading after a Switch has Occurred.....	62
8.1.1.1 Identify Cause of Error and Restructure the Standby Server.....	64
8.1.1.1.1 Stop Mirroring Controller.....	64
8.1.1.1.2 Recovery of the Mirroring Controller management directory.....	64
8.1.1.1.3 Identify cause of error and perform recovery.....	64
8.1.1.2 Rebuild the Standby Server.....	64
8.1.1.3 Failback of the Primary Server.....	65

8.1.2 Operations when the Server has Started Degrading after a Disconnection has Occurred.....	66
8.1.2.1 Identify Cause of Error and Restructure the Standby Server.....	67
8.1.2.1.1 Stop Mirroring Controller.....	67
8.1.2.1.2 Recovery of the Mirroring Controller management directory.....	68
8.1.2.1.3 Identify cause of error and perform recovery.....	68
8.1.2.2 Rebuild the Standby Server.....	68
8.2 Action Required when Automatic Switch Fails (when the Mirroring Controller Option is Used).....	68
8.3 Action Required when Automatic Disconnection Fails(when the Mirroring Controller Option is Used).....	69
8.4 Action Required when All Servers or Instances Stopped.....	70
Appendix A Creating Resources and Creating/Modifying Cluster Applications.....	75
A.1 Creating Resources and Cluster Applications.....	75
A.1.1 Preparing for Creation.....	75
A.1.2 Creating a Cluster Application and Configuring Its Attributes.....	75
A.1.3 Creating GDS Resources.....	82
A.1.4 Creating Symfoware Resources.....	83
A.1.5 Creating GIs or Takeover Network Resources.....	83
A.1.6 Creating Fsystem Resources.....	84
A.1.7 Creating Resources and Finalizing Cluster Application Creation.....	86
A.2 Modifying Cluster Applications.....	87
A.2.1 Moving to the Cluster Application Edit Window.....	87
A.2.2 Adding a Created Shared Disk Class to the Cluster Application.....	88
A.2.3 Adding a Created File System to the Cluster Application.....	89
A.2.4 Reflecting Modifications Made to the Cluster Application.....	89
Appendix B Parameters.....	91
B.1 Parameters Set on the Primary Server.....	91
B.2 Parameters Set on the Standby Server.....	93
B.3 Network Configuration File.....	93
B.4 Server Configuration File.....	94
Appendix C Supplementary Information on Building the Primary Server and Standby Server on the Same Server.....	96
C.1 Features that Cannot be Used in this Configuration.....	96
C.2 Backup Data Storage Destination Directory.....	96
C.3 How to Execute the mc_ctl Command.....	96
Index.....	98

# Part 1 Overview of Cluster Systems

---

---

This part provides an overview of the cluster systems, and describes the types and operation modes.

<a href="#">Chapter 1 Definition of Cluster Operation.....</a>	<a href="#">2</a>
--	-------------------

# Chapter 1 Definition of Cluster Operation

This chapter explains the features of a cluster system, the types of cluster operations, and the operation mode.

## 1.1 What Is a Cluster System?

A cluster system is a system that joins multiple machines (hereafter, referred to as servers) to achieve high availability and scalability throughout the system.

This allows the creation of systems that simultaneously achieve high availability and scalability.

### High availability

For example, by transferring the role to another server when a server in the cluster system fails, the downtime for the entire system can be made shorter.

This type of operation mode is referred to as a "failover operation".

### Scalability

The load should be distributed, in order to extend the processing capacity across the entire system by dividing multiple application processes amongst multiple servers, regardless of processing content, or to suit the processing content.

## 1.2 Cluster Operation Provided by Symfoware Server

In Symfoware Server, the following failover operation modes are provided in each system:

- Shared disk mode
- Log shipping mode

### Shared disk mode

This operation mode requires integration with the PRIMECLUSTER cluster software, to use the shared disk (GDS) it provides. A server in the cluster system receives the application processing, acting as the active server (active node).

For example, if the active node fails, another server (standby node) inherits the shared disk content and becomes the active node. Using the PRIMECLUSTER feature, the applications to be run on the active node for the database can also be linked with the database and switched. There is also no need to send the updated content of the database on the active node to the standby node, so the processing performance will be the same as of a non-cluster system.



### Note

When integrating PRIMECLUSTER with Symfoware Server, the disk is not referenced by the standby node, and therefore the shared disk mode cannot be used for load distribution.

### Log shipping mode

This operation mode is based on PostgreSQL streaming replication. Other software such as cluster software is not required.

This mode replicates the database on all servers that comprise the cluster system. It achieves this by sending and reflecting the updated transaction log of the database from the server that receives the update (primary server) to another server (standby server).

In addition, the client driver automatically distinguishes between the primary and standby servers, so applications can be connected transparently regardless of the physical server. Accordingly, a flexible system can easily be built, for example a system that offloads connections to the standby server if the connections only perform referencing.

## Note

Symfoware Server Enterprise Edition and Standard Edition support only configurations comprising one standby server per primary server.

Log shipping mode has the following three features:

- Streaming replication feature

The PostgreSQL streaming replication feature can be used as is.

The database will be copied in asynchronous mode, that is, the transaction log will be asynchronously sent to the standby server. This will result in performance equivalent to that of a non-cluster system.

In this case, note that, strictly speaking, a slight delay will occur in the copying of the database of the standby server.

- Database multiplexing feature

Consists of a feature that detects faults in elements that are essential for the continuity of the database operation (such as the database process, disk, and network), as well as a switchover feature to manually switch the primary server and a disconnection feature for the standby server. Note that SQL statements cannot be run on the standby server.

The database will be copied in synchronous mode.

- Mirroring Controller option

This option provides failover functionality to automatically switch and disconnect the primary server when the database multiplexing feature is used. Reference type SQL statements can also be run on the standby server.

## See

- Refer to "[Part 2 Shared Disk Type Failover Mode](#)" for information on the shared disk mode.

- The streaming replication feature is not described in this manual.

Refer to "High Availability, Load Balancing, and Replication" in the PostgreSQL Documentation for information on the streaming replication feature.

- Refer to "[Part 3 Database Multiplexing Mode](#)" for information on the database multiplexing feature.

- Also refer to "[Part 3 Database Multiplexing Mode](#)" when using the Mirroring Controller option.

Additionally, refer to "When the Mirroring Controller option is used" in the various chapters and sections for the description and procedure required specifically for the Mirroring Controller option.

# Part 2 Shared Disk Type Failover Mode

---

---

This part describes the shared disk type failover mode.

<a href="#">Chapter 2 Overview of Failover Operation.....</a>	5
<a href="#">Chapter 3 Failover Operation.....</a>	16
<a href="#">Chapter 4 Procedures Required after a Failover Error.....</a>	22

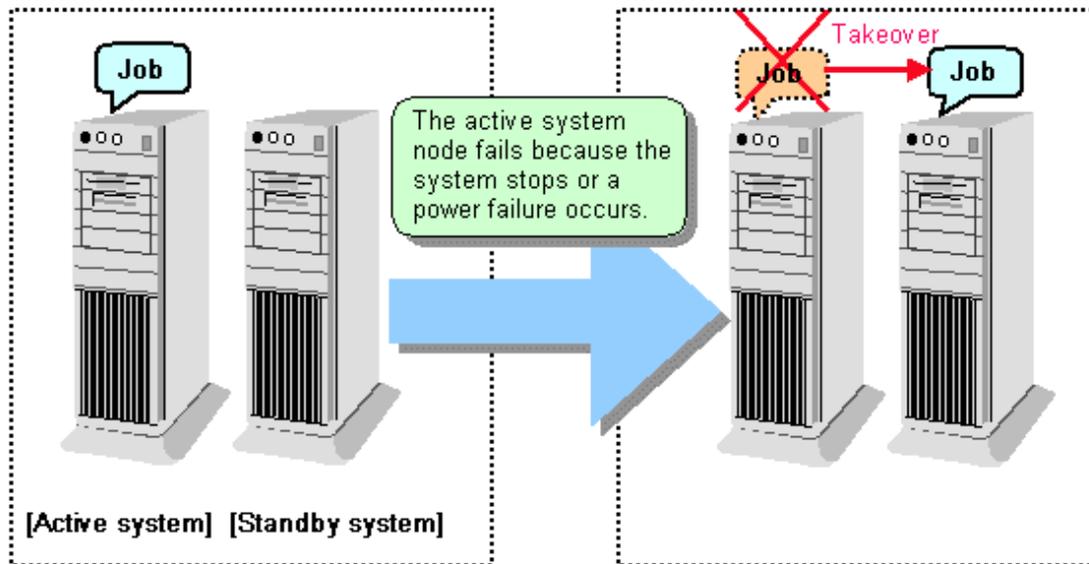
# Chapter 2 Overview of Failover Operation

This chapter provides an overview of failover operation in Symfoware Server.

## 2.1 Definition of Failover Operation

When an abnormality occurs on any of the multiple server devices that make up the cluster system (hereafter referred to as "nodes"), the failover feature transfers a job operating on such a node to another one. Failover can reduce the time for which jobs are stopped when an abnormality occurs, and allows jobs to continue while the node on which the abnormality occurred is recovered.

Figure 2.1 Concept of failover



### Feature of failover operation

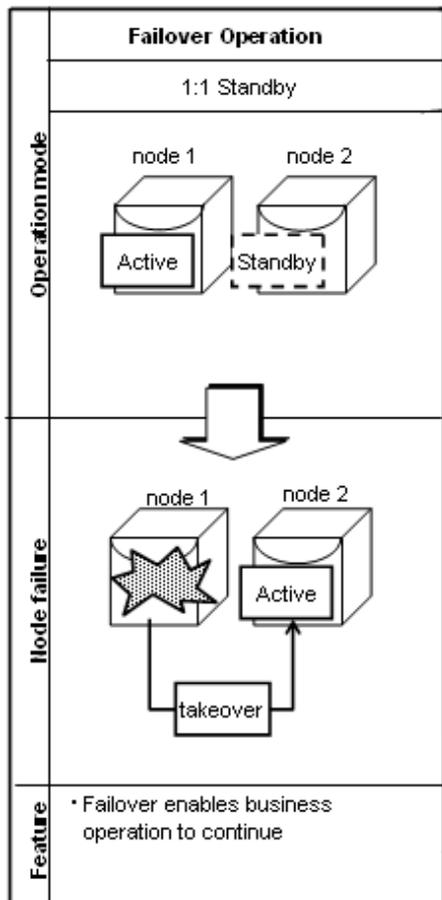
Failover operation in Symfoware Server supports the standby feature.

When an abnormality occurs on a node, the standby feature activates the instance that is the standby system at the time of the switch, and transfers the job to it.

### Operation mode of failover operation

The operation mode supported in failover operation in Symfoware Server is 1:1 standby.

1:1 standby is a mode in which one active system or standby system operates on one node. Because failover allows jobs to continue when an abnormality occurs in the active system, this mode allows highly reliable systems to be constructed.



## 2.2 System Configuration of Failover Operation

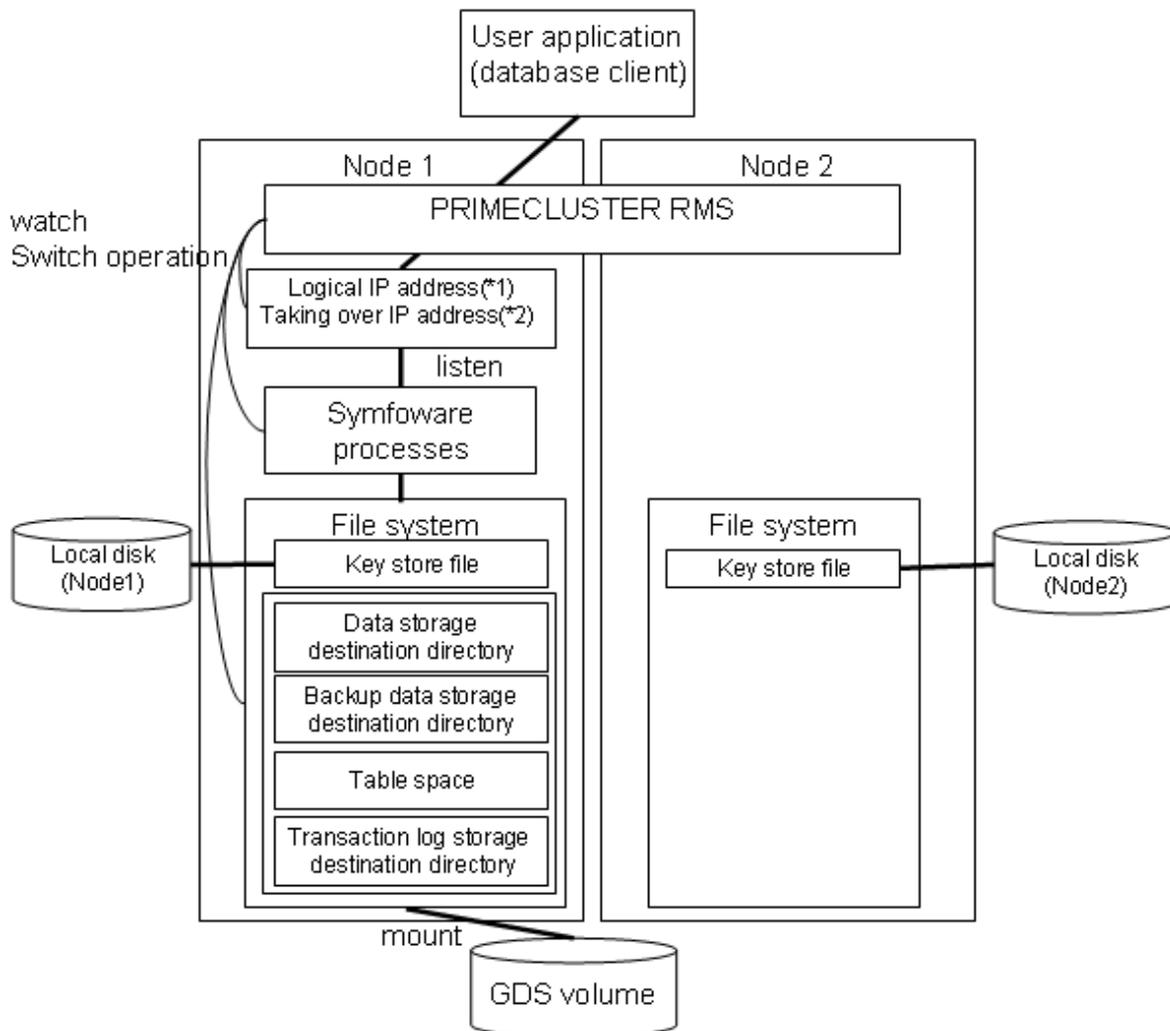
---

In the illustration below, the file system is on a GDS volume (shared disk), mounted on node 1 only.

The following Symfoware Server resources are located in this file system:

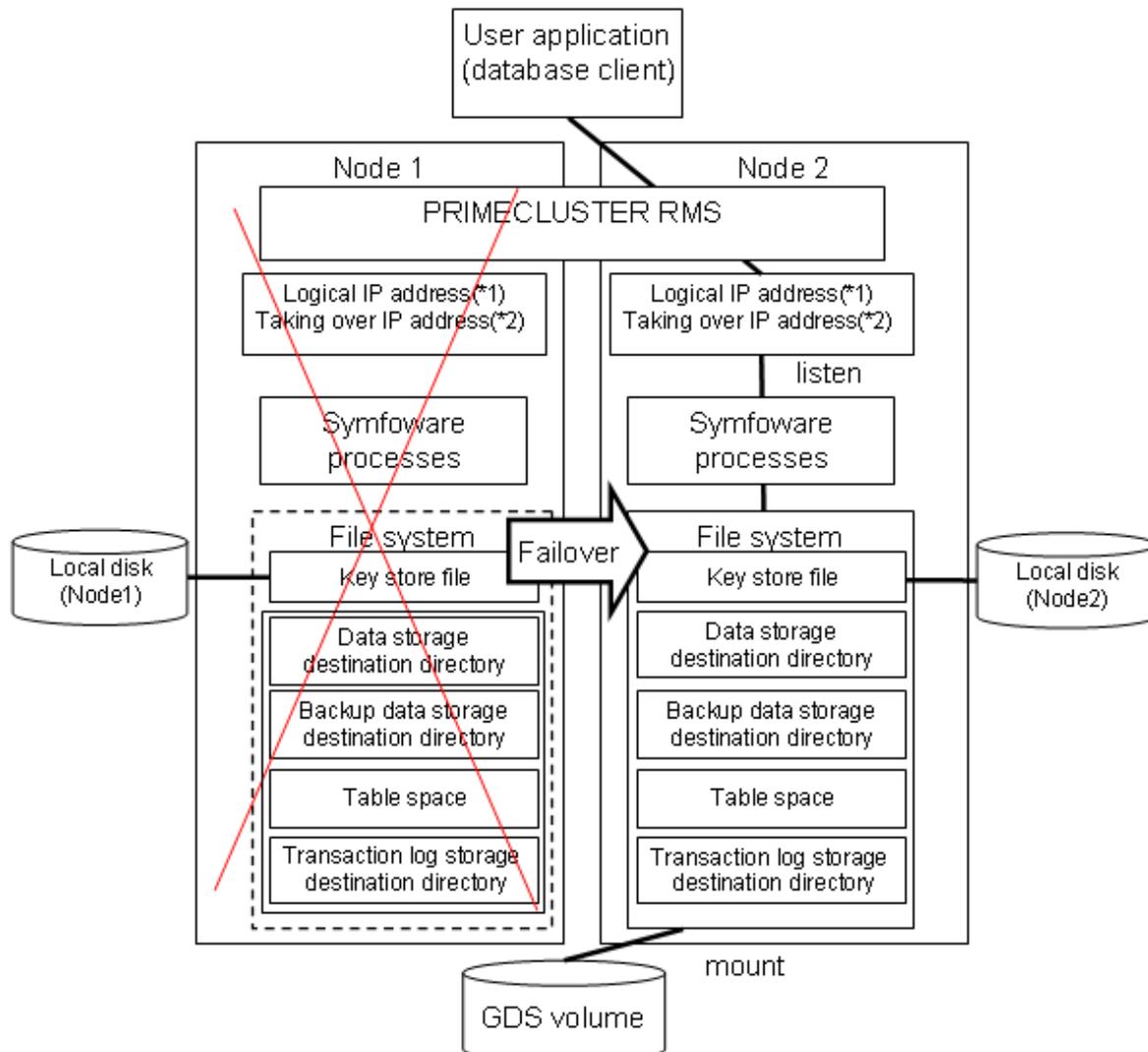
- Data storage destination directory
- Tablespaces
- Backup data storage destination directory
- Transaction log storage destination directory

A Symfoware Server process waits for connections of user applications that access the database, on a logical IP address activated on node 1 only or on a takeover IP address.



- \*1) The function of PRIMECLUSTER GLS(Global Link Services)
- \*2) The function of PRIMECLUSTER Taking over network resource

When an error is detected on node 1, PRIMECLUSTER mounts the file system on node 2, and activates the logical IP address or the takeover IP address on it. The Symfoware Server process is also started on node 2. Since the logical IP address or the takeover IP address remains unchanged, the user application can access Symfoware Server process on node 2 simply by re-establishing the database connection.



\*1) The function of PRIMECLUSTER GLS(Global Link Services)  
 \*2) The function of PRIMECLUSTER Taking over network resource

## 2.3 Setup of Failover Operation

Use the procedure in the table below to perform setup:

Step	Work item		Reference
	Active system	Standby system	
1	Installing PRIMECLUSTER and Symfoware Server		<a href="#">2.3.1 Installing PRIMECLUSTER and Symfoware Server</a>
2	Configuring PRIMECLUSTER		<a href="#">2.3.2 Configuring PRIMECLUSTER</a>
3	Creating a GDS volume (*1)		<a href="#">2.3.3 Creating a GDS Volume</a>
4	Creating a file system (*1)		<a href="#">2.3.4 Creating a File System</a>
5	Creating a user for the operating system to start Symfoware Server		<a href="#">2.3.5 Creating an Operating System</a>

Step	Work item		Reference
	Active system	Standby system	
			User to Start Symfoware Server
6	Mounting the file system		2.3.6 Mounting the File System
7	Creating a Symfoware database cluster(*1)		2.3.7 Creating a Symfoware Database Cluster
8	Registering resource information for the Symfoware database cluster		2.3.8 Registering Resource Information for the Symfoware Database Cluster
9	Configuring storage data protection using transparent data encryption (*1)		2.3.9 Configuring Storage Data Protection Using Transparent Data Encryption
10	Configuring the PL/extJava environment settings		2.3.10 Configuring the PL/extJava Environment Settings
11	Unmounting the file system		2.3.11 Unmounting the File System
12	Creating a cluster application		2.3.12 Creating a Cluster Application
13	Checking operation		2.3.13 Checking Operation

\*1: Some settings and operations must also be performed on the standby node. Refer to the relevant sections for details.

The sections that follow explain each step.

## 2.3.1 Installing PRIMECLUSTER and Symfoware Server

Refer to the manual for each product, and then install each product.



See

- Refer to the PRIMECLUSTER Installation and Administration Guide for information on how to install PRIMECLUSTER.
- Refer to the "Installation and Setup Guide for Server" for information on how to install Symfoware Server.

## 2.3.2 Configuring PRIMECLUSTER

Configure the initial settings for the cluster.

If using GLS, configure the GLS settings.



See

Refer to the PRIMECLUSTER Installation and Administration Guide for information on steps required before setup.

## 2.3.3 Creating a GDS Volume

---

Each of the following Symfaware Server resources must be located in a file system on a shared disk:

- Data storage destination directory
- Tablespaces
- Backup data storage destination directory
- Transaction log storage destination directory

### Note

- Store data storage destination directories or tablespaces and backup data storage destination directories for these in disk classes with different GDS to guard against file system corruption and the MONITORONLY attribute settings at the time of the GDS resource and Fsystem resource creation.
- It is also recommended that database storage destination directories and transaction log storage destination directories be placed in different disk classes, so that the I/O load is distributed.
- Create a directory for each resource on each node. Ensure that the directories to be mounted have the same path on all nodes.
- The monitoring disk area must be secured if using Fsystem resources. Ensure that the GDS volume to be used as the monitoring disk area has been secured when creating the GDS volume.

### See

- Refer to the PRIMECLUSTER Installation and Administration Guide for information on GDS volume creation.
- Refer to "Preparing Directories to Deploy Resources" in the "Installation and Setup Guide for Server" for information on how to create directories.
- Refer to "Setting Up Fsystem Resources" in the PRIMECLUSTER Installation and Administration Guide for information on configuring the monitoring disk area.

## 2.3.4 Creating a File System

---

Create a file system.

### See

- Refer to the PRIMECLUSTER Installation and Administration Guide for information on how to create a file system on a GDS volume.

### 2.3.4.1 Creating a File System

Select a file system configuration from the configuration menu in the [Global Disk Services] window in the PRIMECLUSTER Web-Based Admin View, and then create a file system.

### 2.3.4.2 Editing Configuration Files

Edit /etc/fstab and /etc/fstab.pcl on each node.

- /etc/fstab
- Add the "ro" attribute to the entries for the file system for monitoring, as follows:

```
# cat /etc/fstab
/dev/sfdsk/class0001/dsk/volume0001 /mnt/swdsk1 ext3 noauto 0 0
/dev/sfdsk/class0001/dsk/volume0002 /mnt/monitor1 ext3 noauto,ro 0 0
```

```
/dev/sfdsk/class0001/dsk/volume0003 /mnt/swdsk2 ext3 noauto 0 0
/dev/sfdsk/class0001/dsk/volume0004 /mnt/monitor2 ext3 noauto,ro 0 0
```

- /etc/fstab.pcl

Copy the entries shown above, and then prefix each line with #RMS#, as follows:

```
# cat /etc/fstab.pcl
#RMS#/dev/sfdsk/class0001/dsk/volume0001 /mnt/swdsk1 ext3 noauto 0 0
#RMS#/dev/sfdsk/class0001/dsk/volume0002 /mnt/monitor1 ext3 noauto,ro 0 0
#RMS#/dev/sfdsk/class0001/dsk/volume0003 /mnt/swdsk2 ext3 noauto 0 0
#RMS#/dev/sfdsk/class0001/dsk/volume0004 /mnt/monitor2 ext3 noauto,ro 0 0
```



## Note

Different files are specified depending on the version of the PRIMECLUSTER. Check the version of PRIMECLUSTER being used and edit appropriate files. Refer to the PRIMECLUSTER Installation and Administration Guide for details.

### 2.3.4.3 Checking

Start the GDS volume, and ensure that each file system can be mounted on each node.

```
# mount -t ext3 /dev/sfdsk/class0001/dsk/volume0001 /mnt/swdsk1
# mount -t ext3 /dev/sfdsk/class0001/dsk/volume0002 /mnt/monitor1
# mount -t ext3 /dev/sfdsk/class0001/dsk/volume0003 /mnt/swdsk2
# mount -t ext3 /dev/sfdsk/class0001/dsk/volume0004 /mnt/monitor2
# umount /mnt/swdsk1
# umount /mnt/monitor1
# umount /mnt/swdsk2
# umount /mnt/monitor2
```

### 2.3.5 Creating an Operating System User to Start Symfoware Server

Create an operating system user (an OS user that will become the instance administrator) who is to start Symfoware Server.



## Note

Ensure that the name, UID, and GID of the operating system user (an OS user that will become the instance administrator) who is to start Symfoware Server match on all nodes that make up the cluster system.

### 2.3.6 Mounting the File System

1. Start the GDS volume if it has not already started.
2. Mount all file systems containing the resources required to create database clusters, such as for the data storage destination directory, as shown below:

```
# mount -t ext3 /dev/sfdsk/class0001/dsk/volume0001 /mnt/swdsk1
```



## Note

Create directories in the file system for storing each resource. Each directory must meet certain conditions to be able to create the Symfoware database cluster. Check the notes in "Preparing Directories to Deploy Resources" in the Installation and Setup Guide for Server.

## 2.3.7 Creating a Symfoware Database Cluster

---

1. Create the database cluster.

### Note

---

- If the configuration requires password authentication, create a `.pgpass` file in the home directory of the operating system user who is to start Symfoware Server, and then specify a password for the `template1` database as this is required. This will be used when `PRIMECLUSTER` checks the operation of the Symfoware database server. If authentication fails, a fault will be assumed to have occurred in the database server.
  - The value specified for the `port` parameter in `postgresql.conf` needs to be set in the `/etc/services` files on all nodes.
  - Since `postgresql.conf` is stored in the data directory, it becomes a common configuration file for each node in the shared disk. Therefore, it must be taken into account that the path set for `postgresql.conf` must be valid for the Symfoware Server processes started on each node. For example, if a local directory is specified in the `core_directory` parameter, that local directory must have been created on all nodes.
  - To execute the Symfoware Server commands, you must configure the environment variables on each node.
  - Use the `initdb` command to create a database cluster.
  - Specify "off" for the `restart_after_crash` parameter in `postgresql.conf`.  
If "on" is specified, or the default value is used for the `restart_after_crash` parameter, behaviors equivalent to restarting the Symfoware Server, including crash recovery, will be performed when some server processes go down. When linking with `PRIMECLUSTER`, however, a failover will occur after an error is detected as soon as some server processes go down, and server processes restarting will be forced to stop. This is to inhibit complex behaviors such as processes being canceled in the middle of restarting for no particular meaningful purpose.
- 

2. Check starting, connection, and stopping at each node. The procedure is as follows.

### Note

---

Queries are issued to monitor the Symfoware Server from `PRIMECLUSTER`. With this connection process, if a password is requested, the monitoring process cannot be continued. For this reason, when checking the connection, ensure that connections can be made without entering a password.

---

1. Check the starting, connection, and stopping at node 1.

Use the `pg_ctl` command to start and stop the node.

For connection, execute the following to ensure that connections can be made without entering a password:

```
su - OsUser  
psql -d template1 -p portNum [ -U dbUser]
```

*OsUser*: Operating system user to start the Symfoware Server

*portNum*: Port number specified when the database cluster was created

*dbUser*: Database superuser

If the database superuser is specified at the time of executing the `initdb` command, or if the database superuser has been changed after the `initdb` command was executed, specify the user for the `-U` option. If the database superuser is not specified or changed, omit this.

2. Unmount all file systems containing the resources required to create database clusters (such as the data storage destination directory) on node 1, and stop the GDS volume.
3. Start the GDS volume on node 2, and mount all file systems containing the resources required to create database clusters (such as the data storage destination directory).
4. Check starting, connection, and stopping at node 2.

Perform the same check as step 1.

5. Unmount all file systems containing the resources required to create database clusters (such as the data storage destination directory) on node 2, and stop the GDS volume.
6. Start the GDS volume on node 1, and mount all file systems containing the resources required to create database clusters (such as the data storage destination directory).



See

- Refer to "Using the initdb Command" of "Creating an Instance" in the "Installation and Setup Guide for Server" for information on how to create a database cluster.
- Refer to "The Password File" in "Client Interfaces" in the PostgreSQL Documentation for information on files for which passwords are set.

## 2.3.8 Registering Resource Information for the Symfoware Database Cluster

1. Use the `pgx_pclrsc` command to register the database cluster in PRIMECLUSTER - the following is the simplest execution example: The database cluster does not need to be started when executing the command.

```
# pgx_pclrsc -a -c databaseClusterName -u OsUser -D pgData -w workDir -n memberNodes
```

Item	Explanation
<i>databaseClusterName</i>	Specify the Symfoware Server database cluster name. The database cluster name is case-sensitive. The database cluster name must be as follows: <ul style="list-style-type: none"> <li>- Up to 16 bytes, and</li> <li>- The first byte must be an ASCII letter, and</li> <li>- The other bytes must be ASCII alphanumeric characters, or underscores (_).</li> </ul>
<i>OsUser</i>	Specify the operating system user (an OS user that will become the instance administrator) who can start the Symfoware Server database server.
<i>pgData</i>	Specify the absolute path of the data storage destination directory specified during database cluster creation.
<i>workDir</i>	Specify the directory for storing the temporary files required for Symfoware resource monitoring and state transition, and for storing the trace logs required in case of problems.  If a directory that does not exist is specified, create the directory. The newly created directory path assumes the <i>OsUser</i> specified with the <code>-u</code> option as the owner, and 0700 as the privilege  Required directory size is twice the size that is specified in the <code>--trace-max-file-size</code> option. If the default value is specified in the <code>--trace-max-file-size</code> option, 10240 KB x 2 is required for the directory size.
<i>memberNodes</i>	Specify the names of all nodes that make up the cluster system.

Item	Explanation
	<p>Use a comma (,) to separate each node name.</p> <p>Suffix each cluster node name with the literal "RMS".</p> <p>Specification example: -n node1RMS,node2RMS</p>

Other than the above, the following options can be specified:

Option	Explanation
--db-user=name	Specify the Symfoware database superuser. The user specified in the -u option is the default.
--response-timeout=seconds	<p>In cluster operations, queries are regularly issued to the server to perform error detection and state monitoring.</p> <p>Specify a timeout time for queries issued for the heartbeat monitoring of the Symfoware database server.</p> <p>For queries, "SELECT 1" for the database "template1" is used. If 0 is specified, queries will not time out. The default is 0.</p>
--timeout-retry-count=count	<p>Specify the upper limit for the number of retries when queries for monitoring time out.</p> <p>The default is 6 times.</p>
--trace-max-file-size	Specify the maximum size of the trace file. The default is 10240 KB.
--watch-interval=seconds	Specify the monitoring interval (the interval with which the queries are issued to the server) for a monitoring program to monitor the Symfoware database server. The default is 3 seconds.



See

Refer to the "Reference" for information on the `pgx_pclrsc` command.

## 2.3.9 Configuring Storage Data Protection Using Transparent Data Encryption

If using transparent data encryption, refer to "Protecting Storage Data Using Transparent Data Encryption" in the Operation Guide to configure it. Take note on the following when configuring it:

### Location of the keystore file

Do not specify a shared disk for the `keystore_location` parameter in `postgresql.conf`. Instead, specify a local directory with the same path on all nodes that comprise the cluster application.

If a shared disk is specified, startup from the cluster application will fail. This is because the `keystore.aks` file, which is generated when automatic opening of the keystore is enabled, is obfuscated so that it can only be read on the node where it was generated, and therefore it cannot be shared across the shared disk.

### Distribution of the keystore file

The master encryption key settings must be common across all nodes. For this reason, configure the master encryption key settings on one node, and then copy the keystore file to each node. Also, if the master encryption key or passphrase is changed, you must also copy the keystore file to each node.

### Enabling automatic opening of the keystore

Automatic opening of the keystore must be enabled in all nodes that comprise the cluster application. Note that, when enabling the automatic opening of the keystore, only the directory in which the keystore file is stored will be referenced, therefore tasks such as starting and stopping the GDS volume are not required.

If it is not enabled, then startup from the cluster application will fail. This is because the startup process will not finish, since it waits for the manual input of the passphrase that opens the keystore.

## 2.3.10 Configuring the PL/extJava Environment Settings

---

Note the following points when using the PL/extJava environment:

### Location in which the PL/extJava resources are placed

Place the domain root and the resources of the PL/extJava backup data storage destination directory on a shared disk.

### Environment setup

To use the `pgx_jadmin` command on the standby node, specify the `-cluster-standbynode` option.

### Setup procedure

Use the following procedure to configure the PL/extJava environment:

1. Stop the cluster application on all nodes.
2. On the active node or the standby node, mount the file systems on which the domain root and the PL/extJava backup data storage destination directory have been placed, and all file systems on which the resources of the database cluster (such as the data storage destination directory), have been placed.  
  
Refer to "[2.3.6 Mounting the File System](#)" for details.
3. Configure the PL/extJava environment.  
  
Refer to "Setting up and Operating PL/extJava" in the Operation Guide for details.
4. Unmount all file systems that were mounted on the active node or the standby node.  
  
Refer to "[2.3.11 Unmounting the File System](#)" for details.
5. Restart the cluster application on all nodes.

## 2.3.11 Unmounting the File System

---

1. Unmount all file systems containing the resources required to create database clusters (such as the data storage destination directory).

```
# umount /mnt/swdsk1
```

2. Stop the GDS volume.

## 2.3.12 Creating a Cluster Application

---

Refer to "[Appendix A Creating Resources and Creating/Modifying Cluster Applications](#)".

## 2.3.13 Checking Operation

---

To ensure that the environment settings have been configured correctly, start, switch, and stop from Web-Based Admin View, and check the behavior.

# Chapter 3 Failover Operation

This chapter explains the procedures involved in failover operation.

## 3.1 Adding Tablespaces

This section explains how to add tablespaces to a new file system.

This procedure is not required when you are adding tablespaces to an existing file system.

Perform the following steps when using a new shared disk class:

- Stop RMS on all nodes.
- Perform setup as described from "2.3.3 Creating a GDS Volume" to "2.3.4 Creating a File System".
- Modify the cluster application (refer to "A.2 Modifying Cluster Applications" for details).

Perform the following steps when not using a new shared disk class:

- Stop RMS on all nodes.
- Perform the setup as described in "2.3.4 Creating a File System".
- Modify the cluster application (refer to "A.2 Modifying Cluster Applications" for details).

## 3.2 Modifying Database Cluster Resources

This section explains the following operations:

- Displaying database cluster resource information
- Modifying database cluster resource content
- Deleting a database cluster resource

### Displaying database cluster resource information

Execute the `pgx_pclrsc` command as shown below (note that if `-c` is not specified, the command lists all registered database cluster names):

```
# pgx_pclrsc -p -c databaseClusterName
```

### Modifying database cluster resource content

1. Stop RMS.
2. Execute the `pgx_pclrsc` command as shown below (this is the simplest example):

```
# pgx_pclrsc -d -c databaseClusterName  
# pgx_pclrsc -a -c databaseClusterName -u OsUser -D pgData -w workDir -n memberNodes
```



#### Note

If you are modifying resource content, first delete the Symfoware resource from the cluster application, and then re-create it.



#### See

Refer to "Reference" for information on the `pgx_pclrsc` command.

## Deleting a database cluster resource

1. Stop RMS.
2. Delete the Symfoware resource from the cluster application.
3. Execute the `pgx_pclrsc` command as shown below:

```
# pgx_pclrsc -d -c databaseClusterName
```



The directory specified in the `-w` option of the `pgx_pclrsc` command during registration will not be deleted. Use operating system commands to delete it if necessary.

## 3.3 Changing the PL/extJava Environment

---

Perform the following steps to change the PL/extJava environment:

1. Stop the cluster application on all nodes.
2. On the active node or the standby node, mount the file systems on which the domain root and the PL/extJava backup data storage destination directory have been placed, and all file systems on which the resources of the database cluster (such as the data storage destination directory), have been placed.  
  
Refer to "[2.3.6 Mounting the File System](#)" for details.
3. Change the PL/extJava environment.  
  
Refer to "[2.3.10 Configuring the PL/extJava Environment Settings](#)" for details.
4. Unmount all file systems that were mounted on the active node or the standby node.  
  
Refer to "[2.3.11 Unmounting the File System](#)" for details.
5. Restart the cluster application on all nodes.

## 3.4 Operations under Normal Circumstances

---

### Starting and stopping

For cluster operations, perform starting and stopping from the cluster application. If starting and stopping are performed by using the `pg_ctl` command or WebAdmin during failover operation, the cluster application will misjudge that services have gone down, resulting in unexpected behaviors.

### Switching

There are two ways to switch between the active node and the standby node, as shown below.

Under normal circumstances, switch using mutual switch. In an emergency, for example when there is no response from the active node, perform forced switch. Forced switch should only be used in emergency scenarios because the differences in Symfoware stop modes when the active node is offline will cause the statistics to be initialized, and the load will increase as crash recovery is performed after the switch.

- Mutual switch

Refer to "[3.6.2.4 Mutual Switch](#)".

Symfoware is stopped with the "fast" mode of the `pg_ctl` command.

- Forced switch

Right-click the cluster application object, and select [Switch] from the menu. From the switchable nodes displayed in the drop-down menu, select a target node, and switch the cluster application to that node.

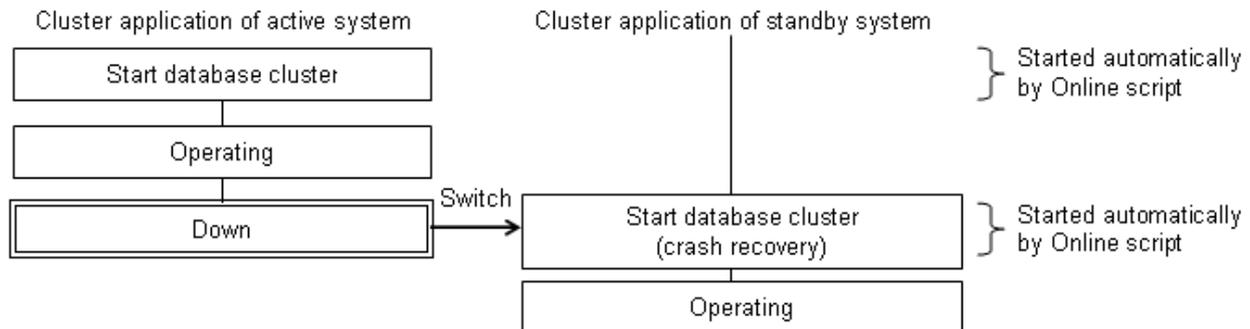
Symfoware is stopped with the "immediate" mode of the `pg_ctl` command.

## 3.5 Operation at State Transition

---

The workflow for operation at state transition is shown below.

The activation operation during failover operation will be performed automatically by the Online script.



### Non-transferrable feature

- All transactions being processed by the cluster application on the active system will be rolled back.
- The statistics Symfoware Server has collected will be initialized.

The statistics will be initialized in any of the following cases. Refer to the notes outlined in "Starting and Stopping an Instance" in the Operation Guide for details.

- When the node goes down because of a kernel panic, for example
- When forced switch is performed

Statistics are initialized because the "immediate" mode of the `pg_ctl` command is used to stop Symfoware Server.

Refer to "[3.4 Operations under Normal Circumstances](#)" for details.

- When an error occurs in the cluster application resources

Statistics are initialized because the "immediate" mode of the `pg_ctl` command is used to stop Symfoware Server to perform an immediate switch.

### Errors during state transition

If an error occurs in a Symfoware Server script during cluster application state transition, the following message will be displayed in the switchlog of Cluster Admin (the registered database cluster name will be displayed in the `pgxtest` part shown below):

```
2012-05-14 11:08:22.060:(DET, 3): ERROR: FAULT REASON: Resource <Symfoware_pgxtest> transitioned to a Faulted state due to a script failure.
```

To display the switchlog, select [Tools] >> [View switchlog] in the [Cluster Admin] window.

If the resource displayed in the error message is a Symfoware resource name, a message indicating the cause will be output either before the message above or to the system log.

## 3.6 Maintenance Tasks

---

When you need to perform tasks such as configuration changes, patch application, and hardware parts replacement that may require restarting, there are two ways of node maintenance. Performing maintenance on both nodes by stopping both active and standby nodes, and performing maintenance on one node at a time by leaving the active node running to perform maintenance on the standby node first, and then switch the active node to perform maintenance on the other node. In this section, the former is referred to as simultaneous stopped node maintenance, whereas the latter is referred to as mutual switch maintenance.

Refer to the tasks outlined below in either "Figure 3.1 Workflow for simultaneous stopped node maintenance" or "Figure 3.2 Workflow for mutual switch maintenance", and perform your maintenance tasks accordingly.

Figure 3.1 Workflow for simultaneous stopped node maintenance

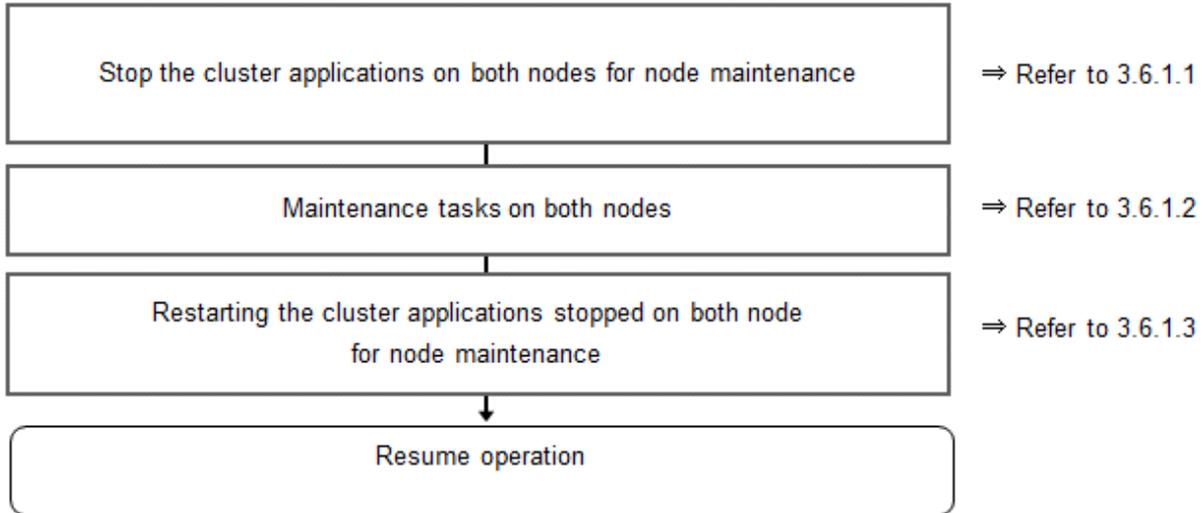
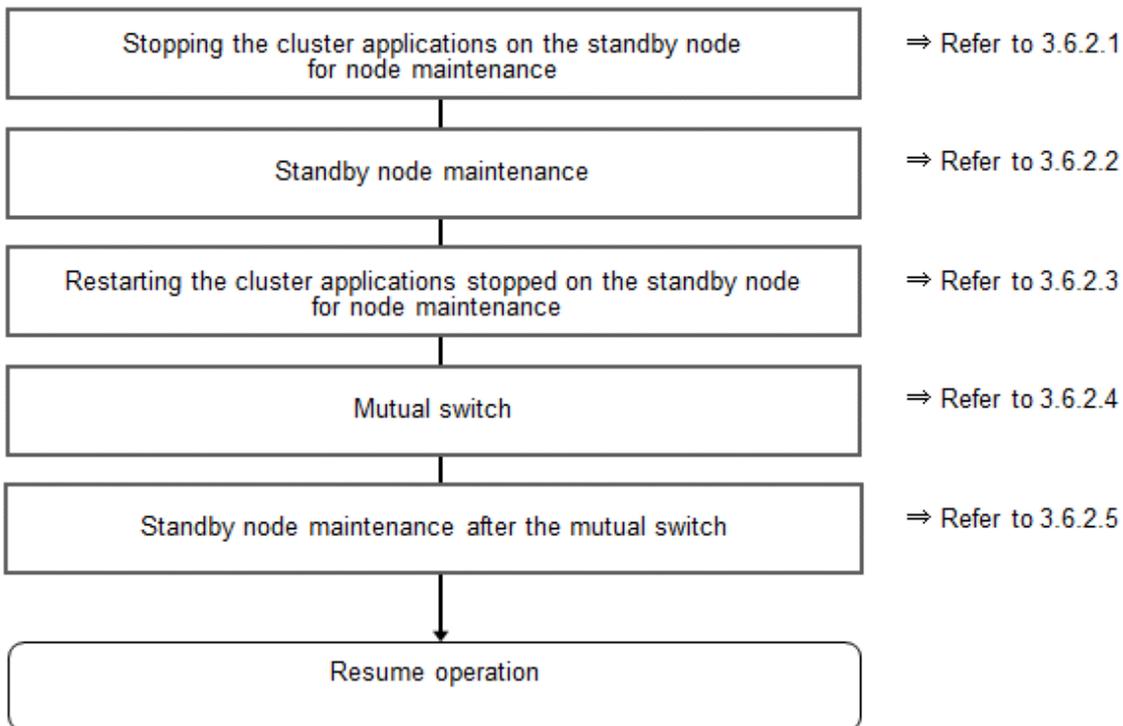


Figure 3.2 Workflow for mutual switch maintenance



### 3.6.1 Simultaneous Stopped Node Maintenance

---

This section explains the procedure to perform maintenance on both active and standby nodes simultaneously.

#### 3.6.1.1 Stopping the Cluster Applications on Both Nodes for Node Maintenance

Follow the steps shown below to stop the cluster application on the active and standby nodes:

1. Start the [Cluster Admin] window.

2. Stop the cluster application.

On the RMS tree in the [Cluster Admin] window, right-click the standby cluster application to be stopped, and select [Offline] from the pop-up menu to stop it.

Stop the cluster application on both the active and standby nodes.

3. Stop RMS.

On the RMS tree in the [Cluster Admin] window, right-click the system node where the cluster application to be stopped exists, and select [Shutdown RMS] from the pop-up menu to stop RMS. To perform maintenance without stopping RMS, use the `hvutil -d` command to change the cluster application to "Deact" state.



- If applying the Symfaware Server update patches, RMS on the node (onto which the patches are applied) must be stopped. This is because the Symfaware monitoring process that runs when linked with PRIMECLUSTER must be stopped.
- By stopping RMS, all cluster applications on the system node where RMS is stopped will stop. When you have multiple cluster applications configured, ensure that stopping all other cluster applications will not cause any problems, before stopping RMS.

### 3.6.1.2 Maintenance Tasks on Both Nodes

Perform maintenance tasks such as configuration changes, patch application, hardware and parts replacement that may require restarting.

### 3.6.1.3 Restarting the Cluster Applications Stopped on Both Nodes for Node Maintenance

Follow the steps shown below to restart the cluster application on the active and standby nodes.

1. Start the [Cluster Admin] window.
2. If RMS has not been started, select [Tools] >> [Start RMS] in the [Cluster Admin] window, and select the node to be started and click [Ok].

Note that if the cluster applications were stopped for node maintenance using the `hvutil -d` command, execute the `hvutil -a` command before starting RMS.

3. Start the cluster application.

On the RMS tree in the [Cluster Admin] window, right-click the standby cluster application, and from the pop-up menu, if you are starting the node as the active node, select [Online], or if you are starting it as the standby node, select [Standby]. If the state is [Fault], then select [Clear fault].

This operation is not required if the cluster application has been configured to start automatically when RMS is started.

## 3.6.2 Mutual Switch Maintenance

---

This section explains the procedure to perform maintenance on the standby node while running the active node at the same time.

### 3.6.2.1 Stopping the Cluster Applications on the Standby Node for Node Maintenance

Follow the steps shown below to stop the cluster application on the standby node only:

1. Edit the `/opt/SMAW/SMAWRrms/bin/hvenv.local` file, and add "export HV\_RCSTART=0".

If the `hvenv.local` file does not exist, then create it.

```
# vi /opt/SMAW/SMAWRrms/bin/hvenv.local
Add export HV_RCSTART=0
```

2. On the RMS tree in the [Cluster Admin] window, right-click the cluster application on the standby node to be stopped, and select [Offline] from the pop-up menu to stop it.

3. On the RMS tree in the [Cluster Admin] window, right-click the system node (where the cluster application on the standby node that was stopped exists), and select [Shutdown RMS]. At this stage, select [Keep local Apps] as the option, and stop RMS.

### Note

- Ensure that the steps up to step 3 have been completed properly before performing the maintenance tasks. If the active node goes down at the point when steps 1 and 2 have been completed, Symfoware will be started on the supposedly stopped standby node, and it becomes the active node instead. As a result, this will compete with the maintenance tasks and unexpected behavior may occur.
- If applying the Symfoware Server update patches, RMS on the node (onto which the patches are applied) must be stopped. This is because the Symfoware monitoring process that runs when linked with PRIMECLUSTER must be stopped.
- By stopping RMS, all cluster applications on the system node where RMS is stopped will stop. When you have multiple cluster applications configured, ensure that stopping all other cluster applications will not cause any problems, before stopping RMS.

## 3.6.2.2 Standby Node Maintenance

Perform maintenance tasks such as patch application and hardware parts replacement.

## 3.6.2.3 Restarting the Cluster Applications Stopped on the Standby Node for Node Maintenance

Follow the steps shown below to restart the cluster application on the standby node only (that was stopped for maintenance):

1. Edit the /opt/SMAW/SMAWRrms/bin/hvenv.local file, and delete "export HV\_RCSTART=0".

```
# vi /opt/SMAW/SMAWRrms/bin/hvenv.local
Delete export HV_RCSTART=0
```

2. Select [Tools]-[Start RMS] in the [Cluster Admin] window, and select the node to be started and click [Ok].
3. On the RMS tree in the [Cluster Admin] window, right-click the cluster application to be started as the standby node, and select [Standby] from the pop-up menu to start it. If the state is [Fault], then select [Clear fault].

## 3.6.2.4 Mutual Switch

To perform mutual switch between the active and standby nodes, perform the steps shown below:

1. Select and right-click the cluster application on the active node, and select [Offline] from the pop-up menu to stop it.
2. Select and right-click the cluster application on the standby node, and select [Online] from the pop-up menu to start it.
3. Select and right-click the cluster application that was previously on the active node, and select [Standby] from the pop-up menu to start it as standby.

## 3.6.2.5 Standby Node Maintenance after the Mutual Switch

To perform maintenance tasks on the new standby node after mutual switch, perform the steps from "[3.6.2.1 Stopping the Cluster Applications on the Standby Node for Node Maintenance](#)" to "[3.6.2.3 Restarting the Cluster Applications Stopped on the Standby Node for Node Maintenance](#)".

## Chapter 4 Procedures Required after a Failover Error

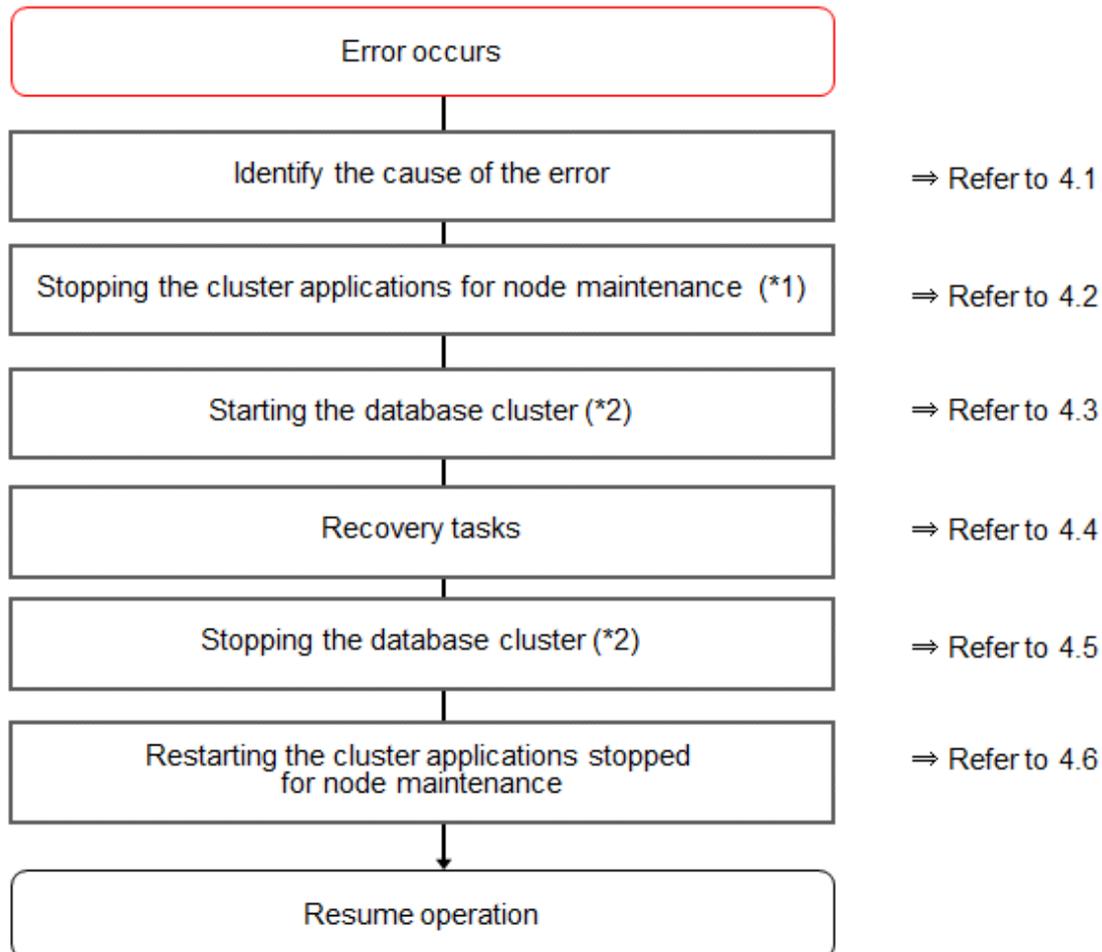
This chapter explains how to perform recovery when an abnormality occurs during failover operation.

When an error occurs during failover operation, refer to the system log to identify the cause and stop the RMS to perform recovery, separately from the RMS management. Normal operations can be resumed by restarting the RMS once the recovery is completed.

Note that the database cluster may be started or stopped without using cluster applications when performing recovery.

Refer to the tasks outlined below in "[Figure 4.1 Operation flow when a failover error occurs](#)", and perform recovery.

Figure 4.1 Operation flow when a failover error occurs



\*1) If the machine is already stopped when the error occurred, this step is not necessary.

\*2) Depending on the recovery tasks, start or stop the database cluster as required.

### 4.1 Identifying the Cause of an Error

When an error occurs during failover operation, refer to the system log to identify the cause. When cluster applications fail to start, refer to the server.log file to identify the cause.

The server.log file is a file created under the directory specified in the -w option when executing the pgx\_pclrsc command. This file has almost the same content as the file specified in the -l option when executing the pg\_ctl command to start the database cluster, and errors that are not output to the system log are output to these files, including startup errors.

## 4.2 Stopping the Cluster Applications for Node Maintenance

---

When an error occurs during failover operation, stop the RMS to perform recovery, separately from the RMS management. Refer to "[3.6 Maintenance Tasks](#)" for information on how to stop cluster applications for node maintenance.

Note that this task is not required in cases such as when the RMS is stopped because the machine is stopped due to an error.

## 4.3 Starting the Database Cluster

---

To perform recovery when an error occurs, it is necessary to start the Symfoware Server without using cluster applications.

Follow the procedure below to start the database cluster if required.

Check if the database cluster needs to be started by referring to the recovery tasks written in "Actions When an Error Occurs" in the Operation Guide.

1. Start the GDS volume.
2. Mount the necessary file systems such as those where the data storage destination directory is placed.
3. Start the database cluster by using the `pg_ctl` command. Refer to "Using Server Commands" in "Starting an Instance and Creating a Database" in the Operation Guide for information on how to start the database cluster.

## 4.4 Recovery Tasks

---

Refer to "Actions When an Error Occurs" in the Operation Guide for information on how to perform recovery. If there are two ways available to perform recovery; using WebAdmin or commands, use commands.

## 4.5 Stopping the Database Cluster

---

If a database cluster is started, stop the database cluster.

1. Stop the database cluster using the `pg_ctl` command. Refer to "Using Server Commands" in "Starting an Instance and Creating a Database" in the Operation Guide" for information on how to stop the database cluster.
2. Unmount the file systems such as those where the data storage destination directory is placed.
3. Stop the GDS volume.

## 4.6 Restarting the Cluster Applications Stopped for Node Maintenance

---

Refer to "[3.6 Maintenance Tasks](#)" to restart the cluster applications stopped for node maintenance.

# Part 3 Database Multiplexing Mode

---

This part describes database multiplexing mode.

<a href="#">Chapter 5 Overview of Database Multiplexing Mode.....</a>	25
<a href="#">Chapter 6 Setting up Database Multiplexing Mode.....</a>	33
<a href="#">Chapter 7 Operations in Database Multiplexing Mode.....</a>	49
<a href="#">Chapter 8 Action Required when an Error Occurs in Database Multiplexing Mode.....</a>	62

# Chapter 5 Overview of Database Multiplexing Mode

This chapter provides an overview of database multiplexing mode.

## Point

In this and subsequent chapters, the word "Mirroring Controller" may be used in the process or management directory name or explanation, even when database multiplexing mode does not use the Mirroring Controller option.

Cases where the Mirroring Controller option is used are denoted by the text "when the Mirroring Controller option is used".

## 5.1 What is Database Multiplexing Mode

This operation mode is based on PostgreSQL streaming replication. Other software such as cluster software is not required.

This mode replicates the database on all servers that comprise the cluster system. It achieves this by transferring the updated transaction logs of the database from the server that receives the updates (primary server) to another server (standby server), and then reflecting them on the standby server.

It consists of a feature that detects faults in the elements that are essential for the continuity of the database operation (such as the database process, disk, and network), as well as simplified switchover and standby server disconnection features. The database will be copied in synchronous mode.

### Monitoring using database multiplexing mode

- Operating system or server failures, and no-response state

By generating a heartbeat between Mirroring Controller on each server, operating system or server errors are detected and acknowledged between the relevant servers.

- Database process failures, and no-response state

Mirroring Controller periodically accesses the database processes and checks the status. A process error is detected by monitoring whether an access timeout occurs.

Mirroring Controller periodically accesses the database processes and checks the status. A process error is detected by monitoring whether an access timeout occurs.

- Disk failure

Mirroring Controller periodically creates files on the data storage destination disk. A disk error is detected when an I/O error occurs.

Failures that can be detected are those that physically affect the entire system, such as disk header or device power failures.

Figure 5.1 Failover from the primary server to the standby server

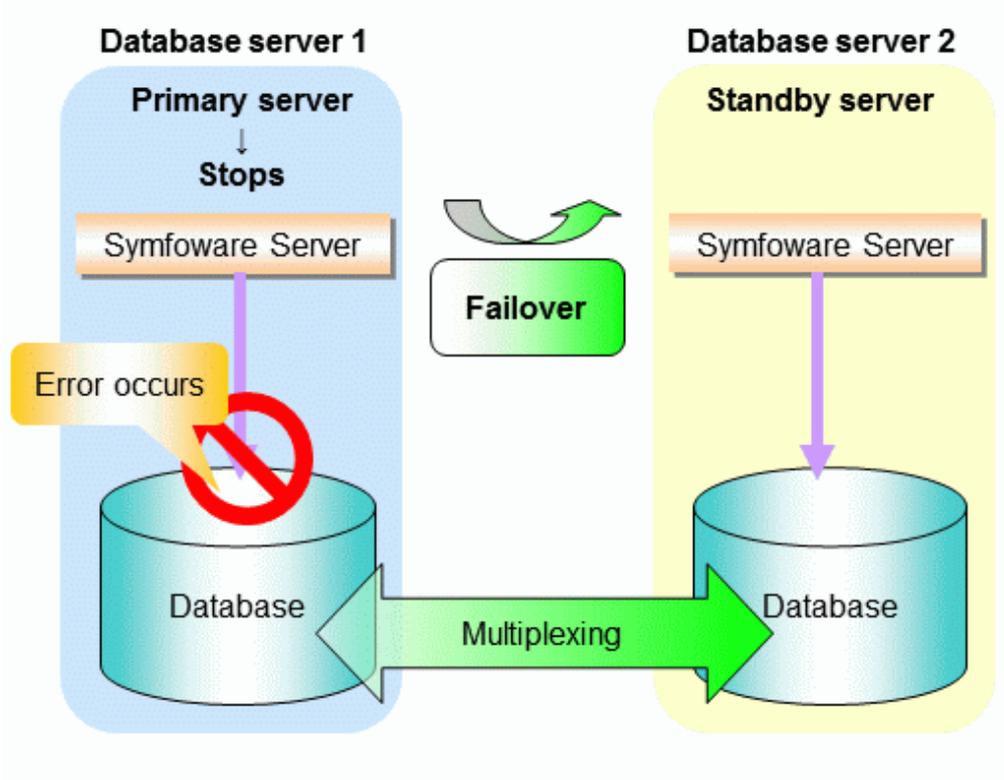
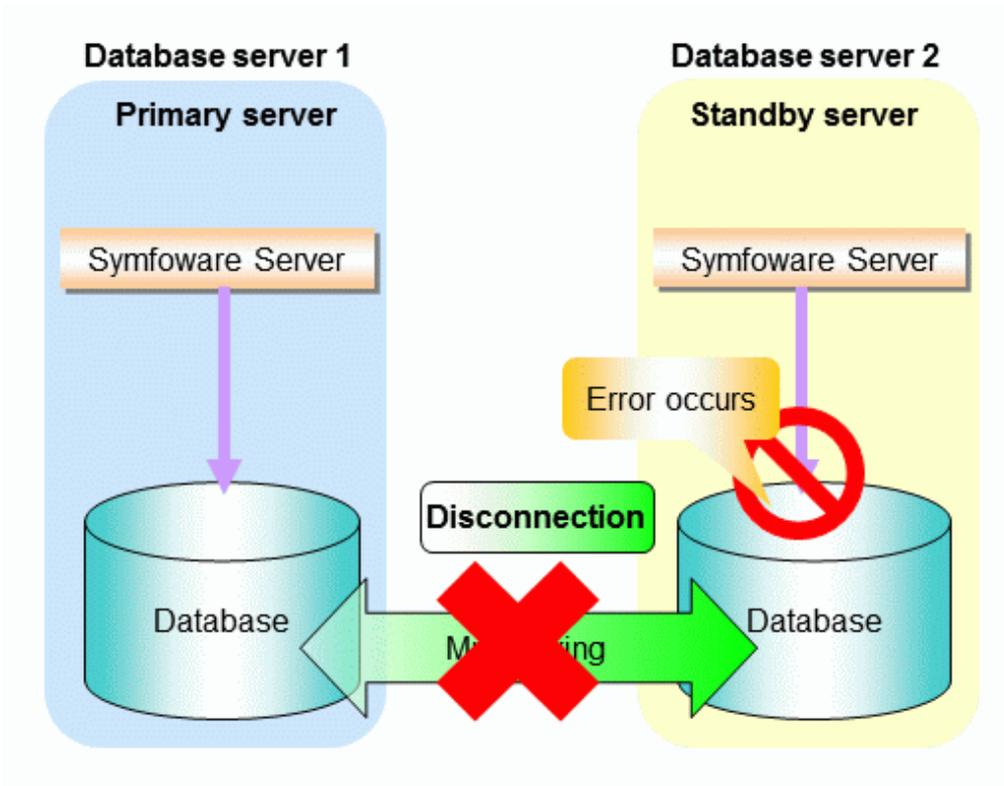


Figure 5.2 Standby server disconnection



### Mirroring Controller option

When database multiplexing is used to perform monitoring and an error is detected, this option will automatically switch the primary server or disconnect the standby server.

Reference jobs can also be executed on the standby server.

### Note

If the role of primary server was switched to another server and then starts degrading, the original primary server will not become the standby server automatically. Remove the cause of the error, and then change the role of the original primary server to the server currently acting as standby server. Refer to "[8.1 Action Required when Server Degradation Occurs](#)" for details.

### See

- Refer to "[5.2.4 Notes on Executing Reference Jobs on the Standby Server \(when the Mirroring Controller Option is Used\)](#)" for information on reference jobs.
- The features that can be used will depend on the whether the Mirroring Controller option is used. Refer to "Features that can be Installed" in the Installation and Setup Guide for Server for details.

### Point

If the standby server was disconnected, Mirroring Controller will automatically delete the target standby server from the synchronous\_standby\_names parameter in the postgresql.conf file of the primary server. Accordingly, you can prevent the application processing for the primary server being stopped.

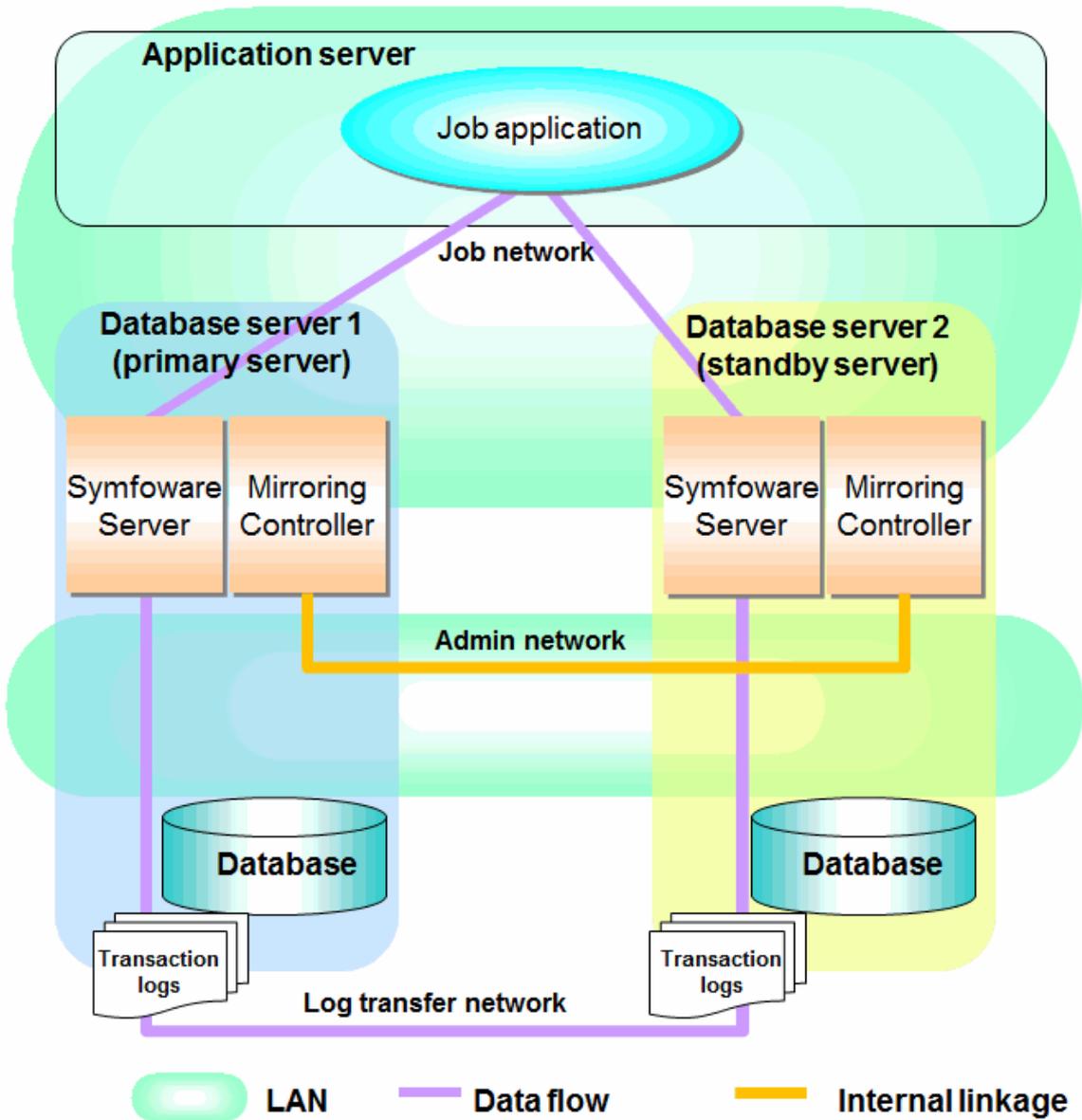
## 5.2 System Configuration for Database Multiplexing Mode

This section explains the products, features, and networks that are part of a database multiplexing system.

The following table shows the network types uses by database multiplexing systems.

Network type	Description
Job network	Network between the application that accesses the database, and the database server.
Admin network	Network used by the primary server and the standby server to monitor each other using Mirroring Controller, and to control Mirroring Controller of other servers.
Log transfer network	Network used to transfer the transaction logs of the database, which is part of database multiplexing.

Figure 5.3 System configuration for database multiplexing mode



### 5.2.1 Mirroring Controller Resources

The only Mirroring Controller resource is the Mirroring Controller management directory, which stores the files that define the Mirroring Controller behavior, and the temporary files that are created when Mirroring Controller is active.

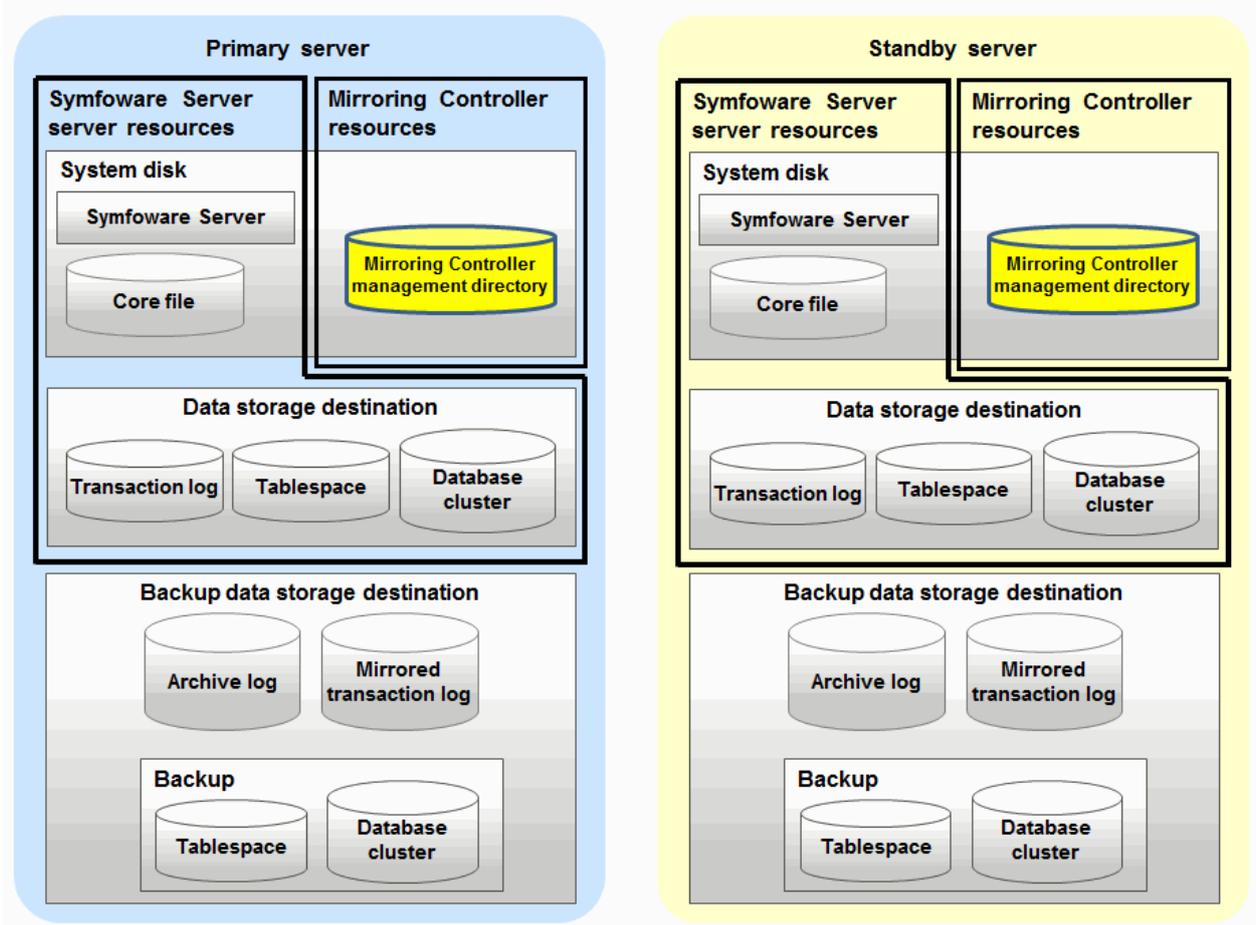
 **Note**

- Do not create the Mirroring Controller management directory in a directory managed by Symfoware Server, otherwise it may be deleted by mistake or may cause unexpected problems when Symfoware Server recovery is performed (such as old version of files being restored).
- Refer to "Preparing Directories for Resource Deployment" in the Installation and Setup Guide for Server for information on the directories managed by Symfoware Server.
- The backup methods described in "Backing Up the Database" in the Operation Guide cannot be used to back up the Mirroring Controller resources. Therefore, users must obtain their own backup of Mirroring Controller resources, in addition to Symfoware Server server resources.

The image below shows the configuration when backing up Mirroring Controller resources

The content on the primary server will be backed up. You cannot tell which server is the primary server to be backed up, because switching and failback may be performed between the servers. It is also impossible to tell which server is to be restored using the backed up data. Accordingly, ensure that you create a backup of each server when it is working as the primary server.

Figure 5.4 Configuration when backing up Mirroring Controller resources



## 5.2.2 Redundancy of the Admin and Log Transfer Networks

The admin network is an important one, because it is used by Mirroring Controller to check the status of each server.

Additionally, the log transfer network is an important one, because it is necessary to ensure data freshness.

Accordingly, configure a failure-resistant network by implementing network redundancy via channel bonding provided by the operating system or network driver vendor.

## 5.2.3 Notes on CPU Architecture and Products

Use the same CPU architecture (endian) for the primary server and the standby server. Additionally, use the same bit architecture (32-bit/64-bit) for the Symfoware Server installation.

A server using only PostgreSQL streaming replication cannot be specified as the database multiplexing system log transfer destination.

## 5.2.4 Notes on Executing Reference Jobs on the Standby Server (when the Mirroring Controller Option is Used)

In database multiplexing mode that uses the Mirroring Controller option, switch and disconnection are performed automatically. If a switch occurs when executing a reference job for the standby server, it may affect the performance of that job. The reason is that, on the new primary server (that is, the original standby server), both the main job that was being executed on the original primary server and the

reference job that was being executed on the original standby server will be processed. Therefore, before executing a job using this configuration, give careful consideration to the server resource estimates, and the likely impact on performance.

Accordingly, if you are concerned about the impact on performance, pause the reference job on the original standby server, change the original primary server to the new standby server, and then resume the reference job for the new standby server.

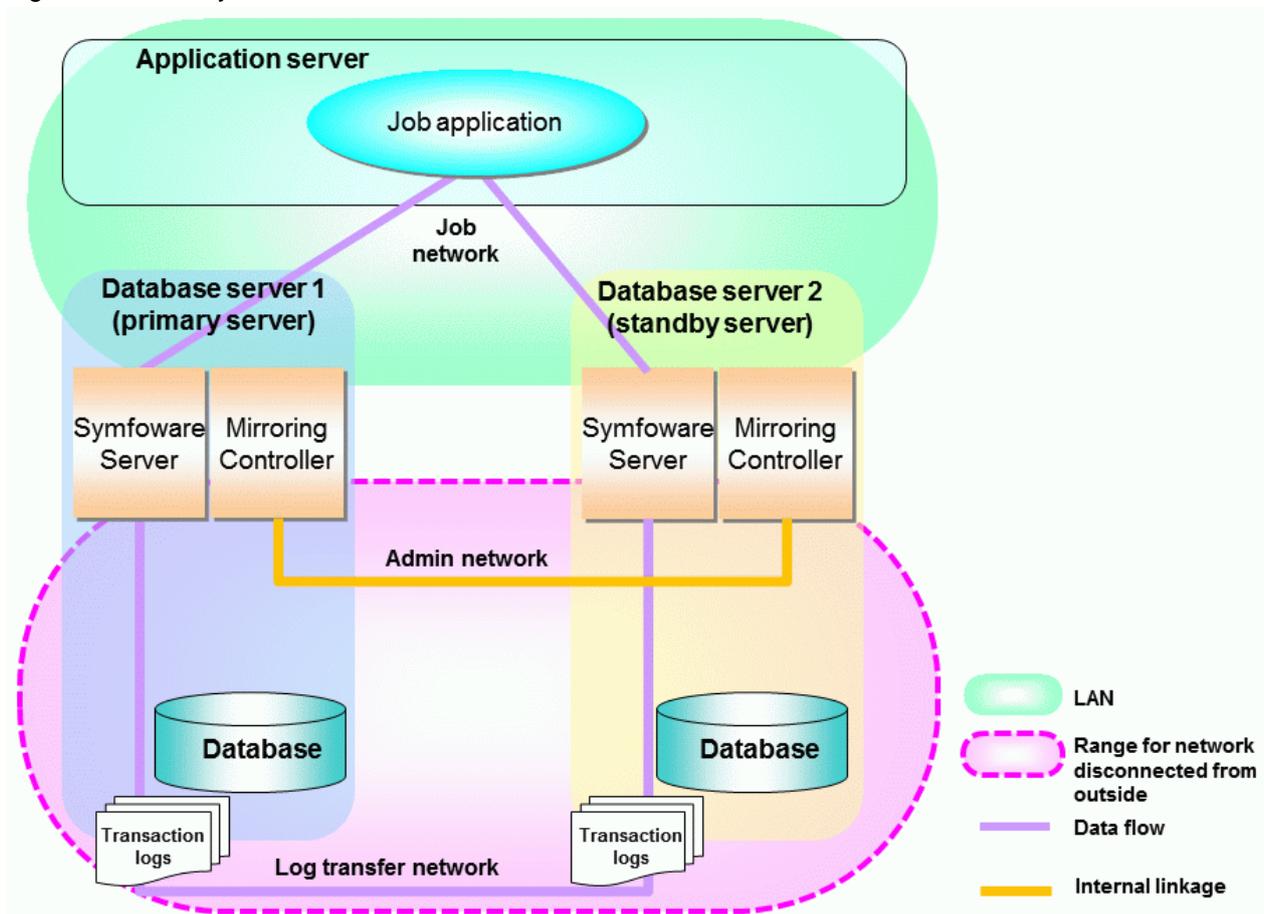
## 5.3 Security in Database Multiplexing

Database multiplexing mode replicates the database on all servers that comprise the cluster system. It achieves this by transferring and reflecting the updated transaction logs of the database from the primary server to the standby server.

To safeguard the database against unauthorized access and protect the confidentiality of data when transferring the transaction logs, carefully consider security and take note of the following:

- Do not use trust authentication when using replication connection.
- Configure the admin network and the log transfer network so that they cannot be connected from the outside, as shown below.

Figure 5.5 Security



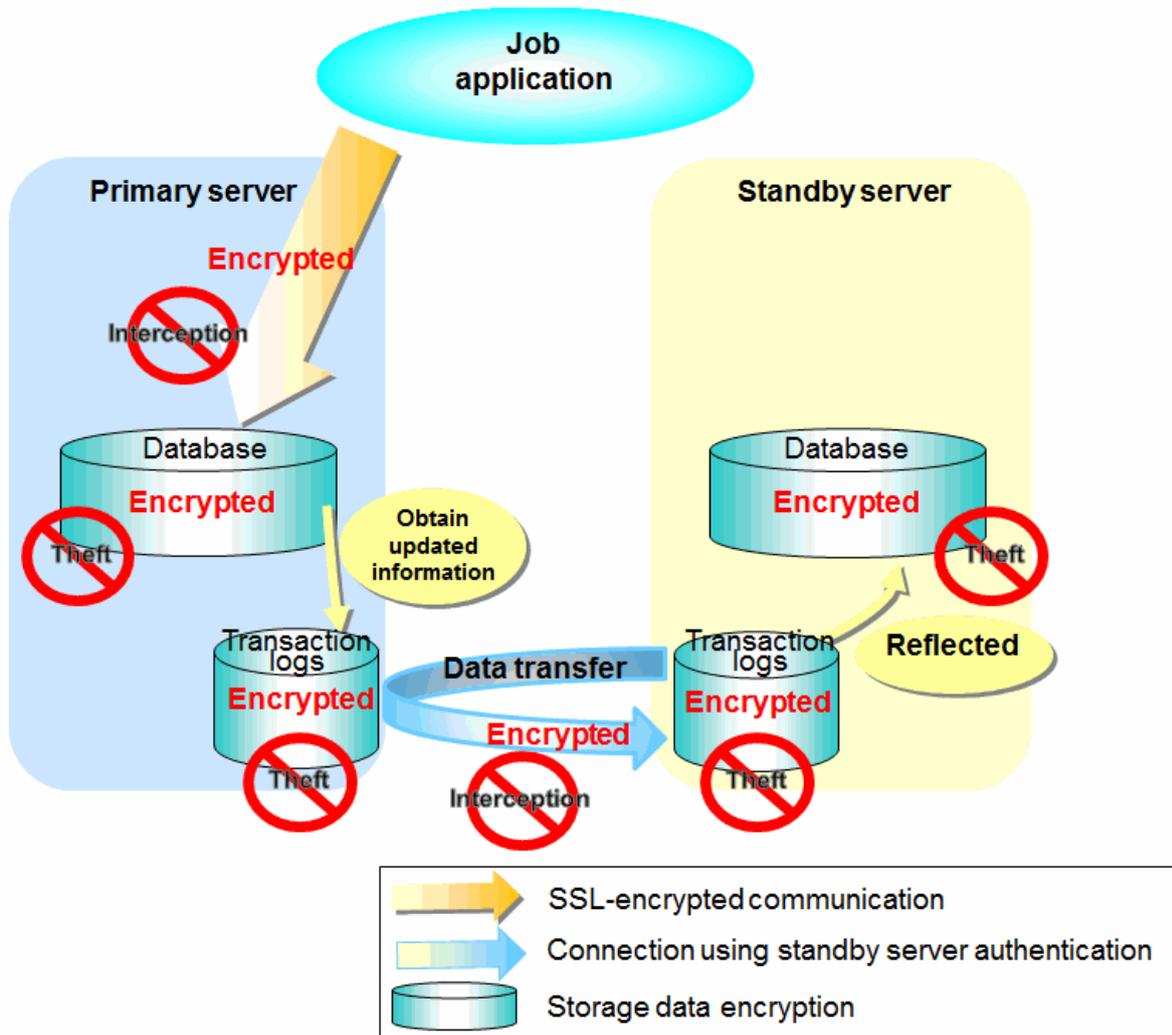
However, it may not always be possible to adopt the configuration mentioned above. For example, you may want to place the servers in a nearby/neighbor office to minimize network delays.

In this case, combine the following features to enhance security:

- [Authentication of the standby server](#)
- [Encryption of transaction logs transferred to the standby server](#)

When these features are combined, security will be achieved as shown below.

Figure 5.6 Security achieved when standby server authentication is combined with transaction log encryption



 See

Refer to "Performing Database Multiplexing" under "Configuring Secure Communication Using Secure Sockets Layer" in the Operation Guide for information on encrypting SSL communications.

### 5.3.1 Authentication of the Standby Server

You can prevent spoofing connections from an external server purporting to be the standby server by using authentication with a user name and password.

Configure the setting in the primary server `pg_hba.conf` file so that authentication is performed for connections from the standby server in the same way as for connections from the client.

 See

Refer to "Client Authentication" in the PostgreSQL Documentation for information on content that can be configured in `pg_hba.conf`.

## 5.3.2 Encryption of Transaction Logs Transferred to the Standby Server

---

In case the authentication of the standby server is breached so that a malicious user purporting to be the standby server can spoof data, the transaction log data can be encrypted to prevent it from being deciphered. The transparent data encryption feature is used to encrypt the data.



See

.....  
Refer to "Protecting Storage Data Using Transparent Data Encryption" in the Operation Guide for details.  
.....

# Chapter 6 Setting up Database Multiplexing Mode

This chapter describes how to set up database multiplexing mode, and how to check it.

## Users who perform setup and operations

Setup and operations of the database multiplexing mode must be performed by the instance administrator user.

The instance administrator user must also be registered as a database superuser.



When executing the `initdb` command, specify the `-U` option, and only perform operations in which the instance administrator user (who is the command execution user) is also a database superuser.

If an operation in which the instance administrator user is not a database superuser is performed, Mirroring Controller will be unable to connect to the instance, and therefore will not be able to perform error monitoring for the database multiplexing mode.

## Matching the system times

Before starting the setup, ensure that the times in both servers match, by using the operating system time synchronization feature, for example.

The tolerated difference is approximately one second.

If the system times are not synchronized (because the tolerated difference is exceeded, for example), problem investigation may be affected.

## Setup

Perform the procedure in the table below to set up database multiplexing mode.

Database multiplexing mode uses the streaming replication feature. If you know how to set up streaming replication, the setup of database multiplexing mode will be easy.

Step	Task		Refer to
	Primary server	Standby server	
1	Installation		<a href="#">6.1 Installation</a>
2	Preparing the backup disk		<a href="#">6.2 Preparing the Backup Disk</a>
3	Setting up database multiplexing mode		<a href="#">6.3.1 Setting up Database Multiplexing Mode on the Primary Server</a>
4	Creating, setting, and registering the instance		<a href="#">6.3.2 Creating, Setting, and Registering the Primary Server Instance</a>
5		Setting up database multiplexing mode	<a href="#">6.4.1 Setting up Database Multiplexing Mode on the Standby Server</a>
6		Creating, setting, and registering the instance	<a href="#">6.4.2 Creating, Setting, and Registering the Standby Server Instance</a>
7	Confirming the streaming replication status		<a href="#">6.5 Confirming the Streaming Replication Status</a>
8	Setting up the PL/extJava environment		<a href="#">6.6 Setting up the PL/extJava Environment</a>
9	Creating applications		<a href="#">6.7 Creating Applications</a>

Step	Task		Refer to
	Primary server	Standby server	
10	Checking the behavior		<a href="#">6.8 Checking the Behavior</a>

Explanations for each step are provided below.

### Information

- The setup procedure is also the same when changing the mode on a single server to database multiplexing mode. In this case, omit the installation of Symfoware Server and the creation of the instance.  
Refer to "[7.7.2 Changing from Single Server Mode to Database Multiplexing Mode](#)" for details.
- The primary and standby server can be pseudo-configured on the same server for system testing, for example. In this case, the setup can be performed using the same procedure, however there will be some supplementary steps.  
Before performing the setup, refer to "[Appendix C Supplementary Information on Building the Primary Server and Standby Server on the Same Server](#)".

## 6.1 Installation

Refer to the manuals of each product, and then install the product.

To use features such as automatic switch or disconnection that extend database multiplexing, select the Mirroring Controller option and then perform the installation.

### See

- Refer to the Installation and Setup Guide for Server for details on how to install Symfoware Server.
- The features that can be used will depend on whether Mirroring Controller is used. Refer to "Features that can be installed" in the Installation and Setup Guide for Server for details.

## 6.2 Preparing the Backup Disk

In Mirroring Controller, backup allows recovery to be performed even if all server disks are corrupted.

The content on the primary server will be backed up. However, through switching and failback, the standby server may also become the primary server. Accordingly, prepare a backup disk device for the primary and standby servers. Ensure that backup is performed using the backup of the primary server.

## 6.3 Setting up the Primary Server

This section explains how to set up the primary server.

### 6.3.1 Setting up Database Multiplexing Mode on the Primary Server

This section explains how to set up database multiplexing mode on the primary server.

In database multiplexing, the files that are required for operations are managed in the Mirroring Controller management directory.

There is one Mirroring Controller management directory for each instance.

### Note

- Do not place the Mirroring Controller management directory in a directory managed by Symfoware Server, otherwise it may be deleted by mistake with the directories managed by Symfoware Server, and an old version of files may be restored.

- Start Mirroring Controller first, from the primary server. If it is started from the standby server, a switch will occur.



- Refer to "Preparing Directories for Resource Deployment" in the Installation and Setup Guide for Server for details on the directories that are managed by Symfoware Server.
- Refer to "mc\_ctl" in Reference for information on the command.
- Refer to "[Appendix B Parameters](#)" for details on each parameter to be edited for the setup.

Perform the following procedure:

1. Log in to the primary server.
2. Create the Mirroring Controller management directory that will store the files required by database multiplexing.  
Use ASCII characters excluding spaces in the Mirroring Controller management directory.  
Additionally, grant "Write" permission to the instance administrator user for the Mirroring Controller management directory.
3. In the network.conf file, define the network configuration that will link between the Mirroring Controller processes.  
Create the network.conf file in the Mirroring Controller management directory, based on the sample file. In the network.conf file, set "Read" permission for the instance administrator user only.  
If users other than the instance administrator user are granted access, the mc\_ctl command will not work. In this way, users other than the instance administrator user are prevented from operating Mirroring Controller.

Path of the sample file

```
/install/Dir/share/mc_network.conf.sample
```

In the network.conf file, specify the host name (or IP address) and port number of the primary server and standby server, and define the network configuration that will link between the Mirroring Controller processes.

Refer to "[B.3 Network Configuration File](#)" for details.

A definition example is shown below.

In this example, the Mirroring Controller default port number 26540 is specified, and "server1" and "server2" represent any server identifiers.

Example)

```
server1 192.0.2.100 26540  
server2 192.0.2.110 26540
```

Ensure that the port numbers for the primary and standby server do not conflict with other software.

Register in /etc/services the port number of the primary server, because programs such as WebAdmin use it to search for available port numbers.

Register any name as the service name.

4. Define the information related to Mirroring Controller monitoring and control in the *serverIdentifier.conf* file.  
Create the *serverIdentifier.conf* file in the Mirroring Controller management directory, based on the sample file.  
In the *serverIdentifier.conf* file, set "Read" permission for the instance administrator user only. If users other than the instance administrator user are granted access, the mc\_ctl command will not work.  
As the file name for the *serverIdentifier.conf* file, use the server identifier name that was specified in the network.conf file in step 3.

Path of the sample file

```
/InstallDir/share/mc_server.conf.sample
```

Set the parameters shown in the table below in the *serverIdentifier.conf* file.

Table 6.1 Parameters

Parameter	Content specified	Remarks
db_instance	'dataStorageDestinationDir'	Enclose the directory within single quotation marks ('). Use ASCII characters excluding spaces to specify this parameter.
db_instance_password	'Password of the instance administrator user'	If password authentication is performed, you must specify this parameter in the settings used when Mirroring Controller connects to a database instance. Use ASCII characters excluding spaces to specify this parameter. Enclose the name within single quotation marks (').
syslog_ident	'programName'	Specify the program name to be used to identify the Mirroring Controller messages in the system log. Use ASCII characters excluding spaces to specify this parameter. Use the same program name as the parameter in the postgresql.conf file ensures that the Mirroring Controller output content can be referenced transparently, so log reference is easy. Enclose the name within single quotation marks (').
heartbeat_interval	800 (milliseconds)	The time until an operation, such as the switch of the primary server, is performed after it was detected that there was no response, is calculated according to the following formula: $\text{heartbeat\_timeout (seconds)} \times \text{heartbeat\_retry (count)} = 1 \times 2 = 2 \text{ (seconds)}$ According to this setting, errors are monitored at 800-millisecond intervals, and if there were two successive occurrences when there was no response for a duration of one second, the operation to switch the primary server, or disconnect the standby server from the primary server, is performed.
heartbeat_timeout	1 (second)	
heartbeat_retry	2 (count)	

## 6.3.2 Creating, Setting, and Registering the Primary Server Instance

This section explains how to create, set, and register the primary server instance.

### Note

When database multiplexing mode is used, the server command cannot be used together with WebAdmin.

Additionally, instances that run in database multiplexing mode cannot be accessed from WebAdmin, even if the database multiplexing mode is stopped and operations are changed to single server mode.



- Refer to "Client Authentication" in the PostgreSQL Documentation for information on the `pg_hba.conf` file.
- Refer to "B.1 Parameters Set on the Primary Server" for information on the `postgresql.conf` file.
- Refer to "mc\_ctl" in Reference for information on the command.

Perform the following procedure:

1. Refer to "Setup" in the Installation and Setup Guide for Server, and then perform the Symfoware Server setup and create the Symfoware Server instance.

Use ASCII characters excluding spaces in the data storage destination directory.

2. Configure the encryption settings for the storage data.

Create the keystore file.

Refer to "Database Multiplexing Mode" in the Operation Guide for details, and then configure the settings.

3. To connect from the primary server to the standby server to send the transaction logs, add the following entries to the `pg_hba.conf` file:

Copy the file to the standby server later.

#	TYPE	DATABASE	USER	ADDRESS	METHOD
	host	replication	symfo	<i>standbyServerAddress</i>	<i>authenticationMethod</i>
	host	replication	symfo	<i>primaryServerAddress</i>	<i>authenticationMethod</i>

For the primary and standby server addresses, specify the IP address that will connect to the log transfer network.

Additionally, all servers can be used as the primary server or the standby server, so add entries for the addresses of all servers that comprise the database multiplexing system.

As the authentication method, set a method other than trust authentication.

So that authentication can be performed automatically, in the home directory of the instance administrator user, create the `.pgpass` file and then specify a password for the replication database.

Accordingly, the OS user account of the instance administrator user and the user registered in the database will be the same, so you can verify that the connection was not made by an unspecified user. Additionally, the password that was set beforehand will be used in the authentication, so that the connection will be automatic.

When building the standby server, each option of the `mc_ctl` command setup mode must match the user name and password specified in the `.pgpass` file.



If trust authentication is set, all OS users who can log in to the primary server will be able to connect, and if one of these is a malicious user, then that user can corrupt the standby server data, or cause the job system to fail, by sending an erroneous transaction log. Refer to "Authentication Methods" in the PostgreSQL Documentation for details on the authentication methods that can be set.

4. Configure this setting to enable the instance administrator user of the primary server to connect as a database application.

This setting enables the connection to the instance using the user name of the instance administrator user, so that Mirroring Controller can monitor instance errors. Configure this setting to enable the connection to the postgres database.

As the authentication method, set a method other than trust authentication.

- If password authentication is used

In the `db_instance_password` parameter of the `serverIdentifier.conf` file, specify the password for the instance administrator user. This password is used to connect to the database instance. If a password is not specified in the `db_instance_password` parameter, the connection to the database instance from Mirroring Controller will fail, and it will not be possible to perform the process monitoring of the instance.

- If password authentication is not used

There is no need to specify the password in the `db_instance_password` parameter.

Even if the password for the instance administrator user is specified in the `db_instance_password` parameter, it will be ignored.

An example of setting the authentication method is shown below.

#	TYPE	DATABASE	USER	ADDRESS	METHOD
	host	postgres	symfo	127.0.0.1/32	<i>authenticationMethod</i>

Refer to the note in step 3 for notes relating to trust authentication.

- To use database multiplexing mode, specify the parameters shown in the table below in the `postgresql.conf` file.

The `postgresql.conf` file is copied when the standby server instance is created. Accordingly, set the required parameters in the standby server.

To use database multiplexing mode, specify the parameters shown in the table below in the `postgresql.conf` file. After editing the `postgresql.conf` file, restart the instance.

**Table 6.2 Parameters**

Parameter	Content specified	Remarks
<code>wal_level</code>	Literal "hot_standby"	
<code>max_wal_senders</code>	Number of standby servers + 1	The number of standby servers (n) is 1, so 2 must be specified.
<code>synchronous_standby_names</code>	' <i>standbyServerName</i> '	Specify the name that will identify the standby server.  Enclose the name within single quotation marks ('').  The value specified for this parameter must also be specified in the <code>--standby-names</code> parameter when the <code>mc_ctl</code> command is executed in setup mode.  Do not change this parameter while Mirroring Controller is running.
<code>hot_standby</code>	on	Specify this to execute reference jobs on the standby server. (*1)
<code>wal_receiver_status_interval</code>	Time interval (seconds)	This value must be smaller than the value set for <code>replication_timeout</code> in the <code>postgresql.conf</code> file of the primary server.
<code>wal_keep_segments</code>	Number of file segments	If a delay exceeding the value set in this parameter occurs, the WAL segment required later by the primary server may be deleted.  Additionally, if you stop a standby server (for maintenance, for example), consider the stop time and set a value that will not cause the WAL segment to be deleted.  Setting this parameter requires consideration regarding stabilization of the database

Parameter	Content specified	Remarks
		<p>multiplexing mode (refer to "6.9.1 Tuning to Stabilize the Database Multiplexing Mode" for details).</p> <p>Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL segment.</p>
replication_timeout	Timeout (seconds)	<p>Specify the time period after which it is determined that an error has occurred in the transaction log transfer.</p> <p>By aligning this value with the value of (heartbeat_interval + heartbeat_timeout) x heartbeat_retry of the <i>serverIdentifier.conf</i> file, you can unify the time after which it is determined that an error has occurred.</p>
archive_mode	on	Specify the archive log mode.
archive_command	'installDir/bin/pgx_xlogcopy.cmd "%p" "backupDataStorageDestinationDirector y/archived_xlog/%f"'	Specify the command and storage destination to save the transaction log.
backup_destination	Backup data storage destination directory	<p>Specify the name of directory where to store the backup data.</p> <p>Set the privileges so that only the instance administrator user can access the specified directory.</p> <p>Specify the same full path on all servers, so that the backup data of other servers can be used to perform recovery.</p>
max_connections	The limit value for the number of standby server connections + number of simultaneous executions of mc_ctl status (*2) + 2 + number of simultaneous client connections to the instance + superuser_reserved_connections	<p>The value specified is also used to restrict the number of connections from client applications and the number of connections for the management of instances.</p> <p>Refer to "When an Instance was Created with the initdb Command" in the Installation and Setup Guide for Server, and "Connections and Authentication" in the PostgreSQL Documentation, for details.</p>

\*1: Used in the Mirroring Controller option feature.

\*2: Number of simultaneous executions of the mc\_ctl command in the status mode.

## 6. Start the Mirroring Controller process.

When the Mirroring Controller option is used

As the instance administrator user, execute the mc\_ctl command in start mode with the -f option specified. This enables automatic switch or disconnection.

If you start Mirroring Controller and the instance without specifying the -f option, automatic switch and disconnection will not be enabled. To enable both, start Mirroring Controller and then execute the mc\_ctl command in enable-failover mode, or restart Mirroring Controller with the -f option specified.

Example)

```
$ mc_ctl start -M /mdir/inst1 -f
```

When the Mirroring Controller option is not used

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example )

```
$ mc_ctl start -M /mdir/inst1
```

7. Obtain the backup.

Use the `pgx_dmpall` command to collect the backup.

## 6.4 Setting up the Standby Server

---

This section explains how to set up the standby server.

### 6.4.1 Setting up Database Multiplexing Mode on the Standby Server

---

This section explains how to set up database multiplexing mode on the standby server.

In database multiplexing, the files that are required for operations are managed in the Mirroring Controller management directory.

There is one Mirroring Controller management directory for each instance.



- Do not place the Mirroring Controller management directory in a directory managed by Symfoware Server, otherwise it may be deleted by mistake with the directories managed by Symfoware Server, and an old version of files may be restored.
- Start Mirroring Controller first, from the primary server. If it is started from the standby server, a switch will occur.
- When creating a standby server for a large database, stop job system operations, or specify a large value for the `wal_keep_segments` parameter.  
This is because WALs generated after the standby server is built using the `mc_ctl` command in setup mode, but before it is started, need to be retained. However, the number of WAL segments that can be retained is constrained by the `wal_keep_segments` parameter. Additionally, setting the `wal_keep_segments` parameter requires consideration regarding stabilization of the database multiplexing mode (refer to "[6.9.1 Tuning to Stabilize the Database Multiplexing Mode](#)" for details).



- Refer to "Preparing Directories for Resource Deployment" in the Installation and Setup Guide for Server for details on the directories that are managed by Symfoware Server.
- Refer to "`mc_ctl`" in Reference for information on the command.
- Refer to "[Appendix B Parameters](#)" for details on each parameter to be edited for the setup.

Perform the following procedure:

1. Log in to the standby server.
2. Create the Mirroring Controller management directory that will store the files required by database multiplexing .  
Use ASCII characters excluding spaces in the Mirroring Controller management directory.  
Additionally, grant "Write" permission to the instance administrator user for the Mirroring Controller management directory.
3. Copy, and then deploy, the `network.conf` file of the primary server.  
Copy the `network.conf` file that was defined in the primary server setup, and deploy it to the Mirroring Controller management directory of the standby server.

Set "Read" permission for the instance administrator user only. If users other than the instance administrator user are granted access, the mc\_ctl command will not work. Accordingly, users other than the instance administrator user are prevented from operating Mirroring Controller.

Register in /etc/services the port number of the standby server that was specified in the network.conf file, because programs such as WebAdmin use it to search for available port numbers.

Register any name as the service name.

4. Copy, and then deploy, the *serverIdentifier.conf* file of the primary server.

Copy the *serverIdentifier.conf* file that was defined in the primary server setup, and deploy it to the Mirroring Controller management directory of the standby server.

Set "Read" permission for the instance administrator user only. If users other than the instance administrator user are granted access privileges, the mc\_ctl command will not work.

## 6.4.2 Creating, Setting, and Registering the Standby Server Instance

This section explains how to create, set, and register the standby server instance.



See

- Refer to "Appendix B Parameters" for details on each parameter.
- Refer to "mc\_ctl" in Reference for information on the command.

Perform the following procedure:

1. Set the kernel parameters.

Refer to "Configuring Kernel Parameters" in the Installation and Setup Guide for Server for details.

2. Configure the encryption settings for the storage data.

Deploy a copy of the keystore file of the primary server on the standby server.

Refer to "Database Multiplexing Mode" in the Operation Guide for details.

3. Execute the mc\_ctl command in setup mode to create a copy of the primary server instance on the standby server.

Example)

```
$ mc_ctl setup -M /mcdir/inst1 -r standby -U userName -h primaryServerHostName -p
primaryServerPortNumber --standby-name=standbyServerName
```

4. Set the parameters shown in the table below in the postgresql.conf file.

Table 6.3 Parameters

Parameter	Content specified	Remarks
synchronous_standby_names	<i>'primaryServerName'</i>	Required after switching the primary server and then changing the original primary server to the new standby server.  Enclose the name within single quotation marks (').  Do not change this parameter while Mirroring Controller is running.

5. Start the Mirroring Controller process.

After ensuring that the Mirroring Controller process of the primary server has started, start the Mirroring Controller process.

### When the Mirroring Controller option is used

As the instance administrator user, execute the `mc_ctl` command in start mode with the `-f` option specified. This action enables automatic switch or disconnection.

If you start Mirroring Controller and the instance without specifying the `-f` option, automatic switch and disconnection will not be enabled. To enable both, start Mirroring Controller and then execute the `mc_ctl` command in enable-failover mode or restart Mirroring Controller with the `-f` option specified.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -f
```

### When the Mirroring Controller option is not used

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
$ mc_ctl start -M /mcdir/inst1
```

### 6. Check the status of the Mirroring Controller process.

As the instance administrator user, execute the `mc_ctl` command in status mode. Ensure that "mirroring status" is switchable.

Example)

```
$ mc_ctl status -M /mcdir/inst1
```

## 6.5 Confirming the Streaming Replication Status

Before performing the setup of the database multiplexing mode, ensure that the prerequisite streaming replication feature has been set up correctly.

Perform the following procedure:

### 1. On the primary server, ensure that single-row searches can be performed using the `pg_stat_replication` statistics view.

An example output of the `psql` command is shown below.

Example)

```
postgres=# select * from pg_stat_replication;
-[ RECORD 1 ]-----+-----
pid           | 16706
usesysid     | 10
username     | symfo
application_name | standby1
client_addr  | 192.0.2.210
client_hostname |
client_port  | 26500
backend_start | 2014-04-25 14:21:30.097644+09
state        | streaming
sent_location | 0/3003C30
write_location | 0/3003C30
flush_location | 0/3003C30
replay_location | 0/3003C30
sync_priority | 1
sync_state   | sync
```

### 2. Confirm the search results of step 1.

Ensure that the connection established with the intended standby server is in synchronous mode.

Table 6.4 Items to be checked

Item	Required value
application_name	Value specified for synchronous_standby_names parameter in the postgresql.conf file of the primary server.
client_addr	IP address of the standby server.
state	"streaming".
sync_state	"sync".



See

- Refer to "The Statistics Collector" in "Server Administration" in the PostgreSQL Documentation for information on the pg\_stat\_replication statistics view.
- Note that the pg\_stat\_replication statistics view may change in the future.

## 6.6 Setting up the PL/extJava Environment

This section explains how to set up PL/extJava.



See

Refer to "Setting up and Operating PL/extJava" in the Operation Guide for details.

Perform the following procedure:

1. Set up the primary server

Set up the PL/extJava environment on the primary server.

2. Set up the standby server

Set up the PL/extJava environment on the standby server, using the same port number, user name, and directory name as the primary server.

Perform the following setup on the standby server:

- Create the domain
- Configure the database cluster  
Configure the postgresql.conf file and the pg\_hba.conf file. There is no need to add the users who connect to the database.
- Create the container  
Create the container, and add the server instance (JavaVM). There is no need to extend PL/extJava using the CREATE EXTENSION statement.
- Register the Java functions  
Store the Java application. There is no need to register the Java functions using the CREATE FUNCTION and GRANT statements.

## 6.7 Creating Applications

This section explains how to create applications using database multiplexing, and points that should be noted when you create the applications.

## 6.7.1 Application Connection Server Settings

---

If database multiplexing is used and a failover occurs, it will be necessary to switch the application connection server. Accordingly, use the application connection switch feature to create applications.



See

.....  
Refer to "Application Connection Switch Feature" in the Application Development Guide for details.  
.....

## 6.7.2 Notes on Indexes

---

Do not use hash indexes.

Even if a hash index is updated, a transaction log will not be output and the hash index will not be reflected on the standby server.

## 6.8 Checking the Behavior

---

To check if the environment setup was performed correctly, start the application and then check the behavior of the switch and rebuild.

## 6.9 Tuning

---

This section explains how to tune database multiplexing mode.

### 6.9.1 Tuning to Stabilize the Database Multiplexing Mode

---

When large amounts of data are updated, the write-to load for the database will become great, and the multiplexing state may become unstable.

Accordingly, by editing the parameters below in the postgresql.conf file, a stable multiplexing state can be maintained. Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on transaction log space requirements.

Table 6.5 Parameters

Parameter	Content
wal_keep_segments	Refer to " <a href="#">6.3.2 Creating, Setting, and Registering the Primary Server Instance</a> " for details.
checkpoint_segments	The transaction log is written out according to the checkpoint trigger. If a transaction log with the capacity of the value specified in this parameter is generated, the checkpoint will be executed. If a large value is specified in this parameter, the time required for crash recovery will increase. If a small value is specified in this parameter, many checkpoints will be generated, which will affect the performance of the applications that connect to the primary server.

### 6.9.2 Tuning to Stabilize Queries on the Standby Server

---

Queries made using reference jobs on the standby server may be canceled by jobs executed on the primary server.

To reduce the possibility of a job being canceled, specify as large a value as possible for the max\_standby\_archive\_delay parameter in the postgresql.conf file.



See

.....  
- Refer to "Handling Query Conflicts" in the PostgreSQL Documentation for details.  
.....

- Refer to "Standby Servers" in the PostgreSQL Documentation for details on the max\_standby\_archive\_delay parameter.

### 6.9.3 Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server)

If jobs are updated on the primary server regularly and frequently, it will be easy for the query made by the reference job on the standby server to be canceled. In this case, edit one of the postgresql.conf file parameters shown in the table below.

Parameter	Description
hot_standby_feedback	When "on" is set, the deletion (vacuum) of the data area that was deleted or updated on the primary server is suppressed. Accordingly, the query on the standby server will not be canceled. (*1)
vacuum_defer_cleanup_age	The deletion (vacuum) of the data area that was deleted or updated on the primary server is delayed until the specified number of transactions is processed. Accordingly, the probability that the query on the standby server will be canceled decreases.

\*1: Because the vacuum is delayed, the data storage destination disk space of the primary server comes under pressure.

Additionally, if there is conflict between accesses and queries executed on the standby server, transaction logs indicating this conflict will be transferred.

Accordingly, specify as large a value as possible for the max\_standby\_archive\_delay parameter so that access conflicts do not occur.



See

- Refer to "Standby Servers" in the PostgreSQL Documentation for details on the hot\_standby\_feedback parameter.

- Refer to "Master Server" in the PostgreSQL Documentation for details on the vacuum\_defer\_cleanup\_age parameter.

## 6.10 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances

Multiplexed instances and Mirroring Controller can be started and stopped automatically in line with the starting and stopping of the operating system of the database server.



Note

To guarantee the startup sequence of Mirroring Controller on the primary and standby servers, first confirm that the primary server has started, and then start the standby servers in sequence.

If using an rc script

Perform the following procedure:

1. Set the OS user account of the instance administrator user

If you logged in using the OS user account of the instance administrator user, set the environment variables required for starting the instance.

This setting is required for executing the "su -" command.

2. Edit the shell script

Copy the sample shell script, and edit the environment variables in the shell script to suit the environment.

Path of the sample shell script

```
/install/Dir/share/rc_mcoi.sample
```

3. Copy the revised shell script and set access permissions

As the OS superuser, copy the revised shell script to /etc/rc.d/init.d, and then set and register the access rights.

In the following example, the shell script was created as "rc\_mc\_inst1".

Example)

```
# cp rc_mc_inst1 /etc/rc.d/init.d/
# chmod 755 /etc/rc.d/init.d/rc_mc_inst1
```

4. Register and enable automatic start and stop

As the OS superuser, execute the chkconfig command to register and enable the script. To register and enable the script, specify the options shown in the table below.

Option	Description
--add	Registers the shell script as the startup script.
--level	Sets and enables the startup script for each run level.

In the following example, the shell script is registered, and then enabled for run levels 3 and 5.

Example)

```
# chkconfig --add rc_mc_inst1
# chkconfig --level 35 rc_mc_inst1 on
```

If using systemd

Perform the following procedure:

1. Create a unit file

Copy the unit file sample stored in the directory below, and revise it to match the target instance.

Path of the sample file

```
/installDir/share/mcoi.service.sample
```

Example)

In the following example, the installation directory is "/opt/symfoser64", and the instance name is "inst1".

```
# cp /opt/symfoser64/share/mcoi.service.sample /usr/lib/systemd/system/mcoi_inst1.service
```

Revise the underlined portions of the options below in the unit file.

Section	Option	Specified value	Description
Unit	Description	Symfoware Server (Open Interface)	Specifies the feature overview.
		MirroringController <u>instanceName</u>	Specifies the name of the target instance. (*1)
Service	ExecStart	/bin/bash -c ' <u>installDir</u> /bin/mc_std start <u>installDir</u> <u>MirroringControllerManagementDir</u> ' <u>mc_ctlOption</u>	Command to be executed when the service is started.  Specify the option you want to add when the mc_ctl option is executed in the mc_ctl option.  Note that the content specified in this mc_ctl option is carried over from the mc_std command to the mc_ctl command.
	ExecStop	/bin/bash -c ' <u>installDir</u> /bin/mc_std stop <u>installDir</u>	Command to be executed when the service is stopped.

Section	Option	Specified value	Description
		<i>MirroringControllerManagementDir</i> ' <i>mc_ctlOption</i>	Specify the option you want to add when the mc_ctl command is executed in the mc_ctl option.  Note that the content specified in this mc_ctl option is carried over from the mc_std command to the mc_ctl command.
	User	<u>User</u>	OS user account of the instance administrator user.
	Group	<u>Group</u>	Group to which the instance administrator user belongs.

\*1: The instance name should be as nameThatIdentifiesTheInstance.  
The naming conventions for identifying the instance are as follows:

- Up to 16 bytes
- The first character must be an ASCII alphabetic character
- The other characters must be ASCII alphanumeric characters

## 2. Enable automatic start and stop

As the OS superuser, use the systemctl command to enable automatic start and stop.

Example)

```
# systemctl enable mcoi_inst1.service
```

## 6.11 Backup Operation

This section explains the backup operation for database multiplexing mode.

### 6.11.1 Backing up Database Multiplexing Mode Information

When changing the Mirroring Controller settings, in addition to backing up the database, back up the configuration file in the Mirroring Controller management directory so that the Mirroring Controller settings are not lost.

### 6.11.2 Database Backup Operation

Using database multiplexing mode is the same as obtaining the backup data on the standby server as a safeguard against a disk failure. Note that all server disks may be corrupted due to some cause.

As a safeguard against this type of case, execute the pgx\_dmpall command on the primary server to create the backup data.

However, it is not definite as to which server runs as the primary server, so ensure that the pgx\_dmpall command is executed periodically on all servers, so that the backup data will be obtained. For example, create a script to obtain the backup data, and set it in the operation management software.



#### Point

When the pgx\_dmpall command is executed on the standby server, it will not match the statuses, however the error message shown below will be output and return the value "1".

If a script that ignores only this type of error is executed on all servers, the backup data of the primary server can be obtained.

Error message

```
ERROR:recovery is in progress (10095)
```

### Note

- Consider the possibility that the server that runs as the primary server may be destroyed alongside the backup data, so it is recommended to promote another server to become the primary server, and then back up the data on the new primary server without waiting for the next scheduled backup.
- Specify the same backup directory name for the primary and standby servers. If different backup directory names are specified, and recovery is performed using the backup data of the other server, the recovery cannot be performed correctly.
- In the database multiplexing mode, data cannot be recovered to a past point using backup data.

### See

- Period backups allow shorter recovery time and reduction in disk usage. Refer to "Backing Up the Database" in the Operation Guide for details on the backup operation.
- Refer to "[Chapter 8 Action Required when an Error Occurs in Database Multiplexing Mode](#)" for details on recovery based on the backup data that was obtained using the `pgx_dmpall` command.

# Chapter 7 Operations in Database Multiplexing Mode

This chapter describes the periodic operations that are performed when running database multiplexing mode.

The periodic operations are the same as the operations on a single server.



See

Refer to "Periodic Operations" in the Operation Guide for information on the periodic operations.

## 7.1 Starting and Stopping Instances

When database multiplexing mode is used, use the `mc_ctl` command to start and stop the instance and Mirroring Controller at the same time.

Do not start or stop the instance by itself.

When the Mirroring Controller option is used

Execute the `mc_ctl` command with the `-f` option specified.

If you start Mirroring Controller and the instance without specifying the `-f` option, start Mirroring Controller and then execute the `mc_ctl` command in enable-failover mode, or restart Mirroring Controller with the `-f` option specified.

When only the instance is started and stopped, the following will happen:

- When only the instance is started
  - Features such as automatic switch and automatic disconnection will not work until Mirroring Controller is started.
- When only the instance is stopped
  - Mirroring Controller determines that an error has occurred in the instance, and performs an unnecessary automatic switch.
  - Automatic switch may also stop working correctly in some cases.

When the Mirroring Controller option is not used

When only the instance is started and stopped, the following will happen:

- When only the instance is started
  - Errors indicated in "5.1 What is Database Multiplexing Mode" will not be detected until Mirroring Controller is started.
- When only the instance is stopped
  - Mirroring Controller determines that an error has occurred in the instance, and outputs an error to the system log.

## 7.2 Checking the Database Multiplexing Mode Status

Check the multiplexed database status by executing the `mc_ctl` command in status mode.

Additionally, errors can be detected by monitoring the Mirroring Controller messages. If the status or messages are monitored periodically, you can react quickly following an automatic switch failure.

When the `mc_ctl` command is executed, the details of the multiplexing configuration, information about whether switch is possible following the error, and location and details of the error that caused the switch or disconnection are displayed.

After starting database multiplexing mode, execute the `mc_ctl` command in status mode to check the multiplexing status.

An example of the status displayed when the `mc_ctl` command is executed is shown below.

Example)

```
$ mc_ctl status -M /mdir/inst1
mirroring status
```

```

-----
switchable
server_id host_role      host                host_status db_proc_status      disk_status
-----
----
nd1       primary      192.0.2.100        normal      normal               normal
nd2       standby     192.0.2.110        normal      normal               normal

```

Additionally, by referencing the `pg_stat_replication` statistics view on the primary server, the data synchronization status can be confirmed. However, when creating the monitoring program, note that the content of `pg_stat_replication` may be changed in the future.

The following example shows that the locations of the transaction log after it is sent and received (`sent_location`, `replay_location`) match, and that they are fully synchronized.

Example)

```

postgres=# select * from pg_stat_replication;
-[ RECORD 1 ]-----+-----
pid          | 16706
usesysid     | 10
username     | symfo
application_name | standby1
client_addr  | 192.0.2.210
client_hostname |
client_port  | 26500
backend_start | 2014-04-25 14:21:30.097644+09
state        | streaming
sent_location | 0/3003C30
write_location | 0/3003C30
flush_location | 0/3003C30
replay_location | 0/3003C30
sync_priority | 1
sync_state   | sync

```



- Refer to "mc\_ctl" in Reference for information on the command.
- Refer to "Notes on Application Compatibility" in the Application Development Guide for information on retaining application compatibility.
- Refer to "The Statistics Collector" in "Server Administration" in the PostgreSQL Documentation for details on `pg_stat_replication`.

## 7.3 Manually Switching the Primary Server

The primary server cannot be switched automatically in the following cases:

- When the Mirroring Controller option is not used
- When the Mirroring Controller option is used, but automatic switch/disconnection has not been enabled

To manually switch the primary server, execute the `mc_ctl` command in switch mode on either the primary server or the standby server.

Example)

```

$ mc_ctl switch -M /mcdire/inst1

```

## 7.4 Manually Disconnecting the Standby Server

The standby server cannot be disconnected automatically in the following cases:

- When the Mirroring Controller option is not used

- When the Mirroring Controller option is used, but automatic switch/disconnection has not been enabled

To disconnect manually, execute the mc\_ctl command in stop mode on the standby server.

Example)

```
$ mc_ctl stop -M /mcdir/inst1
```

## 7.5 Monitoring Mirroring Controller Messages

The messages that are output by Mirroring Controller are output to the system log. If the automatic switch fails, for example, an important message related to the continuation of the operation may be output, so ensure that the system log messages are monitored.



### Point

To monitor message types considered to be important, a setting must be configured in the system log beforehand.

Refer to the operating system manuals, check if the message is of a message type that is monitored to be output to the system log, and configure the setting if required.

### Display format

```
programName[processId]: messageType: messageText (messageNumber)
```

Specify the program name in the syslog\_ident parameter of the serverIdentifier.conf file.

The message types output by Mirroring Controller, their severity, and their corresponding value in the system log are shown in the table below.

Table 7.1 Message type, severity, and corresponding value in the system log

Message type	Severity	Meaning	System log
INFO	Information	Provides information that does not fall under LOG or NOTICE.	INFO
LOG		Provides information recognized as a particularly important event in tracing the operation history. (Example: Automatic switch is complete)	
NOTICE	Notice	Outputs information that takes into account the user instructions within the program in response to an executed or automatically executed process.	NOTICE
WARNING	Warning	Provides a warning, for example it will soon be impossible to maintain the multiplexing state.	WARNING
ERROR	Error	Reports that an error other than FATAL or PANIC has occurred.	ERROR
FATAL		Reports that an abnormality was detected in one of the multiplexed database systems requiring recovery of this system, and also the content and cause of the abnormality.	CRIT
PANIC		Reports that an abnormality was detected in all multiplexed database systems requiring immediate recovery of this system, and also the content and cause of the abnormality.	ALERT

The message severity has the following meanings:

- Information

Informational status. A message that was reported by the system is displayed. No action is required.

- Notice

Informational status, but a message that should be noted is displayed. If necessary, take the actions described in the "Action" section of the message.

- Warning

No error has occurred, but the user is requested to check, and take action. Take the actions described in the "Action" section of the message.

- Error

An error has occurred. Take the actions described in the "Action" section of the message.

## 7.6 Server Maintenance

---

To perform maintenance tasks such as periodic server inspections and the application of updates for software products including the operating system, you must perform a planned stop of the server, and then perform the maintenance.

### 7.6.1 Rolling Updates

---

In database multiplexing mode, rolling updates, that perform the maintenance for the servers that comprise the cluster system, can be performed while jobs continue.

First, perform the maintenance for the standby server, and then switch the standby server to the primary server. Then, perform the maintenance for the original primary server that was switched to the standby server. This enables maintenance to be performed while jobs continue.

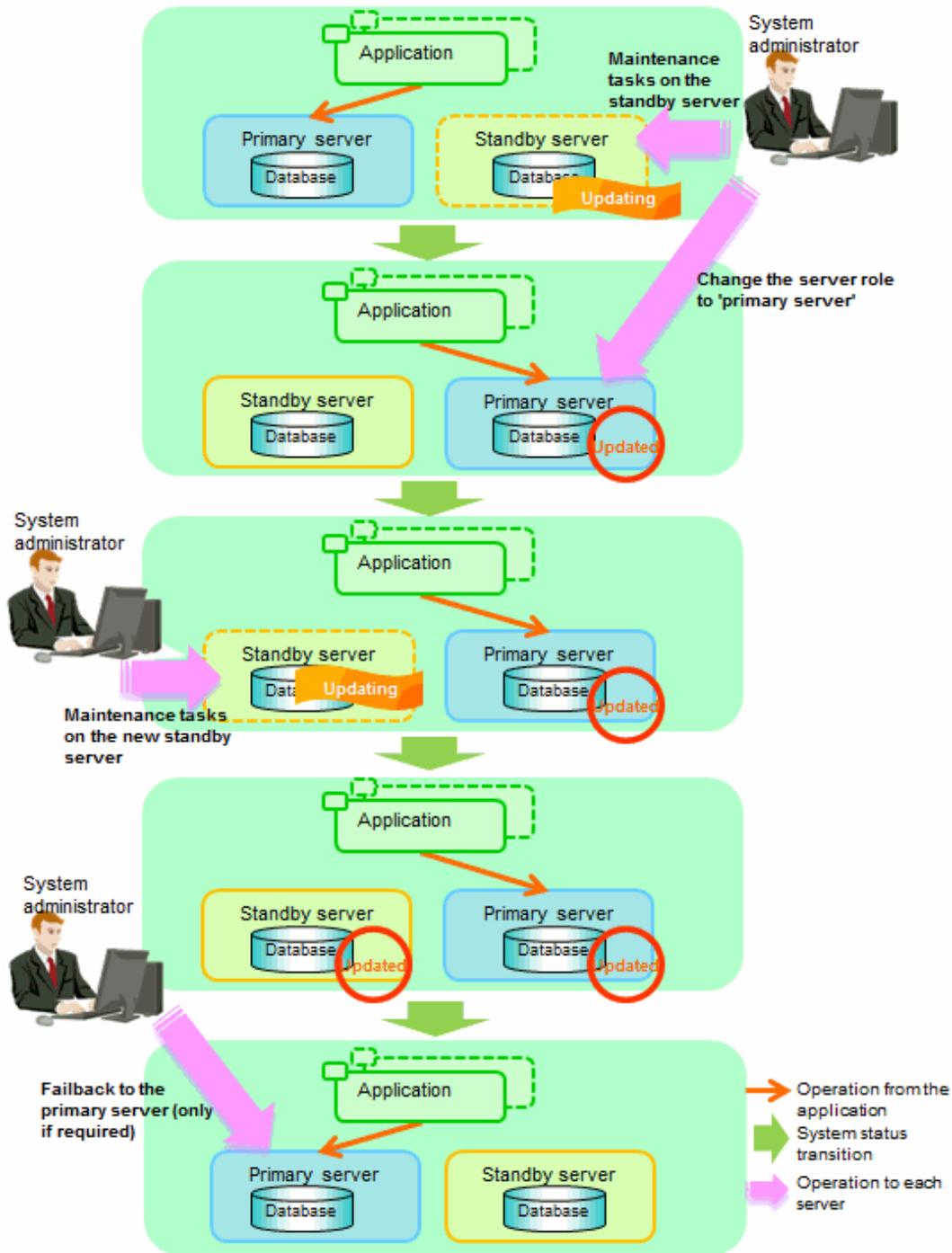


See

.....  
If the downtime due to the maintenance of the standby server is expected to be long, refer to "Standby server downtime" in "[7.7.1 Changes Required when the Standby Server is Stopped](#)".  
.....

The flow of a rolling update is shown below.

Figure 7.1 Performing a Rolling Update



### Standby server maintenance tasks

1. To perform the maintenance on the standby server, stop Mirroring Controller.

If you have set automatic start and stop beforehand and you want to turn the server off for maintenance, the server will stop automatically when the operating system is shut down. Accordingly, this step is not necessary.

Example)

```
$ mc_ctl stop -M /mdir/inst1
```

2. Ensure that Mirroring Controller has completely stopped.

If the multiplexed instances and Mirroring Controller have been configured on the standby server to start and stop automatically when the operating system of the database server is started or stopped, cancel the setting to start and stop automatically.



.....  
Refer to "6.10 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for information on how to configure the multiplexed instances and Mirroring Controller to start and stop automatically when the operating system of the database server start and stops.  
.....

#### If using an rc script

As the OS superuser, execute the `chkconfig` command to disable automatic start and stop.

In the example below, the shell script for automatic start and stop was created as "rc\_mc\_inst1" and is disabled for run levels 3 and 5.

Example)

```
# chkconfig --level 35 rc_mc_inst1 off
```

#### If using systemd

As the OS superuser, execute the `systemctl` command to disable automatic start and stop.

The example below disables automatic start and stop of "mcoi\_inst1.service".

Example)

```
# systemctl disable mcoi_inst1.service
```

3. Perform maintenance tasks.
4. Create a copy of the primary server instance on the standby server.

Execute the `mc_ctl` command in setup mode to create data in the standby server by synchronizing with the primary server.

Example)

```
$ mc_ctl setup -M /mcdir/inst1 -r standby -U userName -h primaryServerHostName -p  
primaryServerPortNumber --standby-name=standbyServerName
```



.....  
The procedure for copying the primary server instance to the standby server is the same as the procedure for setting up the standby server.

Refer to "6.4.1 Setting up Database Multiplexing Mode on the Standby Server", and then perform the recovery.  
.....

5. Check the settings for automatic start and stop of the multiplexed instances and Mirroring Controller.

If the multiplexed instances and Mirroring Controller were configured in step 2 to not start and stop automatically when the operating system of the database server starts and stops, then change the settings back. This step can be skipped if automatic start and stop are not required.

#### If using an rc script

As the OS superuser, execute the `chkconfig` command to enable the shell script.

In the example below, the shell script for automatic start and stop was created as "rc\_mc\_inst1" and is enabled for run levels 3 and 5.

Example)

```
# chkconfig --level 35 rc_mc_inst1 on
```

### If using systemd

As the OS superuser, execute the systemctl command to enable automatic start and stop.

The example below disables automatic start and stop of "mcoi\_inst1.service".

Example)

```
# systemctl enable mcoi_inst1.service
```

### 6. Start (rebuild) Mirroring Controller on the standby server.

This operation is required when determining the maintenance tasks on the standby server.

#### When the Mirroring Controller option is used

As the instance administrator user, execute the mc\_ctl command in start mode with the -f option specified. This enables automatic switch and disconnection.

If you start Mirroring Controller and the instance without specifying the -f option, automatic switch and disconnection will not be enabled. To enable both, start Mirroring Controller and then execute the mc\_ctl command in enable-failover mode, or restart Mirroring Controller with the -f option specified.

Example)

```
$ mc_ctl start -M /mcdire/inst1 -f
```

#### When the Mirroring Controller option is not used

As the instance administrator user, execute the mc\_ctl command in start mode.

Example)

```
$ mc_ctl start -M /mcdire/inst1
```

## Switching to the primary server

To perform the maintenance on the primary server, execute the mc\_ctl command in the switch mode on the primary server or the standby server.

Example)

```
$ mc_ctl switch -M /mcdire/inst1
```

When the switch is complete, the synchronous\_standby\_names parameter in the postgresql.conf file of the new primary server will be commented as follows:

Example)

```
#synchronous_standby_names = 'primary'
```

## New standby server maintenance tasks

### 1. Stop the Mirroring Controller.

On the new standby server (the primary server before the switch), execute the mc\_ctl command in stop mode.

Example)

```
$ mc_ctl stop -M /mcdire/inst1
```

### 2. Ensure that Mirroring Controller has completely stopped.

If the multiplexed instances and Mirroring Controller have been configured on the new standby server to start and stop automatically when the operating system of the database server is started or stopped, cancel the setting to start and stop automatically now.



See

Refer to "6.10 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances" for information on how to configure the multiplexed instances and Mirroring Controller to start and stop automatically when the operating system of the database server starts and stops.

#### If using an rc script

As the OS superuser, execute the chkconfig command to disable automatic start and stop.

In the example below, the shell script for automatic start and stop was created as "rc\_mc\_inst1" and is disabled for run levels 3 and 5.

Example)

```
# chkconfig --level 35 rc_mc_inst1 off
```

#### If using systemd

As the OS superuser, execute the systemctl command to disable automatic start and stop.

The example below disables automatic start and stop of "mcoi\_inst1.service".

Example)

```
# systemctl disable mcoi_inst1.service
```

3. Perform the maintenance on the new standby server that was stopped.

4. Create a copy of the new primary server instance on the new standby server.

Execute the mc\_ctl command in setup mode to create data in the new standby server by synchronizing with the new primary server.

Example)

```
$ mc_ctl setup -M /mdir/inst1 -r standby -U userName -h newPrimaryServerHostName -p  
newPrimaryServerPortNumber --standby-name=newStandbyServerName
```



See

The procedure for copying the primary server instance to the standby server is the same as the procedure for setting up the standby server.

Refer to "6.4.1 Setting up Database Multiplexing Mode on the Standby Server", and then perform the recovery.

5. Check the settings for automatic start and stop of the multiplexed instances and Mirroring Controller.

If the multiplexed instances and Mirroring Controller were configured in step 2 to not start and stop automatically when the operating system of the database server starts and stops, then change the settings back. This step can be skipped if automatic start and stop are not required.

#### If using an rc script

As the OS superuser, execute the chkconfig command to enable the shell script.

In the example below, the shell script for automatic start and stop was created as "rc\_mc\_inst1" and is enabled for run levels 3 and 5.

Example)

```
# chkconfig --level 35 rc_mc_inst1 on
```

#### If using systemd

As the OS superuser, execute the systemctl command to enable automatic start and stop.

The example below disables automatic start and stop of "mcoi\_inst1.service".

Example)

```
# systemctl enable mcoi_inst1.service
```

6. After the maintenance is complete, edit the following parameters in the postgresql.conf file of the standby server as required.

Copying an instance results in the value of the synchronous\_standby\_names parameter becoming the specified value on the primary server. Therefore, correct it to the specified value on the standby server. If the parameter was commented out, then you must uncomment it.

7. On the standby server, start (rebuild) Mirroring Controller.

When the Mirroring Controller option is used

As the instance administrator user, execute the mc\_ctl command in start mode with the -w and -f options specified. This enables automatic switch and disconnection.

If you start Mirroring Controller and the instance without specifying the -f option, automatic switch and disconnection will not be enabled. To enable both, start Mirroring Controller and then execute the mc\_ctl command in enable-failover mode, or restart Mirroring Controller with the -f option specified.

Example)

```
$ mc_ctl start -M /mdir/inst1 -w -f
```

When the Mirroring Controller option is not used

As the instance administrator user, execute the mc\_ctl command in start mode with the -w option specified.

Example)

```
$ mc_ctl start -M /mdir/inst1 -w
```

## Failback of the Primary Server

Revert the primary server and standby server to the original server configuration. Do this to execute the main job on the previous primary server. Refer to "[8.1.1.3 Failback of the Primary Server](#)" for details.



### Note

Obtain a backup as soon as this task is complete.

## 7.6.2 Stopping for Maintenance

Perform this procedure to stop all servers for periodic inspections, for example. On the server on which Mirroring Controller is running, execute the mc\_ctl command in stop mode to stop the instance and Mirroring Controller.

Example)

```
$ mc_ctl stop -M /mdir/inst1 -a
```

## 7.7 Changes in Operation

The following changes in operation may be required:

- Changes required when the standby server is stopped
- Changing from single server mode to database multiplexing mode
- Changing from database multiplexing mode to single server mode

## 7.7.1 Changes Required when the Standby Server is Stopped

---

### Operation when the standby server is stopped

Before performing maintenance for the primary server instance when the standby server has been stopped, comment out the `synchronous_standby_names` parameter in the `postgresql.conf` file of the primary server, and then execute the `pg_ctl` command in reload mode.

If this operation is not performed, operations performed on the primary server for the instance will remain in a wait state.



See

.....  
Refer to "pg\_ctl" in Reference for information on the command.  
.....

### Standby server downtime

If you specified the `synchronous_standby_names` parameter of the `postgresql.conf` file and then the standby server instance is stopped, consider the points below.

- The `replication_timeout` parameter in the `postgresql.conf` file

If the standby server is stopped after the timeout set in this parameter was exceeded, an error stating that the transaction log could not be received may be output to the standby server system log, and all transaction logs that should be transferred to the standby server may be lost.

- The `wal_keep_segments` parameter in the `postgresql.conf` file

If a transaction log that exceeds the value set in this parameter was generated while the standby server was stopped, the transaction log may be deleted.

Additionally, setting this parameter requires consideration regarding stabilization of the database multiplexing mode. Refer to "[6.9.1 Tuning to Stabilize the Database Multiplexing Mode](#)" for details.



Note

.....  
The standby server must be rebuilt in both cases above.

Take the action advised in the recovery operation that starts from "[8.1.1.1.3 Identify cause of error and perform recovery](#)" through to "[8.1.1.2 Rebuild the Standby Server](#)".  
.....

## 7.7.2 Changing from Single Server Mode to Database Multiplexing Mode

---

The procedure for switching single server mode to database multiplexing mode for the purposes of high reliability and load distribution of the system is explained below.

This procedure is equivalent to the setup procedure explained in "[Chapter 6 Setting up Database Multiplexing Mode](#)".



Note

.....  
**If the data storage destination directory name is not comprised of ASCII characters excluding spaces**

Stop the application job and then migrate to a directory with a name that uses only ASCII characters excluding spaces:

1. Stop the database instance on the primary server.
  2. Change the name of the data storage destination directory to one that uses only ASCII characters excluding spaces.
- .....



See

When encrypting the storage data, refer to "Database Multiplexing Mode" in the Operation Guide, and then perform the setup for encryption on the primary and standby servers.

1. Install on the standby server

Install Symfoware Server on the server to be started as the standby server.

Refer to "Installation" in the Installation and Setup Guide for Server for information on how to install Symfoware Server.

Use ASCII characters excluding spaces in the data storage destination directory.

2. Stop the application jobs

Stop the application jobs to be connected to the primary server.

3. Change the primary server settings

To allow connections from the server to be started as the standby server, configure the settings in step 2 and thereafter of "[6.3.2 Creating, Setting, and Registering the Primary Server Instance](#)" on the primary server.

4. Set up database multiplexing mode on the standby server

Refer to "[6.4.1 Setting up Database Multiplexing Mode on the Standby Server](#)" for details.

5. Create the standby server instance and start it

Refer to "[6.4.2 Creating, Setting, and Registering the Standby Server Instance](#)" for details.

After the above steps are completed, refer to the remaining explanations in "[Chapter 6 Setting up Database Multiplexing Mode](#)" and ensure that the required settings and operations are completed.

## 7.7.3 Changing from Database Multiplexing Mode to Single Server Mode

The procedure for stopping database multiplexing mode and changing to single server mode is explained below.

1. Determine the server for which the instance is to be stopped, and switch this server

Determine the server that is to be excluded as the database multiplexing mode target, and for which the instance is to be stopped.

If the server for which the instance is to be stopped is the primary server, execute the `mc_ctl` command in the switch mode to switch the standby server to the primary server.

The standby server after the switch is complete will be the server for which the instance is to be stopped.

If the server for which the instance is to be stopped is the standby server, there is no need to perform the switch operation.

Example)

```
$ mc_ctl switch -M /mcdire/inst1
```

2. Stop Mirroring Controller and the instance, and delete the file resources

On the server that was determined in step 1, execute the `mc_ctl` command in stop mode to stop Mirroring Controller and the instance.

Example)

```
$ mc_ctl stop -M /mcdire/inst1
```

Then, delete the following file resources:

- Data storage destination directory
- Mirroring Controller management directory

Example)

```
$ rm -rf /database/inst1
$ rm -rf /mdir/inst1
```



See

Refer to "Security-Related Notes" in the Operation Guide for details on deleting the data securely.

### 3. Stop Mirroring Controller on the primary server

Execute the mc\_ctl command in stop mode on the primary server.

Example)

```
$ mc_ctl stop -M /mdir/inst1
```

### 4. Delete the database multiplexing mode settings that were configured for the primary server instance.

Each parameter and resource is shown in the table below.

- Database multiplexing parameters in postgresql.conf and recovery.conf
- Primary server settings

File	Parameter	Action
postgresql.conf	wal_level	Delete the "hot_standby" value that was specified.
	max_wal_senders	Revert the value to the one set before database multiplexing mode was set.
	synchronous_standby_names	Delete.
	replication_timeout	
	listen_addresses	Revert the value to the one set before database multiplexing mode was set.
	max_connections	

- Standby server settings

File	Parameter	Action
postgresql.conf	synchronous_standby_names	Delete.
	hot_standby	
	wal_receiver_status_interval	
recovery.conf	standby_mode	Delete. Note that this file is created automatically when the mc_ctl command is executed in setup mode.
	primary_conninfo	

- Mirroring Controller management directory

Additionally, if the backup operation was performed, delete the following resources:

- Mirroring Controller management directory backup data obtained in database multiplexing mode
- Instance backup data obtained in database multiplexing mode

After these actions are performed, ensure that the backup data is collected when starting the single operation.



Note

**If using together with WebAdmin**

Instances that run in database multiplexing mode cannot be accessed from WebAdmin, even if the database multiplexing mode is stopped and operations are changed to single server mode. Using WebAdmin, create a new instance.

Refer to "Setup" in the Installation and Setup Guide for Server for information on creating instances.



## See

- Refer to "Security-Related Notes" in the Operation Guide for details on deleting the data securely.
- Refer to "[6.11 Backup Operation](#)" for details on the backup operation.
- Refer to "[Appendix B Parameters](#)" for details on the postgresql.conf file parameters.

## Chapter 8 Action Required when an Error Occurs in Database Multiplexing Mode

This chapter describes the action required if an error occurs in database multiplexing mode.

In database multiplexing mode, when an error is detected, the switch or disconnection of the standby server is performed automatically, so that only the primary server starts degrading. In this case, the recovery tasks will be required for the standby server on which the switch or disconnection was performed.

Other possible cases are as follows:

- When automatic switch fails
- When automatic disconnection fails
- When all servers or instances were stopped

### 8.1 Action Required when Server Degradation Occurs

If the server has started degrading, the recovery tasks will vary depending on whether the cause was the switch (failover or switchover), or the disconnection.

Execute the `mc_ctl` command in status mode, or refer to the system log, and check if the cause of the server degradation was the switch or the disconnection.

In the example below, the `mc_ctl` command is executed in status mode.

If a switch has occurred, "switched" (the switch is complete and the server is in a degrading state) is displayed for "mirroring status".

Example)

```
$ mc_ctl status -M /mdir/inst1
mirroring status
-----
switched
:
```

If a disconnection has occurred, "not-switchable" (disconnection was performed so the server cannot be switched) is displayed for "mirroring status".

Example)

```
$ mc_ctl status -M /mdir/inst1
mirroring status
-----
not-switchable
:
```

#### 8.1.1 Operations when the Server has Started Degrading after a Switch has Occurred

This section explains the operations when the server has started degrading after a switch has occurred.



- After a switch has occurred as a result of an abnormality on the primary server, the database will not have a multiplexed configuration until the standby server is rebuilt. Remove the cause of the error as quickly as possible, and then rebuild the standby server.

- If the reference job was executed on the standby server, and the servers are switched because an error occurred on the primary server, the load is concentrated on the new primary server. Accordingly, pause the reference job on the original standby server, rebuild the original primary server as the new standby server, and then resume the reference job for the new standby server.
- If the instance on the new primary server is stopped before the original primary server where the error occurred is rebuilt as the new standby server, a split brain occurs at startup from the instance on the original primary server. Therefore, start the instance on the new primary server before rebuilding the standby server.

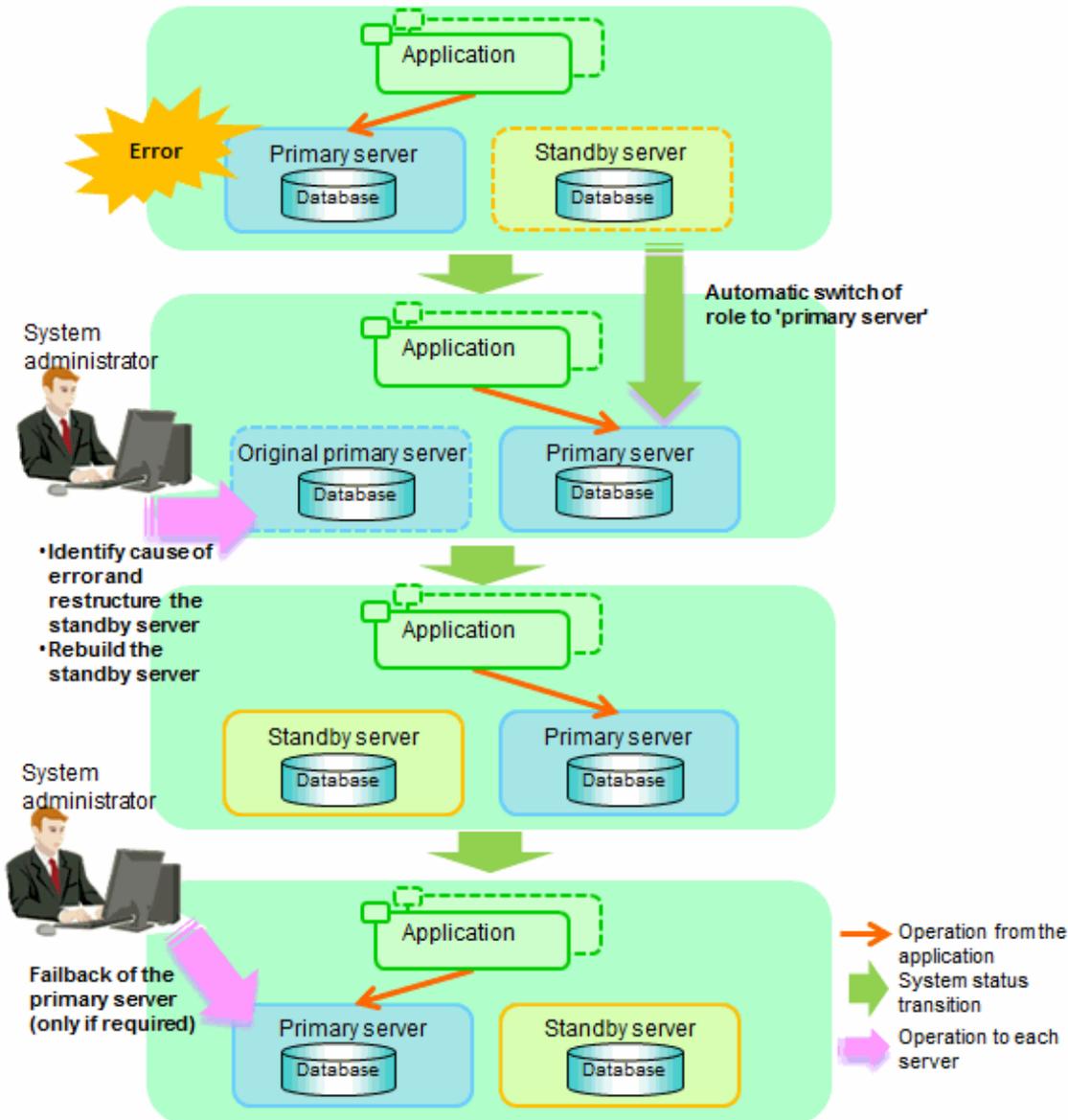
.....

If the switch occurred and the server has started degrading, perform the following operations to recover the standby server and revert it to its original state:

- [Identify Cause of Error and Restructure the Standby Server](#)
- [Rebuild the Standby Server](#)
- [Failback of the Primary Server](#) (only if required)

The flow of these operations is shown in the figure below.

Figure 8.1 Flow of operations



## 8.1.1.1 Identify Cause of Error and Restructure the Standby Server

Perform the recovery according to the following procedure:

1. [Stop Mirroring Controller](#)
2. [Recovery of the Mirroring Controller management directory](#)
3. [Identify cause of error and perform recovery](#)

### 8.1.1.1.1 Stop Mirroring Controller

Execute the `mc_ctl` command in stop mode for the original primary server on which the error occurred.

Example)

```
$ mc_ctl stop -M /mcdire/inst1
```

This also stops the instance that is required to perform the recovery.



If the instance does not stop, refer to "Actions in Response to Failure to Stop an Instance" in the Operation Guide, and then stop the instance.

Then, specify the `-e` option in the above command to forcibly stop Mirroring Controller.

### 8.1.1.1.2 Recovery of the Mirroring Controller management directory

Copy the files in the Mirroring Controller management directory from the backup data, and then perform the recovery.

### 8.1.1.1.3 Identify cause of error and perform recovery

Refer to the system log of the primary server and the standby server to identify the cause of the error, and then perform recovery.

Execute the `mc_ctl` command in setup mode to perform recovery by synchronizing data in the original primary server with the new primary server.

Example)

```
$ mc_ctl setup -M /mcdire/inst1 -r standby -U userName -h primaryServerHostName -p  
primaryServerPortNumber --standby-name=standbyServerName
```



This recovery procedure is the same as the procedure for setting up the standby server.

Refer to "[6.4.2 Creating, Setting, and Registering the Standby Server Instance](#)", and then perform the recovery.

## 8.1.1.2 Rebuild the Standby Server

The starting of the recovered original primary server as the standby server is referred to as the "standby server rebuild".

On the original primary server, start Mirroring Controller and the instance.

When the Mirroring Controller option is used

As the instance administrator user, execute the `mc_ctl` command in start mode with the `-f` option specified. This enables automatic switch and disconnection.

If you start Mirroring Controller and the instance without specifying the `-f` option, automatic switch and disconnection will not be enabled. To enable both, start Mirroring Controller and then execute the `mc_ctl` command in enable-failover mode, or restart Mirroring Controller with the `-f` option specified.

Example)

```
$ mc_ctl start -M /mcdire/inst1 -f
```

When the Mirroring Controller option is not used

As the instance administrator user, execute the mc\_ctl command in start mode.

Example)

```
$ mc_ctl start -M /mcdire/inst1
```

### 8.1.1.3 Failback of the Primary Server

To revert the primary server and standby server to the original server configuration after rebuilding the standby server, perform failback for the primary server.

Do this to execute the main job on the previous primary server.

Perform the following procedure:

#### 1. Failback of the primary server

Execute the mc\_ctl command in switch mode on the primary server or the standby server.

Example)

```
$ mc_ctl switch -M /mcdire/inst1
```

After executing the mc\_ctl command in switch mode, the status will be as follows:

Example)

```
$ mc_ctl status -M /mcdire/inst1
mirroring status
-----
switched
server_id  host_role                host          host_status  db_proc_status  disk_status
-----
nd1        primary                   192.0.2.100  normal      abnormal(postmaster) normal
nd2        none(inactivated primary) 192.0.2.110  normal      abnormal(postmaster) normal
```

#### 2. Stop the original primary server

On the original primary server, execute the mc\_ctl command in stop mode to stop Mirroring Controller and the instance.

Example)

```
$ mc_ctl stop -M /mcdire/inst1
```

#### 3. Create a copy of the new primary server instance in the original primary server (new standby server)

Execute the mc\_ctl command in setup mode to create data in the new standby server by synchronizing with the new primary server.

Example)

```
$ mc_ctl setup -M /mcdire/inst1 -r standby -U userName -h newPrimaryServerHostName -p
newPrimaryServerPortNumber --standby-name=newStandbyServerName
```



The procedure for copying the new primary server instance to the new standby server is the same as the procedure for setting up the new standby server.

Refer to "6.4.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.

#### 4. Rebuild the standby server

On the standby server, start Mirroring Controller and the instance.

When the Mirroring Controller option is used

As the instance administrator user, execute the `mc_ctl` command in start mode with the `-w` and `-f` options specified. This enables automatic switch and disconnection.

If you start Mirroring Controller and the instance without specifying the `-f` option, automatic switch and disconnection will not be enabled. To enable both, start Mirroring Controller and then execute the `mc_ctl` command in enable-failover mode, or restart Mirroring Controller with the `-f` option specified.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -w -f
```

When the Mirroring Controller option is not used

As the instance administrator user, execute the `mc_ctl` command in start mode with the `-w` option specified.

Example)

```
$ mc_ctl start -M /mcdir/inst1 -w
```

## 8.1.2 Operations when the Server has Started Degrading after a Disconnection has Occurred

---

This section explains the operations when the server has started degrading after a disconnection has occurred.



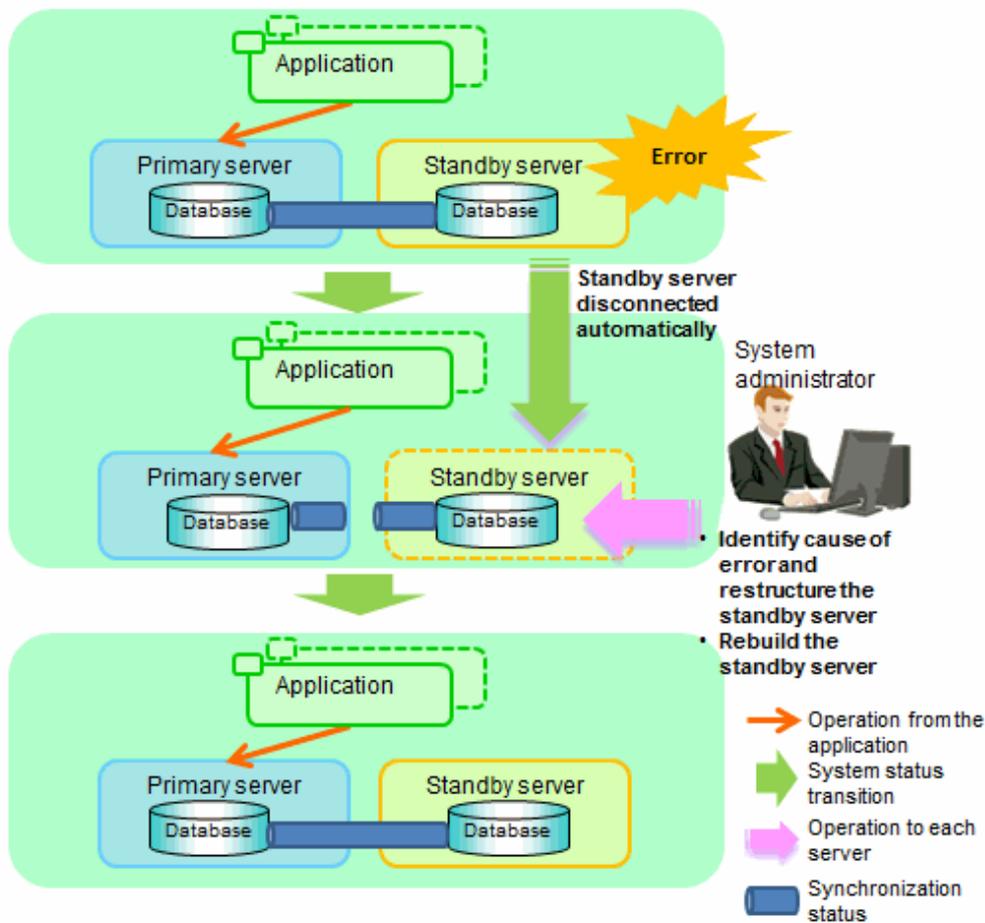
After a disconnection has occurred as a result of an abnormality on the standby server, the database will not have a multiplexed configuration until the standby server is rebuilt. Remove the cause of the error as quickly as possible, and then rebuild the standby server.

If the disconnection occurred and the server has started degrading, perform the following operations to recover the standby server and revert it to its original state:

- [Identify Cause of Error and Restructure the Standby Server](#)
- [Rebuild the Standby Server](#)

The flow of these operations is shown in the figure below.

Figure 8.2 Flow of operations



### 8.1.2.1 Identify Cause of Error and Restructure the Standby Server

Perform the recovery according to the following procedure:

1. [Stop Mirroring Controller](#)
2. [Recovery of the Mirroring Controller management directory](#)
3. [Identify cause of error and perform recovery](#)

#### 8.1.2.1.1 Stop Mirroring Controller

Execute the `mc_ctl` command in stop mode for the standby server on which the error occurred.

Example)

```
$ mc_ctl stop -M /mdir/inst1
```

This also stops the instance that is required to perform the recovery.



#### Note

If the instance does not stop, refer to "Actions in Response to Failure to Stop an Instance" in the Operation Guide, and then stop the instance.

Then, specify the `-e` option in the above command to forcibly stop Mirroring Controller.

### 8.1.2.1.2 Recovery of the Mirroring Controller management directory

Copy the files in the Mirroring Controller management directory from the backup data, and then perform the recovery.

### 8.1.2.1.3 Identify cause of error and perform recovery

Refer to the system logs of the primary server and the standby server to identify the cause of the error, and then perform recovery.

Execute the `mc_ctl` command in setup mode to perform recovery by synchronizing data in the primary server with the standby server.

Example)

```
$ mc_ctl setup -M /mcdire/inst1 -r standby -U userName -h primaryServerHostName -p  
primaryServerPortNumber --standby-name=standbyServerName
```



See

.....  
This recovery procedure is the same as the procedure for setting up the standby server.

Refer to "6.4.2 Creating, Setting, and Registering the Standby Server Instance", and then perform the recovery.  
.....

### 8.1.2.2 Rebuild the Standby Server

Start the Mirroring Controller and the instance of the standby server, and rebuild the standby server.

When the Mirroring Controller option is used

As the instance administrator user, execute the `mc_ctl` command in start mode with the `-f` option specified. This enables automatic switch and disconnection.

If you start Mirroring Controller and the instance without specifying the `-f` option, automatic switch and disconnection will not be enabled. To enable both, start Mirroring Controller and then execute the `mc_ctl` command in enable-failover mode, or restart Mirroring Controller with the `-f` option specified.

Example)

```
$ mc_ctl start -M /mcdire/inst1 -f
```

When the Mirroring Controller option is not used

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
$ mc_ctl start -M /mcdire/inst1
```

## 8.2 Action Required when Automatic Switch Fails (when the Mirroring Controller Option is Used)

If the system behavior is unstable, for example there are insufficient temporary system resources, the Mirroring Controller automatic switch may fail.

Perform the switch manually using one of the following methods:

- Execute the `mc_ctl` command in switch mode

Execute the `mc_ctl` command in switch mode on the primary server or the standby server.

Example)

```
$ mc_ctl switch -M /mcdire/inst1
```

## Note

If the server fails during execution of the `mc_ctl` command, automatic switch, or automatic disconnection, or if communication between the primary server and the standby server is lost, a message indicating that the `mc_ctl` command is being executed may be output even though the command is not running elsewhere, and the command may end abnormally.

In this case, switch the server in accordance with the "Execute the `mc_ctl` command in stop mode" procedure below.

- Execute the `mc_ctl` command in stop mode

Execute the `mc_ctl` command in stop mode on the server on which the error occurred.

Example)

```
$ mc_ctl stop -M /mcdire/inst1
```

If the instance does not stop, refer to "Actions in Response to Failure to Stop an Instance" in the Operation Guide, and then stop the instance.

Then, specify the `-e` option in the `mc_ctl` command to forcibly stop Mirroring Controller.

Example)

```
$ mc_ctl stop -M /mcdire/inst1 -e
```

## See

### Recovery to database multiplexing mode

Refer to "[8.1.1.2 Rebuild the Standby Server](#)" and "[8.1.1.3 Failback of the Primary Server](#)" for information on recovery to database multiplexing mode.

## 8.3 Action Required when Automatic Disconnection Fails (when the Mirroring Controller Option is Used)

If the system behavior is unstable, for example there are insufficient system resources such as available memory or free disk space, automatic disconnection using Mirroring Controller may not be possible.

In this case, perform the following procedure to disconnect the primary server manually.

1. On the standby server, execute the `mc_ctl` command in stop mode with the `-e` option specified.

Example)

```
$ mc_ctl stop -M /mcdire/inst1 -e
```

2. On the primary server, comment out the `synchronous_standby_names` parameter in the `postgresql.conf` file.

Example)

```
:  
# synchronous_standby_names = 'standby1 '
```

3. Execute the `pg_ctl` command in reload mode on the primary server.

Example)

```
$ pg_ctl reload -D /database/inst1
```

 See

.....  
Refer to "pg\_ctl" in Reference for information on the command.  
.....

 See

.....  
**Recovery to database multiplexing mode**

Refer to "[8.1.2.2 Rebuild the Standby Server](#)" for information on recovery to database multiplexing mode.  
.....

## **8.4 Action Required when All Servers or Instances Stopped**

---

This section explains what happens when all servers or instances on the server have stopped, so jobs cannot continue.

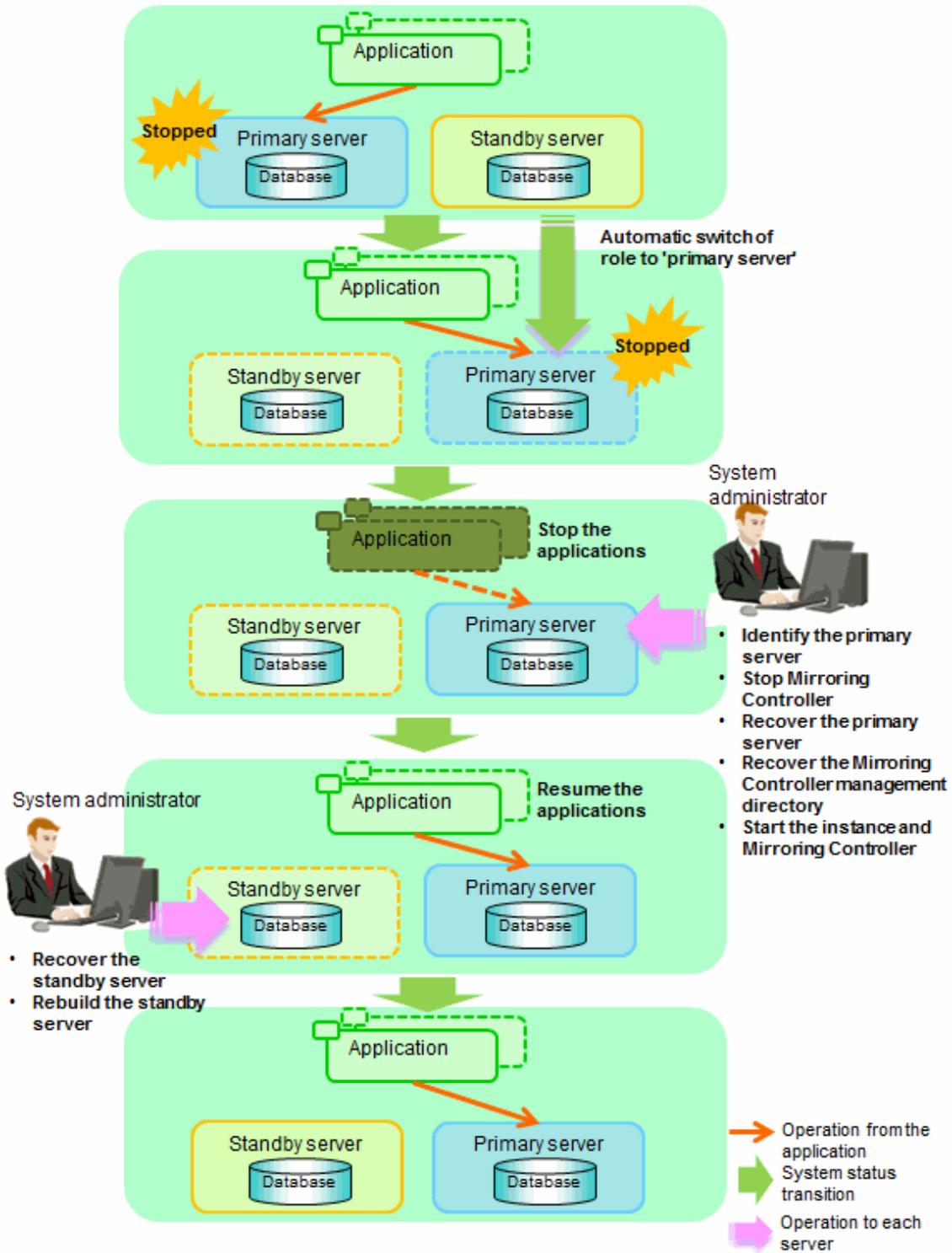
 See

.....  
**Recovery to database multiplexing mode**

Refer to "[8.1.1.2 Rebuild the Standby Server](#)" and "[8.1.1.3 Failback of the Primary Server](#)" for information on recovery to database multiplexing mode.  
.....

The flow of these recovery operations is shown in the figure below.

Figure 8.3 Flow of operations



Perform the following procedure:

1. Stop the applications  
Stop the applications from running.
2. Identify the primary server

Use one of the following methods to identify the primary server that was running before the servers or instances stopped:

- Refer to the system log on each server and identify the server where the following message was output.

Message:

```
MirroringControllerOpen[30017]: LOG: promotion processing completed (MCA00062)
```

- On each server, execute the `mc_ctl` command in status mode to search the servers for which "none(inactivated primary)" is displayed.

### 3. Stop Mirroring Controller on the primary server

Execute the `mc_ctl` command in stop mode on the primary server.

Example)

```
$ mc_ctl stop -M /mdir/inst1
```



#### Forcibly stopping Mirroring Controller

If Mirroring Controller does not stop, specify the `-e` option in the stop mode of the `mc_ctl` command and then execute the command.

Example)

```
$ mc_ctl stop -M /mdir/inst1 -e
```

### 4. Recover the primary server

First, refer to "Actions when an Error Occurs" in the Operation Guide, and then identify the cause of the error and perform recovery.

Next, recover the primary server using the recovery method that uses the `pgx_rcvall` command based on the backup data.

If the backup operation was performed using the `pgx_dmpall` command based on the instructions in [6.11.2 Database Backup Operation](#), perform the following procedure for the recovery:

- Perform the following operations on both the primary server and the standby server, and check the server containing the backup data and the archive log that show the latest date.
  - Execute the `pgx_rcvall` command with the `-l` option specified and identify the backup data that shows the latest date.
  - Identify the archive log that shows the latest date, as shown below.

Example)

```
$ ls -ltr backupDataStorageDestinationDirectory/*_xlog
```

- If the latest backup data exists on the standby server, copy (\*1) the backup data and overwrite (\*2) it to each backup storage destination directory on the primary server.
- If the latest archive log exists on the standby server, copy (\*1) the archive log and overwrite (\*2) it to the backup storage destination directory on the primary server.
- Execute the `pgx_rcvall` command on the primary server, specifying the backup storage destination directory you used in step 3.



\*1: The backup data may contain a symbolic link, so copy the backup data so that the symbolic link is not converted to an ordinary file (with the `tar` command, for example).

\*2: If you can save a copy of the backup storage destination directory, do so without overwriting it.



See

Refer to "Actions when an Error Occurs" in the Operation Guide for information on the `pgx_rcvall` command.

5. Recover the Mirroring Controller management directory

Copy the files in the Mirroring Controller management directory from the backup data, and then perform the recovery.

6. Start the primary server instance and Mirroring Controller

When the Mirroring Controller option is used

As the instance administrator user, execute the `mc_ctl` command in start mode with the `-f` option specified. This enables automatic switch and disconnection.

If you start Mirroring Controller and the instance without specifying the `-f` option, automatic switch and disconnection will not be enabled. To enable both, start Mirroring Controller and then execute the `mc_ctl` command in enable-failover mode, or restart Mirroring Controller with the `-f` option specified.

Example)

```
$ mc_ctl start -M /mdir/inst1 -f
```

When the Mirroring Controller option is not used

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

```
$ mc_ctl start -M /mdir/inst1
```

7. Resume applications

Resume the applications.

8. Stop Mirroring Controller on the standby server

Execute the `mc_ctl` command in stop mode on the standby server.

Example)

```
$ mc_ctl stop -M /mdir/inst1
```

9. Recover the standby server

Refer to "[6.4.2 Creating, Setting, and Registering the Standby Server Instance](#)", and then recover (set up) the standby server from the primary server.

10. Rebuild the standby server

On the standby server, start Mirroring Controller and the instance.

When the Mirroring Controller option is used

As the instance administrator user, execute the `mc_ctl` command in start mode with the `-f` option specified. This enables automatic switch and disconnection.

If you start Mirroring Controller and the instance without specifying the `-f` option, automatic switch and disconnection will not be enabled. To enable both, start Mirroring Controller and then execute the `mc_ctl` command in enable-failover mode, or restart Mirroring Controller with the `-f` option specified.

Example)

```
$ mc_ctl start -M /mdir/inst1 -f
```

When the Mirroring Controller option is not used

As the instance administrator user, execute the `mc_ctl` command in start mode.

Example)

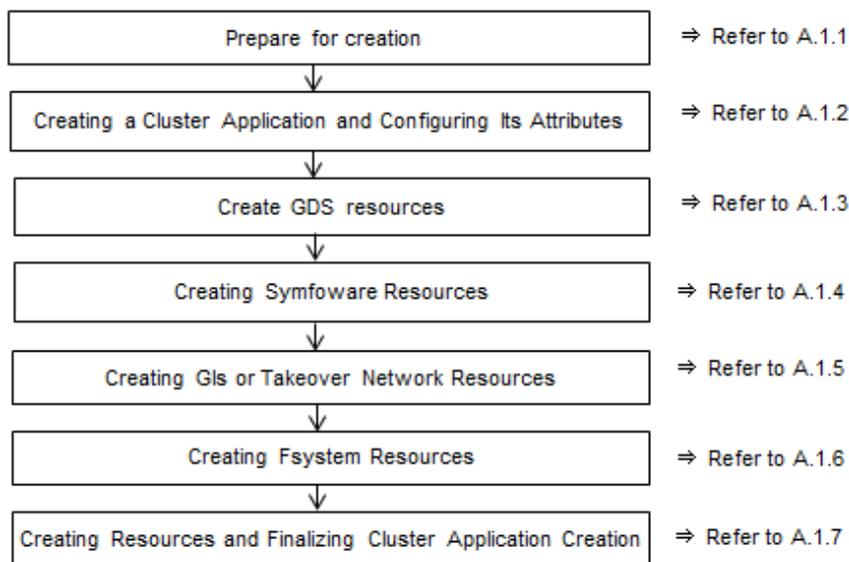
```
$ mc_ctl start -M /mcdir/inst1
```

# Appendix A Creating Resources and Creating/Modifying Cluster Applications

This appendix explains how to create and modify cluster applications that include Symfoware resources in PRIMECLUSTER.

## A.1 Creating Resources and Cluster Applications

Use the following procedure to create resources and a cluster application:



### A.1.1 Preparing for Creation

1. Execute the `hvgdsetup` command for all shared disk classes to be used by all file systems to be registered in the cluster application.

```
# /usr/opt/reliant/bin/hvgdsetup -a class0001
```

2. From the Web-Based Admin View, stop RMS on all nodes.

3. Execute the `hvw` command.

```
# /opt/SMAW/SMAWRrms/bin/hvw
```



See

Refer to the PRIMECLUSTER Installation and Administration Guide for information on the `hvgdsetup` command.

### A.1.2 Creating a Cluster Application and Configuring Its Attributes

1. Select [Application-Create].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP
 2) QUIT
 3) Application-Create
 4) Application-Edit
 5) Application-Remove
 6) Application-Clone
10) Configuration-Remove
11) Configuration-Freeze
12) Configuration-Thaw
13) Configuration-Edit-Global-Settings
14) Configuration-Consistency-Report
15) Configuration-ScriptExecution
```

```

7) Configuration-Generate          16) RMS-CreateMachine
8) Configuration-Activate          17) RMS-RemoveMachine
9) Configuration-Copy
Choose an action: 3

```

2. Select [STANDBY].

```

Creation: Application type selection menu:
1) HELP                          7) SCALABLE
2) QUIT                          8) STANDBY
3) RETURN
4) OPTIONS
5) DEMO
6) GENERIC
Application Type: 8

```

3. Select [Machines+Basics].

Specify any application name that is within the scope of the PRIMECLUSTER constraints.

```

Settings of turnkey wizard "STANDBY" (APPL:not yet consistent)
1) HELP                          4) REMOVE+EXIT
2) NO-SAVE+EXIT                 5) ApplicationName=APPL
3) SAVE+EXIT                   6) Machines+Basics(-)
Choose the setting to process: 6

```

4. Select [AdditionalMachine].

```

Machines+Basics (appl:consistent)
1) HELP                          14) (AutoStartUp=no)
2) -                             15) (AutoSwitchOver=No)
3) SAVE+EXIT                     16) (PreserveState=no)
4) REMOVE+EXIT                   17) (PersistentFault=0)
5) AdditionalMachine             18) (ShutdownPriority=)
6) AdditionalConsole             19) (OnlinePriority=)
7) Machines[0]=pcl-vm13RMS       20) (StandbyTransitions=)
8) (PreCheckScript=)            21) (LicenseToKill=no)
9) (PreOnlineScript=)           22) (AutoBreak=yes)
10) (PostOnlineScript=)         23) (AutoBreakMaintMode=no)
11) (PreOfflineScript=)         24) (HaltFlag=no)
12) (OfflineDoneScript=)        25) (PartialCluster=0)
13) (FaultScript=)              26) (ScriptTimeout=)
Choose the setting to process: 5

```

5. Select a standby node.

```

1) HELP
2) RETURN
3) pcl-vm13RMS
4) pcl-vm14RMS
Choose a machine for this application: 4

```

6. Ensure that all nodes that make up the cluster application are displayed in [Machines]. Note that the item displayed in Machines[0] needs to be the active node, and Machines[1] to be the standby node. If these are not set correctly, modify the setting.

```

Machines+Basics (appl:consistent)
1) HELP                          15) (AutoStartUp=no)
2) -                             16) (AutoSwitchOver=No)
3) SAVE+EXIT                     17) (PreserveState=no)
4) REMOVE+EXIT                   18) (PersistentFault=0)
5) AdditionalMachine             19) (ShutdownPriority=)
6) AdditionalConsole             20) (OnlinePriority=)
7) Machines[0]=pcl-vm13RMS       21) (StandbyTransitions=)
8) Machines[1]=pcl-vm14RMS       22) (LicenseToKill=no)

```

```

9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
23) (AutoBreak=yes)
24) (AutoBreakMaintMode=no)
25) (HaltFlag=no)
26) (PartialCluster=0)
27) (ScriptTimeout=)
Choose the setting to process:

```

7. Configure attributes for cluster applications from step 8.

Refer to the PRIMECLUSTER Installation and Administration Guide for information on the attributes for cluster applications.

Configure the attributes as shown below:

Attribute	Value
AutoStartUp (Automatically starts cluster applications when the RMS is started.)	Yes
AutoSwitchOver (Automatically switches nodes when a fault occurs in cluster applications.)	- HostFailure - ResourceFailure - ShutDown
PersistentFault (Selects whether to keep the Faulted state of the cluster application after restarting the RMS.)(*1)	0 or 1
OnlinePriority (Selects whether to use the same online nodes as those used before all clusters were restarted or follow the configured order for nodes, when all clusters were restarted.)(*2)	0 or 1
StandbyTransitions (Timing to perform standby state transition.)	- ClearFaultRequest - StartUp - SwitchRequest
HaltFlag (Selects whether to terminate the node forcibly when another error occurs during the Fault process.)	Yes

\*1: If 0 is specified, the Faulted state is automatically canceled when the RMS is started. If 1 is specified, the Faulted state is kept.

\*2 If 0 is specified and the placement differs from the initial configuration due to reasons, such as switching, cluster applications are restarted using the placement from the initial configuration. If 1 is specified, cluster applications are restarted using the placement from before the restart.

8. Select [AutoStartUp].

```

Machines+Basics (appl:consistent)
1) HELP
2) -
3) SAVE+EXIT
4) REMOVE+EXIT
5) AdditionalMachine
6) AdditionalConsole
7) Machines[0]=pcl-vm13RMS
8) Machines[1]=pcl-vm14RMS
15) (AutoStartUp=no)
16) (AutoSwitchOver=No)
17) (PreserveState=no)
18) (PersistentFault=0)
19) (ShutdownPriority=)
20) (OnlinePriority=)
21) (StandbyTransitions=)
22) (LicenseToKill=no)

```

```

 9) (PreCheckScript=)                23) (AutoBreak=yes)
10) (PreOnlineScript=)              24) (AutoBreakMaintMode=no)
11) (PostOnlineScript=)            25) (HaltFlag=no)
12) (PreOfflineScript=)           26) (PartialCluster=0)
13) (OfflineDoneScript=)          27) (ScriptTimeout=)
14) (FaultScript=)
Choose the setting to process: 15

```

9. Select [yes].

```

1) HELP
2) RETURN
3) yes
4) no
Set the AutoStartUp mode: 3

```

10. Select [AutoSwitchOver].

```

Machines+Basics (appl:consistent)
 1) HELP                15) (AutoStartUp=yes)
 2) -                  16) (AutoSwitchOver=No)
 3) SAVE+EXIT          17) (PreserveState=no)
 4) REMOVE+EXIT        18) (PersistentFault=0)
 5) AdditionalMachine  19) (ShutdownPriority=)
 6) AdditionalConsole  20) (OnlinePriority=)
 7) Machines[0]=pcl-vm13RMS  21) (StandbyTransitions=)
 8) Machines[1]=pcl-vm14RMS  22) (LicenseToKill=no)
 9) (PreCheckScript=)    23) (AutoBreak=yes)
10) (PreOnlineScript=)  24) (AutoBreakMaintMode=no)
11) (PostOnlineScript=) 25) (HaltFlag=no)
12) (PreOfflineScript=) 26) (PartialCluster=0)
13) (OfflineDoneScript=) 27) (ScriptTimeout=)
14) (FaultScript=)
Choose the setting to process: 16

```

11. Select [HOSTFAILURE], then [RESOURCEFAILURE], then [SHUTDOWN], and then select [SAVE+RETURN] when the following window is displayed:

```

Set flags for AutoSwitchOver: Currently set: HOSTFAILURE,RESOURCEFAILURE,SHUTDOWN (HRS)
1) HELP                4) DEFAULT                7) NOT:RESOURCEFAILURE(R)
2) -                  5) NO(N)                 8) NOT:SHUTDOWN(S)
3) SAVE+RETURN        6) NOT:HOSTFAILURE(H)
Choose one of the flags: 3

```

12. Select [PersistentFault].

```

Machines+Basics (appl:consistent)
 1) HELP
 2) -
 3) SAVE+EXIT
 4) REMOVE+EXIT
 5) AdditionalMachine
 6) AdditionalConsole
 7) Machines[0]=pcl-vm13RMS
 8) Machines[1]=pcl-vm14RMS
 9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
15) (AutoStartUp=yes)
16) (AutoSwitchOver=HostFailure|ResourceFailure|ShutDown)
17) (PreserveState=no)

```

```
18) (PersistentFault=0)
19) (ShutdownPriority=)
20) (OnlinePriority=)
21) (StandbyTransitions=)
22) (LicenseToKill=no)
23) (AutoBreak=yes)
24) (AutoBreakMaintMode=no)
25) (HaltFlag=no)
26) (PartialCluster=0)
27) (ScriptTimeout=)
Choose the setting to process: 18
```

13. Select [0] or [1] (the example below selects [1]).

```
1) HELP
2) RETURN
3) 0
4) 1
Enable persistent fault feature for this application: 4
```

14. Select [OnlinePriority].

```
Machines+Basics (appl:consistent)
1) HELP
2) -
3) SAVE+EXIT
4) REMOVE+EXIT
5) AdditionalMachine
6) AdditionalConsole
7) Machines[0]=pcl-vm13RMS
8) Machines[1]=pcl-vm14RMS
9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
15) (AutoStartUp=yes)
16) (AutoSwitchOver=HostFailure|ResourceFailure|ShutDown)
17) (PreserveState=no)
18) (PersistentFault=1)
19) (ShutdownPriority=)
20) (OnlinePriority=)
21) (StandbyTransitions=)
22) (LicenseToKill=no)
23) (AutoBreak=yes)
24) (AutoBreakMaintMode=no)
25) (HaltFlag=no)
26) (PartialCluster=0)
27) (ScriptTimeout=)
Choose the setting to process: 20
```

15. Select [0] or [1] (the example below selects [1]).

```
1) HELP
2) RETURN
3) 0
4) 1
Enable Online Priority (Active Standby) feature for this application: 4
```

16. Select [StandbyTransitions].

```
Machines+Basics (appl:consistent)
1) HELP
```

```

2) -
3) SAVE+EXIT
4) REMOVE+EXIT
5) AdditionalMachine
6) AdditionalConsole
7) Machines[0]=pcl-vm13RMS
8) Machines[1]=pcl-vm14RMS
9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
15) (AutoStartUp=yes)
16) (AutoSwitchOver=HostFailure|ResourceFailure|ShutDown)
17) (PreserveState=no)
18) (PersistentFault=1)
19) (ShutdownPriority=)
20) (OnlinePriority=1)
21) (StandbyTransitions=)
22) (LicenseToKill=no)
23) (AutoBreak=yes)
24) (AutoBreakMaintMode=no)
25) (HaltFlag=no)
26) (PartialCluster=0)
27) (ScriptTimeout=)
Choose the setting to process: 21

```

17. Select [CLEARFAULTREQUEST], then [STARTUP], then [SWITCHREQUEST], and then select [SAVE+RETURN] when the following window is displayed:

```

Set flags for StandbyTransitions: Currently set: ClearFaultRequest,StartUp,SwitchRequest (CTW)
1) HELP                    5) NOT: CLEARFAULTREQUEST(C)
2) -                      6) NOT: STARTUP(T)
3) SAVE+RETURN            7) NOT: SWITCHREQUEST(W)
4) NO(N)
Choose one of the flags: 3

```

18. Select [HaltFlag].

```

Machines+Basics (appl:consistent)
1) HELP
2) -
3) SAVE+EXIT
4) REMOVE+EXIT
5) AdditionalMachine
6) AdditionalConsole
7) Machines[0]=pcl-vm13RMS
8) Machines[1]=pcl-vm14RMS
9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
15) (AutoStartUp=yes)
16) (AutoSwitchOver=HostFailure|ResourceFailure|ShutDown)
17) (PreserveState=no)
18) (PersistentFault=1)
19) (ShutdownPriority=)
20) (OnlinePriority=1)
21) (StandbyTransitions=ClearFaultRequest|StartUp|SwitchRequest)
22) (LicenseToKill=no)

```

```
23) (AutoBreak=yes)
24) (AutoBreakMaintMode=no)
25) (HaltFlag=no)
26) (PartialCluster=0)
27) (ScriptTimeout=)
Choose the setting to process: 25
```

19. Select [yes].

```
1) HELP
2) RETURN
3) yes
4) no
Set the Halt mode: 3
```

20. Ensure that all attributes are displayed as set in the previous steps, and then select [SAVE+EXIT].

```
Machines+Basics (appl:consistent)
1) HELP
2) -
3) SAVE+EXIT
4) REMOVE+EXIT
5) AdditionalMachine
6) AdditionalConsole
7) Machines[0]=pcl-vm13RMS
8) Machines[1]=pcl-vm14RMS
9) (PreCheckScript=)
10) (PreOnlineScript=)
11) (PostOnlineScript=)
12) (PreOfflineScript=)
13) (OfflineDoneScript=)
14) (FaultScript=)
15) (AutoStartUp=yes)
16) (AutoSwitchOver=HostFailure|ResourceFailure|ShutDown)
17) (PreserveState=no)
18) (PersistentFault=1)
19) (ShutdownPriority=)
20) (OnlinePriority=1)
21) (StandbyTransitions=ClearFaultRequest|StartUp|SwitchRequest)
22) (LicenseToKill=no)
23) (AutoBreak=yes)
24) (AutoBreakMaintMode=no)
25) (HaltFlag=yes)
26) (PartialCluster=0)
27) (ScriptTimeout=)
Choose the setting to process: 3
```

21. The following window will be displayed:

```
Settings of turnkey wizard "STANDBY" (APP1:not yet consistent)
1) HELP
2) -
3) SAVE+EXIT
4) -
5) ApplicationName=APP1
6) Machines+Basics(appl)
7) CommandLines(-)
8) Procedure:Application(-)
9) Procedure:BasicApplication(-)
10) Symfoware(-)
11) Procedure:SystemState3(-)
12) Procedure:SystemState2(-)
13) Gls:Global-Link-Services(-)
14) IpAddresses(-)
15) LocalFileSystems(-)
16) Gds:Global-Disk-Services(-)
Choose the setting to process:
```

## A.1.3 Creating GDS Resources

1. Select [Gds:Global-Disk-Services].

```
Settings of turnkey wizard "STANDBY" (APP1:not yet consistent)
1) HELP                                9) Procedure:BasicApplication(-)
2) -                                    10) Symfoware(-)
3) SAVE+EXIT                            11) Procedure:SystemState3(-)
4) -                                    12) Procedure:SystemState2(-)
5) ApplicationName=APP1                 13) Gls:Global-Link-Services(-)
6) Machines+Basics(app1)                14) IpAddresses(-)
7) CommandLines(-)                     15) LocalFileSystems(-)
8) Procedure:Application(-)             16) Gds:Global-Disk-Services(-)
Choose the setting to process: 16
```

2. Select [AdditionalDiskClass].

```
Volume management (Gds_APP1:not yet consistent)
1) HELP                                4) REMOVE+EXIT                    7) (StandbySupport=no)
2) -                                    5) AdditionalDiskClass            8) (AutoRecover=no)
3) SAVE+EXIT                            6) (ClassNameFilter=)            9) (Timeout=1800)
Choose the setting to process: 5
```

3. Select a disk class.

```
1) HELP
2) RETURN
3) FREECHOICE
4) class0001
Choose the disk class: 4
```

4. Follow the steps above for all shared disk classes to be used by all file systems to be registered in the cluster application. When doing so, configure the MONITORONLY attribute as shown in the following table:

Purpose of file system	MONITORONLY attribute
Data storage destination directory	No
Backup data storage destination directory	Yes
Table space	No
Transaction log storage destination directory	No

5. To configure the disk class, select [StandbySupport] and specify "no" - this means that upon failure, the disk will switch according to the purpose of the disk class.

```
Volume management (Gds_APP1:consistent)
1) HELP                                6) DiskClasses[0]=class0001
2) -                                    7) (ClassNameFilter=)
3) SAVE+EXIT                            8) (StandbySupport=no)
4) REMOVE+EXIT                          9) (AutoRecover=no)
5) AdditionalDiskClass                   10) (Timeout=1800)
Choose the setting to process: 8
```

6. Select [SAVE+EXIT].

```
Volume management (Gds_APP1:consistent)
1) HELP                                6) DiskClasses[0]=class0001
2) -                                    7) (ClassNameFilter=)
3) SAVE+EXIT                            8) (StandbySupport=no)
4) REMOVE+EXIT                          9) (AutoRecover=no)
```

```

5) AdditionalDiskClass          10) (Timeout=1800)
Choose the setting to process: 3

```

## A.1.4 Creating Symfoware Resources

1. Select [Symfoware].

```

Settings of turnkey wizard "STANDBY" (APPl:consistent)
1) HELP                      9) Procedure:BasicApplication(-)
2) -                          10) Symfoware(-)
3) SAVE+EXIT                 11) Procedure:SystemState3(-)
4) -                          12) Procedure:SystemState2(-)
5) ApplicationName=APPl      13) Gls:Global-Link-Services(-)
6) Machines+Basics(appl)    14) IpAddresses(-)
7) CommandLines(-)         15) LocalFileSystems(-)
8) Procedure:Application(-) 16) Gds:Global-Disk-Services(Gds_APPl)
Choose the setting to process: 10

```

2. Select [AdditionalResource].

```

Resource (Symfoware_APPl:not yet consistent)
1) HELP                      4) REMOVE+EXIT
2) -                          5) AdditionalResource
3) SAVE+EXIT
Choose the setting to process: 5

```

3. Select the database cluster name specified in the `pgx_pclsrc` command in "2.3.8 Registering Resource Information for the Symfoware Database Cluster".

```

1) HELP
2) RETURN
3) FREECHOICE
4) pgxtest
Choose the resource: 4

```

4. Select [SAVE+EXIT].

```

Resource (Symfoware_APPl:consistent)
1) HELP                      6) ResourceName[0]=pgxtest
2) -                          7) Flags[0]=T900
3) SAVE+EXIT                 8) OnlineTimeout[0]=3600
4) REMOVE+EXIT               9) OfflineTimeout[0]=1800
5) AdditionalResource        10) FaultScript[0]=''
Choose the setting to process: 3

```

## A.1.5 Creating GIs or Takeover Network Resources

Create GIs or takeover network resources to use takeover network resources. This section provides an example for configuring GI resources.



See

- Refer to the PRIMECLUSTER Installation and Administration Guide for information on creating takeover network resources.
- Refer to the PRIMECLUSTER Installation and Administration Guide for information on creating GI resources.

1. Select [Gls:Global-Link-Services].

```

Settings of turnkey wizard "STANDBY" (APPl:consistent)
1) HELP                      9) Procedure:BasicApplication(-)
2) -                          10) Symfoware(Symfoware_APPl)
3) SAVE+EXIT                 11) Procedure:SystemState3(-)

```

```

4) -
5) ApplicationName=APP1
6) Machines+Basics(app1)
7) CommandLines(-)
8) Procedure:Application(-)
Choose the setting to process: 13

12) Procedure:SystemState2(-)
13) Gls:Global-Link-Services(-)
14) IpAddresses(-)
15) LocalFileSystems(-)
16) Gds:Global-Disk-Services(Gds_APP1)

```

2. Select [AdditionalTakeoverIpaddress].

```

Gls (Gls_APP1:not yet consistent)
1) HELP
2) -
3) SAVE+EXIT
Choose the setting to process: 5

4) REMOVE+EXIT
5) AdditionalTakeoverIpaddress
6) (Timeout=60)

```

3. Select a takeover IP.

```

1) HELP
2) RETURN
3) FRECHOICE
4) takeoverip
Choose a takeover IP address for Gls: 4

```

4. Select [SAVE+RETURN].

```

Set a flag for takeover IP address: takeoverip
Currently set:
1) HELP
2) -
3) SAVE+RETURN
4) DEFAULT
Choose additionally one of the flags: 3

5) AUTORECOVER(A)

```

5. Select [SAVE+EXIT].

```

Gls (Gls_APP1:consistent)
1) HELP
2) -
3) SAVE+EXIT
4) REMOVE+EXIT
Choose the setting to process: 3

5) AdditionalTakeoverIpaddress
6) TakeoverIpaddress[0]=N,takeoverip
7) (Timeout=60)

```

## A.1.6 Creating Fsystem Resources

1. Select [LocalFileSystem].

```

Settings of turnkey wizard "STANDBY" (APP1:consistent)
1) HELP
2) -
3) SAVE+EXIT
4) -
5) ApplicationName=APP1
6) Machines+Basics(app1)
7) CommandLines(-)
8) Procedure:Application(-)
Choose the setting to process: 15

9) Procedure:BasicApplication(-)
10) Symfoware(Symfoware_APP1)
11) Procedure:SystemState3(-)
12) Procedure:SystemState2(-)
13) Gls:Global-Link-Services(Gls_APP1)
14) IpAddresses(-)
15) LocalFileSystems(-)
16) Gds:Global-Disk-Services(Gds_APP1)

```

2. Select [AdditionalMountPoint].

```

File systems (Lfs_APP1:not yet consistent)
1) HELP
2) -
4) REMOVE+EXIT
5) AdditionalMountPoint
7) (Timeout=180)

```

```
3) SAVE+EXIT          6) (Filter=)
Choose the setting to process: 5
```

3. Select all mount points allocated for the data storage destination directory, backup data storage destination directory, tablespace, and transaction log storage destination directory, as well as the mount point for the file systems used to monitor these areas.

Note that settings for the mount point for file systems used for monitoring must be configured before those of the mount point for file systems that will be used for the job. Accordingly, to register multiple file systems for monitoring, repeat Steps 3 and 4 before proceeding to configure settings for file systems used for the job.

```
1) HELP                5) /mnt/monitor1
2) RETURN              6) /mnt/monitor2
3) FREECHOICE          7) /mnt/swdsk1
4) ALL                 8) /mnt/swdsk2
Choose a mount point: 5
```

4. Select [SAVE+RETURN].

```
Set flags for mount point: /mnt/monitor1 Currently set: LOCAL,AUTORECOVER (LA)
1) HELP                4) DEFAULT          7) SHARE(S)
2) -                   5) SYNC(Y)         8) MONITORONLY(M)
3) SAVE+RETURN         6) NOT:AUTORECOVER(A)
Choose one of the flags: 3
```

5. Select the mount point of the file system that is used for a job.

If registering multiple file systems to be used for a job, repeat Steps 5 to 7.

```
1) HELP                5) /mnt/monitor1
2) RETURN              6) /mnt/monitor2
3) FREECHOICE          7) /mnt/swdsk1
4) ALL                 8) /mnt/swdsk2
Choose a mount point: 7
```

6. Configure the MONITORONLY attribute.

Purpose of the file system	MONITORONLY attribute
Data storage destination directory	No
Backup data storage destination directory	Yes
Table space	No
Transaction log storage destination directory	No

Select MONITORONLY to specify Yes to the MONITORONLY attribute.

```
Set flags for mount point: /mnt/swdsk1 Currently set: LOCAL,AUTORECOVER (LA)
1) HELP                4) DEFAULT          7) SHARE(S)
2) -                   5) SYNC(Y)         8) MONITORONLY(M)
3) SAVE+RETURN         6) NOT:AUTORECOVER(A)
Choose one of the flags: 8
```

7. Select [SAVE+RETURN].

```
Set flags for mount point: /mnt/swdsk1 Currently set:
LOCAL,AUTORECOVER, MONITORONLY (LAM)
1) HELP                4) DEFAULT          7) SHARE(S)
2) -                   5) SYNC(Y)         8) NOT:MONITORONLY(M)
3) SAVE+RETURN         6) NOT:AUTORECOVER(A)
Choose one of the flags: 3
```

- After configuring all mount points, select [SAVE+EXIT].

```
File systems (Lfs_APP1:consistent)
1) HELP
2) -
3) SAVE+EXIT
4) REMOVE+EXIT
5) AdditionalMountPoint
6) MountPoints[0]=LA:/mnt/monitor1
7) MountPoints[1]=LA:/mnt/monitor2
8) MountPoints[2]=LA:/mnt/swdsk1
9) MountPoints[3]=LA:/mnt/swdsk2
10) (Filter=)
11) (Timeout=180)
Choose the setting to process: 3
```

- Select [SAVE+EXIT].

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
1) HELP
2) -
3) SAVE+EXIT
4) -
5) ApplicationName=APP1
6) Machines+Basics(app1)
7) CommandLines(-)
8) Procedure:Application(-)
9) Procedure:BasicApplication(-)
10) Symfoware(Symfoware_APP1)
11) Procedure:SystemState3(-)
12) Procedure:SystemState2(-)
13) Gls:Global-Link-Services(Gls_APP1)
14) IpAddresses(-)
15) LocalFileSystems(Lfs_APP1)
16) Gds:Global-Disk-Services(Gds_APP1)
Choose the setting to process: 3
```



See

Refer to the PRIMECLUSTER Installation and Administration Guide for information on creating Fsystem resources.

## A.1.7 Creating Resources and Finalizing Cluster Application Creation

- Select [Configuration-Generate].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
1) HELP
2) QUIT
3) Application-Creat
4) Application-Edit
5) Application-Remove
6) Application-Clone
7) Configuration-Generate
8) Configuration-Activate
9) Configuration-Copy
10) Configuration-Remove
11) Configuration-Freeze
12) Configuration-Thaw
13) Configuration-Edit-Global-Settings
14) Configuration-Consistency-Report
15) Configuration-ScriptExecution
16) RMS-CreateMachine
17) RMS-RemoveMachine
Choose an action: 7
```

- Select [Configuration-Activate].

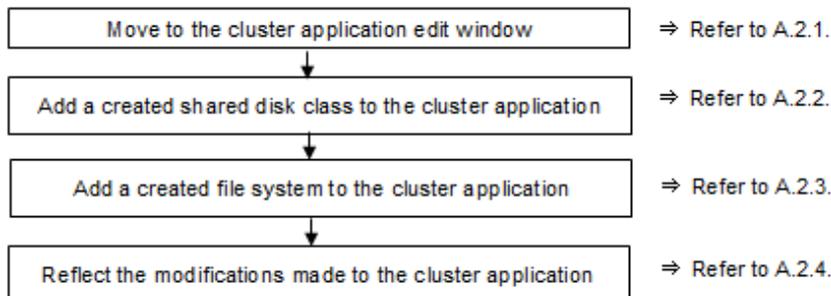
```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
1) HELP
2) QUIT
3) Application-Creat
4) Application-Edit
5) Application-Remove
6) Application-Clone
7) Configuration-Generate
8) Configuration-Activate
9) Configuration-Copy
10) Configuration-Remove
11) Configuration-Freeze
12) Configuration-Thaw
13) Configuration-Edit-Global-Settings
14) Configuration-Consistency-Report
15) Configuration-ScriptExecution
16) RMS-CreateMachine
17) RMS-RemoveMachine
Choose an action: 8
```

3. Select [QUIT].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                      10) Configuration-Remove
 2) QUIT                      11) Configuration-Freeze
 3) Application-Create        12) Configuration-Thaw
 4) Application-Edit          13) Configuration-Edit-Global-Settings
 5) Application-Remove        14) Configuration-Consistency-Report
 6) Application-Clone         15) Configuration-ScriptExecution
 7) Configuration-Generate    16) RMS-CreateMachine
 8) Configuration-Activate    17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 2
```

## A.2 Modifying Cluster Applications

Follow the steps shown below to add a new file system to a cluster application. Refer to the PRIMECLUSTER Installation and Administration Guide for information on other modifications on cluster applications. Stop the RMS on all nodes when modifying cluster applications.



### A.2.1 Moving to the Cluster Application Edit Window

1. Execute the hvw command.

```
# hvw
```

2. Select [Application-Edit].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                      10) Configuration-Remove
 2) QUIT                      11) Configuration-Freeze
 3) Application-Create        12) Configuration-Thaw
 4) Application-Edit          13) Configuration-Edit-Global-Settings
 5) Application-Remove        14) Configuration-Consistency-Report
 6) Application-Clone         15) Configuration-ScriptExecution
 7) Configuration-Generate    16) RMS-CreateMachine
 8) Configuration-Activate    17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 4
```

3. Select the user application to be used.

```
Edit: Application selection menu (restricted):
 1) HELP
 2) QUIT
 3) RETURN
 4) OPTIONS
```

```
5) APP1
Application Name: 5
```

## A.2.2 Adding a Created Shared Disk Class to the Cluster Application

1. Select [Gds:Global-Disk-Services].

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
1) HELP                                9) Procedure:BasicApplication(-)
2) READONLY                            10) Symfoware(Symfoware_APP1)
3) SAVE+EXIT                           11) Procedure:SystemState3(-)
4) -                                    12) Procedure:SystemState2(-)
5) ApplicationName=APP1                13) Gls:Global-Link-Services(Gls_APP1)
6) Machines+Basics(app1)              14) IpAddresses(-)
7) CommandLines(-)                    15) LocalFileSystems(Lfs_APP1)
8) Procedure:Application(-)           16) Gds:Global-Disk-Services(Gds_APP1)
Choose the setting to process: 16
```

2. Select [AdditionalDiskClass].

```
Volume management (Gds_APP1:consistent)
1) HELP                                6) DiskClasses[0]=class0001
2) NO-SAVE+EXIT                        7) (ClassNameFilter=)
3) SAVE+EXIT                           8) (StandbySupport=no)
4) REMOVE+EXIT                         9) (AutoRecover=no)
5) AdditionalDiskClass                 10) (Timeout=1800)
Choose the setting to process: 5
```

3. Select the class of the created disk.

```
1) HELP
2) RETURN
3) FRECHOICE
4) class0001
5) class0002
Choose the disk class: 5
```

4. Select [SAVE+EXIT].

```
Volume management (Gds_APP1:consistent)
1) HELP                                7) DiskClasses[1]=class0002
2) NO-SAVE+EXIT                        8) (ClassNameFilter=)
3) SAVE+EXIT                           9) (StandbySupport=no)
4) REMOVE+EXIT                         10) (AutoRecover=no)
5) AdditionalDiskClass                 11) (Timeout=1800)
6) DiskClasses[0]=class0001
Choose the setting to process: 3
```

5. Select [SAVE+EXIT] to quit or [LocalFileSystems] to add local file systems.

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
1) HELP                                9) Procedure:BasicApplication(-)
2) -                                    10) Symfoware(Symfoware_APP1)
3) SAVE+EXIT                           11) Procedure:SystemState3(-)
4) -                                    12) Procedure:SystemState2(-)
5) ApplicationName=APP1                13) Gls:Global-Link-Services(Gls_APP1)
6) Machines+Basics(app1)              14) IpAddresses(-)
7) CommandLines(-)                    15) LocalFileSystems(Lfs_APP1)
8) Procedure:Application(-)           16) Gds:Global-Disk-Services(Gds_APP1)
Choose the setting to process:
```

## A.2.3 Adding a Created File System to the Cluster Application

1. Select [LocalFileSystems].

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
1) HELP                                9) Procedure:BasicApplication(-)
2) -                                    10) Symfoware(Symfoware_APP1)
3) SAVE+EXIT                            11) Procedure:SystemState3(-)
4) -                                    12) Procedure:SystemState2(-)
5) ApplicationName=APP1                 13) Gls:Global-Link-Services(Gls_APP1)
6) Machines+Basics(app1)                14) IpAddresses(-)
7) CommandLines(-)                     15) LocalFileSystems(Lfs_APP1)
8) Procedure:Application(-)             16) Gds:Global-Disk-Services(Gds_APP1)
Choose the setting to process: 15
```

2. Select [AdditionalMountPoint].

```
File systems (Lfs_APP1:consistent)
1) HELP                                7) MountPoints[1]=LA:/mnt/monitor2
2) NO-SAVE+EXIT                        8) MountPoints[2]=LA:/mnt/swdsk1
3) SAVE+EXIT                            9) MountPoints[3]=LA:/mnt/swdsk2
4) REMOVE+EXIT                         10) (Filter=)
5) AdditionalMountPoint                 11) (Timeout=180)
6) MountPoints[0]=LA:/mnt/monitor1
Choose the setting to process: 5
```

3. After registering the file systems to be used and the file system for the areas for monitoring them, select [SAVE+EXIT].

```
File systems (Lfs_APP1:consistent)
1) HELP                                8) MountPoints[2]=LA:/mnt/swdsk1
2) NO-SAVE+EXIT                        9) MountPoints[3]=LA:/mnt/swdsk2
3) SAVE+EXIT                            10) MountPoints[3]=LA:/mnt/monitor3
4) REMOVE+EXIT                         10) MountPoints[3]=LA:/mnt/swdsk3
5) AdditionalMountPoint                 11) (Filter=)
6) MountPoints[0]=LA:/mnt/monitor1     12) (Timeout=180)
7) MountPoints[1]=LA:/mnt/monitor2
Choose the setting to process:
```

## A.2.4 Reflecting Modifications Made to the Cluster Application

1. Select [SAVE+EXIT].

```
Settings of turnkey wizard "STANDBY" (APP1:consistent)
1) HELP                                9) Procedure:BasicApplication(-)
2) -                                    10) Symfoware(Symfoware_APP1)
3) SAVE+EXIT                            11) Procedure:SystemState3(-)
4) -                                    12) Procedure:SystemState2(-)
5) ApplicationName=APP1                 13) Gls:Global-Link-Services(Gls_APP1)
6) Machines+Basics(app1)                14) IpAddresses(-)
7) CommandLines(-)                     15) LocalFileSystems(Lfs_APP1)
8) Procedure:Application(-)             16) Gds:Global-Disk-Services(Gds_APP1)
Choose the setting to process: 3
```

2. Select [RETURN].

```
Edit: Application selection menu (restricted):
1) HELP
2) QUIT
3) RETURN
4) OPTIONS
5) APP1
Application Name: 3
```

3. Select [Configuration-Generate].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                                10) Configuration-Remove
 2) QUIT                                11) Configuration-Freeze
 3) Application-Create                  12) Configuration-Thaw
 4) Application-Edit                   13) Configuration-Edit-Global-Settings
 5) Application-Remove                  14) Configuration-Consistency-Report
 6) Application-Clone                   15) Configuration-ScriptExecution
 7) Configuration-Generate              16) RMS-CreateMachine
 8) Configuration-Activate              17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 7
```

4. Select [Configuration-Activate].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                                10) Configuration-Remove
 2) QUIT                                11) Configuration-Freeze
 3) Application-Create                  12) Configuration-Thaw
 4) Application-Edit                   13) Configuration-Edit-Global-Settings
 5) Application-Remove                  14) Configuration-Consistency-Report
 6) Application-Clone                   15) Configuration-ScriptExecution
 7) Configuration-Generate              16) RMS-CreateMachine
 8) Configuration-Activate              17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 8
```

5. Select [QUIT].

```
pcl-vm13: Main configuration menu, current configuration: config
No RMS active in the cluster
 1) HELP                                10) Configuration-Remove
 2) QUIT                                11) Configuration-Freeze
 3) Application-Create                  12) Configuration-Thaw
 4) Application-Edit                   13) Configuration-Edit-Global-Settings
 5) Application-Remove                  14) Configuration-Consistency-Report
 6) Application-Clone                   15) Configuration-ScriptExecution
 7) Configuration-Generate              16) RMS-CreateMachine
 8) Configuration-Activate              17) RMS-RemoveMachine
 9) Configuration-Copy
Choose an action: 2
```

## Appendix B Parameters

This appendix describes the configuration files and parameters required by the database multiplexing mode.



See

Refer to "Server Configuration" in the PostgreSQL Documentation for information on the postgresql.conf file.

### B.1 Parameters Set on the Primary Server

The content for the parameters set in the postgresql.conf file of the primary server is shown in the table below.

Table B.1 postgresql.conf file

Parameter	Value set	Explanation
wal_level	Literal "hot_standby"	Specify the output level for the transaction log. Specify "hot_standby".
max_wal_senders	Number of standby servers + 1	Specify the number of standby servers + 1. In Symfoware Server Enterprise Edition and Standard Edition, 1:1 standby is used, so specify 2.
synchronous_standby_names	' <i>standbyServerName</i> '	Use single quotation marks (') to enclose the name that will identify the standby server. Any name can be specified. The value specified for this parameter must also be specified in the --standby-names parameter when the mc_ctl command is executed in setup mode. Do not change this parameter while Mirroring Controller is running.
hot_standby	on	Specify whether queries can be run on the standby server. Specify this to execute reference jobs on the standby server. (*1)
wal_receiver_status_interval	Time interval (seconds)	This value must be smaller than the value set for replication_timeout in the postgresql.conf file of the primary server.
wal_keep_segments	Number of file segments	If a delay exceeding the value set in this parameter occurs, the WAL segment required later by the primary server may be deleted. Additionally, if you stop a standby server (for maintenance, for example), consider the stop time and set a value that will not cause the WAL segment to be deleted. Setting this parameter requires consideration regarding stabilization of the database multiplexing mode (refer to " <a href="#">6.9.1 Tuning to Stabilize the Database Multiplexing Mode</a> " for details).

Parameter	Value set	Explanation
		Refer to "Estimating Transaction Log Space Requirements" in the Installation and Setup Guide for Server for information on estimating the WAL segment.
replication_timeout	Timeout (seconds)	<p>Specify the time period after which it is determined that the receiver process (walreceiver) of the transaction log is in an abnormal state.</p> <p>The specified value must be larger than the value set for the wal_receiver_status_interval parameter set in the postgresql.conf file of the standby server.</p> <p>By aligning this value with the value of (heartbeat_interval + heartbeat_timeout) x heartbeat_retry of the serverIdentifier.conf file, you can unify the time after which it is determined that an error has occurred.</p>
archive_mode	on	Specify the archive log mode.
archive_command	<code>'installDir/bin/pgx_xlogcopy.cmd "%p" "backupDataStorageDestinationDirectory/archived_xlog/%f"'</code>	Specify the command and storage destination to save the transaction log.
backup_destination	Backup data storage destination directory	<p>Specify the name of directory where to store the backup data.</p> <p>Set the permissions so that only the instance administrator user can access the specified directory.</p> <p>Specify the same full path on all servers, so that the backup data of other servers can be used to perform recovery.</p>
listen_addresses	Standby server IP address, host name, or "*"	<p>Specify the IP address or host name of the standby server. Specify the IP address or host name of the standby server. Specify the IP address or corresponding host name that will be used to connect to the log transfer network.</p> <p>The content specified is also used to allow connections from client applications.</p> <p>To receive the connection and the transaction log from any client or standby server, specify "*".</p> <p>Refer to "Connections and Authentication" in the PostgreSQL Documentation for details.</p>
max_connections	The limit value for the number of standby server connections + number of simultaneous executions of mc_ctl status (*2) + 2 + number of simultaneous client connections to the instance + superuser_reserved_connections value	<p>Specify the maximum value for the number of standby server connections.</p> <p>The value specified is also used to restrict the number of connections from client applications and the number of connections for the management of instances.</p>

Parameter	Value set	Explanation
		Refer to "When an Instance was Created with the initdb Command" in the Installation and Setup Guide for Server, and "Connections and Authentication" in the PostgreSQL Documentation, for details.

\*1: Used in the Mirroring Controller option feature.

\*2: Number of simultaneous executions of the mc\_ctl command in the status mode.

## B.2 Parameters Set on the Standby Server

This section explains the content of the file and parameters set on the standby server. After editing postgresql.conf file, start the instance.

The content for the parameters specified in postgresql.conf file is shown in the table below.

Table B.2 postgresql.conf file

Parameter	Value set	Explanation
synchronous_standby_names	<i>'primaryServerName'</i>	Use single quotation marks (') to enclose the name that will identify the primary server. Any name can be specified.  This name will be required to rebuild the original primary server as the new standby server after the primary server was switched.  The value specified for this parameter must also be specified in the --standby-names parameter when the mc_ctl command is executed in setup mode.  Do not change this parameter while Mirroring Controller is running.
backup_destination	Backup data storage destination directory	Specify the name of the backup data storage directory.  Set the permissions so that only the instance administrator user can access the specified directory.  Specify the same full path on all servers so that the backup data of other servers can be used to perform recovery.
archive_command	<i>'installDir/bin/pgx_xlogcopy.cmd "%p" "backupDataStorageDestinationDirectory/archived_xlog/%f"'</i>	Specify the command and storage destination to save the transaction log.

## B.3 Network Configuration File

Define the network configuration that will link the Mirroring Controller processes in the network configuration file (network.conf file).

Define the same content on the primary server and standby server.

Table B.3 network.conf file

Format specified	Explanation
<i>serverIdentifier hostName portNumber</i> <i>Or</i>	Specify any identifier (*1) for the server, a host name (or IP address), and port number (*2), using a space as the separator.

Format specified	Explanation
<i>serverIdentifier ipAddress portNumber</i>	<p>Specify the IP address or corresponding host name (*3) that will be used to connect to the admin network.</p> <p>Define the same content on the primary server and standby server.</p> <p>Example)</p> <p>The literal <i>&lt;space&gt;</i> represents a space.</p> <p>[For IPv4]</p> <pre>server1 &lt;space&gt;192.0.2.100 26540 server2 &lt;space&gt;192.0.2.110 &lt;space&gt;26540</pre> <p>[For IPv6]</p> <pre>server1 2001:258:8404:1217:250:56ff:fea7:559f 27000 server2 2001:258:8404:1217:250:56ff:fea7:55a0 27100</pre>

\*1: The maximum length of the server identifier is 64 bytes. Characters other than Japanese may be specified. Use ASCII characters excluding spaces to specify this parameter.

\*2: The port number must be 0 to 65535. Ensure that the port number does not conflict with other software. Do not specify an ephemeral port that may temporarily be assigned by another program. Note that the value specified in this parameter must also be set in the */etc/services* file.

\*3: Use ASCII characters excluding spaces to specify the host name.

## B.4 Server Configuration File

Define the information related to Mirroring Controller monitoring and control in the *serverIdentifier.conf* file. The maximum length of the server identifier is 64 bytes. Use ASCII characters excluding spaces to specify this parameter.

If the primary server and standby server environments are different, define content that is different, according to the environment.

Table B.4 *serverIdentifier.conf* file

Parameter	Value set	Explanation
<i>db_instance</i>	<p><i>'dataStorageDestinationDir'</i></p> <p>[Example]</p> <pre>db_instance = '/database1/inst1'</pre>	<p>Specify using single quotation marks (') to enclose the data storage destination directory used to identify the monitoring target instance.</p> <p>Use ASCII characters excluding spaces to specify this parameter.</p>
<i>db_instance_password</i>	<p>'Password of the instance administrator user'</p>	<p>Specify the password used when Mirroring Controller connects to a database instance.</p> <p>Use ASCII characters excluding spaces to specify this parameter.</p> <p>If password authentication is performed, you must specify this parameter in the settings used when Mirroring Controller connects to a database instance.</p> <p>If you specify this parameter when password authentication is not performed, the parameter will be ignored.</p> <p>Enclose the name within single quotation marks (').</p>
<i>remote_call_timeout</i>	Admin communication timeout	Specify the timeout value (milliseconds) of the Mirroring Controller agent process for communication between servers.

Parameter	Value set	Explanation
		Specify a value between 0 and 2147483647. The value 0 indicates that there is no timeout limit. The default is 70000 milliseconds (70 seconds).
core_file_path	Core file output directory	Specify the directory to which the core file is to be output. Use ASCII characters excluding spaces to specify this parameter. If this parameter is omitted, it will be assumed that the Mirroring Controller management directory was specified.
syslog_facility	Specify LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, or LOCAL7.	When the import of logs to the syslog is enabled, the value of this parameter will be used for "facility" of the syslog. The default is "LOCAL0".
syslog_ident (*1)	' <i>programName</i> '	Specify using single quotation marks (') to enclose the program name used to identify the Mirroring Controller message in the system log. Use ASCII characters excluding spaces to specify this parameter. The default is 'MirroringControllerOpen'.
heartbeat_interval	Interval time for operating system/ server and process heartbeat monitoring, and disk abnormality monitoring (milliseconds)	Servers will contact each other and perform abnormality monitoring at the specified interval. Specify a value between 1 and 2147483647. The default is 800 milliseconds.
heartbeat_timeout	Timeout for operating system/server and process heartbeat monitoring, and disk abnormality monitoring (seconds)	If there is no response for at least the number of seconds specified, it will be assumed that an error has occurred that requires the primary server to be switched, or the standby server to be disconnected. Specify a value between 1 and 2147483647. The default is 1 second.
heartbeat_retry	Number of retries for operating system/ server and process heartbeat monitoring, and disk abnormality monitoring (number of times)	Specify the number of retries when an error has been detected that requires the primary server to be switched, or the standby server to be disconnected. If an error is detected in succession for at least the number of times specified, the switch or the disconnection will be performed. Specify a value between 0 and 2147483647. The default is 2 times.

\*1: By specifying the syslog\_ident parameter of the postgresql.conf file, the Mirroring Controller output content can be referenced transparently, so log reference is easy.

## Appendix C Supplementary Information on Building the Primary Server and Standby Server on the Same Server

The primary server and standby server can be pseudo-configured on the same server for system testing, for example. Out of consideration for performance and reliability, do not use this type of configuration for any other purposes. For this reason, do not use this type of configuration in a production environment.

Note that the setup and operations is the same as if the primary and standby servers are built on different servers.

This appendix provides supplementary information explaining how to configure the primary server and standby server on the same server.

### C.1 Features that Cannot be Used in this Configuration

The following feature cannot be used:

#### Tablespace

In Symfoware Server, the storage destination for database objects such as tables and indexes can be changed to another disk using the tablespace. However, this feature cannot be used if the primary server and standby server are built on the same server.

The reason is that, when the updates on the primary server are reflected on the standby server, the same directory will be updated, and the updates on the primary server may be overwritten.

### C.2 Backup Data Storage Destination Directory

It is not a problem if the same backup data storage destination directory is used on the primary server and standby server.

### C.3 How to Execute the mc\_ctl Command

When executing the mc\_ctl command, specify the server identifier in the --local-server option in order to identify the operation destination server.

Below is an example of starting Mirroring Controller of the server "server1" defined in the network.conf file. For mc\_ctl command operations using another mode, also specify the --local-server option.

Define two server identifiers for the same IP address with different port numbers in the network.conf file.

Example)

```
server1 192.0.2.100 26540
server2 192.0.2.100 26541
```

Ensure that the port numbers of both primary server and standby server do not conflict with any other software.

When the Mirroring Controller option is used

Start Mirroring Controller of the server "server1":

Example)

```
$ mc_ctl start -M /mcdir/inst1 -f --local-server server1
```

Stop Mirroring Controller of the server "server1":

Example)

```
$ mc_ctl stop -M /mcdir/inst1 --local-server server1
```

When the Mirroring Controller option is not used

Start Mirroring Controller of the server "server1":

Example)

```
$ mc_ctl start -M /mcdire/inst1 --local-server server1
```

Stop Mirroring Controller of the server "server1":

Example)

```
$ mc_ctl stop -M /mcdire/inst1 --local-server server1
```

## Note

If using an rc script

Append the `--local-server` option to the `mc_ctl` command call defined in the auto-startup shell script.

If using `systemd`

Add the `--local-server` option to the `mc_ctl` option specification for `ExecStart` and `ExecStop` of the unit file for `systemd`.

Refer to "[6.10 Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances](#)" for details.

# Index

[A]	[D]
About Database Multiplexing Mode..... 25	Database Backup Operation..... 47
Action Required when All Servers or Instances Stopped..... 70	Definition of Cluster Operation..... 2
Action Required when an Error Occurs in the Database Multiplexing Mode..... 62	Definition of Failover Operation..... 5
Action Required when Automatic Disconnection Fails..... 69	Deleting a database cluster resource..... 17
Action Required when Automatic Switch Fails..... 68	Displaying database cluster resource information..... 16
Action Required when Server Degradation Occurs..... 62	
Adding a Created File System to the Cluster Application..... 89	[E]
Adding a Created Shared Disk Class to the Cluster Application..... 88	Editing Configuration Files..... 10
Adding Tablespaces..... 16	Encryption of Transaction Logs Transferred to the Standby Server..... 32
Application Connection Server Settings..... 44	Errors during state transition..... 18
Authentication of the Standby Server..... 31	
	[F]
[B]	Failback of the Primary Server..... 65
Backing up Database Multiplexing Mode Information..... 47	Failover Operation..... 16
Backup Data Storage Destination Directory..... 96	Feature of failover operation..... 5
Backup Operation..... 47	Features that Cannot be Used..... 96
[C]	[H]
Changes in Operation ..... 57	High availability..... 2
Changes Required when the Standby Server is Stopped..... 58	How to Execute the mc_ctl Command..... 96
Changing from Database Multiplexing Mode to Single Server Mode..... 59	
Changing from Single Server Mode to Database Multiplexing Mode..... 58	[I]
Changing the PL/extJava Environment..... 17	Identify cause of error and perform recovery..... 64,68
Checking Operation..... 15	Identify Cause of Error and Restructure the Standby Server..... 64,67
Checking the Behavior..... 44	Identifying the Cause of an Error..... 22
Checking the Database Multiplexing Mode Status..... 49	Installation..... 34
Configuring Storage Data Protection Using Transparent Data Encryption..... 14	Installing PRIMECLUSTER and Symfoware Server..... 9
Cluster Operation Provided by Symfoware Server..... 2	
Configuration for Database Multiplexing Mode..... 27	[L]
Configuring PRIMECLUSTER..... 9	Log shipping mode..... 2
Configuring the PL/extJava Environment Settings..... 15	
Confirming the Streaming Replication Status..... 42	[M]
Creating, Setting, and Registering the Primary Server Instance..... 36	Maintenance Tasks..... 18
Creating, Setting, and Registering the Standby Server Instance..... 41	Manually Disconnecting the Standby Server..... 50
Creating a Cluster Application..... 15	Manually Switching the Primary Server..... 50
Creating a Cluster Application and Configuring Its Attributes..... 75	Matching the system times..... 33
Creating a File System..... 10	Mirroring Controller option..... 26
Creating a GDS Volume..... 10	Mirroring Controller Resources..... 28
Creating an Operating System User to Start Symfoware Server..... 11	Modifying Cluster Applications..... 87
Creating Applications..... 43	Modifying database cluster resource content..... 16
Creating a Symfoware Database Cluster..... 12	Modifying Database Cluster Resources..... 16
Creating Fsystem Resources..... 84	Monitoring Mirroring Controller Messages..... 51
Creating GDS Resources..... 82	Monitoring using database multiplexing mode..... 25
Creating GIs or Takeover Network Resources..... 83	Mounting the File System..... 11
Creating Resources and Cluster Applications..... 75	Moving to the Cluster Application Edit Window..... 87
Creating Resources and Creating/Modifying Cluster Applications..... 75	Mutual Switch Maintenance..... 20
Creating Resources and Finalizing Cluster Application Creation..... 86	
Creating Symfoware Resources..... 83	[N]
	network.conf file..... 93
	Non-transferrable feature..... 18
	Notes on CPU Architecture and Products..... 29
	Notes on Executing Reference Jobs on the Standby Server (when the Mirroring Controller Option is Used)..... 29
	Notes on Indexes..... 44

	[O]		
Operation at State Transition.....		18	
Operation mode of failover operation.....		5	
Operations in Database Multiplexing Mode.....		49	
Operations under Normal Circumstances.....		17	
Operations when the Server has Started Degrading after a Disconnection has Occurred.....		66	
Operations when the Server has Started Degrading after a Switch has Occurred.....		62	
Overview of Database Multiplexing Mode.....		25	
Overview of Failover Operation.....		5	
	[P]		
Parameters.....		91	
Parameters Set on the Primary Server.....		91	
Parameters Set on the Standby Server.....		93	
postgresql.conf file.....		91,93	
Preparing for Creation.....		75	
Preparing the Backup Disk.....		34	
Procedures Required after a Failover Error.....		22	
	[R]		
Rebuild the Standby Server.....		64,68	
Recovery of the Mirroring Controller management directory .....		64,68	
Recovery Tasks.....		23	
Redundancy of the Admin and Log Transfer Networks.....		29	
Reflecting Modifications Made to the Cluster Application.....		89	
Restarting the Cluster Applications Stopped for Node Maintenance.....		23	
Rolling Updates.....		52	
	[S]		
Scalability.....		2	
Security in Database Multiplexing.....		30	
Server Configuration File.....		94	
serverIdentifier.conf file.....		94	
Server Maintenance.....		52	
Setting Automatic Start and Stop of Mirroring Controller and Multiplexed Instances.....		45	
Setting up Database Multiplexing Mode.....		33	
Setting up Database Multiplexing Mode on the Primary Server.....		34	
Setting up Database Multiplexing Mode on the Standby Server.....		40	
Setting up the PL/extJava Environment.....		43	
Setting up the Primary Server.....		34	
Setting up the Standby Server.....		40	
Setup.....		33	
Setup of Failover Operation.....		8	
Shared disk mode.....		2	
Simultaneous Stopped Node Maintenance.....		19	
Starting and Stopping Instances.....		49	
Starting the Database Cluster.....		23	
Stop Mirroring Controller.....		64,67	
Stopping for Maintenance .....		57	
Stopping the Cluster Applications for Node Maintenance.....		23	
Stopping the Database Cluster.....		23	
Supplementary Information on Building the Primary Server and Standby Server on the Same Server.....		96	
			System Configuration of Failover Operation..... 6
	[T]		
			Tuning.....44
			Tuning to Stabilize Queries on the Standby Server.....44
			Tuning to Stabilize Queries on the Standby Server (when Performing Frequent Updates on the Primary Server)..... 45
			Tuning to Stabilize the Database Multiplexing Mode.....44
	[U]		
			Unmounting the File System..... 15
			Users who perform setup and operations..... 33
	[W]		
			What is a cluster system? .....2