FUJITSU

FUJITSU Software
Interstage Application Server

High Availability System Guide

Windows/Solaris/Linux

Preface

Purpose of this Document

This manual provides information on how to set up and operate Interstage for high availability.

Note

Throughout this manual Interstage Application Server is referred to as Interstage.

Intended Readers

This document is intended for users installing and operating Interstage Application Server.

It is assumed that readers of this manual have a basic knowledge of the following:

- The Internet

- Basic knowledge of the OS used

Structure of This Document

The structure of this manual is as follows:

Chapter 1 For a High-performance, High-reliability System

This chapter describes functions that were implemented for a system with high reliability.

Chapter 2 Designing Systems for High-reliability

This chapter outlines the high-reliability system and describes features of the system.

Chapter 3 Load Distribution Using IPCOM

This chapter explains about load distribution using IPCOM.

Chapter 4 Environment Setup Procedure for Cluster Service

This chapter describes the setup procedure for cluster service environments.

Conventions

Representation of Platform-specific Information

In the manuals of this product, there are parts containing content that relates to all products that run on the supported platform. In this case, an icon indicating the product platform has been added to these parts if the content varies according to the product. For this reason, refer only to the information that applies to your situation.

Windows32	Indicates that this product (32-bit) is running on Windows.
Windows64	Indicates that this product (64-bit) is running on Windows.
Windows32/64	Indicates that this product (32/64-bit) is running on Windows.
Solaris32	Indicates that this product (32-bit) is running on Solaris.
Solaris64	Indicates that this product (64-bit) is running on Solaris.
Solaris32/64	Indicates that this product (32/64-bit) is running on Solaris.
Linux32	Indicates that this product (32-bit) is running on Linux.
Linux64	Indicates that this product (64-bit) is running on Linux.
Linux32/64	Indicates that this product (32/64-bit) is running on Linux.

Abbreviations

Read occurrences of the following Components as their corresponding Service.

Service	Component
CORBA Service	ObjectDirector
Component Transaction Service	TransactionDirector

Export Controls

Exportation/release of this document may require necessary procedures in accordance with the regulations of the Foreign Exchange and Foreign Trade Control Law of Japan and/or US export control laws.

Trademarks

Trademarks of other companies are used in this documentation only to identify particular products or systems.

Product Trademarks/Registered Trademarks
Microsoft, Active Directory, ActiveX, Excel, Internet Explorer, MS-DOS, MSDN, Visual Basic, Visual C++, Visual Studio, Windows, Windows NT, Windows Server, Win32 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Other company and product names in this documentation are trademarks or registered trademarks of their respective owners.

Copyrights

Contents

Chapter 1 For a High-performance, High-reliability System

Reliability and scalability are extremely important for core business servers. From the beginning, Interstage offered the following functions to achieve this goal:

Now the Enterprise Edition offers the following additional functions for larger scale and more reliability:

1.1 Load Distribution Using IPCOM

IPCOM can be linked to distribute the load on a server machine.

1.1.1 Protocols Supporting Load Distribution

The following two Interstage communication protocols are available:

- HTTP protocol

- IIOP protocol

Linkage with IPCOM is associated with these two communication protocols.

This item describes the IIOP load distribution. For HTTP load distribution, refer to the IPCOM manual.

1.1.2 Load Distribution Policy

Load distribution can be performed with the following policies:

- Standard Load distribution policy of IPCOM (*)

    - Round robin method

    - Load measurement (CPU, memory, disk I/O) method

    - Load measurement (number of connections, response time) method

    - Load measurement (data communication quantity) method

    - Static weighting method

    (*) Load distribution can be performed for EJB applications deployed in the IJServer of the following operation types:

    - Web applications and EJB applications run in separate Java VM

    - EJB Applications only

 - Load distribution policy specific to Interstage (IPCOM linkage function)

    - Number of wait messages

    - Communication buffer use ratio

For an explanation of the IPCOM standard load distribution policies, refer to the IPCOM manuals.

The following explains the load distribution policy specific to Interstage.

Load distribution policies specific to Interstage can be applied to applications operated as a WorkUnit. The following table lists applications to which such specific policies can be applied:

Table 1.1 Applications to which Specific Policies can be Applied Windows32 Solaris32 Linux32

	CORBA Application	Transaction Application	EJB Application
Number of wait messages	Y	Y	N

	CORBA Application	Transaction Application	EJB Application
Use ratio of communication buffer	N	Y	N

Y: Applicable

N: Not applicable

1.1.3 Number of Wait Messages `Windows32` `Solaris32` `Linux32`

Applicable to the following applications that operate as a WorkUnit. The number of staying queues of wait messages is monitored.

- CORBA Application

  The number of wait messages for the implementation repository of CORBA applications is monitored.

- Transaction Application

  The number of wait messages for objects of a transaction application is monitored.

Note

EJB applications deployed in the IJServer cannot be monitored.

This is effective when the load amount can be judged by the number of waiting queues in a business operation mode where requests from a client application are received by a specific object and such requests are distributed to other objects.

Figure 1.1 Number of Wait Messages
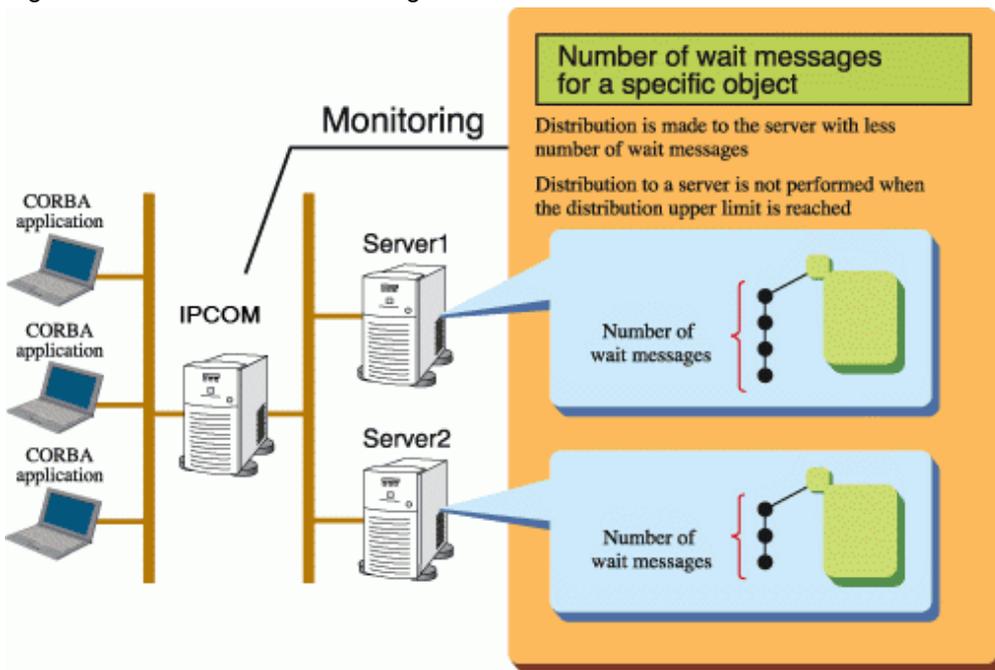
1.1.4 Communication Buffer Use Ratio

Using this ratio enables the communication load status to be monitored for all transaction applications in the server. This item is valid when an application is executed while multiple objects receive requests from the client application.

Figure 1.2 Communication Buffer Use Ratio
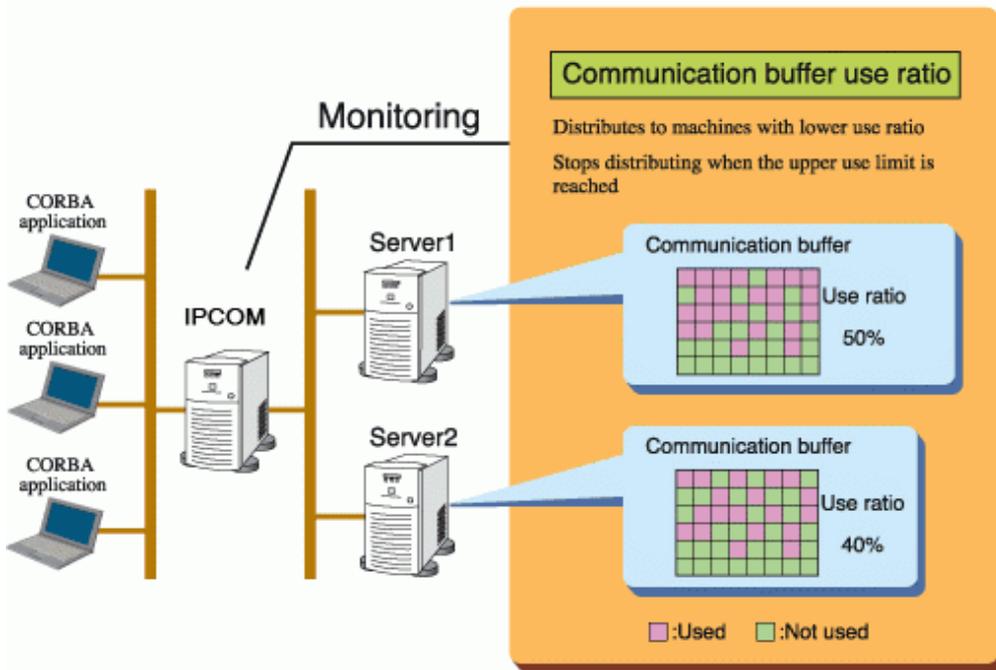
1.1.5 Operation when the Distribution Upper Limit is Reached

When a load measurement item value reaches the distribution upper limit, no distribution occurs for the server.

When the load measurement item values of all servers for distribution have reached the distribution upper limit, the subsequent distribution to a server is not performed. In this case, the request from the client is returned with a communication error (COMM_FAILURE).

When the load measurement item value is less than the restoration value, distribution to the server restarts.

1.1.6 Load Distribution for a WorkUnit

For the load distribution of applications operated as a WorkUnit, one of the following two modes can be selected for each application:

For load distribution for EJB applications of IJServer WorkUnits, Fujitsu recommends that load distribution in method calls be used.

- Load distribution in method calls

  Load is distributed for each method call. This mode can be used for communication of question and answer exchanges.

  For EJB applications, a target server of load distribution that operates the IJServer and a server that manages the target server are required (for the IJServer types for which load distribution can be performed, refer to "1.1.2 Load Distribution Policy").

  For EJB applications, load is distributed at execution of a method of the Home interface (such as the create method). Because the same server is accessed at execution of a method of the Remote interface, distribute load by executing a Home interface method at appropriate timing.

- Load distribution when an object reference to the Naming Service is obtained

  Load is distributed when an object reference to the Naming Service is obtained. In this mode, subsequent communication is conducted with the server to which an object reference acquisition request of the Naming Service is allocated.

  This mode can be used when a series of requests from the same client should be allocated to the same server. An operation like a continuous session can be performed.

Figure 1.3 Load Distribution for a WorkUnit
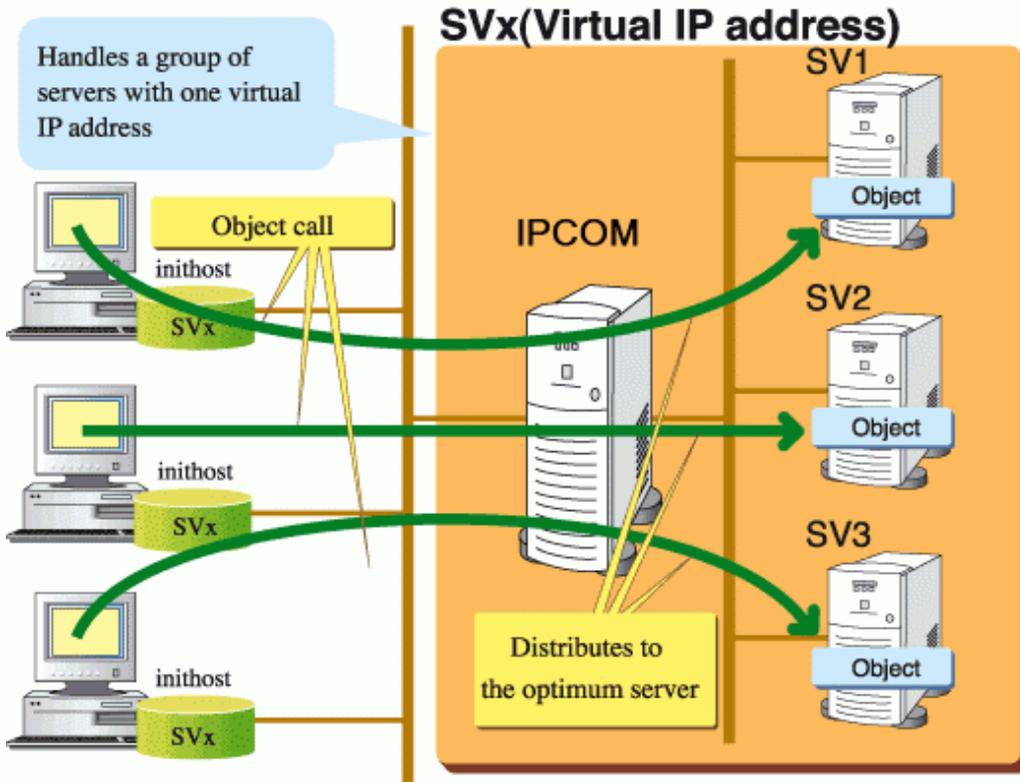
In the load distribution of the WorkUnit, a server can be disconnected from the load distribution server, and diminished operation can be executed when a particular monitored WorkUnit ends abnormally. Refer to "1.1 Load Distribution Using IPCOM" for details.

Note

The Naming Service must be used for load distribution to each target server machine.

1.2 Cluster Service Function

A high reliability system is needed to prepare for unplanned events such as hardware and software failure, and system crashes.

To prepare for unplanned problems, operate an extra server in standby mode (a standby server) and have it share disks with the normal operating servers. Map data onto the shared disks. If, due to a hardware or some other failure, the operational system goes down, operations will be shifted to the standby server which in turn will have access to the disks. Service will continue, uninterrupted. This is called the Cluster Service function.

The Cluster Service function inheritance process uses functions such as network inheritance (IP address inheritance). It supports many types of system construction hardware and software.

1.2.1 Cluster System

The Cluster Service is supplied by Interstage Application Server Enterprise Edition, and runs on the following cluster system:

Windows32/64

- Microsoft(R) Cluster Service (MSCS) (used as the Microsoft(R) cluster system) or Microsoft(R) Windows Server(R) Failover Clustering.

Solaris32/64

- The Fujitsu cluster system PRIMECLUSTER

Linux32/64

- The Fujitsu cluster system PRIMECLUSTER

To use the cluster service function, the above knowledge is required. For more information, refer the cluster system manual.

Windows32/64

In this document, for Microsoft(R) Windows Server(R) 2008 or later, read "Failover Clustering" for "MSCS".

1.2.2 Operating Mode

The following operating modes are provided for active and standby machines:

- 1:1 Fixed Standby Mode

  In this mode, if an active node becomes faulty, a standby node takes over the jobs automatically.

  Figure 1.4 The Cluster Service Function (1:1 Fixed Standby Mode)
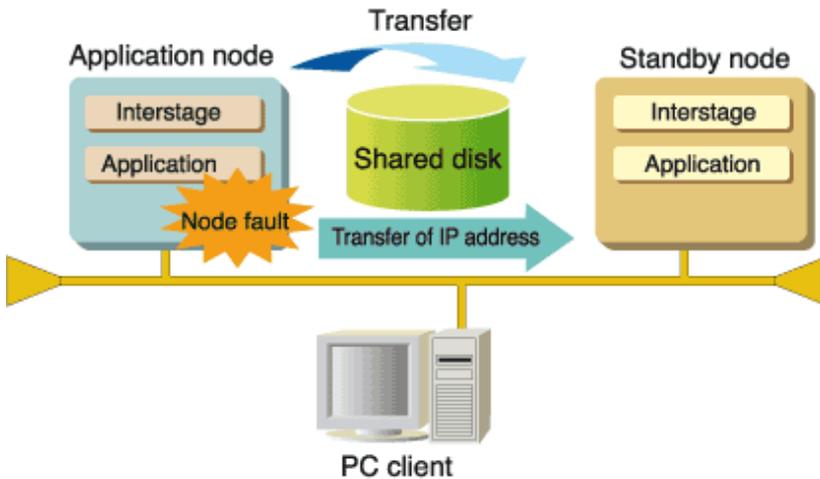
- Mutual Standby Mode　Solaris32

  In this mode, two nodes perform jobs. If one node becomes faulty, the other node takes over the jobs.

  Figure 1.5 The Cluster Service Function (Mutual Standby Mode)
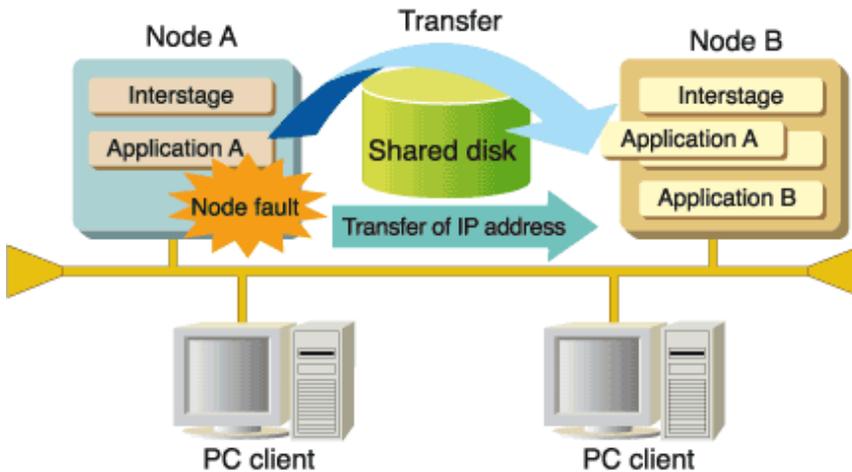
1.2.3 Services that can be Used in a Cluster System

The following table indicates whether the service of the Interstage-provided service group can be used in a cluster system.

Table 1.2 Services that can be used in a Cluster System

Service Name	Can the service be used?					
	Windows32	Windows64	Solaris32	Solaris64	Linux32	Linux64
Interstage HTTP Server	Y	Y	Y (*1)	Y (*1)	Y	Y
Interstage HTTP Server 2.2	Y	Y	Y (*1)	Y (*1)	Y	Y
Servlet Service	Y	Y	Y (*2)	Y (*2)	Y	Y
EJB Service	Y	Y	Y	Y	Y	Y
Interstage JMS	Y	Y	Y	Y	Y	Y
CORBA Service (ObjectDirector)	Y	Y	Y	Y	Y	Y
Event Service	Y	Y	Y (*1)	Y (*1)	Y	Y
Component Transaction Service (TransactionDirector)	Y	-	Y	-	Y	-
Database Linkage Service (ObjectTransactionService)	Y	Y	Y	-	Y	Y
Interstage Single Sign-on	Y	Y	Y (*1)	Y (*1)	Y	Y
Interstage Directory Service	Y	Y	Y (*1)	Y (*1)	Y	Y

Y: Can be used.

-: The function is not supported.

*1 Cannot be used in mutual standby mode.

*2 Session Registry Server cannot be used in mutual standby mode.

Session Registry Server is not valid for Linux (64 bit).

Note

- Windows32/64 Solaris32 Linux32/64

  If a Combined Server is run on a cluster system, mutual standby mode cannot be used. Use fixed standby mode.

- Windows32/64 Solaris32 Linux32/64

  Do not create multilevel IJServers on Managed Servers operating in a cluster system.

Chapter 2 Designing Systems for High-reliability

To implement a high-reliability Interstage system, the following patterns are available:

2.1 Implementing a High-reliability System with the IPCOM

High-reliability Method

Multiple servers act as a single system, executing multiple processes in parallel. Client jobs are automatically distributed. When a server fails, it is disconnected from the system and operation continues with the remaining servers. Using the IPCOM, a similar operation can be performed when a specific WorkUnit terminates abnormally.

Figure 2.1 High-reliability Method
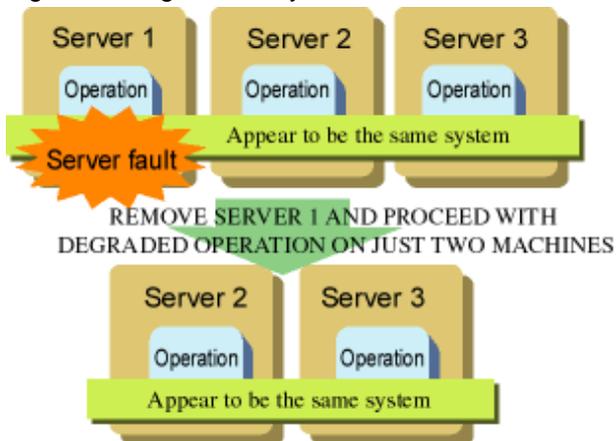
Purpose

To distribute loads among servers automatically.

Features

Multiple servers can be used as a single system, enabling the application load to be distributed, making this function suitable to construct large-scale systems.

It is possible to construct a system that is highly reliable, even in the event of server machine errors. Client loads can be automatically distributed. Servers can be easily added. It is recommended that the server in which IPCOM is set be made highly reliable by using the cluster system.

Time Needed for Operational Recovery

If you are not sharing a DB, operation will not stop just because a node fails. When you do share a DB, it will take less than a minute to recover tables that were being used on a server when it failed. However, this time may increase in line with the number of transactions.

Operation Method

To construct a load balancing system that uses IPCOM, prepare one server on which IPCOM runs for multiple servers. IPCOM performs monitoring. If a node goes down, IPCOM causes degradation.

If the server on which IPCOM is running stops, this stop affects the entire system. It is recommended that the server on which IPCOM runs is duplicated. For the duplication of the server on which IPCOM runs, refer to the IPCOM manual.

Choices for System Configuration

To construct a high-reliability system by using degradation of IPCOM, the following choices are available:

- **DB usage types**

  Any one of the following patterns can be selected: "DB not used" pattern for not using DB, "DB not shared" pattern for not sharing DB between nodes, and "DB shared" pattern for sharing DB between nodes

- **DBMS product used**

  Select a DBMS product according to the DB usage type.

Refer to the following table for details of combinations for degraded operations using IPCOM.

Table 2.1 Combinations for Degraded Operations Using IPCOM

DB usage type	DB not used	DB not shared	DB shared
DBMS used	-	DBMS product	DBMS product that supports DB sharing between nodes
Pattern	Pattern 1	Pattern 2	Pattern 3

Note

You must have a cluster system when you use the sharing DB type.

Mandatory Products

Table 2.2 Mandatory Products

Mandatory products
IPCOM (*1)
DBMS product that supports a cluster system (*2)
Cluster system products to be used and additional products the cluster system requires (*3)

*1 IPCOM for IIOP load balancing is required. For details on IIOP load balancing, refer to one of the IPCOM manuals.

*2 It is unnecessary at the time of DB intact.

Moreover, in DB common use, the DBMS product corresponding to DB common use between nodes is required.

*3 Only in a DB common use type case, it is required.

Pattern 1 (DB Not Used)

The following explains the case in which DBMS is not used for the degradation by IPCOM.

Figure 2.2 Pattern 1 (DB Not Used)
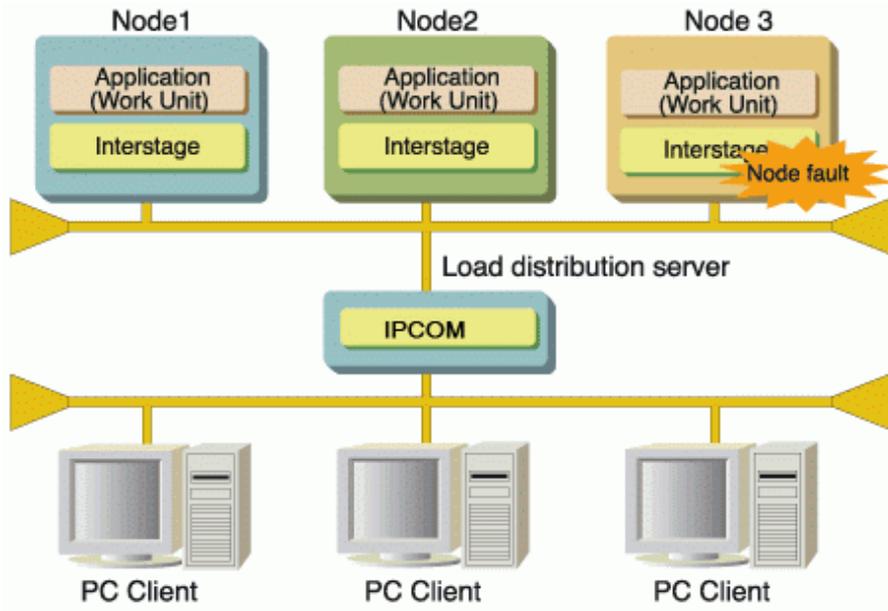
Pattern 2 (DB Not Shared)

The following explains the case in which DBMS is not shared (between nodes) for the degradation by IPCOM:

Figure 2.3 Pattern 2 (DB Not Shared)
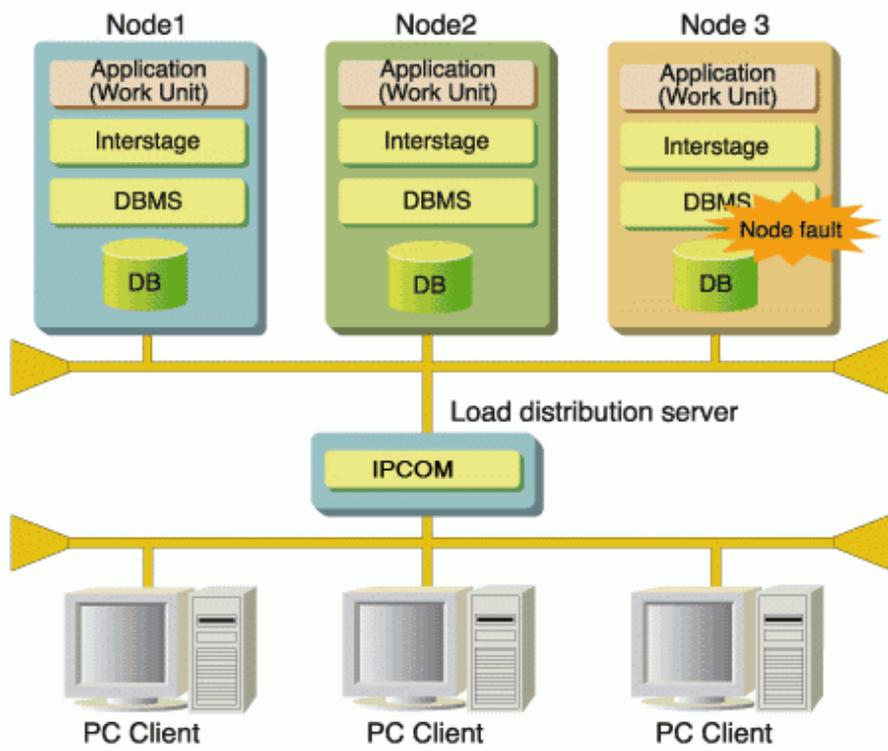
Pattern 3 (DB Shared)

The following explains the case in which DBMS is shared (between nodes) for the degradation by IPCOM:

Figure 2.4 Pattern 3 (DB Shared)
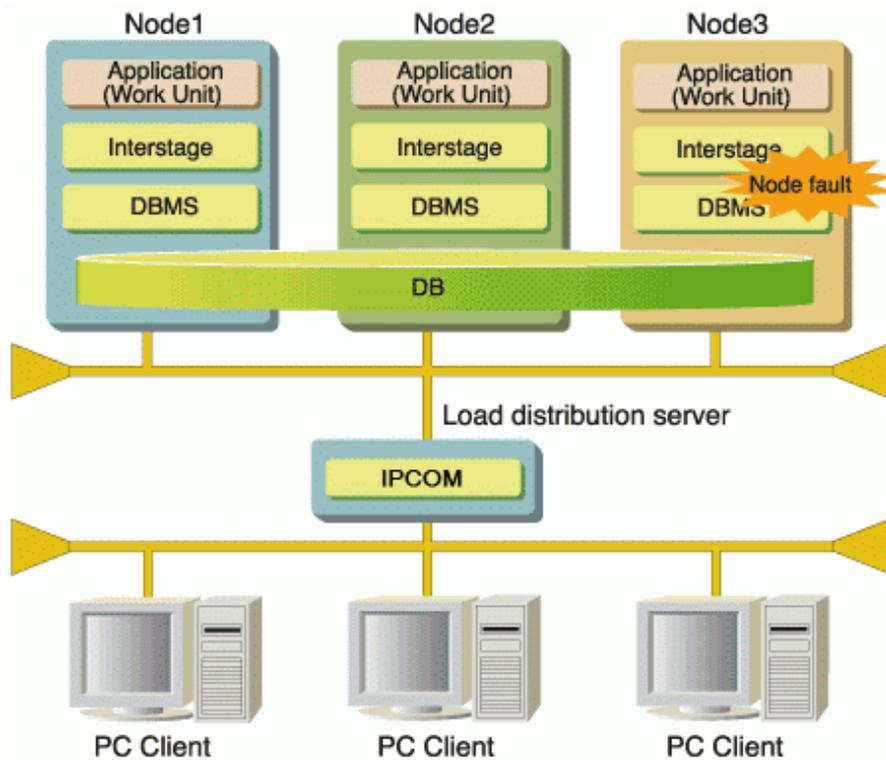
2.2 Implementing a High-reliability System with the Cluster Service Function

High-reliability Method

Prepare a standby server with the same environment as the operational server. When the operational server has a fault, the standby unit takes over operations.

Figure 2.5 High-reliability Method
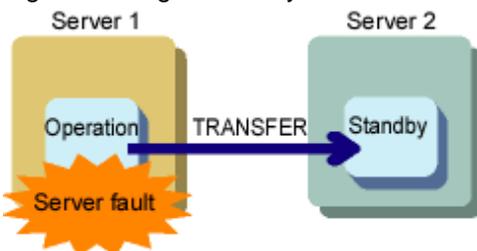
Purpose

To improve overall system robustness.

Features

In addition to server faults, it can also monitor things such as system loops and Interstage failure, making the entire system highly reliable. However, this does require a cluster system environment, making it more expensive than other methods.

Time Needed for Operational Recovery

If an error (such as a hardware error) occurs in a system in which the cluster service function is not used, it takes generally two to three hours before the system can be recovered and the business can be restarted. If the cluster service function is used, the business can be restarted much earlier.

Windows32/64

About 2 to 5 minutes.

Solaris32/64 **Linux32/64**

About 10 to 15 minutes

Operation Method

A cluster system is required when you use the Interstage Application Server Enterprise Edition's cluster service function. A cluster system monitors its nodes and, when it detects a fault on one, it shifts work to the others. Interstage Application Server Enterprise Edition's cluster service function uses a cluster system to make a high-reliability system.

Options for System Configuration

If you use this function to make a high-reliability system, there are the following options.

- Cluster system

  Windows32/64

  MSCS can be used.

  Solaris32/64 **Linux32/64**

  PRIMECLUSTER can be used.

- Cluster mode

  Solaris32/64

  Select either 1:1 fixed standby or mutual standby. However, mutual standby is supported only by PRIMECLUSTER.

- DBMS product

  Select a DBMS product that supports a cluster system.

Refer to the following table for details of cluster system combinations.

Table 2.3 Cluster System Combinations

Cluster product	PRIMECLUSTER		MSCS
Cluster type	1:1 fixed standby	mutual standby	1:1 fixed standby
DBMS	DBMS product that supports a cluster system.		
Pattern	Pattern 1	Pattern 2	Pattern 3

Mandatory Products

Table 2.4 Mandatory Products

Mandatory products
Cluster system products to be used and additional products the cluster system requires
DBMS product that supports a cluster system

Pattern 1 (PRIMECLUSTER + 1:1 Fixed Standby + DBMS)

The following explains an example of a cluster service function configuration that uses the 1:1 fixed standby option of the Fujitsu cluster product PRIMECLUSTER, and DBMS in a cluster system.

Set up the database resource in shared disks. This will allow data inheritance when you switch. The Naming Service and Interface Repository are set up in the local disk.

Figure 2.6 Pattern 1 (PRIMECLUSTER + 1:1 Fixed Standby + DBMS)
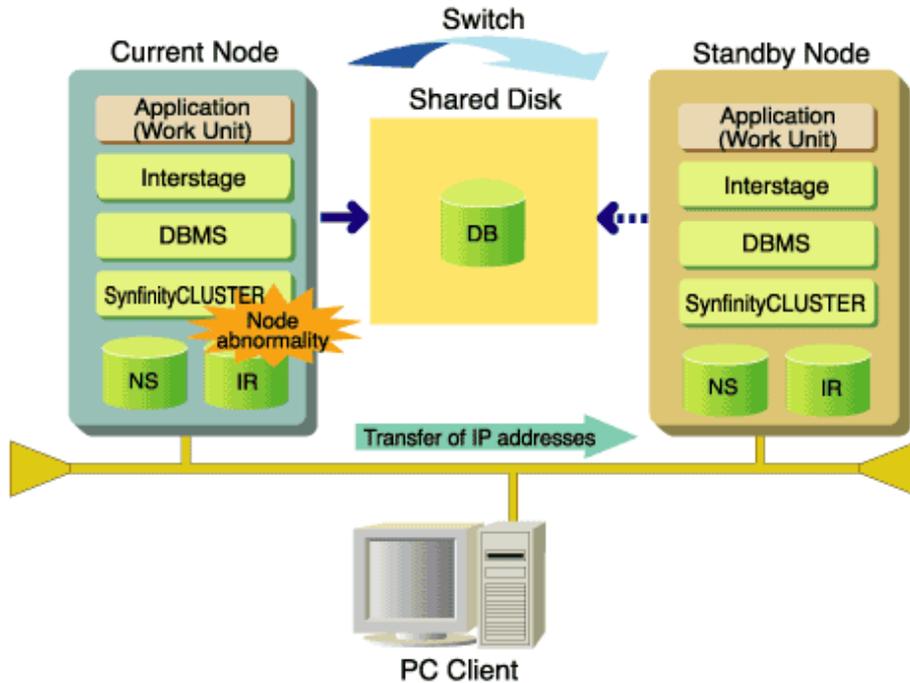
Pattern 2 (PRIMECLUSTER + Mutual Standby + DBMS)

The following explains an example of a cluster service function configuration that uses the mutual standby option of the Fujitsu cluster product PRIMECLUSTER and DBMS in a cluster system.

The Database Linkage Service cannot be used with mutual standby.

Figure 2.7 Pattern 2 (PRIMECLUSTER + Mutual Standby + DBMS)
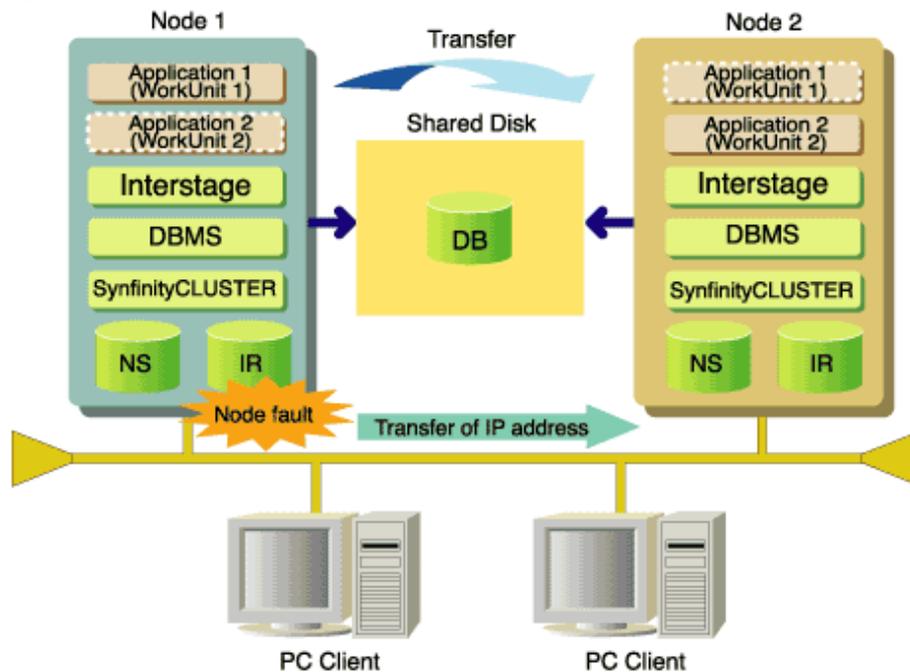
Pattern 3 (MSCS + DBMS)

The following explains an example of a high-availability configuration that uses the Microsoft cluster product MSCS, and DBMS in a cluster system.

Set up the database resource in shared disks. This will allow data inheritance when you switch. The Naming Service and Interface Repository are set up in the local disk.

Figure 2.8 Pattern 3 (MSCS + DBMS)
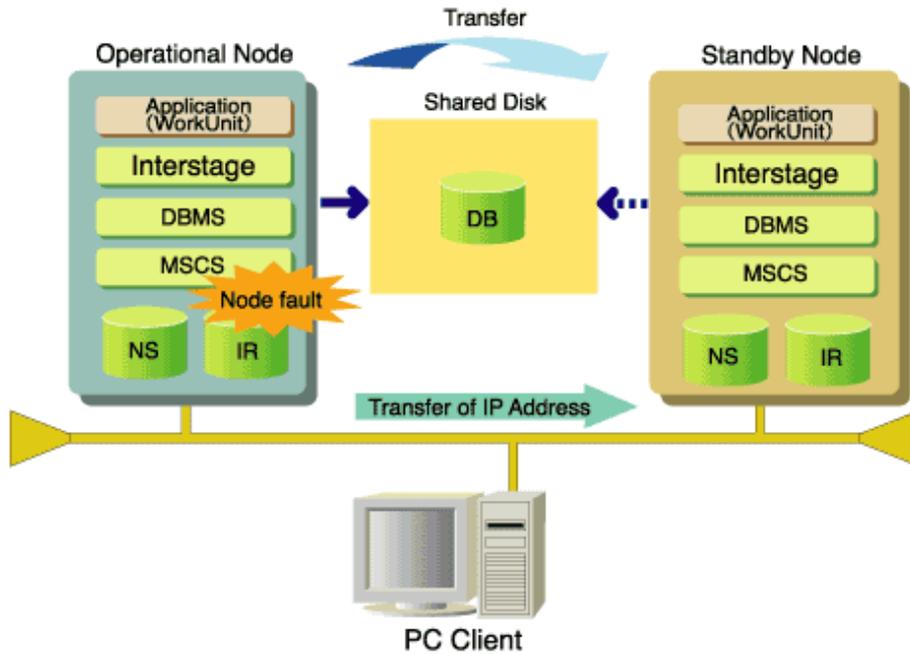
2.3 Recommended High-reliability System Type

To some extent database mapping will determine how you construct a high-reliability system. This section summarizes typical application server system configurations and the recommended high-reliability system to accompany each.

- Using Multiple Servers as a Single System. Mapping the Database Onto Each Server.

- Using Multiple Servers as a Single System. One Database is used with Multiple Servers.

- Using Multiple Servers as a Single System. No Database Used.

- Accessing a Database in a Single Server.

Using Multiple Servers as a Single System. Mapping the Database Onto Each Server.

System type

High-reliability system using degradation by IPCOM

Figure 2.9 High-reliability System using Degradation by IPCOM
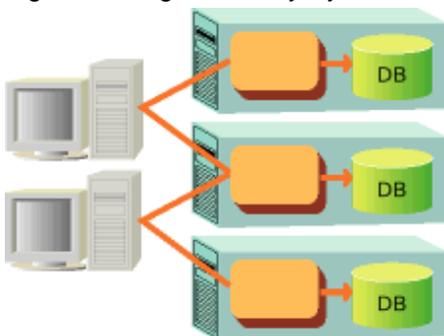
Features

Client loads can be distributed. Good for database referencing.

IPCOM distributes the load dynamically, making it suitable for large-scale systems.

Using Multiple Servers as a Single System. One Database is used with Multiple Servers.

System type

High-reliability system using degradation by IPCOM.

Figure 2.10 High-reliability System using Degradation by IPCOM
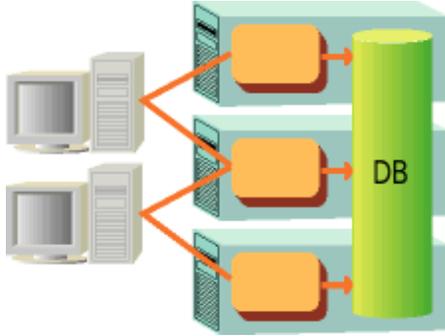
Features

Client loads can be distributed. Suitable for database referencing and updating. However, tables in use when a server fails will enter exclusive wait status until the database system auto-recovers.

IPCOM distributes the load dynamically, making it suitable for large-scale systems.

Using Multiple Servers as a Single System. No Database Used.

System type

High-reliability system using degradation by IPCOM.

Figure 2.11 High-reliability System using Degradation by IPCOM
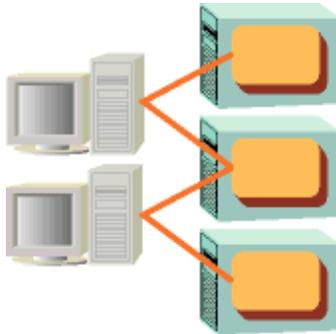
Features

Client load can be distributed. Suited to application server use.

IPCOM distributes the load dynamically, making it suitable for large-scale systems.

Accessing a Database in a Single Server.

System type

High-reliability system using the cluster service function.

Figure 2.12 High-reliability System Using the Cluster Service Function
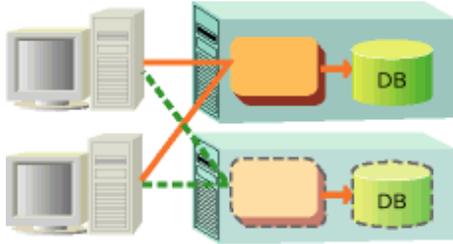
Features

Good for database referencing and updating. The cluster service function makes for very robust system construction.

Chapter 3 Load Distribution Using IPCOM

The load distribution system can be constructed by linking with IPCOM.

IPCOM enables the server group to be seen by a client as a single virtual server machine. When the client calls an object for the server group, IPCOM selects the optimum server machine from the server group and distributes the object call.

When building a load distribution system using Interstage, you are advised to adhere to this system.

The following figure shows an overview of the load distribution environment for transaction application.

Figure 3.1 Load Distribution for a Small System Using IPCOM
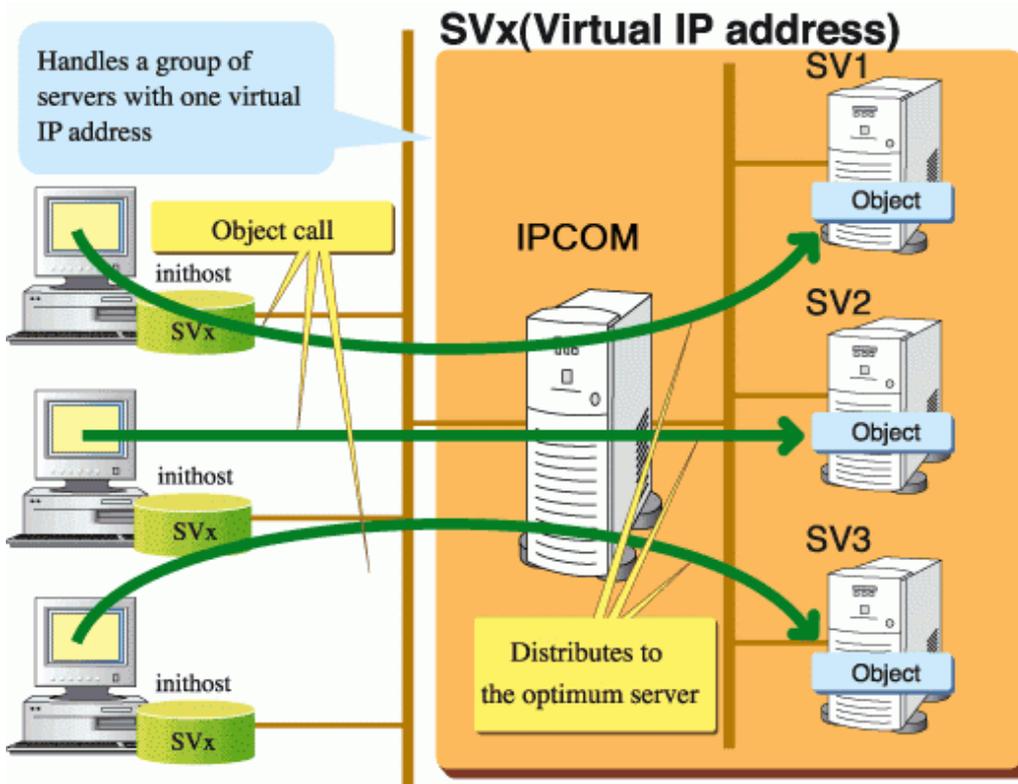
This section explains the following items:

- Programming design

- Environment setup procedure

- Operation procedure

- Operation for server down

- Degraded operation at the time of WorkUnit termination

If the machine on which IPCOM is running stops during operation, the entire system stops. The following describes load distribution of IIOP. For more details, refer to "Protocols Supporting Load Distribution" in the "For a High-performance, High-reliability System" chapter.

Note

- The load distribution function using the IPCOM can only be used in the Enterprise Edition.

- A load balancing function that uses the Event Service is not provided in this product.

3.1 Operation Model

This section explains the following load balancing types:

-

3.1.1 J2EE Model Load Balancing

Interstage Application Server can be linked with IPCOM to set up a load balancing system between Web applications and EJB applications.

If load balancing is executed for an IJServer that runs only Web applications and for an IJServer that runs only EJB applications on separate server machines, it is possible to monitor the operating status of the balancing server machine.

3.1.2 CORBA WorkUnit Load Balancing

Interstage Application Server can be linked with IPCOM to set up a load balancing system between CORBA client and server applications.

3.1.3 Transaction Application Load Balancing `Windows32` `Solaris32` `Linux32`

Interstage Application Server can be linked with IPCOM to set up a load balancing system between transaction application client and server applications.

3.2 Environment Setup

This section explains the following environment settings methods:

3.2.1 J2EE Model

3.2.1.1 Setting the Server

Refer to the IPCOM manual.

3.2.1.2 Setting the IPCOM

Setting the Load Distribution Policy

Refer to the IPCOM manual.

Carry out load distribution policy setting in the following way:

- When configuring the session maintenance settings for the load balancing policy (to guarantee uniqueness), it is recommended that balancing is set for each connection.

- In load balancing for EJB applications, configure the settings so that load balancing is performed per method call.

3.2.1.3 Interstage Setup

Interstage Application Server Setup (Application Server)

Configure Interstage environment settings for each application server according to the following procedure. For details of the Interstage environment settings procedure, refer to the Interstage Management Console Help.

1. Configure the hosts file

2. Stop Interstage

3. Configure the Naming Service

Naming Service Mapping

Interstage Application Server Setup (Load distribution Server Side)

1. Configure the hosts file

   Add the declaration of virtual IP address host name aliases to the host name and IP address of the local server defined in the following hosts file.

   `Windows32/64`

   ```
   Windows installation folder\system32\drivers\etc\hosts
   ```

   `Solaris32/64` `Linux32/64`

   ```
   /etc/hosts
   ```

   After the declaration is added, confirm that the host name is correct by using by the *ping* command.

   The following shows an example of such a description when the host name of a server to be initialized is host1 and the host name of the virtual IP address is vhost. The IP address '192.0.2.21' is the real IP address of host1.

   ```
   192.0.2.21  host1   vhost
   ```

2. Stop Interstage

   In the Interstage Management Console, click [Interstage Application Server] > [System] > [View System Status]. Click [Stop Interstage], and then select Forced System Stop and click [Stop Interstage].

3. Configure the Naming Service

   In the Interstage Management Console, click [Interstage Application Server] > [System]. Click the [Update System Settings] tab > {Detailed Settings [Show]}, and make the following settings in [Naming Service Settings [Show]].

Item		Description
Use Remote Host for Naming Service?	Remote host	Select a remote host
	Server host name	Specify the virtual host name that has been set for the load balancing policy of the IPCOM sites and which corresponds with the virtual IP address.
	Port Number	Specify the representative port value that has been set for the IPCOM CORBA Service. Valid range : 1 to 65535
		If 'Use' is selected for the IPCOM method load balancing settings and the delegate port is omitted, the default value '8002' is used.

Naming Service Mapping

   Use a remote host naming service for the application server.

Interstage Application Server Setup (Load Balancing Server)

When setting the Interstage environment for the servers on which load distribution is performed, follow the procedure described below on every server on which load distribution is performed.

1. Configure the hosts file

   Add the declaration of virtual IP address host name aliases to the host name and IP address of the local server defined in the following hosts file.

   `Windows32/64`

   ```
   Windows installation folder\system32\drivers\etc\hosts
   ```

   `Solaris32/64` `Linux32/64`

   ```
   /etc/hosts
   ```

   After the declaration is added, confirm that the host name is correct by using by the *ping* command.

The following shows an example of such a description when the host name of a server to be initialized is host1 and the host name of the virtual IP address is vhost. The IP address '192.0.2.21' is the real IP address of host1.

```
192.0.2.21  host1   vhost
```

2. Interstage environment setting

The Interstage system environment of the load balancing server can be run using the default environment settings when Interstage is installed.

Naming Service Mapping

Allocate the naming service to all load balancing server machines.

3.2.2 CORBA WorkUnit

3.2.2.1 Setting the Server

Refer to the IPCOM manual.

3.2.2.2 Setting the IPCOM

Setting the Load Distribution Policy

Refer to the IPCOM manual.

Carry out load distribution policy settings as follows:

- When configuring the session maintenance settings for the load balancing policy (to guarantee uniqueness), it is recommended that balancing is set for each connection.

Windows32　Solaris32　Linux32

- When specifying the number of waiting messages as a load distribution policy, specify the following as the monitoring target of waiting messages:

    - Creating a CORBA WorkUnit using the Interstage Management Console

    Click [Interstage Application Server] > [System] > [WorkUnit] > [WorkUnit Name]. Click the [Deploy] tab, and specify the implementation repository ID that was specified in the deployment settings.

    - Creating a CORBA WorkUnit using the *isaddwudef* command

    The implementation repository ID specified in 'impl ID' in the [Application Program] section in the work unit definition.

- Set appropriate values to 'Upper limit number of waiting messages ', 'Recovery number of waiting messages', 'Limit value of communication buffer usage ratio', and 'Recovery value of communication buffer use ratio'.

Interstage Setup Procedure

Set up this product according to the following procedure:

(1) Interstage setup

1. Configure the hosts file

2. Stop Interstage

3. Generate the Interstage system definition

4. Register the Interstage system definition file

5. Customize the Interstage operating environment definition Solaris32/64　Linux32/64

6. Initialize Interstage

(2) Register an object reference (Load Distribution Mode Setting)

One of the following methods is used for this setting:

- Load distribution when an object reference of the Naming Service is obtained

- Load distribution in method calls

(3) Set the client environment

Note

When the Naming Context is used and the 'load distribution in method calls' method is used, set up this product using the following procedure:

(1) Interstage Setup

1. Configure the hosts file

2. Stop Interstage

3. Generate the Interstage system definition

4. Register the Interstage system definition file

5. Set the host name

6. Initialize Interstage

(2) Register an object reference (Load Distribution Mode Setting)

This product is set up using the 'load distribution in method calls' method.

(3) Set the client environment

The setup procedure details are shown below.

Interstage Setup

When setting the Interstage environment for the servers on which load distribution is performed, follow the procedure described below on every server on which load distribution is performed.

Figure 3.2 Interstage Setup
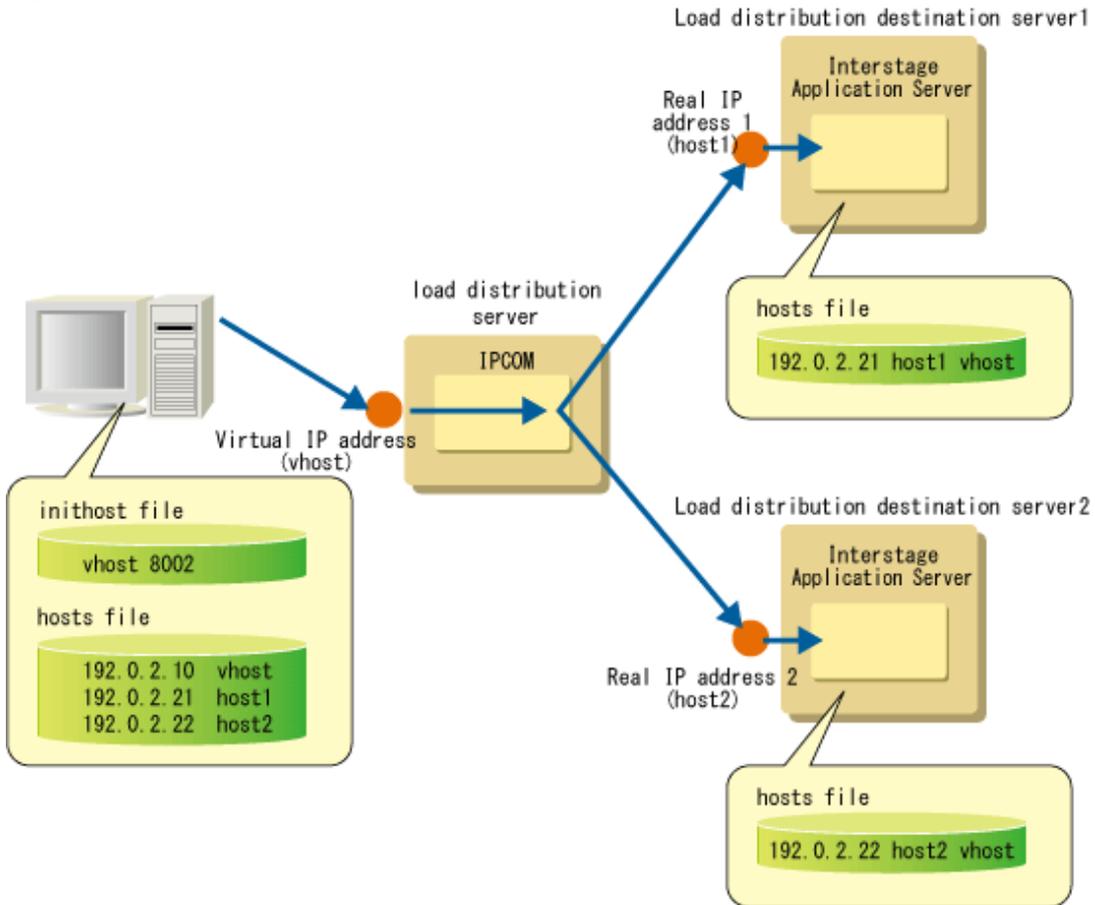
1. Configure the hosts file

   Add the declaration of virtual IP address host name aliases to the host name and IP address of the local server defined in the following hosts file.

   Windows32/64

   ```
   Windows(R) installation folder\system32\drivers\etc\hosts
   ```

   Solaris32/64 Linux32/64

   ```
   /etc/hosts
   ```

   After the declaration is added, confirm that the host name is correct by using by the *ping* command.

   The following shows an example of such a description when the host name of a server to be initialized is host1 and the host name of the virtual IP address is vhost. The IP address '192.0.2.21' is the real IP address of host1.

   ```
   192.0.2.21   host1    vhost
   ```

2. Stop Interstage

   Execute the *isstop-f* command and stop Interstage.

   An example of the *isstop-f* command is shown below.

   ```
   isstop -f
   ```

3. Generate the Interstage system definition

   Execute the *isgendef* command to generate the Interstage system definition.

   The following shows an example of executing the *isgendef* command when the system scale is small:

```
isgendef small
```

4. Register the Interstage system definition file

   Execute the *isregistdef* command to register the Interstage system definition file.

   The following shows an example of executing the *isregistdef* command:

```
isregistdef
```

5. Customize the Interstage operating environment definition `Solaris32/64` `Linux32/64`

   Interstage Operating Environment Definition is stored as follows.

   `Solaris32/64`

```
/opt/FSUNtd/etc/isreg/isinitdef.txt
```

   `Linux32/64`

```
/opt/FJSVtd/etc/isreg/isinitdef.txt
```

   Set the following items for the Interstage operating environment definition:

   - CORBA HOST NAME

   - CORBA PORT NUMBER

   The following is a specification example for each definition item.

```
CORBA HOST NAME=host1
CORBA PORT NUMBER=8003
```

   Note

   Configure this setting if the Naming Context is not going to be used.

6. Set the host name

   Execute the OD_set_env command to set the host name that will be embedded when the object reference is generated.

   The following shows an example of executing the *OD_set_env* command:

```
OD_set_env -n vhost
```

   Note

   Configure this setting if the Naming Context is going to be used.

7. Initialize Interstage

   Execute the *isinit* command to initialize Interstage. For initialization of Interstage, set up the Naming Service on the local server.

   The following shows an example of initialization when the Interstage operating mode is type1.

```
isinit type1
```

Registering an Object Reference (Load Distribution Mode Setting)

The registration method of object reference depends on the mode of load distribution.

Load distribution when an object reference of the Naming Service is obtained

- Creating a CORBA WorkUnit using the Interstage Management Console

  Click [Interstage Application Server] > [System] > [WorkUnit] > [WorkUnit Name]. Click the [Deploy] tab > Show Details [Show]] to display the detailed settings, and set the following items in the [CORBA Application [Show]] settings window.

Item		Description
Use IPCOM load balancing?	Enabled	Select 'Disabled'.
	Disabled	
	Virtual Host Name	Omit this. If set, it will be ignored.
	Representative port	Omit this.If this is set, it will be ignored.

- Creating a CORBA WorkUnit using the isaddwudef command

  Register an object reference by the manual registration by using the *OD_or_adm* command

  Do not specify the object reference host name. If the -h option is specified, set your own host in the object location.

Load distribution in method calls

- Creating a CORBA WorkUnit using the Interstage Management Console

  Click [Interstage Application Server] > [System] > [WorkUnit] > [WorkUnit Name]. Click the [Deploy] tab > Show Details [Show]] to display the detailed settings, and set the following items in the [CORBA Application [Show]] settings window.

Item		Description
Use IPCOM load balancing?	Enabled	If load balancing of IPCOM methods is to be used, select Enabled,' and specify the virtual host name and representative port.
	Disabled	
	Virtual Host Name	Specify the virtual host name that has been set for the load balancing policy of the IPCOM sites and which corresponds with the virtual IP address.
	Representative port	Specify the representative port value that has been set for the IPCOM CORBA Service.
		If 'Use' is selected for the IPCOM method load balancing settings and the delegate port is omitted, the default value '8002' is used.

- Creating a CORBA WorkUnit using the *isaddwudef* command

  Register an object reference by the manual registration by using the *OD_or_adm* command

  When registering a server application object reference in the naming service using the OD_or_adm command, specify the virtual IP address host name in the location specified by the -h option.

Note

If the Naming Context has been used, it must be registered. Copy the information registered in the Naming Service from the copy source server system to the load distribution target server. Accordingly, a Naming Context of the same configuration as the copy source server system Naming Service can be registered as the load distribution target server Naming Service.

The procedure used for the operation to copy the information registered in the Naming Service from the copy source server system "host1" to the load distribution target server"host2" is shown below.

Procedure used for the operation on host1

1. Stop Interstage

   Execute the *isstop -f* command and stop Interstage.

   An example of the *isstop -f* command is shown below.

   ```
   isstop -f
   ```

2. Copy the information registered in the Naming Service

   Copy the directory which stores the information registered in the Naming Service to the working directory.

   An example of how the information registered in the Naming Service is copied to the current directory (working directory) is shown below.

   Windows32/64

```
xcopy C:\Interstage\ODWIN\etc\CosNaming CosNaming_host1
```

Copy the above directory "CosNaming_host1" to the host2 working directory.

Solaris32/64

```
cp -r -p /opt/FSUNod/etc/CosNaming CosNaming_host1
```

Using the ftp command, transfer the above directory "CosNaming_host1" to the host2 working directory in binary format.

Linux32/64

```
cp -r -p /opt/FJSVod/etc/CosNaming CosNaming_host1
```

Using the ftp command, transfer the above directory "CosNaming_host1" to the host2 working directory in binary format.

3. Start Interstage

Execute the *isstart* command and start Interstage.

An example of the *isstart* command is shown below.

```
isstart
```

4. Check the content registered for the Naming Context

Execute the odlistns -l -R command to display the Naming Context registered in the host1 Naming Service. Compare the following [Display Result] and host2 odlistns -l -R command [Display Result], and check whether the Naming Context has been registered using the same configuration in the host1 and host2 Naming Service. Accordingly, it is possible to determine whether the information registered in the Naming Service has been copied correctly.

An example of the *odlist -l -R* command and this display result is shown below.

```
odlistns -l -R
[Display Result]
Name(Type)  Object information(detail)
            Default object information(detail)
NC001(c)    IDL:CosNaming/NamingContextExt:1.0, IDL:CosNaming/NamingContext:1.0, (vhost:
8002:1.0:), context.1225331880.0
```

From the above [Display Result], it is possible to confirm that the Naming Context "NC001" with the Context ID "context. 1225331880.0" has been registered in the host1 Naming Service.

Procedure used for the operation on host2

1. Stop Interstage

Execute the *isstop -f* command and stop Interstage.

An example of the *isstop -f* command is shown below.

```
isstop -f
```

2. Back up and copy the information registered in the Naming Service

Back up the directory which stores the information registered in the host2 Naming Service, then copy the information registered in the host1 Naming Service.

An example of how the information registered in the host1 Naming Service is stored in the current directory (working directory) is shown below.

Windows32/64

```
rename C:\Interstage\ODWIN\etc\CosNaming CosNaming_backup
xcopy CosNaming_host1 C:\Interstage\ODWIN\etc\CosNaming
```

Solaris32/64

```
mv /opt/FSUNod/etc/CosNaming /opt/FSUNod/etc/CosNaming_backup
cp -r -p CosNaming_host1 /opt/FSUNod/etc/CosNaming
chown root /etc/opt/FSUNod/CosNaming
chown sys /etc/opt/FSUNod/CosNaming
```

Linux32/64

```
mv /opt/FSJSVod/etc/CosNaming /opt/FSJSVod/etc/CosNaming_backup
cp -r -p CosNaming_host1 /opt/FSJSVod/etc/CosNaming
chown root /etc/opt/FSJSVod/CosNaming
chown sys /etc/opt/FSJSVod/CosNaming
```

3. Start Interstage

   Execute the *isstart* command and Start Interstage.

   An example of the *isstart* command is shown below.

   ```
   isstart
   ```

4. Check the Naming Context

   Execute the odlistns -l -R command to display the Naming Context registered in the host2 Naming Service.

   An example of the *odlist -l -R* command and this display result is shown below.

   ```
   odlistns -l -R
   [Display Result]
   Name(Type)  Object information(detail)
               Default object information(detail)
   NC001(c)    IDL:CosNaming/NamingContextExt:1.0, IDL:CosNaming/NamingContext:1.0, (vhost:
   8002:1.0:), context.1225331880.0
   ```

   From the above [Display Result], it is possible to confirm that the Naming Context "NC001" with the Context ID "context. 1225331880.0" has also been registered in the host2 Naming Service.

   Accordingly, it is possible to confirm that the Naming Context has been registered using the same configuration in the host1 and host2 Naming Service, and determine that the information registered in the Naming Service has been copied correctly.

Setting the Client Environment

Set in the following files a virtual IP address host name for the reference destination host name of the Naming Service to be set:.

Windows32/64

```
inithost file
```

Solaris32/64 Linux32/64

```
initial_hosts file
```

Note

No IP address can be specified.

Example of correct specification

```
vhost       8002
```

Example of incorrect specification

```
192.0.2.10   8002
```

Also, declare in the following host files the virtual IP addresses and corresponding host names, as well as the host name and the corresponding IP address, of every server to be performing load distribution.

After the declaration is added, confirm that the host name is correct by using the *ping* command.

Windows install folder\system32\drivers\etc\hosts

etc/hosts

The following figure explains the load distribution at the time of obtaining the object reference of the Naming Service.

Figure 3.3 Setting the Client Environment
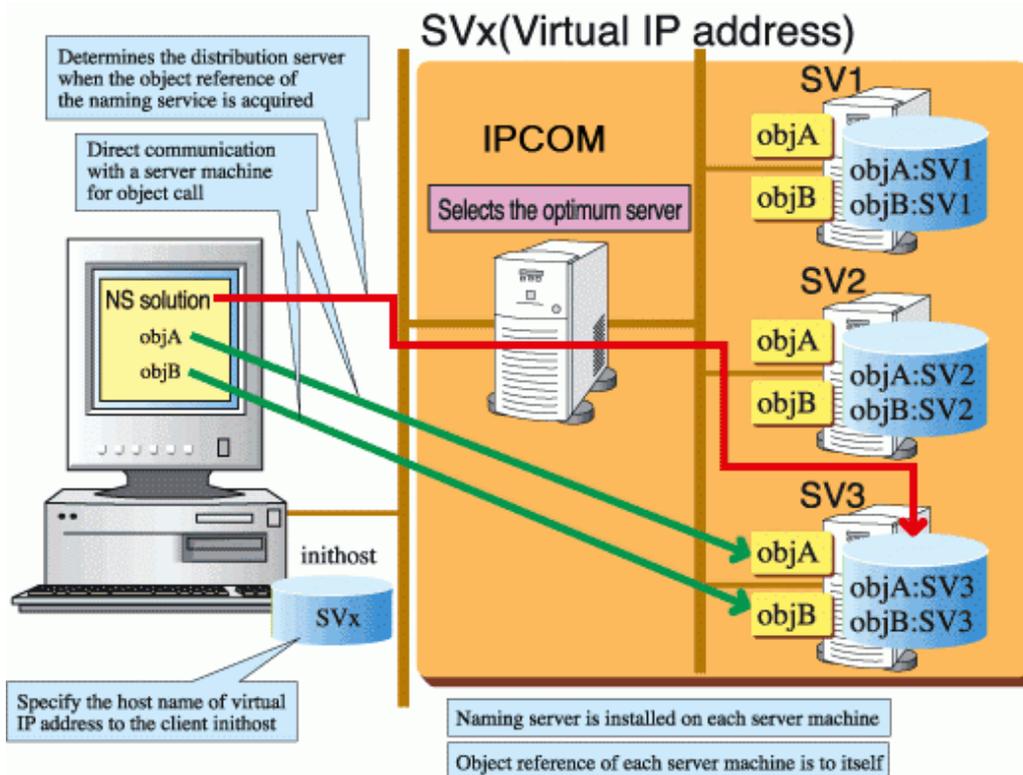
Notes

- To use a server that runs Interstage Application Server (in an environment initialized by the Interstage integration command) as a client, specify the virtual host name as the reference destination host of the Naming Service at initialization.

  For specification details, refer to "Managing the System" in the "System" chapter of the Operator's Guide.

3.2.3 Transaction Application Windows32 Solaris32 Linux32

3.2.3.1 Setting the IPCOM

Setting the Load Distribution Policy

Refer to the IPCOM manual.

Carry out load distribution policy setting in the following way.

- When configuring the session maintenance settings for the load balancing policy (to guarantee uniqueness), it is recommended that balancing is set for each connection.

- When specifying the number of waiting messages as a load distribution policy, specify the following as the monitoring target of waiting messages:

    - The object name specified in 'Destination' in the [Application Program] section in the work unit definition.

- Set appropriate values to 'Upper limit number of waiting messages ', 'Recovery number of waiting messages', 'Limit value of communication buffer usage ratio', and 'Recovery value of communication buffer use ratio'.

Interstage Setup

When setting the Interstage environment for the servers on which load distribution is performed, follow the procedure described below on every server on which load distribution is performed.

Figure 3.4 Interstage Setup
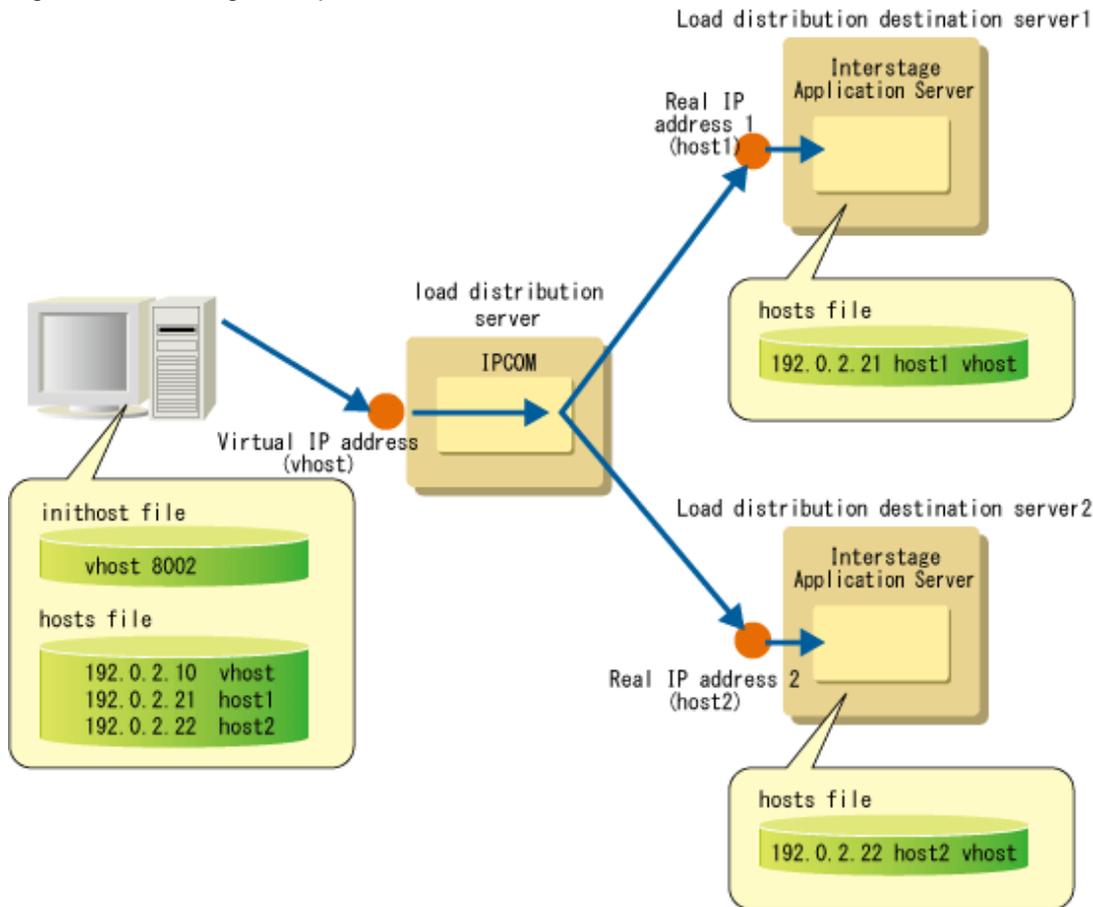
1. Configure the hosts file

   Add the declaration of virtual IP address host name aliases to the host name and IP address of the local server defined in the following hosts file.

   Windows32

   ```
   Windows installation folder\system32\drivers\etc\hosts
   ```

   Solaris32 **Linux32**

   ```
   /etc/hosts
   ```

   After the declaration is added, confirm that the host name is correct by using by the *ping* command.

   The following shows an example of such a description when the host name of a server to be initialized is host1 and the host name of the virtual IP address is vhost. The IP address '192.0.2.21' is the real IP address of host1.

   ```
   192.0.2.21  host1   vhost
   ```

2. Stop Interstage

   Execute the *isstop-f* command and stop Interstage.

   An example of the *isstop-f* command is shown below.

```
isstop -f
```

3. Generate the Interstage system definition

   Execute the *isgendef* command to generate the Interstage system definition.

   The following shows an example of executing the *isgendef* command when the system scale is small:

```
isgendef small
```

4. Register the Interstage system definition file

   Execute the *isregistdef* command to register the Interstage system definition file.

   The following shows an example of executing the *isregistdef* command:

```
isregistdef
```

5. Customize the Interstage operating environment definition Solaris32   Linux32

   Set the following items for the Interstage operating environment definition:

   - CORBA HOST NAME

   - CORBA PORT NUMBER

   The following is a specification example for each definition item.

```
CORBA HOST NAME=host1
CORBA PORT NUMBER=8003
```

6. Initialize Interstage

   Execute the *isinit* command to initialize Interstage. For initialization of Interstage, set up the Naming Service on the local server.

   The following shows an example of initialization when the Interstage operating mode is type1.

```
isinit type1
```

Naming Service Mapping

In the IPCOM linkage function, the load distribution method is determined by the registration of the object reference of the Naming Service.

Registering an Object Reference (Load Distribution Mode Setting)

The registration method of object reference depends on the mode of load distribution.

Load distribution in method calls

Register an object reference by the manual registration by using the *OD_or_adm* command

When registering an object reference of the server application with the Naming Service placed on each server machine using the *OD_or_adm* command, do not set any other host as the object address specified by the -h option.

Registration example of the object reference of the transaction application

```
OD_or_adm -c IDL:TDsample/tdtest:1.0 -a FUJITSU-Interstage-TDLC -n
TDsample::tdtest -h vhost -p 8002
(vhost is a virtual IP address host name)
```

Load distribution when an object reference of the Naming Service is obtained

To register the object reference, use either of the following methods. The registration method can be specified in 'Registration to Naming Service' in the [Control Option] section in the work unit definition.

- Automatic registration by starting the WorkUnit

- Manual registration by *OD_or_adm* command

    When registering an object reference of the server application with the Naming Service placed on each server machine using the *OD_or_adm* command, do not specify the reference destination host name of the object. When specifying the *-h* option, set the local host for the object location. Do not specify the object reference host name. If the *-h* option is specified, set your own host in the object location.

Setting the Client Environment

Set in the following files a virtual IP address host name for the reference destination host name of the Naming Service to be set:.

Windows32

```
inithost file
```

Solaris32　Linux32

```
initial_hosts file
```

Note

No IP address can be specified.

Example of correct specification

```
vhost        8002
```

Example of incorrect specification

```
192.0.2.10   8002
```

Also, declare in the following host files the virtual IP addresses and corresponding host names, as well as the host name and the corresponding IP address, of every server to be performing load distribution.

After the declaration is added, confirm that the host name is correct by using the *ping* command.

Windows32

```
Windows install folder\system32\drivers\etc\hosts
```

Solaris32　Linux32

```
etc/hosts
```

The following figure explains the load distribution at the time of obtaining the object reference of the Naming Service.

Figure 3.5 Setting the Client Environment
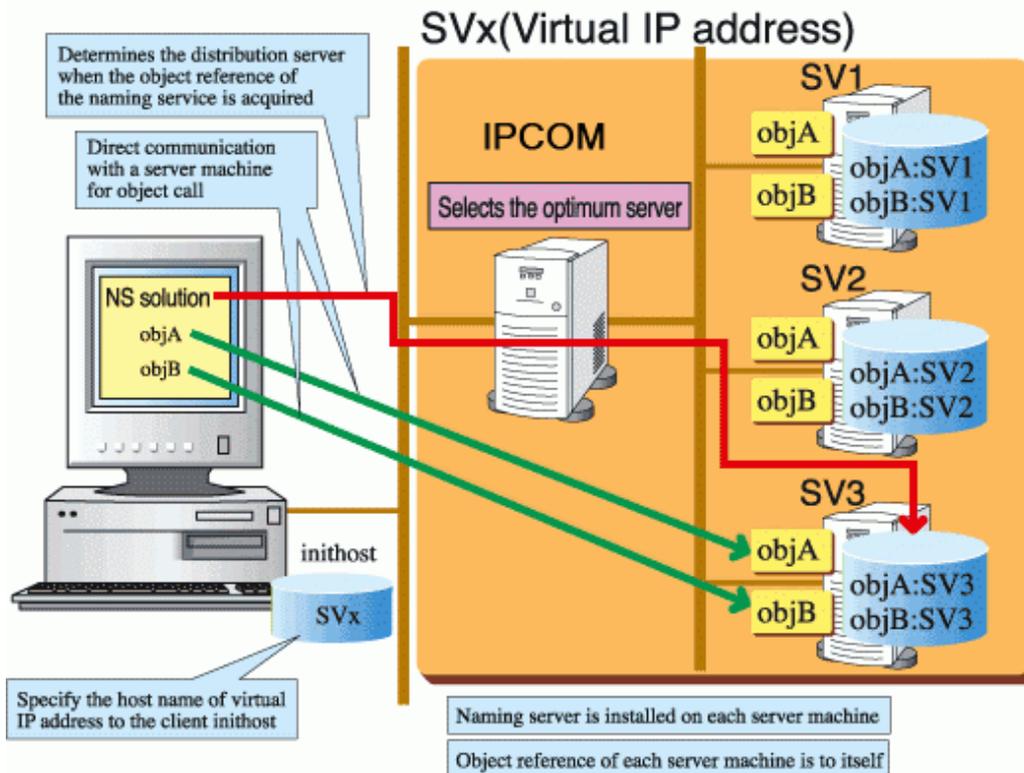
Notes

- The environment setting procedure for linking with IPCOM is different from Interstage Application Server 4.0 and earlier. Though the existing procedure can also be used, the function range is the existing one in this case.

- To use a server that runs Interstage Application Server (in an environment initialized by the Interstage integration command) as a client, specify the virtual host name as the reference destination host of the Naming Service at initialization.

  For specification details, refer to "Managing the System" in the "System" chapter of the Operator's Guide.

3.3 Designing Method of Application

This chapter explains how to design load balancing system applications for linkage with IPCOM in:

- 3.3.1 J2EE Applications
- 3.3.2 CORBA Applications

3.3.1 J2EE Applications

This section explains the deployment and programming method for J2EE applications.

3.3.1.1 Creation of IJServer

The environment settings required for EJB applications to link with IPCOM are explained below.

Environment Setup Procedure

Create an IJServer and specify a virtual host and representative port.

When the IJServer is created (and the virtual host name and representative port corresponding to the virtual IP address for the load balancing policy are specified), the EJB application that is deployed in the created IJServer is automatically set up, so that the EJB application can be linked to IPCOM.

Once the virtual host name and representative port are specified for the IJServer environment settings, the environment settings can be updated.

Application Server

- Select 'Web applications only' as the application server IJServer type and create the server.

Load balancing Server

- Select 'EJB applications only' as the IJServer type for the server intended for load balancing. Specify a virtual host name and delegate port.

   Note

   - Load balancing can only be used if 'EJB applications only' is selected as the IJServer type.

Creating IJServer

Set the following [EJB Container Settings] items from the [Create a new WorkUnit] tab > [Detailed Settings [Show]] of the [WorkUnit] in the Interstage Management Console:

Item		Description
Use IPCOM load balancing?	Enabled	If load balancing of IPCOM methods is to be used, select Enabled,' and specify the virtual host name and representative port. 'No' is the default.
	Virtual Host Name	Specify the virtual host name that has been set for the load balancing policy of the IPCOM sites and which corresponds with the virtual IP address.
	Representative port	Specify the representative port value that has been set for the IPCOM CORBA Service. If 'Use' is selected for the IPCOM method load balancing settings and the delegate port is omitted, the default value '8002' is used.

The EJB Container Settings can also be set using the *isj2eeadmin* command. For details, refer to the Reference Manual (Command Edition).

For Interstage Management Console details, refer to the Interstage Management Console Help.

If an EJB application deployment or an IJServer update fails, a detailed message will be output to the Interstage Management Console.

Note

Load balancing for IJServer WorkUnit EJB applications can only be executed by calling the method.

3.3.1.2 Deployment of Applications

Deployment Procedure

Select [WorkUnit] > [WorkUnit name] on the Interstage Management Console and use the [Deploy] tab. For details, refer to the Interstage Management Console Help.

3.3.1.3 Programming Method

Server-side Design

No special note.

Client-side Design

The following explains the client-side design for each load distribution mode:

Load Distribution in Method Calls

The following describes the timing of load distribution and provides notes for programming on the client side, when load distribution is used in the unit of method calling for EJB applications.

Load distributing timing

The specific server for which load distribution is performed depends on when a *Home interface* method was executed. When an EJB object is acquired by a *Home interface* method, the object is processed in the server to which the *Home interface* method is allocated.

During processing calling a Session Bean, allocation to the same server is assured from the time of execution of a *create* method until execution of a *Remote interface remove* method. Thus, business methods issued between the *create* method and the *remove* method are allocated to the same server.

Notes on programming

As explained above in Load distributing timing, a client application that implements the following pattern will not be load distributed.

- After calling the create method only once, all ensuing processing is executed using the same EJB object.

As a result, the client application should be written to re-call the create method when any new processing requires execution in order to take advantage of load distribution.

3.3.2 CORBA Applications

3.3.2.1 Programming Method

Server-side Design

No special note.

Client-side Design

The following explains the client-side design for each load distribution mode:

Load Distribution in Method Calls

The following describes the timing of load distribution and provides notes for programming on the client side, when load distribution is used in the unit of method calling for EJB applications.

Load Distribution when an Object Reference of the Naming Service is Obtained

The distribution destination server of the object to be called from the client is determined when the Naming Service object reference is acquired. The object reference of the object is acquired by using the object reference of the acquired Naming Service. So, an object call from the client is distributed to the same server until the client re-executes the acquisition of the object reference of the Naming Service.

You are recommended to develop a client application in the following way to re-distribute requests when a destination server stopped working, or to re-adjust distribution when the server that had failed recovers.

- Acquire the object reference of the naming server at any interval.

- Use the object reference of the Naming Service which was acquired regularly and carry out the acquisition process of the object reference for an object.

Exception Processing

In addition to exceptions that may ordinarily occur in communication linked with IPCOM, exceptions specific to IPCOM as shown below may be returned:

- System exception: COMM_FAILURE

- Minor code: 0x464a0114, 0x464a0914

The above minor code is returned when an error is detected in IPCOM (when all the load distribution target servers stop or a communication error is detected in IPCOM).

For details, refer to the "Exception Information Minor Codes Reported from the CORBA Service" chapter of the Messages manual.

Figure 3.6 Client Design
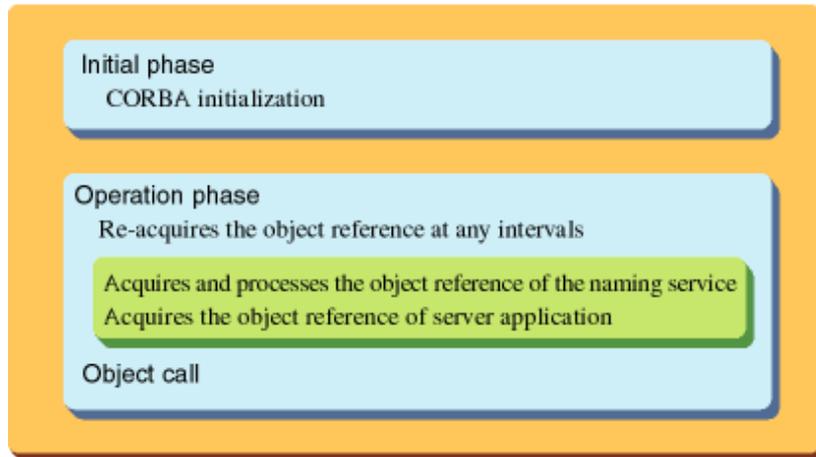
Notes

The following functions cannot be used:

Table 3.1 Functions that Cannot be Used

Unavailable function	Action when this function is used
Function for which iswitch=ON has been specified in the CORBA application information definition file.	The function for which iswitch=ON has been specified in the CORBA application information definition file does not work. Assignment is performed by IPCOM.
Process bind function of the Component Transaction Service	The process bind function does not operate. Assignment is performed by IPCOM and the minor code 10007 is restored to the client application.

3.3.3 Transaction Applications Windows32 Solaris32 Linux32

3.3.3.1 Programming Method

Notes

The following functions cannot be used:

Table 3.2 Functions that Cannot be Used

Unavailable function	Action when this function is used
Process bind function of the Component Transaction Service	The process bind function does not operate. Assignment is performed by IPCOM and the minor code 10007 is restored to the client application.

3.4 Operation Setup

This section explains the following topics:

3.4.1 Degraded Operation at the Time of WorkUnit Termination

If, in a load distributed system whose WorkUnits are linked to the IPCOM, a specific monitored WorkUnit stops abnormally, WorkUnits that are targets of load balancing can continue under degraded operation by separating the server from the load-balancing servers.

When the job operation is performed in the mode where a request from a client application is received by a specific object, and the request is distributed to other objects, the WorkUnit of the object that receives the request from the client application is to be monitored. Another method allows WorkUnits important for jobs to be monitored.

The following diagram illustrates the degraded operation when the WorkUnit terminates abnormally.

Figure 3.7 Degraded Operation upon Abnormal WorkUnit Termination
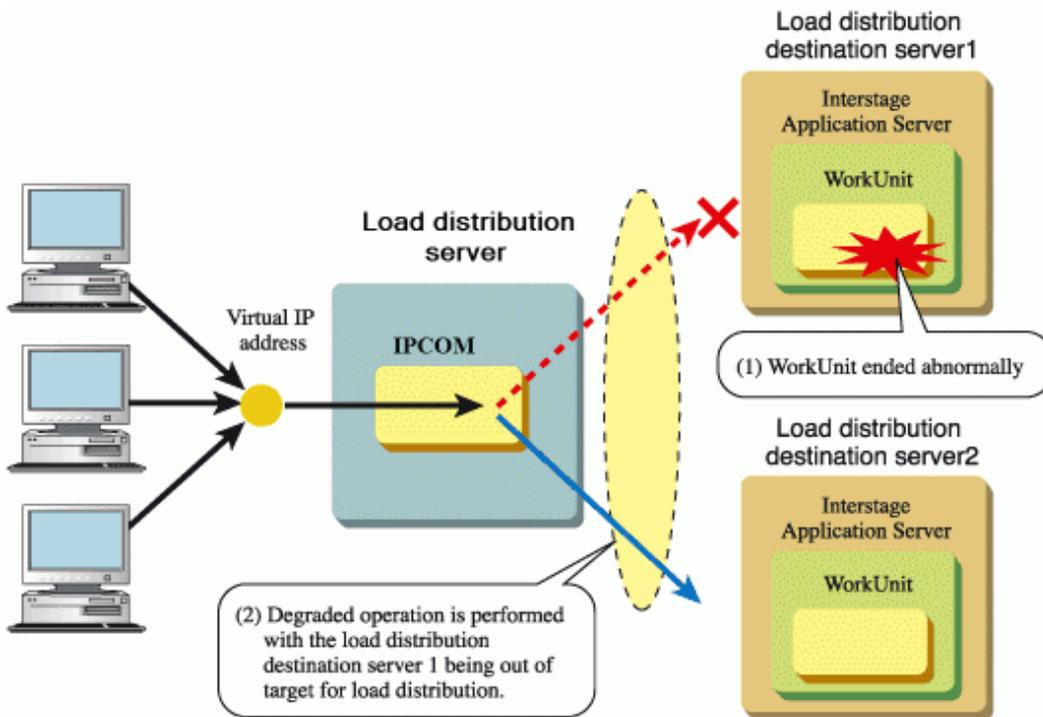
WorkUnits Monitoring Setting

This section explains the settings for switched status monitoring for WorkUnits using IPCOM. Specify whether or not to invoke monitoring for a WorkUnit. Using these settings, multiple WorkUnits can be configured to be monitored.

In the Interstage Management Console, click [WorkUnit] > [Create a new WorkUnit] or [WorkUnit name]. Click the [Detailed Settings [Show]], and configure the following settings in [EJB Container Settings] for IJServer, or in [Settings] tab > [WorkUnit Settings [Show]] for CORBA WorkUnits.

Item		Description
Monitor operating status of the WorkUnit by IPCOM Use IPCOM for monitoring?	Enabled/Disabled Yes/No	Specify whether or not to invoke switched status monitoring for WorkUnits using IPCOM. If 'Yes' or 'Enabled' is selected, and the WorkUnit is stopped, that server is separated from servers that are targets of load balancing and degeneracy is executed. If 'No' or 'Disabled' is selected, degeneracy is not executed, even if the WorkUnit is stopped. For this reason, if IPCOM method load balancing is used, it is recommended that 'Yes' or 'Enabled' is selected for switched status monitoring for WorkUnits. It is also possible to execute switched status monitoring for WorkUnits that are not targets of IPCOM method load balancing.

In IJServers, the WorkUnit settings can also be made using the *isj2eeadmin* command. For details, refer to the Reference Manual (Command Edition).

| Windows32 | Solaris32 | Linux32 |

For transaction applications, set [Traffic Director Monitor Mode: Monitor when using IPCOM linkage] of the WorkUnit definition to [YES]. For details of WorkUnit definitions, refer to the OLTP Server Operator's Guide.

Note

It is only possible to monitor switched status for an IJServer WorkUnit if the IJServer type is as follows:

- IJServer type environment in which EJB applications are operated

- IJServer type environment in which the Web and EJB applications are operated in different JavaVMs

3.4.2 Server-down Operation

In the load distribution at the time of obtaining the object reference of the Naming Service, an object call causes a communication error (COMM_FAILURE) if the assignment destination server is down when the object is called from the client. During re-execution, distribution occurs to a server being operated.

When a server is restored from an error status, a client must re-perform the distribution. So, re-execute the acquisition of the object reference of the Naming Service.

Figure 3.8 Server-down Operation
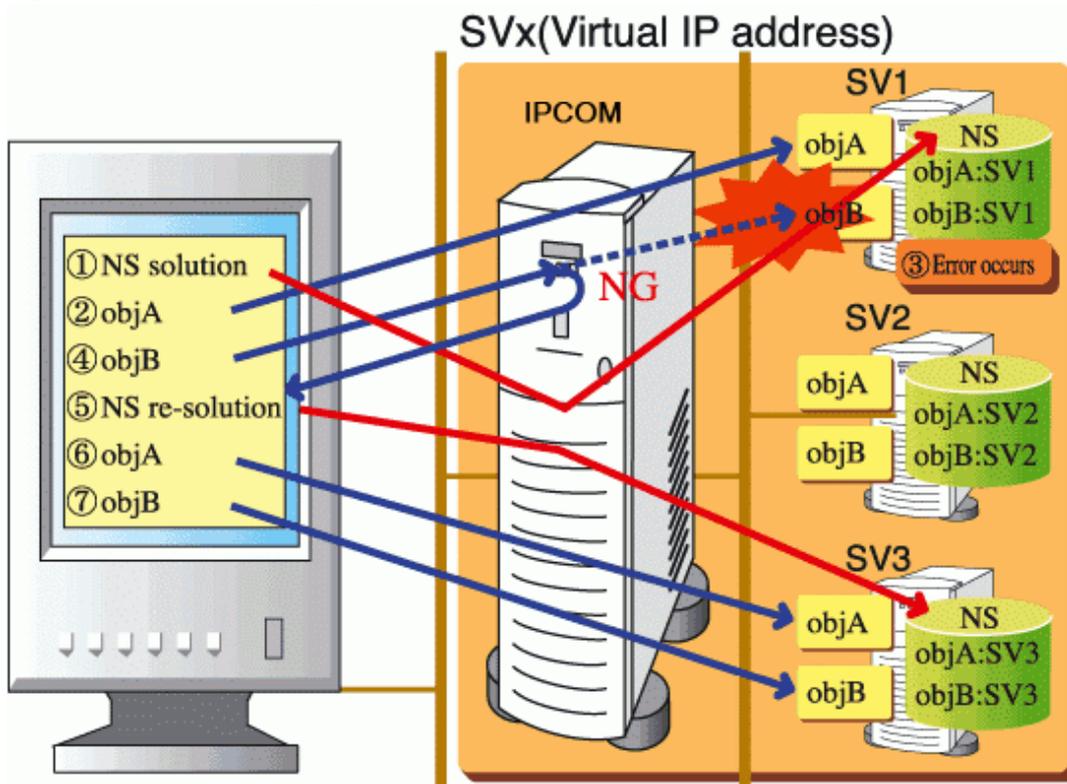
During restoration from server down, switch the down server to a maintenance phase, set the server machine status to operable (in which Interstage and the WorkUnit start and the working environment is prepared), and then release the maintenance phase.

1. Server down

2. Switching to the maintenance phase

3. Remove the cause of the down.

4. Start Interstage.

5. Start the WorkUnit.

6. Release the maintenance phase.

Chapter 4 Environment Setup Procedure for Cluster Service

This chapter describes the cluster service environment setup procedure.

The cluster service is referred to as userApplication in PRIMECLUSTER.

Windows32/64

In this document, for Microsoft(R) Windows Server(R) 2008 or later, read "Failover Clustering" for "MSCS".

4.1 Interstage Resource Structure

An operational node and a standby node contain the following resources:

Figure 4.1 Interstage Resource Structure
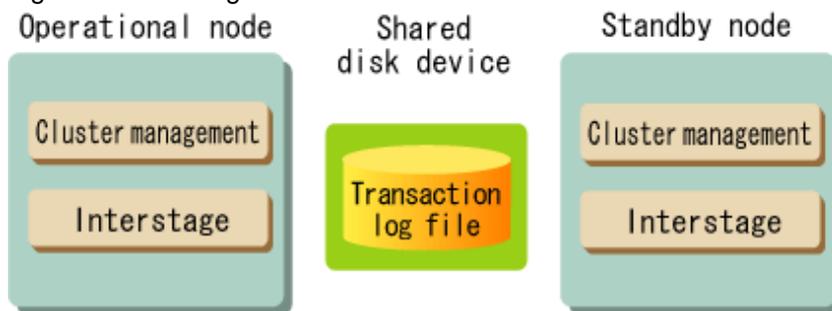
The following Interstage resources are managed in a shared disk unit:

- Transaction log file

- File related to persistent information on Event Channel of Event Service

- Persistent file of the Event Service unit

- JMS persistent file of Interstage JMS

- Interstage Directory Service database

- Interstage Directory Service access log

- Configuration Manager Repository Windows32/64 Solaris32 Linux32/64

- The serialized file of the of Session Recovery function of Servlet Service

- Interstage Single Sign-on encryption information (service ID) file (only when session management is used)

4.2 Environment Setup Procedure

The operation flow for performing an operation in a cluster system is described below.

Operations, such as start, for performing operations in a cluster system are automatically performed by the status transition procedure (method).

Figure 4.2 Operation Flow in a Cluster System
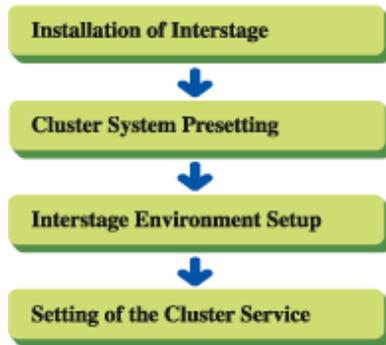
In Solaris Container environments, of the configurations shown in 1-7 of the PRIMECLUSTER Installation and Administration Guide for Oracle Solaris, section H.2.3 "Resource Configuration", all is supported except where "Allocation of non-global zone images" of configuration 5 is "non-shared".

The flow for building a PRIMECLUSTER system in a Solaris Container environment is shown below. Perform the steps that are marked "Y" for the PRIMECLUSTER resource configuration. For single node cluster operations in container environments, configure the settings for Interstage using the "Shared non-global zone images" procedure.

The non-global zone Interstage will start when the global zone RMS is started after the procedure in (6).

Note that the "Application" in "Application monitoring" (the application monitored by PRIMECLUSTER) refers to Interstage Application Server on the non-global zone.

Figure 4.3 Flow for Building a PRIMECLUSTER System in a Solaris Container Environment

Global zone tasks	Non-global Application zone images monitoring			Non-global zone tasks	Non-global Application zone images monitoring			Explanatory Section
	Non-shared	Shared	None		Non-shared	Shared	None	
(1) Creating the global zone								
↓ Refer to the PRIMECLUSTER manuals before configuring the settings.								
(2) Creating the application on the global zone								
↓ Refer to the PRIMECLUSTER manuals before configuring the settings.								
(3) Preparation required before migrating an existing Solaris 10 environment to the non-global zone								
↓ This configuration is not supported, therefore no action is required.								
(4) Creating the non-global zone (if images are non-shared, this task must be performed for the number of nodes that will be used)								
Web-Based Admin View Settings. Refer to the PRIMECLUSTER manuals before configuring the settings. This direction is valid up until this point. The procedure from this point is as shown below.				GLS Settings. Refer to the PRIMECLUSTER manuals before configuring the settings. This direction is valid up until this point. The procedure from this point is as shown below.				
				Installing Interstage	Y	Y	Y	(*6)
				Disabling the Interstage automatic startup settings	Y	Y	-	(*7)
				Interstage environment settings	-	(*1)(*2)	Y (*3)	(*8)
				Cmdline resource/state transition procedure sample correction	Y	Y	-	(*9)
				Registering the state transition procedure as a procedure resource in the resource database	Y	Y	-	(*10)
				Building a cluster application for Interstage	Y (*4)	Y	-	
				Sharing non-global zone configuration information				
(5) Resetting the cluster application on the global zone								
↓ Refer to the PRIMECLUSTER manuals before configuring the settings.								
(6) Setting up Interstage (if images are not shared, this task must be performed for the number of nodes that will be used)								
Taking cluster applications Online	Y	-	-					
				Interstage environment settings	Y (*2)(*5)	-	-	(*8)
							-	
Stopping RMS	Y	-	-					
Starting non-global zones	Y	-	-					
				Stopping RMS	Y	-	-	
				Deleting the cluster application for Interstage		-	-	
				Registering the Cmdline resource/procedure resource in RMS	Y	-	-	(*11)
				Creating the cluster application for Interstage again	Y	-	-	(*12)
Stopping non-global zones	Y	-	-					

PRIMECLUSTER operations   Interstage operations   OS operations

Y: Required

-: Not set

Blank space: Set according to the PRIMECLUSTER Installation and Administration Guide for Oracle Solaris

*1 Set according to "4.5 Interstage Environment Setup for the Server Function". There is no need to start the cluster application.

*2 Even if the multilanguage service or J2EE compatibility are used, when the IP address that is associated with the primary host name (the name that is displayed in the [hostname] command) of the non-global zone and the failover IP address have the same configuration, there is no need to perform the procedure in "4.5.5 Presetting Interstage".

For example,

[Cases in which non-global zone images are shared, and there are exclusive IP zones]

- If the virtual network interface is created by mirroring the non-global zone network (in other words, Gls resources are created in the non-global zone), "4.5.5 Presetting Interstage" will be required.

- When the IP address that is associated with the primary host name of the non-global zone is used as the failover IP address without mirroring the non-global zone network (in other words, Gls resources are not created in the non-global zone), "4.5.5 Presetting Interstage" will not be required.

[Cases in which non-global zone images are not shared, and there are shared IP zones]

- If the IP address for a non-global zone primary host name is allocated in a non-global network interface for a global zone virtual network interface, "4.5.5 Presetting Interstage" will not be required.

- If the IP address for a non-global zone primary host name is not allocated in a non-global network interface for a global zone virtual network interface, "4.5.5 Presetting Interstage" will be required.

Perform "4.5.5 Presetting Interstage" according to the PRIMECLUSTER resource configuration, network configuration, and Interstage features used.

*3 Perform the normal setup, which is not the cluster configuration.

*4 In cases in which non-global zone images are not shared, the cluster applications that are created will only have Fsystem resources (in cases where Interstage uses shared disks) and Gls resources (in cases where there are exclusive IP zones).

*5 Configure the settings according to "4.5 Interstage Environment Setup for the Server Function". Non-global zone cluster applications will go Online (start) as a result of taking global zone cluster applications Online.

*6 Refer to "4.3 Installing Interstage".

*7 Refer to "4.3 Installing Interstage", "4.3.1 Install".

*8 Refer to "4.5 Interstage Environment Setup for the Server Function".

*9 Refer to "4.6.1.1 Modifying the Status Transition Procedures" in this chapter. Also, refer to "Building a Cluster Service Linkage Environment", "Cluster Service Configuration", "PRIMECLUSTER" in the "Running High Reliability Systems" chapter of the Java EE Operator's Guide.

*10 Refer to steps 1 and 2 in "4.6.1.2 Registering the Status Transition Procedures", "Operation Procedure".

*11 Refer to step 3 in "4.6.1.2 Registering the Status Transition Procedures", "Operation Procedure".

*12 Refer to step 4 in "4.6.1.2 Registering the Status Transition Procedures", "Operation Procedure" in this chapter. Also, refer to "Building a Cluster Service Linkage Environment", "Cluster Service Configuration", "PRIMECLUSTER", "Register the Cmdline resources to PRIMECLUSTER", "Cmdline Resource Registration", in the "Running High Reliability Systems" chapter of the Java EE Operator's Guide.

Note

Environments (such as the database environment in which Interstage is used) must have been set before starting to set the Interstage environment. Other work is handled in the same way as it would be when a cluster system is not used.

4.2.1 Mutual-Standby Mode

Solaris32/64

To use mutual standby, there must be two Interstage resource monitors on one machine. For this reason, multi system environment settings must be configured, and both the default system and extended system registered in the cluster service. At this time, configure settings for two Interstage systems based on an operation standby procedure of 1:1, and configure cluster service settings for each Interstage system.

The setup procedure for each Cluster Service is the same as that for 1:1 standby.

Note

If the Admin Server is used on a cluster system, mutual standby cannot be used. Use the operation standby mode instead.

4.2.2 Admin Server, Managed Server Windows32/64 Solaris32 Linux32/64

If the Admin Server or the Managed Server is used on a cluster system, refer to "4.7 Settings for the Admin Server " or "4.8 Settings for the Managed Server ".

4.3 Installing Interstage

The installer is used to install the Interstage package on the two nodes (the operational node and the standby node) which comprise the cluster. The Interstage package needs to be installed with the same path on each of these nodes.

Note

Install the Interstage package on the local disk in each case.

Interstage Management Console control is not inherited in cluster environments. This means that you must log in to the Interstage Management Console again after switching the operation control server. The Interstage Management Console cannot be used in an environment in which inheritance from an operation node to a standby node is assumed.

4.3.1 Install

Windows32/64

After installing the Interstage package, change to "Manual" because the "Startup type" of the following services is set to "Automatic". Additionally, stop these services if they have been started.

- OD_start

- ObjectTransactionService

- FJapache

- Interstage HTTP Server(Web Server name) (*1)

- Interstage Directory Service (repository name)

- INTERSTAGE

- Interstage Java EE Node Agent

*1 Registered if you are running more than one Web server on Interstage HTTP Server.

Note

Among the above-mentioned services, the action is unnecessary for the one that is not installed.

Solaris32/64 Linux32/64

When the Interstage package is installed, the start shell and the stop shell are saved as symbolic link files. The start shell is invoked when the server is started and the stop shell is invoked when the server is shut down. To ensure that the relevant shell is not invoked when the cluster is used, you will need to move them somewhere else - for example to a different directory.

Solaris32/64

- Move S99startis, found in /etc/rc2.d

- Move S99startod, found in /etc/rc2.d

- Move K00stopis, found in /etc/rc0.d

- Move K00stopod, found in /etc/rc0.d

- Move S99FJSVirep, found in /etc/rc2.d

- Move K00FJSVirep, found in /etc/rc0.d

- Move K17FJapache, found in /etc/rc0.d

- Move K17FJapache, found in /etc/rc1.d

- Move K17FJapache, found in /etc/rc2.d

- Move S51FJapache, found in /etc/rc3.d

- Move K17FJapache, found in /etc/rcS.d

Linux32/64

- Move K00stopis, found in /etc/rc0.d

- Move K00stopis, found in /etc/rc1.d

- Move S99startis, found in /etc/rc2.d

- Move S99startis, found in /etc/rc3.d

- Move S99startis, found in /etc/rc4.d

- Move S99startis, found in /etc/rc5.d

- Move K00stopis, found in /etc/rc6.d

- Move K18FJSVirep, found in /etc/rc0.d (*1)

- Move K18FJSVirep, found in /etc/rc1.d (*1)

- Move S82FJSVirep, found in /etc/rc2.d (*1)

- Move S82FJSVirep, found in /etc/rc3.d (*1)

- Move S82FJSVirep, found in /etc/rc4.d (*1)

- Move S82FJSVirep, found in /etc/rc5.d (*1)

- Move K18FJSVirep, found in /etc/rc6.d (*1)

- Move S99startod, found in /etc/rc2.d

- Move S99startod, found in /etc/rc3.d

- Move S99startod, found in /etc/rc4.d

- Move S99startod, found in /etc/rc5.d

- Move K14FJapache, found in /etc/rc0.d

- Move K14FJapache, found in /etc/rc1.d

- Move S86FJapache, found in /etc/rc2.d

- Move S86FJapache, found in /etc/rc3.d

- Move S86FJapache, found in /etc/rc4.d

- Move S86FJapache, found in /etc/rc5.d

- Move K14FJapache, found in /etc/rc6.d

*1 Move once this file has been saved.

4.4 Cluster System Presetting

Before setting up an Interstage environment, perform all required operations - from cluster system installation to setting up the cluster initialization configuration.

Note

- For more information on each cluster system, refer to the manual of each cluster system.

- For related products used by Interstage, follow the installation directions for each product.

4.4.1 Setting up a Shared Disk Unit

Windows32/64

To use the database linkage service, a shared disk must be set up.

Solaris32/64 Linux32/64

Depending on the function to be used, Interstage may use a shared disk unit. It is recommended to set up a shared disk unit.

Set up the shared disk unit so that it can be used as a file system.

As the shared disk unit is required during setup of the Interstage environment, the unit must be registered in the cluster service. (At PRIMECLUSTER, it is userApplication.)

Note

- If Interstage Directory Service is used, prepare an exclusive path for Interstage Directory Service in the shared disk.

  Example:

  `Windows32/64`

  ```
  E:\repository
  ```

  `Solaris32/64` `Linux32/64`

  ```
  /repository
  ```

- If Interstage Single Sign-on with session management is used, prepare an exclusive path for Interstage Single Sign-on on the shared disk.

  Example:

  `Windows32/64`

  ```
  E:\sso
  ```

  `Solaris32/64` `Linux32/64`

  ```
  /sso
  ```

`Solaris32/64`

- When performing a persistent channel operation of the Notification Service in Mutual-Standby mode, the shared disk unit is necessary for both the default system and extended system.

4.4.2 IP Address

Interstage assumes IP address takeovers, so set up a network for takeover in advance.

As the IP address is required during setup of the Interstage environment, the inherited IP address must be registered in the cluster service (At PRIMECLUSTER, it is userApplication.).

To run Admin Server on a cluster system, specify this IP address in the "IP address used by the Admin Server" that was specified at the time of installation.

4.4.3 Inheriting the Node Name

`Solaris32/64` `Linux32/64`

At PRIMECLUSTER, do not use inheriting the node name. Additionally, do not set the same name for the operation node and the standby node. Check the node name according to the following methods:

`Solaris32/64`

Check the contents of /etc/nodename.

`Linux32/64`

Check the value set in "HOSTNAME" of /etc/sysconfig/network.

Note

Since the transition state procedure offered in Interstage is for hot standby, it does not run normally in the environment for inheriting the node name.

4.4.4 Creating a Database Environment

`Windows32/64`

When using a database, also create a database environment.

`Solaris32` `Linux32/64`

If the Database Linkage Service is used, set up the shared disk so that it can be used as a RAW device. At the same time, create disk partitions to create a system log file. For details about the required area size, refer to the Reference Manual (Command Edition).

4.5 Interstage Environment Setup for the Server Function

Construct the Interstage environment within a cluster environment according to the procedures shown below.

1. Starting up the Cluster Service (At PRIMECLUSTER, it is userApplication.)

2. Using the multisystem function, create an Interstage system on each node. `Solaris32/64`

3. Create an Interstage system definition file on each node.

4. Registering the Interstage system definition file on each node.

5. Presetting of Interstage on each node.

6. Initialize Interstage on node 1, the switch node 2 to active and initialize node 2.

7. Performing Persistent Channel Operations of Notification Service

8. Performing Volatile Channel Operations of the Event Service and the Notification Service

9. Using the Database Linkage Service `Windows32/64` `Solaris32` `Linux32/64`

10. Using the Interstage HTTP Server

11. Using the IJServer

12. Using the Session Recovery function of Servlet Service

13. Using the Interstage JMS

14. Using the Interstage Single Sign-on

15. Using the Interstage Directory Service

16. Using an Interstage certificate environment

4.5.1 Starting up the Cluster Service (userApplication)

For Interstage setup, the IP address should be activated and the shared disk should already exist.

The cluster service (At PRIMECLUSTER, it is userApplication.) to which Interstage is set must be created before setup. Also, the shared disk unit and the inherited IP address must be registered in the cluster service, and the cluster service must be started prior to Interstage environment setup.

The initial node 1 is explained as an operational node in the following.

4.5.2 Creating the Interstage System

`Solaris32/64`

Create an Interstage system for the Cluster Service to be created. This work is needed only if the multi system function is used. The Interstage system need not be created if the default system is used.

The Interstage system must be created on the operational node (node 1) and the standby node (node 2).

For the paths used to create the system, the same paths must be present on the operational node (node 1) and the standby node (node 2). For details about creating the Interstage system, refer to the Reference Manual (Command Edition).

Example

```
iscreatesys system1
```

4.5.3 Generating the Interstage System Definition File

The Interstage system definition file is generated in both the operational node and the standby node. It is necessary to consider the system scale of the Interstage installation, which operates on the cluster.

Example

```
isgendef large
```

4.5.4 Registering the Interstage System Definition File

The generated Interstage system definition file is registered in both the operational node and the standby nodes.

Stop Interstage before registering the Interstage system definition file.

For details, refer to "isregistdef" in the "Interstage Setup Commands" chapter of the Reference Manual (Command Edition).

Example

```
isstop -f      Stopping Interstage
isregistdef    Registering the Interstage system definition file
```

4.5.5 Presetting Interstage

When the object reference is generated in the operational node (node 1) and the standby node (node 2), the IP address must be set using the *OD_set_env* command. It is necessary to set the IP address, followed by the cluster service. After changing the settings using the *OD_set_env* command, Interstage must be reinitialized. Refer to the Reference Manual (Command Edition) for details of the *OD_set_env* command.

Example

```
OD_set_env -v Inherited IP address
```

4.5.6 Initializing Interstage

Initialize Interstage on the operational node (node 1) and Interstage on the standby node (node 2) in that order.

Note

If this product was installed with the standard installation, there is no need to initialize this product.

If custom installation was used and the following features were installed for use on a cluster system, initialize Interstage as necessary.

- Multilanguage Service

- J2EE compatibility

4.5.6.1 Initializing Interstage on the Operational Node (Node 1)

Interstage is initialized in the operational node.

When initializing Interstage (executing the *isinit* command), it is necessary to set the Interstage operating definition in advance depending on the function used. In addition, it is necessary to set the operation type by taking the configuration and functions used into account.

For details on the Interstage initialization, refer to "isinit" in the "Interstage Setup Commands" chapter of the Reference Manual (Command Edition).

Note

- Windows32/64   Solaris32   Linux32/64
  To use the Database Linkage Service, the system log file must be generated on the shared disk.

- If using PRIMECLUSTER, do not specify a host name the CORBA service uses (Corba Host Name) in the Interstage operating environment definition.

Example

```
isinit type1
```

4.5.6.2 Initializing Interstage on the Standby Node (Node 2)

Next, switch Node 2 to the operational node, and then initialize Interstage. When doing so, it is necessary to specify "-w" for the command. Node 2 should now have the same settings as node 1 did.

Example

```
isinit -w type1
```

Note

Windows32/64

When Interstage is initialized, the following services might be generated depending on the type. Because the Startup type of the services is "Automatic", change the Startup type to "Manual." In addition, stop any service that is running. Perform these operations for nodes 1 and 2.

- InterfaceRep_Cache Service

- InterfaceRep_Cache_e Service

- Naming Service

- EventService

- EventFactory

4.5.6.3 Checking the Settings

Use the *OD_or_adm* command to check that the CORBA cluster service environment settings, made when Interstage was initialized are correct. Interstage must be started using the *isstart* command before the *OD_or_adm* command can be used. Check this for nodes 1 and 2.

Example

```
> isstart
> OD_or_adm -l
ImplementationRepository    IDL:FJ/ImplementationRep:1.0
                                               (Inherited IP address:8002:1.0:)
FJ_LightInterfaceRepository IDL:FJ/Repository:1.0      (Inherited IP address:8002:1.0:)
FJ_ORB_admin                IDL:OM_ORB/admin:1.0       (Inherited IP address:8002:1.0:)
nill_oref
InterfaceRep                IDL:CORBA/InterfaceRep:1.0 (Inherited IP address:8002:1.0:)
InterfaceRepLock            IDL:CORBA/IrOBF/backup:1.0 (Inherited IP address:8002:1.0:)
InterfaceRepository         IDL:CORBA/Repository:1.0   (Inherited IP address:8002:1.0:)
NameService                 IDL:CosNaming/NamingContextExt:1.0
                                               (Inherited IP address:8002:1.0:)
FJ_LoadBalancingOption      IDL:ISOD/LBO:1.0           (Inherited IP address:8002:1.0:)
InterfaceRep_e              IDL:CORBA/InterfaceRep:1.0 (Inherited IP address:8002:1.0:)
InterfaceRepository_e       IDL:CORBA/Repository:1.0   (Inherited IP address:8002:1.0:)
```

Check the result of executing the command. Check that the host name such as "NameService" and "InterfaceRepository" is the same as the inherited IP address set in Presetting Interstage. For details of the *OD_or_adm* command, refer to the Reference Manual (Command Edition).

Note

When a multi system is operated, and a default system service is called from an extended system, an error may occur while the default system is being initialized. In this case, make sure that the start priority for the transition state procedure of the extended system is lower than the start priority for the default system. For details of changing the start priority, refer to the Cluster System Manual.

4.5.7 Performing Persistent Channel Operations of Notification Service

When performing the persistent channel operations of the Notification Service, a separate process is needed in addition to setting in the initialization of Interstage (*isinit* command).

The required procedure is described below:

When restructuring the environment, refer to "4.12.2 Precautions When Performing Persistent Channel Operation of Notification Service" and execute the following operations after deleting the unit and the Event Channel.

1) Creating the "persistent information associated file of the Event Channel" on the Shared Disk

Create the "persistent information associated file of the Event Channel" on the shared disk of the operational node (node 1) using the *essetcnf* command.

```
essetcnf -f $SWITCH

$SWITCH:   Directory on the shared disk
```

Note

Set it again by using the *essetcnf* command when you change Event Service configuration information in the operational node (node 1).

```
essetcnf -s -edmax 5000
```

2) Preparing the Unit Definition File

The unit definition is prepared under the following directory in the operational node (node 1).

Windows32/64

```
C:\Interstage\eswin\etc\def
```

Solaris32/64

In the case of a default system:

```
/etc/opt/FJSVes/def
```

Solaris32

In the case of an extended system:

```
/var/opt/FJSVisas/system/System-name/FJSVes/etc/def
```

Linux32/64

```
/etc/opt/FJSVes/def
```

For *trandir*, *sysdir,* and *userdir* in the definition file, specify a storage directory in a persistent file on the shared disk. Do not specify a raw device for *trandir*. In addition, the directory is not related to the directory specified using *essetcnf -f*.

3) Set the Persistent File on the Shared Disk

The persistent file is created on the shared disk by the *esmkunit* command in the operational node (node 1).

```
esmkunit
```

4) Starting Interstage

Interstage is started by the *isstart* command in the operational node (node 1).

```
isstart
```

5) Starting the Unit

When the extended unit is used, the extended unit is started by the *esstartunit* command in the operational node (node 1).

When the standard unit is used, this operation need not be done because it is started automatically in "4) Starting Interstage".

```
esstartunit -unit unit name
```

6) Creating the Event Channel

The Event Channel is created by the *esmkchnl* command in the operational node (node 1).

```
esmkchnl -g group name -c channel name -notify -persist all -unit unit name
```

Note

Set it again by using the *essetcnfchnl* command when you change the Event Channel operating environments.

7) Obtaining Event Channel object references from the Naming Service

The Event Channel object reference is obtained from the Naming Service using the *esgetchnlior* command on the operational node (node 1). The created file is stored under the path specified by the -p option. The filename is that specified by the -g option, and has the extension '.ior'.

```
esgetchnlior -g group name -p path name
```

8) Terminating Interstage

Interstage is stopped by the *isstop* command in the operational node (node 1). At this time, it is necessary to set -f option.

```
isstop -f
```

9) Switching of the Cluster Service

The cluster service is switched: Node 2 is made the operational node, and node 1 is made the standby node.

If the cluster service is set to switch when Interstage is stopped, this operation is unnecessary (it will have switched in step 8).

10) The "persistent information associated file of the Event Channel" is created on the Shared Disk.

essetcnf command is executed by -w specification in the operational node (node 2).

```
essestcnf -f $SWITCH -w

$SWITCH: Folder on the shared disk
```

Note

Set the event service in node 2 using the *essetcnf* command as well as in node 1 when you change the Event Service configuration information in node 1 by performing step 1).

```
essetcnf -s -edmax 5000
```

11) Preparation for the Unit Definition File

The unit definition is prepared under the control of the following directory in the operational node (node 2).

Windows32/64

```
C:\Interstage\eswin\etc\def
```

Solaris32/64

In the case of a default system:

```
/etc/opt/FJSVes/def
```

In the case of an extended system:

```
/var/opt/FJSVisas/system/system-name/FJSVes/etc/def
```

```
/etc/opt/FJSVes/def
```

Note

Write the same settings as those generated in step 2) in the unit definition file.

12) Setting the Persistent File into the Shared Disk

esmkunit command is executed by -w specification in the operational node (node 2):

```
esmkunit -w
```

13) Starting Interstage

The Interstage is started up by the *isstart* command in the operational node (node 2).

```
isstart
```

14) Startup of the Unit

When the extended unit is used, the extended unit is started by the *esstartunit* command in the operational node (node 2).

When the standard unit is used, this operation need not be done because it is started automatically in "13) Starting Interstage".

```
esstartunit -unit unit name
```

15) Register the Naming Service

The Event Channel object reference obtained from node 1 is registered with the Naming Service using the *essetchnlior* command on the operational node (node 2). The file name is the same as that created by the *esgetchnlior* command in step 7).

```
essetchnlior -f file name
```

16) Creating the Event Channel

esmkchnl command is executed in the operational node (node 2) by -w specification.

```
esmkchnl -g group name -c channel name -notify -persist all -unit unit name -w
```

Note

Set the event service in node 2 using the *essetcnfchnl* command as well as in node 1 when you change the Event Channel operating environments in the node 1 by performing step 6).

17) Termination of Interstage

In the operational node (node 2), Interstage is stopped by the *isstop* command. It must be stopped with -f specification at this time.

```
isstop -f
```

4.5.8 Performing Volatile Channel Operations of the Event Service and the Notification Service

When performing the volatile operations of the Event Service and Notification Service, a separate process is needed in addition to setting the initialization of Interstage (*isinit* command).

The required procedure is described below.

When restructuring the environment, refer to "4.12.3 Precautions When Performing Volatile Channel Operation of Event Service and Notification Service", then execute the following operations after deleting the unit and the Event Channel.

1) Configuring the Event Service

To configure the Event Service in the operational node (node 1), use the *essetcnf* command.

```
essetcnf -s -edmax 5000
```

2) Starting Interstage

Interstage is started by the *isstart* command in the operational node (node 1).

```
isstart
```

3) Creation of the Event Channel

The Event Channel is created by the *esmkchnl* command in the operational node (node 1).

For the Notification Service:

```
esmkchnl -g group name -c channel name -notify
```

For the Event Service:

```
esmkchnl -g group name -c channel name
```

Note

Use the *essetcnfchnl* command to change the Event Channel operating environments.

4) Obtaining Event Channel object references from the Naming Service

The Event Channel object reference is obtained from the Naming Service using the *esgetchnlior* command on the operational node (node 1). The created file is stored under the path specified by the -p option. The filename is that specified by the -g option, and has the extension '.ior'.

```
esgetchnlior -g group name -p path name
```

5) Stopping Interstage

Interstage is stopped with the *isstop* command in the operational node (node 1). At this time, specify -f option.

```
isstop -f
```

6) Switching the Cluster Service

The cluster service is switched, node 2 is made the operational node and node 1 is made the standby node.

If the cluster service is set to switch when Interstage is stopped, this operation is unnecessary (it will have switched in step 3).

7) Configuring the Event Service

If the Event Service configuration information in node 1 in step 1 was modified, set the same configuration information as for node 1 in the operational node (node 2) using the *essetcnf* command.

```
essetcnf -s -edmax 5000
```

8) Startup of Interstage

Interstage is started with the *isstart* command in the operational node (node 2).

```
isstart
```

9) Registering the Naming Service

The Event Channel object reference obtained from node 1 is registered with the Naming Service using the *essetchnlior* command on the operational node (node 2). The file name is the same as that created by the *esgetchnlior* command in step 4.

```
essetchnlior -f file name
```

10) Creation of the Event Channel

The Event Channel is created with the *esmkchnl* command in the operational node (node 2). At this time, specify -w option.

For the Notification Service:

```
esmkchnl -g group name -c channel name -notify -w
```

For the Event Service:

```
esmkchnl -g group name -c channel name -w
```

Note

Set the event service in node 2 using the *essetcnfchnl* command as well as in node 1 when you change the Event Channel operating environments in the node 1 by performing step 3.

11) Stopping Interstage

In the operational node (node 2), Interstage is stopped with the *isstop* command. It must be stopped with -f specification at this time.

```
isstop -f
```

4.5.9 Using the Database Linkage Service `Windows32/64` `Solaris32` `Linux32/64`

The Database Linkage Service can only be used if the Operational: Standby mode is 1:1. To use the Database Linkage Service, other settings must be made in addition to the Interstage initialization settings (type2 specification using the *isinit* command).

Note

To use a J2EE application that uses JTS, "type2" and "ejb" must be specified using the *isinit* command. For details, refer to the Reference Manual (Command Edition).

4.5.9.1 Procedure when OTS is used by CORBA Applications/Transaction Applications

1) Creation of XA Linkage Program/Resource Management Program/APM

Create the XA Linkage Program and the Resource Management Program.

They must be created using the operational node (Node 1) and the standby node (Node 2) for the corresponding Cluster Service.

The XA Linkage Program must be created using the *otsmkxapgm* command before creating the Resource Management Program using the *otslinkrsc* command.

`Windows32` `Solaris32` `Linux32`

If you are using a transaction application, APM must be created using the *tdlinkapm* command.

For details on these commands, refer to the Reference Manual (Command Edition).

2) Registration of the Resource Management Program

Register the created resource management program. This must be registered using the operational node (Node 1) and standby node (Node 2) for the corresponding Cluster Service. Register all resource management programs that were created. When registering resource management programs using the standby node (Node 2), add -w as the command option.

Before registering the resource management programs, start Interstage, complete registration of all resource management programs, and then stop Interstage.

1. Check the status of the operational node (Node 1)

   If Node 1 is not operational, switch the cluster so that it changes to operational.

2. Start Interstage.

```
isstart
```

3. Register the resource management program in Node 1

```
otssetrsc -a -rf resource definition file
```

4. Stop Interstage on Node 1

```
isstop -f
```

5. Copy the resource definition file

   Copy files under the directory shown below from Node 1 to the same directory on Node 2.

   Windows32/64

```
C:\Interstage\ots\etc\repository
```

   Solaris32

```
/opt/FSUNots/etc/repository
```

   Linux32/64

```
/opt/FJSVots/etc/repository
```

   Note

   When copying, check that Interstage is not running in either the operational node (Node 1) or the standby node (Node 2).

6. Switch the node state

   Switch Node 2 to operational.

7. Start Interstage on Node 2

```
isstart
```

8. Register the Resource Management Program in Node 2

```
otssetrsc -a -w -rf resource definition file
```

   Note

   Specify the -w option in the *otssetrsc* command. If this is not specified, an error is output informing you that the resource management program has already been registered.

9. Stop Interstage on Node 2

```
isstop -f
```

10. Switch Node 1 to operational

    Switch cluster Node 1 to operational to return it to the initialized state.

4.5.9.2 Procedure when JTS is used by J2EE Applications

1) Registration of the JTS resource definition file

Register the JTS resource definition file.

1. Check the status of the operational node (Node 1)

   If Node 1 is not operational, switch the cluster so that it changes to operational.

2. Start Interstage.

   ```
   isstart
   ```

3. Register the resource definition file in Node 1

   ```
   otssetrsc -a -rf resource definition file
   ```

4. Stop Interstage on Node 1

   ```
   isstop -f
   ```

5. Copy the resource definition file

   Copy the files under the directory shown below from Node 1 to the same directory on Node 2.

   Windows32/64

   ```
   C:\Interstage\ots\etc\repository
   ```

   Solaris32

   ```
   /opt/FSUNots/etc/repository
   ```

   Linux32/64

   ```
   /opt/FJSVots/etc/repository
   ```

   Note

   When copying, check that Interstage is not running in either the operational node (Node 1) or the standby node (Node 2).

2) Setting CLASSPATH

When using the resource management program for JTS, set the path in the following file to the class library required for linking with resources:

Windows32/64

```
C:\Interstage\ots\etc\RMP.properties
```

Solaris32

```
/opt/FSUNots/etc/RMP.properties
```

Linux32/64

```
/opt/FJSVots/etc/RMP.properties
```

Example

Windows32/64

```
RecoveryTarget=
Classpath=C:\oracle\jdbc\lib\ojdbc6.jar
```

```
RecoveryTarget=
Classpath=/oracle/jdbc/lib/ojdbc6.jar
```

4.5.10 Using the Interstage HTTP Server

When using Interstage HTTP Server, a separate process is required in addition to Interstage initialization (*isinit* command).

The required procedure is described below.

4.5.10.1 Setting of Environment Definition

The environment definition of the Interstage HTTP Server is set on both the operational node (node 1) and standby node (node 2). It is necessary to have the same environment definition on both nodes.

The Interstage HTTP Server environment definition files should be assumed to have the same directory configuration on the operational node and the standby node.

Set the name (common hostname or IP address) registered in the Domain Name Server (DNS) as the hostname in the ServerName directive of the Interstage HTTP Server environment definition file (httpd.conf), created by each node.

Note

- When using Interstage HTTP Server on a cluster system (For PRIMECLUSTER), use the *ismodifyservice* command to add the Interstage HTTP Server service to the Interstage operating environment, then use the Interstage setup commands (*isstart*, *isstop*) to start and stop Interstage.

```
ismodifyservice -a FJapache
```

- To stop Interstage from switching to a cluster system when Interstage HTTP Server is not running, set "mode1" for the operating status monitoring mode of Interstage operating status using the *ismodifyservice* command.

```
ismodifyservice -m mode1
```

- All services on the system (including applications) must have different port numbers, otherwise the Web Server may not start when switching to the standby node.

4.5.11 Using IJServer

To use IJServer, in addition to settings during Interstage initialization (using the *isinit* command), other settings are required.

The required procedure is described below.

Setting Environment Definitions

Define an IJServer environment on both the operational node (node 1) and the standby node (node2). The same environment need be set up on the operational node (node 1) and standby node (node 2).

Set up IJServer, which was created on each node, in such a way that it is not started automatically during Interstage startup.

To suppress automatic startup, in the Interstage control console, select [Work Unit] > [IJServer Work Unit Name Selection] > [Environment Settings] tab > [Work Unit Setting] and set "Work Unit Automatic Startup."

The *isj2eeadmin* command can also be used to make these settings. For details, refer to the Reference Manual (Command Edition).

Note

A Web server must be set up before using Interstage HTTP Server. For details, refer to "4.5.10 Using the Interstage HTTP Server".

4.5.12 Using the Session Recovery Function of Servlet Service

To run Session Registry Server on a cluster system, it is necessary to make settings other than adding the Interstage initialization settings (using the *isinit* command).

The procedure is described below.

Environment Definition Settings

Session Registry Server runs as an IJServer. Apart from an IJServer for running business Web applications, there is also a Session Registry Server IJServer.

Set the IJServer for running Session Registry Server so that it can run on a cluster system. For details about the IJServer settings, refer to "4.5.11 Using IJServer".

Session Registry Server Environment Definition File Settings

When Session Registry Server is run on a cluster system, session serialization can be enabled to inherit session information when a switch to the standby node is made.

To enable session serialization, make the following definition in the Session Registry Server environment definition file.

```
<param-name>session.store</param-name>
<param-value>on</param-value>
```

Specify a shared disk as the serialized file storage directory.

Make the following definition in the Session Registry Server environment definition file.

```
<param-name>serialize.file.path</param-name>
<param-value>$SWITCH</param-value>

$SWITCH:  This is a directory on the shared disk
```

Note

Solaris32/64

Session Registry Server only supports 1:1 operation standby. It does not support mutual standby applications.

4.5.13 Using the Interstage JMS

When Interstage JMS is used, static Event Channel operation is performed for message transmission and reception. To do so, an environment for the Notification Service must be defined. The procedure for defining the Notification Service varies depending on the operation mode.

When the following function are used. Refer to "4.5.6.3 Checking the Settings".

- Durable Subscription function

- Message persistent function

- Local transaction function

- Global transaction function Windows32/64 Solaris32 Linux32/64

In other cases refer to "4.5.8 Performing Volatile Channel Operations of the Event Service and the Notification Service".

The required procedure is described below.

1) Defining the Environment

Define an environment for Interstage JMS at the operating node (node 1).

2) Allocating a JMS Persistent File

When the Durable Subscription function is used, enter the *jmssetupcluster* command to allocate a JMS persistent file on a shared disk.

```
jmssetupcluster $SWITCH

$SWITCH:  Folder on shared disk
```

3) Switching of the Node

The cluster service is switched: Node 2 is made the operational node, and node 1 is made the standby node.

4) Defining an Environment

Define an environment for Interstage JMS at the operating node (node 2). Define the same environment as that for the standby node (node 1). Register the Connection Factory definition and Destination definition while specifying the same JNDI name and options as those specified for the standby node (node 1).

5) Allocating a JMS Persistent File

When the Durable Subscription function is used, enter the *jmssetupcluster* command with -w specified.

```
jmssetupcluster $SWITCH -w

$SWITCH:  Folder on shared disk
```

4.5.14 Using Interstage Single Sign-on

To use the Interstage single sign-on function, the following must be first be completed:

- Configure the Web server settings. For details, refer to "4.5.10 Using the Interstage HTTP Server".

- When using SSL communication with the repository server, create an Interstage certificate environment. Refer to "4.5.15 Using an Interstage Certificate Environment" for details.

In addition to these operations, there are other tasks that must also be completed. The procedure is explained below.

Note

In the Interstage Single Sign-on, only the repository server (update system when the repository server is configured on multiple machines) supports the cluster system.

4.5.14.1 Environment Setup Method

1) Setup in the operation node (node 1)

1. In the operation node (node 1), construct the SSO repository using the Interstage Management Console. For details on constructing the SSO repository, refer to "4.5.16 Using Interstage Directory Service".

2. In the operation node (node 1), construct the repository server using Interstage Management Console. For details on constructing the repository server, refer to "Repository Server Setup" in the "Environment Setup (SSO Administrators)" chapter of the Single Sign-on Operator's Guide.

3. Change the encryption information (service ID) file specified for "serviceidpath" of the repository server environment configuration file, to the Interstage Single Sign-on path created on the shared disk (*1). For details on the Interstage Single Sign-on shared disk path, refer to "4.4 Cluster System Presetting".

   Example

   Windows32/64

   If the repository server environment configuration file (C:\Interstage\F3FMsso\ssoatcsv\conf\ssoatcsv.conf) is set as shown below, change the encryption information (service ID) file (C:\Interstage\F3FMsso\ssoatcsv\conf\serviceid) to the Interstage Single Sign-on shared disk path.

   ```
   serviceidpath=C:\Interstage\F3FMsso\ssoatcsv\conf\serviceid
   ```

   Solaris32/64 Linux32/64

   If the repository server environment configuration file (/etc/opt/FJSVssosv/conf/ssoatcsv.conf) is set as shown below, change the encryption information (service ID) file (/etc/opt/FJSVssosv/conf/serviceid) to the Interstage Single Sign-on shared disk path.

   ```
   serviceidpath=/etc/opt/FJSVssosv/conf/serviceid
   ```

4. Change "serviceidpath" set in the repository server environment configuration file and the configuration file for using session management, to the path for the encryption information (service ID) file described in step 3 above (*1) (*2).

Example

Windows32/64

Changing the encryption information (service ID) file to the "E:\sso" shared disk

Change "serviceidpath" set in the repository server environment configuration file (C:\Interstage\F3FMsso\ssoatcsv\conf\ssoatcsv.conf) and the configuration file for using session management (C:\Interstage\F3FMsso\ssoatcsv\conf\ssossmgr.conf) as shown below.

```
serviceidpath=E:\sso\serviceid
```

Solaris32/64 Linux32/64

Changing the encryption information (service ID) file to the "/sso" shared disk

Change "serviceidpath" set in the repository server environment configuration file (/etc/opt/FJSVssosv/conf/ssoatcsv.conf) and the configuration file for using session management (/etc/opt/FJSVssosv/conf/ssossmgr.conf) as shown below.

```
serviceidpath=/sso/serviceid
```

5. Back up the resources of the constructed repository server and Interstage HTTP Server. For details on backup, refer to the "Maintenance (Resource Backup)" chapter of the Operator's Guide.

6. Back up the resources of the constructed SSO repository. For details on backing up the SSO repository resources, refer to "4.5.16 Using Interstage Directory Service".

*1 This is unnecessary if session management is not used.

*2 Only the "serviceidpath" part must be edited. If anything else is edited, it might cause the repository server to stop running correctly. Take care when editing the environment definition file.

2) Setup in the Standby Node (Node 2)

1. Restore the SSO repository resources backed up in the operation node (node 1) to the standby node (node 2). For details on restoring the SSO repository resources, refer to "4.5.16 Using Interstage Directory Service".

2. Restore the repository server resources and Interstage HTTP Server resources backed up in the operation node (node 1) to the standby node (node 2). For details on restoration, refer to the "Maintenance (Resource Backup)" chapter of the Operator's Guide.

Notes

- To change the repository server environment setting in the operation node (node 1) after construction, make the same change in the operation node (node 2) to establish the same repository server environment settings.

- To use the repository created in the cluster system as the "SSO repository", in the Interstage Management Console on the standby node (node 2), select [Security] and then [Single Sign-on] from the System menu. Click [Authentication infrastructure] and [Repository server], and on the [Settings] tab click the [Update] button.

- All services on the system (including applications) must have different port numbers set, otherwise the repository server may not start normally.

4.5.14.2 Environment Deleting Method

1. Use the Interstage Management Console to delete the repository server for the active node (node 1).

2. Switch the cluster service, and then use the Interstage Management Console to delete the repository server for the standby node (node 2).

3. Check whether the shared disk contains the encryption information (service ID) file. If it does, delete the file (*1).

4. Delete the SSO repository for each node. Refer to "4.5.16 Using Interstage Directory Service" for details of deleting the SSO repository.

*1 This is unnecessary if session management is not used.

Note

If the procedure used to delete the SSO repository is incorrect, it might cause the switching of the cluster service to fail.

4.5.15 Using an Interstage Certificate Environment

When same site certificates are used on each node during a service using an Interstage certificate environment, the following procedure is required in addition to the procedures followed to setup the service:

Note

Some Certification Authorities (CAs) do not permit use of same site certificates on different nodes, depending on their operation policy. Other CAs may permit the use under certain conditions. Perform the following procedure after confirming that the CA permits use of the site certificates for the planned operation.

If the CA does not permit the use of the same site certificates on different nodes, configure an Interstage certificate environment in each node and use the site certificates of each node, or obtain a site certificate from an CA permitting the planned operation.

4.5.15.1 Environment Setup Method

4.5.15.1.1 Configure an Interstage Certificate Environment on the Operational Node

Configure an Interstage certificate environment on the operational node. For the configuration method, refer to the "Setting and Use of the Interstage Certificate Environment" chapter of the Security System Guide.

4.5.15.1.2 Configure an Interstage Certificate Environment on the Standby Node

Construct an Interstage certificate environment on the standby node by applying the following procedure.

For more information on each command, refer to the Reference Manual (Command Edition).

1) Create an Interstage Certificate Environment.

Create an Interstage certificate environment by using the *scsmakeenv* command, specifying the "-e" option for not creating CSR and a test certificate:

`Windows32/64`

```
scsmakeenv -e -c
```

`Solaris32/64` `Linux32/64`

```
scsmakeenv -e -c -g iscertg
```

2) Register the CA Certificates.

Register all of the certificates from the CA registered on the operational node by using the *scsenter* command.

`Windows32/64`

```
scsenter -n ca -f C:\my_folder\CA.der
```

`Solaris32/64` `Linux32/64`

```
scsenter -n ca -f /usr/home/my_dir/CA.der
```

3) Export (extract) the Site Certificate.

Export the site certificate and private-key by PKCS#12 data from the Interstage certificate environment on the operational node.

If there are multiple site certificates, export all of the site certificates.

`Windows32/64`

```
scsexppfx -n sitecert -f C:\my_folder\MyCert.p12
```

```
scsexppfx -n sitecert -f /usr/home/my_dir/MyCert.p12
```

4) Import (Register) the Site Certificate.

Import the PKCS#12 data exported in step 3 in the Interstage certificate environment of the standby node.

If there are multiple PKCS#12 data items, import all of these data items.

```
scsimppfx -f C:\my_folder\MyCert.p12
```

```
scsimppfx -f /usr/home/my_dir/MyCert.p12
```

5) Register Certificates of Other Sites.

Register all of the certificates of other sites by using the *scsenter* command.

```
scsenter -n othersite -e -f C:\my_folder\otherSite.der
```

```
scsenter -n othersite -e -f /usr/home/my_dir/otherSite.der
```

6) Register CRLs.

Register all of the CRLs registered on the operational node by using the *scsenter* command.

```
scsenter -c -f C:\my_folder\crl.der
```

```
scsenter -c -f /usr/home/my_dir/crl.der
```

Note

When adding or deleting certificates or CRLs after operation starts, add or delete the certificates or CRLs on all of the nodes.

4.5.16 Using Interstage Directory Service

4.5.16.1 Environment Setup Method

The procedures for creating and operating the repository in a cluster environment are described below.

Note

- Interstage Directory Service supports only fixed standby type clustering.

- The cluster environment setup must already be completed.

- The same procedure is used for each cluster environment (administrator access is required).

- Confirm that the directory structures of node1 and node2 match perfectly.

- When creating a new repository, specify the path of the shared disk in the following specified storage destination.

    - Database storage directory

    - Access log storage directory

- In the shared disk path, specify a path that has been registered as a resource.

- To modify a repository environment that has been operated in a cluster environment, from the Interstage Management Console click [System] > [Service] > [Repository] > [Repository Name] > [Settings]. From the Admin Server, click [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository] > [Repository Name] > [Settings]. Then complete the procedure described in '4.5.16.3 Modifying a Repository' below.

Windows32/64

- Specify the port number to a number after 5001 and ensure this number is not used by other application or service.

- Register a resource using Microsoft(R) Cluster Service (MSCS) after completing the steps described in '4.5.16.2 Creating a Repository'.

- A repository operated in the MSCS cluster environment cannot be modified or deleted because the repository cannot be stopped. First, delete the repository resource registered on MSCS and stop the repository.

4.5.16.2 Creating a Repository

This section describes how to create and set up a repository in the Interstage Directory Service in a cluster environment using standalone mode and replication mode.

4.5.16.2.1 Standalone Mode (Using the Standard Database)

This section describes how to create a repository in the Interstage Directory Service running in standalone mode in a cluster environment using the standard database.

The active and standby nodes are assumed to be Nodes 1 and 2 respectively, in the initial cluster environment setup.

To create a repository:

1. Using the connected Interstage Management Console on the active node (Node 1), access [System] > [Service] > [Repository] (from the Admin Server, access [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository]), then [Create a New Repository] tab, make the repository settings in this window, and click [[Show] Detailed Settings].

2. In [Database Storage Directory] and [Access Log Storage Directory] shown in [Detailed Settings], specify the path to a shared disk and click [Create].

3. Windows32/64

   On the active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 2 to a directory.

   Example

   ```
   irepbacksys -d X:\Backup\irep\rep001_back -R rep001 -confonly
   ```

   On the active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 2 to a file.

   Example

   ```
   irepbacksys -f /backup/irep/rep001_back -R rep001 -confonly
   ```

4. Transfer the created backup file to the standby node (Node 2) using ftp or other method.

5. Windows32/64

   Switch clusters and make Node 2 the active node. On the active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 2 from the backup directory.

   Example

   ```
   ireprestsys -d X:\Backup\irep\rep001_back -R rep001 -confonly
   ```

   Solaris32/64 Linux32/64

   Switch clusters and make Node 2 the active node. On the active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 2 from the backup file.

   Example

```
ireprestsys -f /back/irep/rep001_back.tar.Z -R rep001 -confonly
```

6. Using the connected Interstage Management Console on the active node (Node 2), start the repository.

For details of the *irepbacksys* and *ireprestsys* commands, refer to the "Backup Commands" chapter of the Reference Manual (Command Edition).

4.5.16.2.2 Standalone Mode (Using an RDB)

This section describes how to create a repository in Interstage Directory Service running in standalone mode in a cluster environment using an RDB.

The active and standby nodes are assumed to be Nodes 1 and 2 respectively, in the initial cluster environment setup.

To create a repository:

1. Set up the Symfoware Server cluster system. Then, use the failover application standby function.

   For details on setting up Symfoware Server cluster systems, refer to "Setting up fail-over applications" and "Fail-over applications" in the Symfoware Server Cluster Introduction Operator's Guide.

   For details on setting up the Oracle database cluster system, refer to the Oracle database manual.

   When setting up the RDB cluster system, specify the parameters described in the "Creating Databases" chapter of the Directory Service Operator's Guide.

2. Using the connected Interstage Management Console on the active node (Node 1), access [System] > [Service] > [Repository] (from the Admin Server, access [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository]), then [Create a New Repository] tab, make the repository settings in this window, and click [[Show] Detailed Settings].

3. In [Access Log Storage Directory] shown in [Detailed Settings], specify the path to a shared disk and click [Create].

4. Windows32/64

   On the active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 3 to a directory.

   Example

   ```
   irepbacksys -d X:\Backup\irep\rep001_back -R rep001 -confonly
   ```

   Solaris32/64 Linux32/64

   On the active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 3 to a file.

   Example

   ```
   irepbacksys -f /backup/irep/rep001_back -R rep001 -confonly
   ```

5. Transfer the created backup file to the standby node (Node 2) using ftp or other method.

6. Windows32/64

   Switch clusters and make Node 2 the active node. On the active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 3 from the backup directory.

   Example

   ```
   ireprestsys -d X:\Backup\irep\rep001_back -R rep001 -confonly
   ```

   Solaris32/64 Linux32/64

   Switch clusters and make Node 2 the active node. On the active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 3 from the backup file.

   Example

   ```
   ireprestsys -f /back/irep/rep001_back.tar.Z -R rep001 -confonly
   ```

7. Using the connected Interstage Management Console on the active node (Node 2), start the repository.

For details on the *irepbacksys* and *ireprestsys* commands, refer to the "Backup Commands" chapter of the Reference Manual (Command Edition).

4.5.16.2.3 Configuring Replication in a Cluster Environment (Using the Standard Database)

This section describes how to set up the Interstage Directory Service in replication mode in a cluster environment.

In this section, the active and standby nodes are assumed to be Nodes 1 and 2, respectively, in the initial setting of the cluster environment.

Figure 4.4 Configuring Replication when only the Master is in a Cluster Environment

Configuration Common to both Master and Slave

1. Create a repository.

   Using the connected Interstage Management Console on both master and slave machines in the replication configuration, access [System] > [Service] > [Repository] (from the Admin Server, access [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository]), then [Create a New Repository] tab. Ensure the same settings are used for both the master and slave for the following items, and then click the [Create] button:

   - [Repository Name]

   - [Public Directory]

   - [Database Storage Directory]

   To set up replication in a cluster environment, specify the same shared disk path specified for the database and access log storage directories.

Configuring the Slave

1. Using the connected Interstage Management Console on the slave machine, access [System] > [Service] > [Repository] (from the Admin Server, access [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository]), then [Repository: View Status], click the repository created in step 1 and click [[Show] Detailed Settings] in the [Settings] tab.

2. In [Replication settings], select 'Slave' and, in the resulting [Slave Operation Settings] page, set the host name of the machine that will be the master. In [Master host name], specify the physical host names of the active and standby nodes by delimiting them with a comma as shown below, and then click the [Apply] button.

   Example

   ```
   cluster01,cluster02
   ```

   Note

   In [Master host name] of the [Slave operation settings] page, a maximum of two host names can be specified.

3. From the Interstage Management Console, start the repository.

Configuring the Master

1. Using the connected Interstage Management Console on the master machine, access [System] > [Service] > [Repository] (from the Admin Server, access [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository]), then [Repository : View Status], click the repository created in step 1 of this page, and then click [[Show] Detailed Settings] on the [Settings] tab.

2. In [Replication settings], select 'Master'. A new [Replication destination host list] page is displayed.

3. Click the [Add] button and in the resulting [Replication Connection Settings] page, set the appropriate options.

4. Using Backup/Restore, duplicate the repository environment to the master's standby node (Node 2):

   a) `Windows32/64`

   On the master's active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 1 to a directory.

   Example

   ```
   irepbacksys -d X:\Backup\irep\rep001_back -R rep001 -confonly
   ```

   `Solaris32/64` `Linux32/64`

   On the master's active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 1 to a file.

   Example

   ```
   irepbacksys -f /backup/irep/rep001_back -R rep001 -confonly
   ```

   b) Transfer the backed up file to the master's standby node (Node 2) using ftp or other method.

   c) `Windows32/64`

   Switch clusters and make Node 2 the active node. On the master's active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 1 from the backup directory.

   Example

   ```
   ireprestsys -d X:\Backup\irep\rep001_back -R rep001 -confonly
   ```

   `Solaris32/64` `Linux32/64`

   Switch clusters and make Node 2 the active node. On the master's active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 1 from the backup file.

   Example

   ```
   ireprestsys -f /backup/irep/rep001_back.tar.Z -R rep001 -confonly
   ```

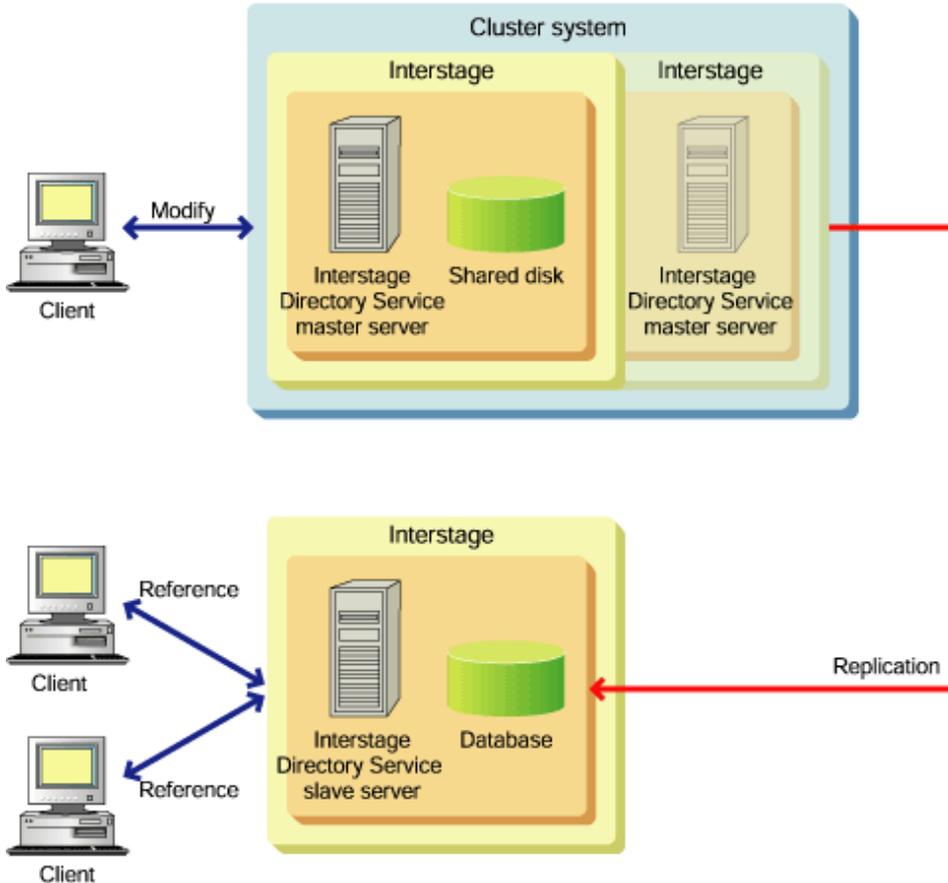5. Using the Interstage Management Console, start the repository on the active node (Node 2).

   For details of the *irepbacksys* and *ireprestsys* commands, refer to the "Backup Commands" chapter of the Reference Manual (Command Edition).

Figure 4.5 Configuring Replication when only the Slave is in a Cluster Environment
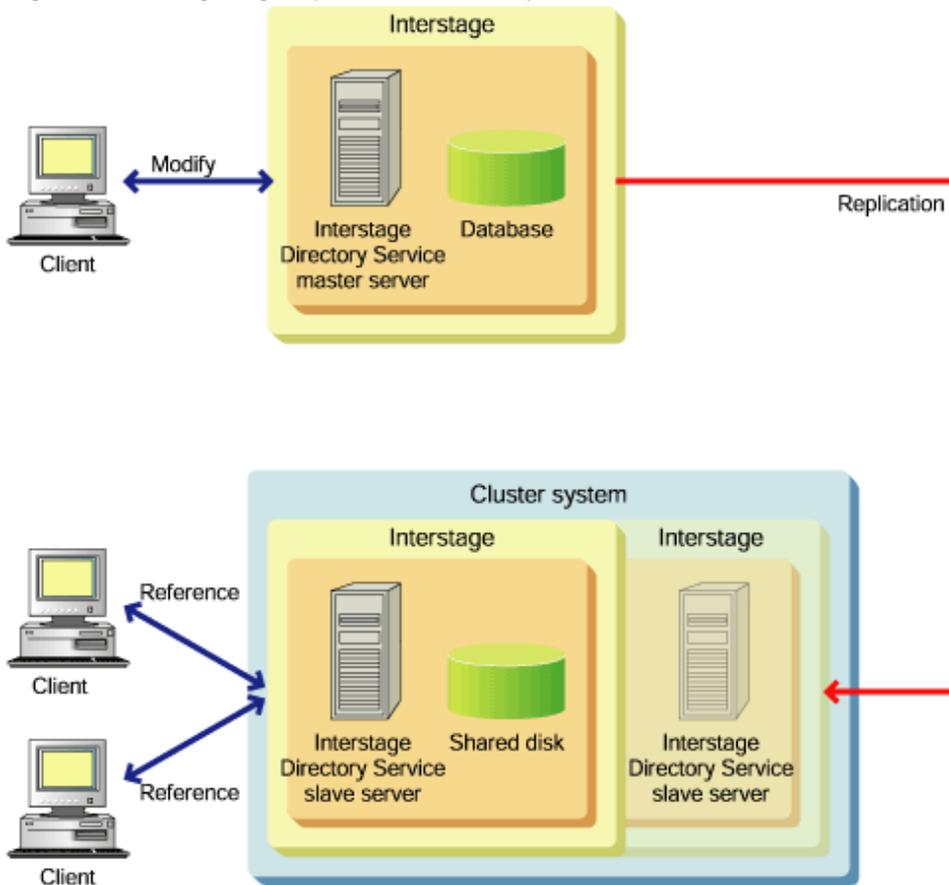
Configuration Common to both the Master and Slave Machines

1. Create a repository.

   Using the connected Interstage Management Console on both master and slave machines in the replication configuration, access [System] > [Service] > [Repository] (from the Admin Server, access [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository]), then [Create a New Repository] tab. Ensure the same settings are used for both the master and slave for the following items, and then click the [Create] button.

   - [Repository Name]

   - [Public Directory]

   - [Database Storage Directory]

   To set up replication in a cluster environment, specify the same shared disk path specified for the database and access log storage directories.

Configuration of the Slave

1. Using the connected Interstage Management Console on the slave machine, access [System] > [Service] > [Repository] (from the Admin Server, access [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository]), then [Repository : View Status], click the repository created in step 1, and then click [[Show] Detailed Settings] in the [Settings] tab.

2. In [Replication settings], select 'Slave' and, in the resulting [Slave Operation Settings] page, set the host name for the master machine.

3. Using Backup/Restore, duplicate the repository environment to the slave's standby node (Node 2):

   a) Windows32/64

   On the slave's active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 1 to a directory.

Example

```
irepbacksys -d X:\Backup\irep\rep001_back -R rep001 -confonly
```

Solaris32/64 Linux32/64

On the slave's active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 1 to a file.

Example

```
irepbacksys -f /backup/irep/rep001_back -R rep001 -confonly
```

b) Transfer the created backup file to the slave's standby node (Node 2) using ftp or other method.

c) Windows32/64

Switch clusters and make Node 2 the active node. On the slave's active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 1 from the backup directory.

Example

```
ireprestsys -d X:\Backup\irep\rep001_back -R rep001 -confonly
```

Solaris32/64 Linux32/64

Switch clusters and make Node 2 the active node. On the slave's active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 1 from the backup file.

Example

```
ireprestsys -f /backup/irep/rep001_back.tar.Z -R rep001 -confonly
```

4. Using the Interstage Management Console, start the repository on the active node (Node 2).

Configuration of the Master

1. Using the connected Interstage Management Console on the master machine, access [System] > [Service] > [Repository] (from the Admin Server, access [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository]), then [Repository : View Status], click the repository created in step 1 and click [[Show] Detailed Settings] in the [Settings] tab.

2. In [Replication settings], select 'Master'. A new [Replication destination host list] page is displayed.

3. Click the [Add] button and, in the resulting [Replication Connection Settings] page, configure the appropriate options. In [Host Name], specify the logical host name of the cluster environment and then click the [Apply] button.

4. Using the Interstage Management Console, start the repository.

For details of the *irepbacksys* and *ireprestsys* commands, refer to the "Backup Commands" chapter of the Reference Manual (Command Edition).

Configuring Replication when both the Master and Slave are in a Cluster Environment

Figure 4.6 Configuring Replication when both the Master and Slave are in a Cluster Environment

Configuration Common to both Master and Slave Machines

1. Create a repository.

   Using the connected Interstage Management Console on both master and slave machines in the replication configuration, access [System] > [Service] > [Repository] (If on the Admin Server, access [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository]), then [Create a New Repository] tab. Ensure that the same settings are used for both the master and slave for the following items, and then click the [Create] button.

   - [Repository Name]

   - [Public Directory]

   - [Database Storage Directory]

   To set up replication in a cluster environment, specify the same shared disk path as the database and access log storage directories.

Configuration on the Slave Machine

1. Using the connected Interstage Management Console on the slave machine, access [System] > [Service] > [Repository] (from the Admin Server, access [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository]), then [Repository: View Status]. Click the repository created in step 1, and then click [[Show] Detailed Settings] in the [Settings] tab.

2. In [Replication settings], select 'Slave' and, in the resulting [Slave Operation Settings], set the host name of the master machine. In [Master host name], specify the physical host names of the master's active node (Node 1) and standby node (Node 2) by delimiting them with a comma as shown below and then click the [Apply] button.

   Example

```
cluster01,cluster02
```

Note

In [Master host name] of [Slave Operation Settings], a maximum of two host names can be specified.

3. Using Backup/Restore, duplicate the repository environment to the slave's standby node (Node 2).

   a) `Windows32/64`

   On the slave's active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 1 to a directory.

   Example

   ```
   irepbacksys -d X:\Backup\irep\rep001_back -R rep001 -confonly
   ```

   `Solaris32/64` `Linux32/64`

   On the slave's active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 1 to a file.

   Example

   ```
   irepbacksys -f /backup/irep/rep001_back -R rep001 -confonly
   ```

   b) Transfer the created backup file to the slave's standby node (Node 2) using ftp or other method.

   c) `Windows32/64`

   Switch clusters and make the slave's Node 2 the active node. On the slave's active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 1 from the backup directory.

   Example

   ```
   ireprestsys -d X:\Backup\irep\rep001_back -R rep001 -confonly
   ```

   `Solaris32/64` `Linux32/64`

   Switch clusters and make the slave's Node 2 the active node. On the slave's active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 1 from the backup file.

   Example

   ```
   ireprestsys -f /tmp/rep001.tar.Z -R rep001 -confonly
   ```

4. Using the Interstage Management Console, start the repository on the slave's active node (Node 2).
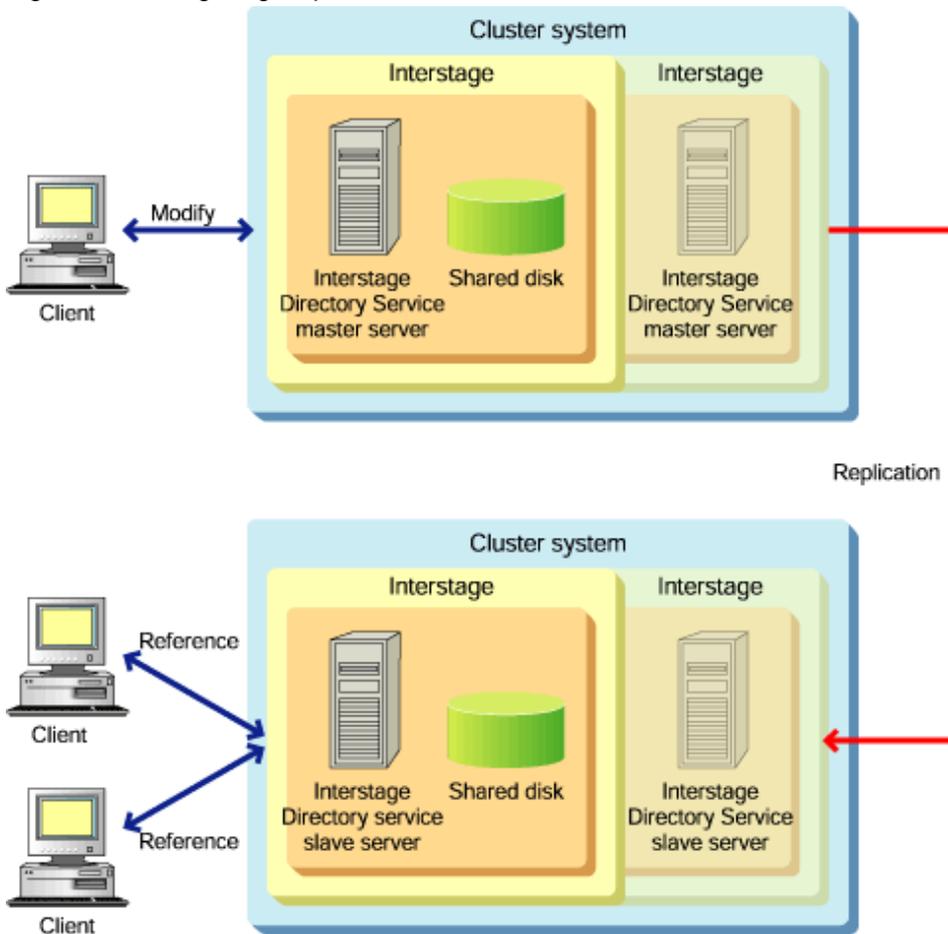
Configuration on the Master Machine

1. Using the connected Interstage Management Console on the master machine, access [System] > [Service] > [Repository] (from the Admin Server, access [Application Management] > [Interstage] > [Interstage Application Server] > [Security] > [Repository]), then [Repository : View Status], click the repository created in step 1, and click [[Show] Detailed Settings] in the [Settings] tab.

2. In [Replication settings], select 'Master'. A new [Replication destination host list] page is displayed.

3. Click the [Add] button and, in the resulting [Replication Connection Settings] page, configure the appropriate options. In [Host Name], specify the logical host name of the cluster environment and click the [Apply] button.

4. Using Backup/Restore, duplicate the repository environment to the master's standby node (Node 2).

   a) `Windows32/64`

   On the master's active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 1 to a directory.

   Example

   ```
   irepbacksys -d X:\Backup\irep\rep001_back -R rep001 -confonly
   ```

On the master's active node (Node 1), execute the *irepbacksys* command to back up the repository information created in step 1 to a file.

Example

```
irepbacksys -f /tmp/rep001 -R rep001 -confonly
```

b) Transfer the created backup file to the master's standby node (Node 2) by ftp or other method.

c) Windows32/64

Switch clusters and make the master's Node 2 the active node. On the master's active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 1 from the backup directory.

Example

```
ireprestsys -d X:Backup\irep\rep001_back -R rep001 -confonly
```

Solaris32/64 Linux32/64

Switch clusters and make the master's Node 2 the active node. On the master's active node (Node 2), execute the *ireprestsys* command to restore the repository information created in step 1 from the backup directory.

Example

```
ireprestsys -f /tmp/rep001.tar.Z -R rep001 -confonly
```

5. Using the Interstage Management Console, start the repository on the master's active node (Node 2).

For details of the *irepbacksys* and *ireprestsys* commands, refer to the "Backup Commands" chapter of the Reference Manual (Command Edition).

4.5.16.3 Modifying a Repository

Active and standby nodes are assumed to be Nodes 1 and 2 respectively, in the initial cluster environment setup.

1. Windows32/64 If modifying a repository operated in the MSCS cluster environment, set the general service resources for the target repository registered to MSCS to offline.

2. Using the Interstage Management Console on the active node (Node 1), stop, modify, and then restart the repository.

3. Switch clusters and make Node 2 the active node. Then, repeat step 1.

4. Windows32/64 If a repository operated in the MSCS cluster environment has been modified, set the resources that were set to offline in step 1 back to online.

4.5.16.4 Deleting a Repository

Active and standby nodes are assumed to be Nodes 1 and 2 respectively, in the initial cluster environment setup.

1. Windows32/64 If deleting a repository operated in the MSCS cluster environment, delete general service resources of the target repository registered to MSCS.

2. Using the Interstage Management Console on the active node (Node 1), stop and delete the repository.

3. Switch clusters and make Node 2 the active node. Then, repeat step 1.

4.6 Setting Cluster Service

This section explains setting the cluster service:

- 4.6.1 For PRIMECLUSTER

- 4.6.2 For MSCS

4.6.1 For PRIMECLUSTER

The environment that relates to the cluster system is set. The following work is performed.

- Modifying the status transition procedures

  They are required at the operational node comprising the cluster system.

- Storing the status transition procedures

  Solaris32/64

  They are required from the cluster employment management view of PRIMECLUSTER.

  Linux32/64

  The procedure need be registered by using the CUI of PRIMECLUSTER.

4.6.1.1 Modifying the Status Transition Procedures

4.6.1.1.1 Sample of the Status Transition Methods

In Interstage, status transition procedures are used to start and terminate Interstage, and to perform switch processing. Samples of status transition procedures are provided in the following directory:

Solaris32/64

```
/opt/FJSVisas/etc/HA/SynfinityCLUSTER
```

Linux32/64

```
/opt/FJSVisas/etc/HA/PRIMECLUSTER
```

You will need to modify the status transition procedures to match the environment.

Interstage provides the following status transition procedure samples:

- IS_INTERSTAGE (for Interstage start/stop)
- OTS_RMP_INTERSTAGE (for the source management program of the Database Linkage Service) Solaris32 Linux32/64
- ODWU_INTERSTAGE (for a CORBA WorkUnit)
- TDWU_INTERSTAGE (for a transition application) Solaris32 Linux32
- UTYWU_INTERSTAGE (for a utility WorkUnit)
- IJSERVER_INTERSTAGE (for IJServer)
- ES_INTERSTAGE (for Event Service)
- IREP_INTERSTAGE (for Interstage Directory Service)
- SSO_INTERSTAGE (for Interstage Single Sign-on) (*1)
- JMX_INTERSTAGE (for operations when the multiserver management feature is used) Solaris32 Linux32/64

*1 This is only used with session management.

4.6.1.1.2 How to Use the Sample

The user modifies the status transition procedure by the following procedure:

1. Copy the status transition procedure.
2. Modify the contents of the status transition procedure.

Note

Give the corrected procedure execution authority..

4.6.1.1.3 Copying the Status Transition Methods

Copy all the status transition procedure samples provided by Interstage to the directory of your choice, with the names as shown below.

```
userApplication name.status transition procedure name
```

It is necessary to correct the following state transition procedures among the state transition procedures copied.

- userApplication name.ODWU_INTERSTAGE

- userApplication name.TDWU_INTERSTAGE   Solaris32   Linux32

- userApplication name.IJSERVER_INTERSTAGE

- userApplication name.OTS_RMP_INTERSTAGE   Solaris32   Linux32/64

- userApplication name.ES_INTERSTAGE

- userApplication name.IS_INTERSTAGE

- userApplication name.UTYWU_INTERSTAGE

- userApplication name.IREP_INTERSTAGE

- userApplication name.SSO_INTERSTAGE (*1)

*1 This is only used with session management.

4.6.1.1.4 Modifying the Content of the Status Transition Methods

How to modify the contents of the copied status transition procedure is described below.

Modifying userApplication name.IS_INTERSTAGE

To prevent switching to standby even if the Naming Service or Interface Repository is stopped due to an error, specify 0 for IS_ISV_WATCH_MODE in the status transition procedure. Normally, 1 is specified and standby is a target of switching. To switch, the value need not be changed.

Example

To not switch the Cluster Service even if the Naming Service or Interface Repository terminates abnormally

```
IS_ISV_WATCH_MODE="0"
```

If the number of file descriptors required to run Interstage HTTP Server exceeds 1024, correct the content so that [ulimit -n <estimated value for the number of file descriptors>] is executed immediately before the Interstage startup process is initiated.

There are two types of startup process. One is used for the active node, and the other for the standby node. Correct both types.

The parameters for the state transition procedures are shown below.

Startup process	Parameter
Active node	$1=START, $2=RUN, $3=AFTER, $5=STARTUP, $6=NONE|SERVICE
Standby node	$1=START, $2=WAIT, $3=AFTER, $5=STARTUP|BUILDIN, $6=NONE|SERVICE

Examples

1. When the estimated number of file descriptors required to run Interstage HTTP Server is 1050

- Active node startup process

  [Before correction]

```
# start
ulimit -c unlimited
$RUN_START_COMMAND > /dev/null 2>&1
```

  [After correction]

```
# start
ulimit -c unlimited
ulimit -n 1050
$RUN_START_COMMAND > /dev/null 2>&1
```

- Standby node startup process

  [Before correction]

```
# wait start
ulimit -c unlimited
$WAIT_STBY_COMMAND > /dev/null 2>&1
```

  [After correction]

```
# wait start
ulimit -c unlimited
ulimit -n 1050
$WAIT_STBY_COMMAND > /dev/null 2>&1
```

4.6.1.1.5 Modifying the userApplication name.ODWU_INTERSTAGE/userApplication name.TDWU_INTERSTAGE/userApplication name.IJSERVER_INTERSTAGE/ userApplication name.UTYWU_INTERSTAGE

The relevant status transition procedure specifies the WorkUnit to be started up on the operational node, and the WorkUnit to be inherited when a switch is performed.

Status transition procedure for every WorkUnit is as follows.

userApplication name.ODWU_INTERSTAGE for a WorkUnit of a CORBA application

userApplication name.TDWU_INTERSTAGE for a WorkUnit of a transaction application   Solaris32   Linux32

userApplication name.IJSERVER_INTERSTAGE for a WorkUnit of an IJServer

userApplication name.UTYWU_INTERSTAGE for a utility WorkUnit.

WorkUnit Names

Code the WorkUnit names in WU_NAME in the status transition procedure as shown below:

Example

To start and advance-start WU1 and WU2:

```
WU_NAME= WU1 WU2
```

Note: Use spaces as delimiters between WorkUnit names.

4.6.1.1.6 Modifying Service Name.OTS_RMP_INTERSTAGE   Solaris32   Linux32/64

The source management program started up in the operational node and the source management program succeeded when switching are set by the state transition procedure. Therefore, describe the source management program name in "DEF_NAME" in the state transition procedure as follows.

Example

To start /home/ots/resource1 with resource1 and /home/otsresource2 with resource2:

```
DEF_NAME=/home/ots/resource1 resource1 /home/ots/resource resource2
```

When using JTS, set "TRUE" as "JTS_USE."

Example

To use JTS in a cluster environment

```
JTS_USE="TRUE"
```

Note: Delimit the described item by a half-width space.

4.6.1.1.7 Modifying Service Name.ES_INTERSTAGE

Use the relevant status transition procedure to specify the Event Channels that are started on the operational node and operate at switching, when setting the Event Channel to 'When the event service starts, the Event Channel is not automatically activated'. Specify the names of the Event Channel groups that are to be started in ES_CHNL during status transition.

The above procedure need not be set when setting all Event Channels to 'when the event service starts, the Event Channel is automatically activated', which is an initial value (default value).

The unit is automatically activated in the Event Channel for persistent channel operation when the Event Channel starts. Hence, it is not necessary to specify the names of the units to be started in ES_UNIT during status transition.

Example

To start the Event Channel groups mix1, mix2, and mix3

```
ES_CHNL="mix1 mix2 mix3"
```

Note: Delimit the described item by a half-width space.

4.6.1.1.8 Modifying userApplication Name.IREP_INTERSTAGE

Specify the path to the shared disk to be used.

Example

Windows32/64

When the shared disk path is specified as "E:\repository"

```
DIRECTORY=E:\repository
```

Solaris32/64

When the shared disk path is specified as "/repository"

```
DIRECTORY=/repository
```

Note: Specify a path that has already been registered as a resource.

4.6.1.2 Registering the Status Transition Procedures

4.6.1.2.1 Setting the Resource Information

When setting the resource information of the state transition procedure, set the "process reactivation frequency" to "0", the "process reactivation interval" to "0", and the "initialization of process reactivation frequency" to "No".

The *claddprocrsc* command is used to set it up. Refer to the PRIMECLUSTER manual about this command.

4.6.1.2.2 Setting the State Transition Instruction Timing

In "Setting the state transition instruction timing", set for the offered state transition procedure to be called from Interstage according to the following timing.

The *claddprocrsc* command is used to set it up. Refer to the PRIMECLUSTER manual about this command.

[START-RUN]

- AFTER

[START-WAIT]

- AFTER

[STOP-RUN]

  - BEFORE

[STOP-WAIT]

  - BEFORE

[FAIL-RUN]

  - BEFORE

[FAIL-WAIT]

  - BEFORE

4.6.1.2.3 Registration to userApplication of State Transition Procedure

State transition procedure is registered into userApplication. As for the "Startup priority of the application", register them so as to be called by the following priority levels.

Solaris32/64

Priorities are set up by 'Resource Creation' in the 'userApplication Configuration wizard'. For details, refer to the PRIMECLUSTER manual.

Linux32/64

A priority is set up by "CUI(RMS Wizard)". For details, refer to the PRIMECLUSTER manual.

Resource start Priorities

- Start priorities of resources to be registered in the procedure class "BasicApplication"

  (Set the resources so that they are called in the following order.)

    - userApplication.JMX_INTERSTAGE (*1) Solaris32   Linux32/64

    - userApplicaion.IREP_INTERSTAGE

    - userApplicaion.IS_INTERSTAGE

    - userApplicaion.SSO_INTERSTAGE (*2)

    - userApplicaion.OTS_RMP_INTERSTAGE Solaris32   Linux32/64

    - userApplicaion.ES_INTERSTAGE (*3)

    - Start priorities of resources to be registered in the procedure class "Application" (option)

    - userApplicaion.TDWU_INTERSTAGE Solaris32   Linux32

    - userApplicaion.IJSERVER_INTERSTAGE

    - userApplicaion.UTYWU_INTERSTAGE

    - userApplicaion.ODWU_INTERSTAGE

*1 This can only be registered for the Admin Server.

*2 This can only be registered for session management.

*3 This state transition procedure need not be registered when setting all the Event Channels to 'when the event service starts, the Event Channel is automatically activated', which is the initial value (default value).

Procedure resource SCRIPTTIMEOUT attribute

The SCRIPTTIMEOUT attribute is used as the start and stop timeout (in seconds) of the procedure resource.

Estimate the SCRIPTTIMEOUT attribute values that will be set as the procedure resources for each state transition procedure as follows:

State transition procedure	How to estimate SCRIPTTIMEOUT
IS_INTERSTAGE	Measure the actual Interstage start and stop times (complete shutdown mode), and double the greater value. The start time is the time from execution of the start command to start completion; the stop time is the time from execution of the stop command to stop completion. To measure the times, wait until the environment settings for each service that will be used in the operations are complete.
`Solaris32` `Linux32` TDWU_INTERSTAGE	Specify the greater of the actual total start and total stop times for all TD WorkUnits. (*1)
ES_INTERSTAGE	Specify the greater of the actual total start time for each unit and event channel defined in the event service and ES_INTERSTAGE, and the actual stop times for each event service. (*1)
IREP_INTERSTAGE	Measure the actual total start and total stop times for all repositories, and double the greater value.
SSO_INTERSTAGE	A minimum of 5 seconds.
`Solaris32` `Linux32/64` OTS_RMP_INTERSTAGE	Measure the actual total start and total stop times for each resource management program, and double the greater value. To use JTS, measure the actual start and stop times for the JTS resource management program and compare them using the value that is calculated.
IJSERVER_INTERSTAGE	Specify the greater of the actual total start and total stop times for all IJServers. (*1)
ODWU_INTERSTAGE	Specify the greater of the actual total start and total stop times for all OD WorkUnits. (*1)
UTYWU_INTERSTAGE	Specify the greater of the actual total start and total stop times for all utility WorkUnits. (*1)
`Solaris32` `Linux32/64` JMX_INTERSTAGE	Measure the actual Interstage JMX Service start and stop times, and double the total.

*1 Repeat these measurements several times, before specifying a timeout value that is large enough.

Operation Procedure

Perform the following operations in each procedure, on all nodes on which the status transition procedure is executed.

1. Register the status transition procedure.

```
clsetproc -c BasicApplication   (*1)
-m IS_INTERSTAGE   (*2)
```

`Solaris32/64`

```
-o /etc/opt/FJSVisas/HA/SynfinityCLUSTER/IS_INTERSTAGE   (*2)
```

`Linux32/64`

```
-o /etc/opt/FJSVisas/HA/PRIMECLUSTER/IS_INTERSTAGE   (*2)
```

2. Register application resources that use the status transition procedure.

```
claddprocrsc -k IS_INTERSTAGE   (*2)
-m IS_INTERSTAGE   (*2)
-c BasicApplication   (*1)
-K AFTER -w  -L AFTER -S BEFORE -T BEFORE -V BEFORE -W BEFORE -u 0
-t 0 -p 100
```

3. Register the resources as PRIMECLUSTER resources among the status transition procedure resources under "userApplication".

Register the resources created in step 2 as PRIMECLUSTER resources.

`Solaris32/64`

Perform the operation in the "userApplication Configuration wizard", starting with "Resource Creation".

Specify the resource names in the form described in the above item "Resource start priorities".

For the "userApplication Configuration wizard", refer to the PRIMECLUSTER manual.

Linux32/64

Perform the operation from CUI (RMS Wizard).

For CUI (RMS Wizard), refer to the PRIMECLUSTER manual.

4. Register the status transition procedure resources under "userApplication."

Register all resources created in step 3 under "userApplication", which was created according to "Starting Cluster Service (userApplication)."

Set 'ClearFaultRequest|StartUp|SwitchRequest' to 'StandbyTransitions' for 'the attribute of userApplication'. Refer to the PRIMECLUSTER manual for information about the registration method to userApplication, and the setting method of the attribute of userApplication.

Note

Parameters to be specified for options:

*1 Omitted for ODWN_INTERSTAGE/TDWU_INTERSTAGE/IJSERVER_INTERSTAGE/UTYWU_INTERSTAGE

*2 Change the contents according to the name of the resource to be registered.

4.6.2 For MSCS

The resources of Interstage are registered to MSCS.

When the service for the WorkUnit or data is used before the resource of Interstage is registered to MSCS, it is necessary to create a batch file for resource management program and the WorkUnit startup. It is also necessary to create the batch file for the Event Channel startup for MSCS when starting up the Event Channel automatically.

4.6.2.1 Creation of Batch File for Database Linkage Service and WorkUnit Startup

To start up the Database Linkage Service and the WorkUnit automatically at the switching of the cluster, the batch file for the Database Linkage Service and the WorkUnit startup is registered to MSCS as a general-purpose application.

The batch file for the Database Linkage Service and the WorkUnit startup is as follows:

```
echo off

otsstart
otsstartrsc  -pg resource management program 1 -n resource definition name 1
otsstartrsc  -pg resource management program 2 -n resource definition name 2
isstartwu  WorkUnit name 1
isstartwu  WorkUnit name 2

pause
```

Other Information

It is necessary to add "pause" at the end of the batch file.

- Describe *otsstartrsc -j* to start up the resource management program for JTS.

4.6.2.2 Creation of Batch File for Event Channel Startup

To start up the Event Channel automatically at the switching of the cluster, the batch file for the Event Channel startup is registered to MSCS as a general-purpose application.

The batch file for the Event Channel startup is as follows:

```
echo off
```

```
esstartunit -unit  UNIT NAME
esstartchnl -g  Event channel group name-c Event Channel name

pause
```

Other Information

- It is necessary to add "pause" at the end of the batch file.

- When volatilizing, it is not necessary to describe the *esstartunit* command.

4.6.2.3 Registering Resources

The resource of Interstage is registered to MSCS.

Register the resources shown in the following table using the cluster administrator. This information applies only when a Symfoware Server database is to be used.

When registering the following resources in Windows Server(R) 2008, delete their setup parameters:

- Interstage HTTP Server services

Execute the following command to delete the setup parameter:

```
Cluster Resource "resource name" /priv StartupParameters=""
```

Table 4.1 Resource Registration in MSCS

Resource	Service name	Resource type	Dependent on	Remarks
Interstage IP address	-	IP address		(*11)
Interstage network name	-	Network name	IP address	(*11)
Shared disk	-	Physical disk	Network name	
Symfoware Service	SymfoWARE RDB (*2)	A general-purpose service	Shared disk	
Interstage Directory Service	Interstage Directory Service(repository name)	A general-purpose service	Shared disk	(*7)
OD_start Service	ODloader	A general-purpose service	Symfoware Service	(*1)
NamingService	Naming	A general-purpose service	OD_start Service	(*1)
InterfaceRep_ Cache Service	InterfaceRep_Cache_s	A general-purpose service	NamingService	(*1)
EventService	esdmnmain	A general-purpose service	InterfaceRep_Cache Service	(*1)
EventFactory Service (*5)	esfactory	A general-purpose service	EventService	(*1)
TransactionDirector Service	TransactionDirector	A general-purpose service	EventService	(*1)
Interstage API Service	INTERSTAGE API	A general-purpose service	TransactionDirector Service	(*1)
ObjectTransactionServ ice (*3)	ObjectTransaction Service	A general-purpose service	Interstage API Service	(*1)

Resource	Service name	Resource type	Dependent on	Remarks
Batch file for Database Linkage Service and WorkUnit startup	-	A general-purpose application	ObjectTransaction Service or Interstage API Service (*4)	(*1)
Batch file for Event Channel startup (*6)	-	A general-purpose application	ObjectTransaction Service or Interstage API Service (*4)	(*1)
Interstage HTTP Server Service	FJapache Interstage HTTP Server (Web Server name)	A general-purpose service	IP address Interstage Directory Service (*8)	
Interstage JMX Service	Interstage Operation Tool	A general-purpose service	Shared disk	(*9)
Application for monitoring the state of the session management server for the Interstage single sign-on function (*10)	-	A general-purpose application	Interstage HTTP Server Service	

Other Information

- Register all of the above resources in the Interstage group.

- Specify an optional string in the resource name.

- Specify the node for the cluster system to an owner who has appropriate authority.

- Specify a service name in the table when you register a general-purpose service.

- As for other setup items, designation is unnecessary.

Notes

*1 Do not select the "Use network name as computer name" check box.

*2 Refer to the Symfoware Server product manual for details on Symfoware Server product service names.

*3 Specify this resource only when using the Database Linkage Service.

*4 If the Database Linkage Service is to be used, specify the ObjectTransactionService service as the dependency relation. If not, specify the Interstage API Service as the dependency relation.

*5 Register the EventFactory service only when performing dynamic generation operations.

*6 Perform registration only when starting the Event Channel.

*7 To stop the repository, delete the repository resources that have been registered in MSCS.

*8 To use an Interstage Single Sign-on repository server, implement dependency settings for the Interstage Directory Service to be used as the SSO Repository.

*9 Specify this resource only when using the Admin Server.

*10 If a repository server is used for the Interstage single sign-on function, register the application that monitors the state of the session management server for the Interstage single sign-on function (C:\Interstage\F3FMsso\ssoatcsv\bin\ssostatssmgr.exe) only when session management operations are performed.

*11 When using Microsoft(R) Windows Server(R) 2008 Failover Clustering, register "Client Access Point" as a resource instead of "Interstage IP address" and "Interstage network name".

4.7 Settings for the Admin Server `Windows32/64` `Solaris32` `Linux32/64`

This section describes the cluster service environment setup procedure for the Admin Server.

Note

- Do not set cluster system environment settings for a Combined Server.

- Mutual standby cannot be used for Managed Servers operating in a cluster system. Use fixed standby mode.

4.7.1 Install Interstage `Windows32/64` `Solaris32` `Linux32/64`

The installer is used to install the Interstage package on the two nodes (the operational node and the standby node) which comprise the cluster. The Interstage package needs to be installed with the same path on each of these nodes.

Note

Install the Interstage package on the local disk in each case.

Interstage Management Console control is not inherited in cluster environments. This means that you must log in to the Interstage Management Console again after switching the operation control server. The Interstage Management Console cannot be used in an environment in which inheritance from an operation node to a standby node is assumed.

If the Admin Server is run on a cluster system, other Interstage Application Server functions cannot be used on that server. Do not install the server functions of Interstage Application Server.

4.7.2 Install `Windows32/64` `Solaris32` `Linux32/64`

`Windows32/64`

After installing the Interstage package, change to "Manual" because the "Startup type" of the following services is set to "Automatic". Additionally, stop these services if they have been started.

- FJapache

- Interstage JServlet(OperationManagement)

- Interstage Operation Tool

- Interstage Operation Tool(FJapache)

- Interstage Directory Service (repository name)

- INTERSTAGE

`Solaris32` `Linux32/64`

When the Interstage package is installed, the start shell and the stop shell are saved as symbolic link files. The start shell is invoked when the server is started and the stop shell is invoked when the server is shut down. To ensure that the relevant shell is not invoked when the cluster is used, you will need to move them somewhere else - for example to a different directory.

`Solaris32`

- Move S99FJSVirep, found in /etc/rc2.d

- Move K00FJSVirep, found in /etc/rc0.d

- Move K17FJapache, found in /etc/rc0.d (*1)

- Move K17FJapache, found in /etc/rc1.d (*1)

- Move K17FJapache, found in /etc/rc2.d (*1)

- Move S51FJapache, found in /etc/rc3.d (*1)

- Move K17FJapache, found in /etc/rcS.d (*1)

- Move S99FJSVjs2su, found in /etc/rc2.d (*1)

- Move S99FJSVjs2su, found in /etc/rc3.d (*1)

- Move S99isstartoptool, found in /etc/rc2.d

- Move S95isjmxstart, found in /etc/rc2.d

- Move K18FJSVirep, found in /etc/rc0.d (*1)

- Move K18FJSVirep, found in /etc/rc1.d (*1)

- Move S82FJSVirep, found in /etc/rc2.d (*1)

- Move S82FJSVirep, found in /etc/rc3.d (*1)

- Move S82FJSVirep, found in /etc/rc4.d (*1)

- Move S82FJSVirep, found in /etc/rc5.d (*1)

- Move K18FJSVirep, found in /etc/rc6.d (*1)

- Move K14FJapache, found in /etc/rc0.d(*1)

- Move K14FJapache, found in /etc/rc1.d(*1)

- Move S86FJapache, found in /etc/rc2.d(*1)

- Move S86FJapache, found in /etc/rc3.d(*1)

- Move S86FJapache, found in /etc/rc4.d(*1)

- Move S86FJapache, found in /etc/rc5.d(*1)

- Move K14FJapache, found in /etc/rc6.d(*1)

*1 Move once this file has been saved.

4.7.3 Cluster System Presetting `Windows32/64` `Solaris32` `Linux32/64`

Before setting up an Interstage environment, perform all required operations - from cluster system installation to setting up the cluster initialization configuration.

Note

- For more information on each cluster system, refer to the manual of each cluster system.

- For related products used by Interstage, follow the installation directions for each product.

4.7.4 Setting up a Shared Disk Unit `Windows32/64` `Solaris32` `Linux32/64`

To use the Admin Server on a cluster system, the shared disk must be setup as a file system.

As the shared disk unit is required during setup of the Interstage environment, the unit must be registered in the cluster service (At PRIMECLUSTER, it is userApplication.).

4.7.4.1 Configuration Manager Repository

Set up the Configuration Manager Repository on an exclusive path in the shared disk. For details on how to set the path to the Configuration Manager Repository, refer to "".

Example

`Windows32/64`

```
E:\jmx\repository
```

`Solaris32` `Linux32/64`

```
/jmx/repository
```

4.7.4.2 Interstage Directory Service

When using the directory service for the management repository of the user logged in to the Interstage Management Console, prepare the exclusive path for the Interstage Directory Service on the shared disk.

Example

`Windows32/64`

```
E:\irep\repository
```

`Solaris32` `Linux32/64`

```
/irep/repository
```

Note

Different paths may be specified for the Configuration Manager and Interstage Directory Service directories.

4.7.5 IP Address `Windows32/64` `Solaris32` `Linux32/64`

Interstage assumes IP address takeovers, so set up a network for takeover in advance.

As the IP address is required during setup of the Interstage environment, the inherited IP address must be registered in the cluster service (At PRIMECLUSTER, it is userApplication.).

To run Admin Server on a cluster system, specify this IP address in the "IP address used by the Admin Server" that was specified at the time of installation.

4.7.6 Interstage Environment Setup for the Admin Server

This section describes setting for Interstage for the Admin Server.

1) Starting the Cluster Service (userApplication)

For Interstage setup, the inherited IP address must be activated and a shared disk is required. This means that a cluster service in which Interstage is set must first be created. In PRIMECLUSTER, this is "userApplication". Register the inherited IP address and shared disk in the cluster service and start the service before implementing the Interstage environment settings.

Node 1 in this document refers to the operation node.

2) Create the Configuration Management Repository in Node 1

Create the Configuration Manager Repository in node1 using the Interstage Management Console Application Management tab. The procedure and the values that should be set are shown below.

1. Log in to the Interstage Management Console.

2. Click the Application Management tab in the tree window.

3. Navigate the following windows:

   Interstage> Interstage Application Server > System > Configuration Management

4. Change "Repository Storage Directory" to a shared disk.

5. Click the Apply button.

3) Create the Repository Managing Login User of the Interstage Management Console in Node 1

This is not valid for Linux (64 bit).

When using the directory service as the repository managing the logon user of the Interstage Management Console, it is necessary to set up the Interstage Directory Service on Node 1.

Setup of the Interstage Directory Service on the Admin Server is the same setup on a Standalone Server. For details, refer to "4.5.16 Using Interstage Directory Service".

The procedure of creating a user repository using a directory service is described in the Operator's Guide - Configuring the Interstage Management Console - Login Authentication for the Interstage Management Console.

4) Back Up the Interstage Resources of Node 1

For details on how to back up resources of the following Interstage services, refer to "Backup/Restore in a Multiserver Environment" in the "Maintenance (Resource Backup)" chapter of the Operator's Guide.

- Interstage Management Console.

- Interstage JMX Service.

- Interstage Directory Service (when using the directory service for authentication)

5) Import the Interstage Resources to Node 2

For details of how to import resources, refer to "Resource Importing Procedure" in the "Maintenance (Resource Backup) chapter of the Operator's Guide.

The Interstage Directory Service resource import procedure is described in "4.5.16 Using Interstage Directory Service".

Note

If the operation is executed in Node 1, Re-execute the procedure from Back Up the Interstage Resources of Node 1, above.

- Customize the Interstage Management Console.

  Refer to the "Configuring the Interstage Management Console" chapter of the Operator's Guide.

- Change the login authentication type for the Interstage Management Console.

  Refer to "Login Authentication for the Interstage Management Console" in the "Configuring the Interstage Management Console" chapter of the Operator's Guide.

4.7.7 Setting the Cluster Service `Windows32/64` `Solaris32` `Linux32/64`

Setting the Cluster Service for the Admin Server is the same as setting up the Interstage Server function. Configure the settings described in the Interstage Server function to the following services. For details, refer to "4.6 Setting Cluster Service".

- Interstage JMX Service

- Interstage Directory Service (when using the directory service for authentication)

4.8 Settings for the Managed Server `Windows32/64` `Solaris32` `Linux32/64`

This section describes the cluster service environment setup procedure for the Managed Server.

The cluster service function can only be used with the Interstage Application Server Enterprise Edition.

Refer to "Cluster Service Function" in the "For a High-performance, High-reliability System" chapter for details of the cluster service function.

The Cluster function can only run on a Managed Server the server was added to the Site as an Independent Managed Server.

For nodes operating in a cluster system, add both the operation node and the standby node to the site as Independent Managed Servers, and register the same resources on both nodes.

Figure 4.7 Cluster System Environment
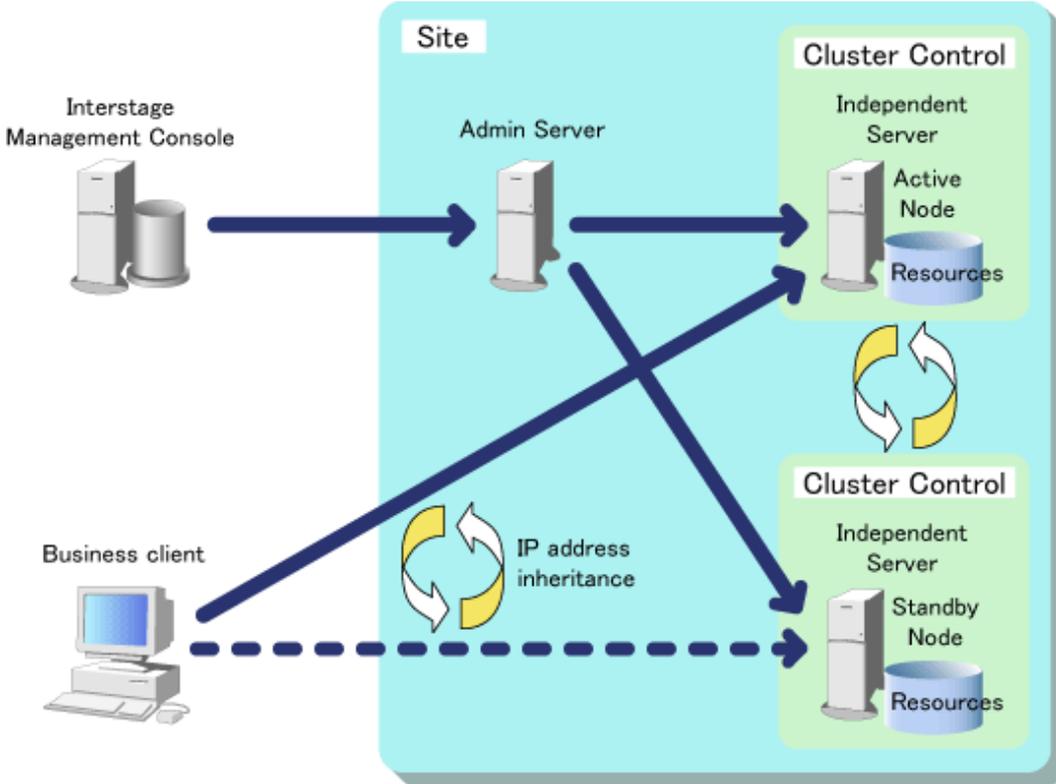
4.8.1 Cluster System Settings Windows32/64 Solaris32 Linux32/64

This section describes the cluster service environment setup procedure for Managed Servers.

Note

- Only Independent Managed Servers can be run on a cluster system. Managed Server that are members of a Server Group cannot be run on a cluster system. If a Managed Server from a Server Group is used on a cluster system, there may be issues with consistency of the Server Group operating status, or unexpected switching of the Cluster Service may result.

- Do not set cluster system environment settings for a Combined Server.

- Mutual standby cannot be used for Managed Servers operating in a cluster system. Use the fixed standby mode.

4.8.1.1 Setup Procedure

Use the following procedure to build the environment to run an Independent Managed Server on a cluster system:

Figure 4.8 Environment Build Steps on a Cluster System

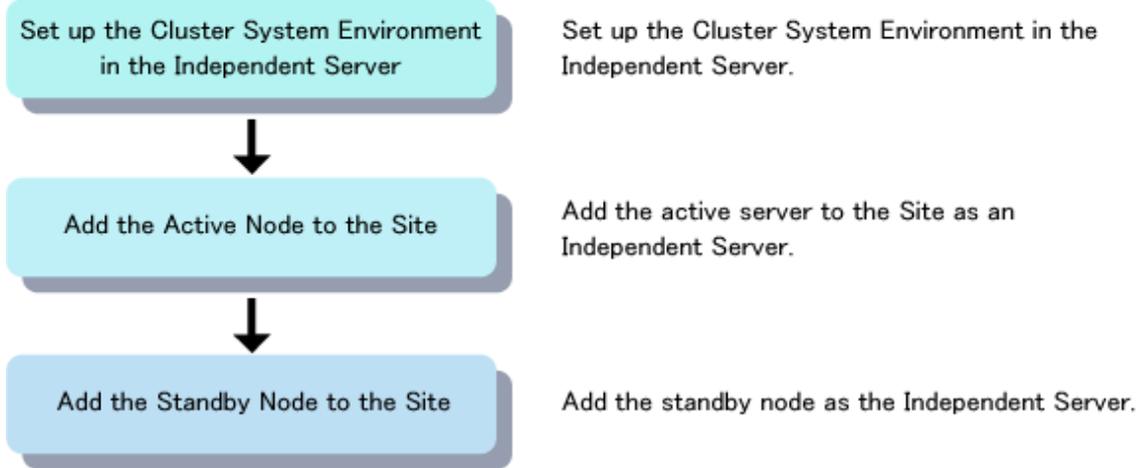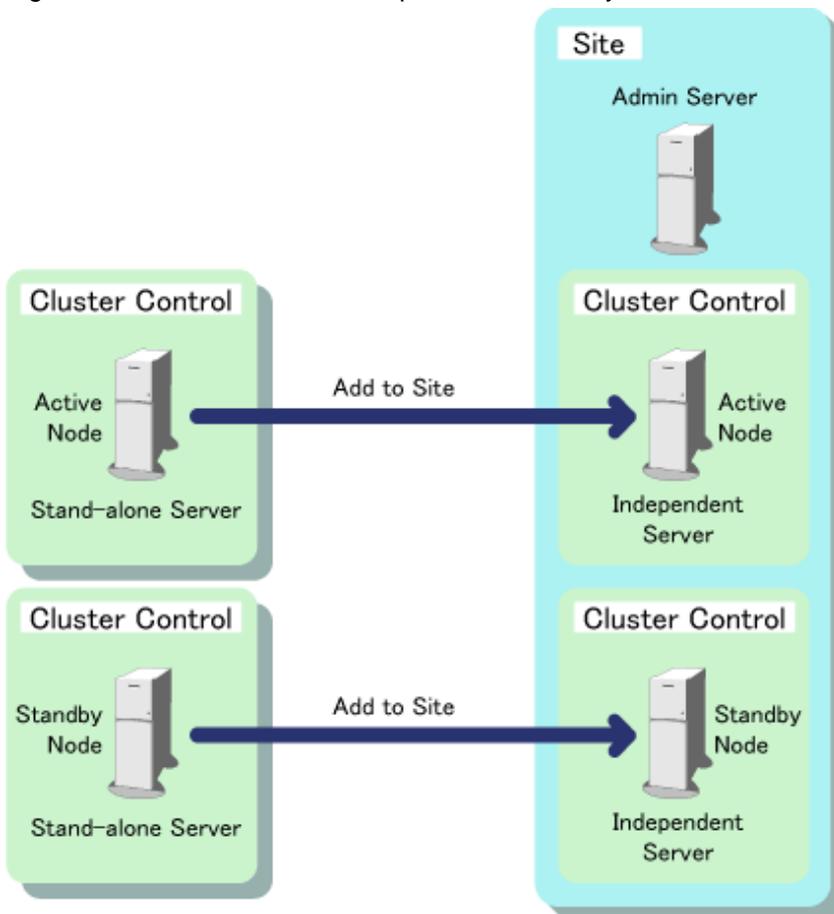|	
Set up the Cluster System Environment in the Independent Server	Set up the Cluster System Environment in the Independent Server.
Add the Active Node to the Site	Add the active server to the Site as an Independent Server.
Add the Standby Node to the Site	Add the standby node as the Independent Server.

Figure 4.9 Environment Build Steps on a Cluster System Illustrated

To run an Independent Managed Server that has already been added to a Site on a cluster system, first remove the Independent Managed Server from the Site to convert it to a Standalone Server. Then setup the Interstage server function environment and re-add it to the Site.

4.8.1.1.1 Building a Cluster System on a Standalone Server

Configure the cluster system environment while the server is a Standalone Server. The task flow for setting up a cluster system is shown below.

Figure 4.10 Setup of a Cluster System
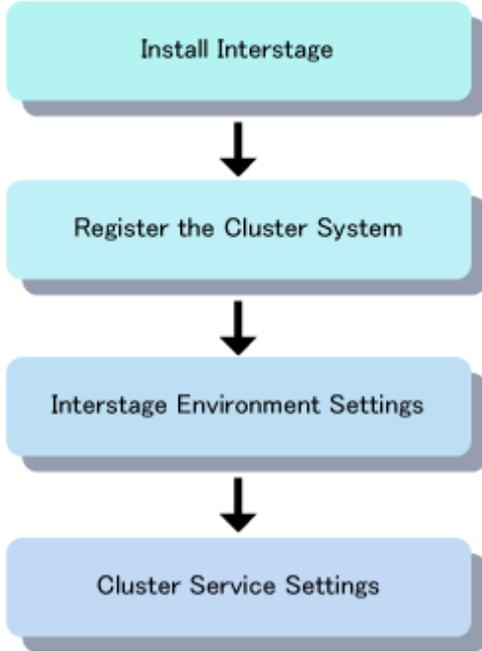
4.8.1.1.2 Install Interstage on a Standalone Server

A server that is to be added to the Site as an Independent Managed Server is installed as a Standalone Server first. For details, refer to "4.3 Installing Interstage".

4.8.1.1.3 Cluster System Presetting

Cluster system presetting for servers that are to be operated as Independent Managed Servers is done in the same way as presetting is done for Standalone Servers. For details, refer to "4.4 Cluster System Presetting".

4.8.1.1.4 Interstage Environment Settings

The Interstage environment settings for servers that are to be operated as Independent Managed Servers are the same as those for Standalone Servers. For details, refer to "4.5 Interstage Environment Setup for the Server Function".

4.8.1.1.5 Cluster Service Settings

The Cluster service settings for servers that are to be operated as Independent Managed Servers are the same as those for Standalone Servers. For details, refer to "4.6 Setting Cluster Service".

4.8.1.2 Adding to a Site

Add the operation node and the standby node to the Site as Independent Managed Servers. Add each node to the Site as a separate operation.

Note

- Do not add servers operating on a cluster system to a Site as a Reserved Managed Servers.

- If a server operating on a cluster system is added to a Site, do not use the IP address inherited by the cluster service for the IP address of the management or business LAN.

4.8.2 Operating Managed Servers in a Cluster System `Windows32/64` `Solaris32` `Linux32/64`

The cluster service functions on an Independent Managed Server are operated by the cluster system functions.

For the standby node, prepare an Independent Managed Server with the same environment as the operation node Independent Managed Server that is being used for business operations. The business operations are transferred to the standby Independent Managed Server if the operation node server fails.

Figure 4.11 Operation of Managed Servers
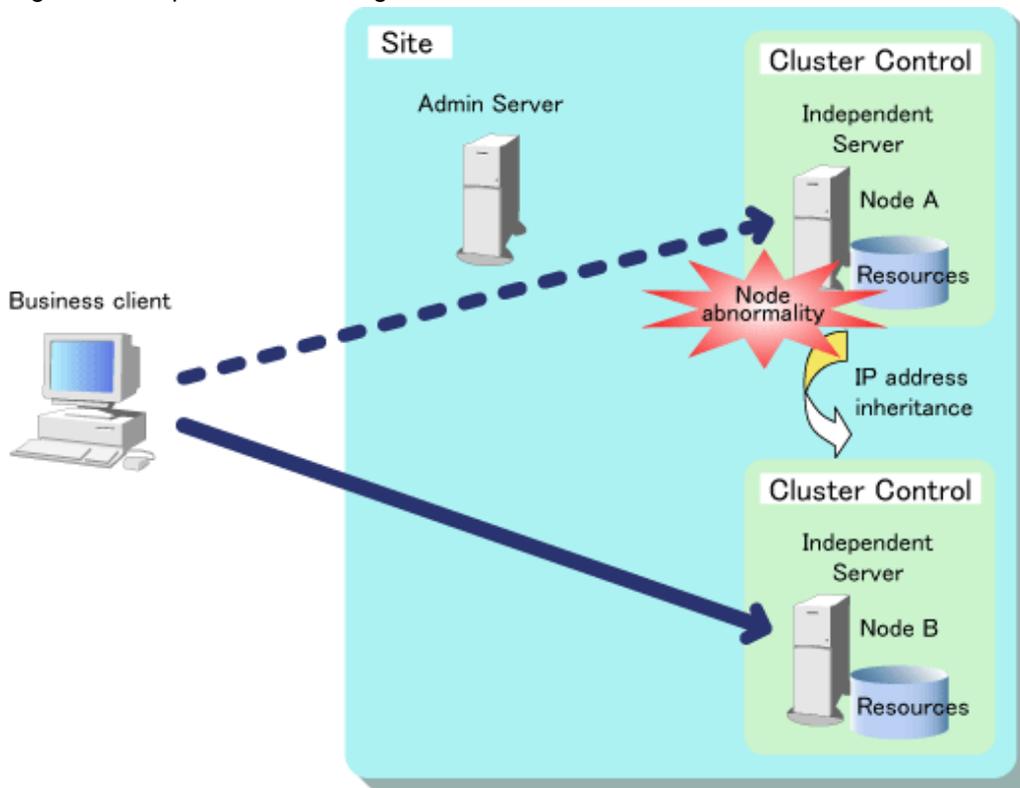
4.8.3 Maintenance Windows32/64 Solaris32 Linux32/64

This section describes application environment settings for Managed Servers and how to rebuild a cluster system.

4.8.3.1 Server Application Environment Settings

The application environments constructed for the operation node and standby node must be exactly the same.

Use the Interstage Management Console on the Admin server to set the application environments for Independent Managed Servers.

Figure 4.12 Setup of Application Environments for Independent Managed Servers
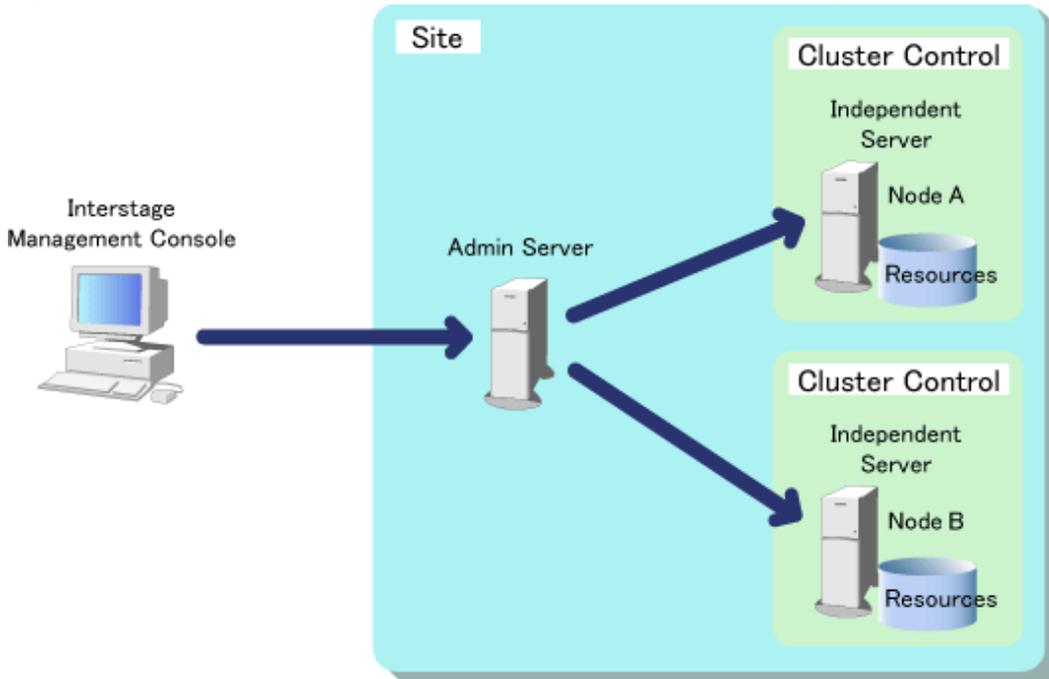
Refer to "4.10.2 Server Application Environment" for details on application environments.

Note

- Do not create multilevel IJServers on Managed Servers operating in a cluster system.

4.8.3.2 System Settings

Remove Managed Servers from the Site before changing the Interstage system structure. Refer to "Removing a Participant from the Site (from the Admin Server)" in the "Site" chapter of the Operator's Guide for details on removing servers from Sites.

4.8.3.3 Cluster System Settings

Remove Managed Servers from the Sites before making changes to the cluster system settings. For details on how to configure the cluster system settings, refer to Figure 4.7 Cluster System Environment.

4.8.4 Deleting an Environment Windows32/64 Solaris32 Linux32/64

To delete the environment of Managed Servers operating on a cluster system, use the procedures below to rebuild the environment.
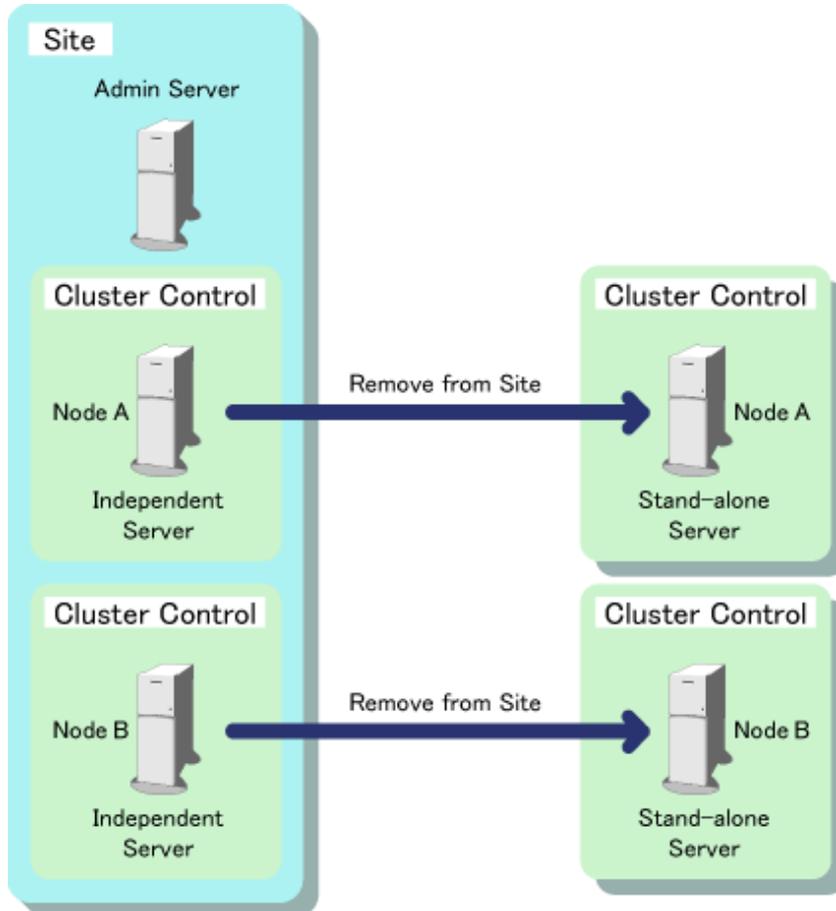
Figure 4.13 Deleting a Cluster System Environment

4.8.4.1 Removing from Site

Remove the operation node and the standby node from the Site. Refer to "Removing a Participant from the Site (from the Admin Server)" in the "Site" chapter of the Operator's Guide for details on removing servers from Sites.

Figure 4.14 Removing Servers from a Site

4.8.4.2 Deleting Cluster System Settings

Delete the status transition procedures registered as cluster service resources. Refer to the documentation for the cluster system being used for information on deleting status transition procedures.

4.8.5 Notes Windows32/64 Solaris32 Linux32/64

Note the following points concerning operating Managed Servers on a cluster system:

- Mutual standby mode cannot be used when Managed Servers operate on a cluster system. Use fixed standby mode.

- Do not operate Managed Servers that belong to a Server Group on a cluster system.

- Managed Servers that are members of a Server Group cannot be run on a cluster system. If a Managed Server from a Server Group is used on a cluster system, there may be issues with consistency of the Server Group operating status, or unexpected switching of the Cluster Service may result.

  If Managed Servers operating on a cluster system were added to a Server Group, use the following procedure to rebuild the environment:

  - Remove the Managed Servers from the Server Group.

  - Remove the Managed Servers from the Site.

  - Add the servers to the Site as Independent Managed Servers.

- If multilevel IJServers have been created on Managed Servers operating on a cluster system, delete those IJServers.

- Do not set cluster system environment settings for a Combined Server.

4.9 Settings for the Combined Server `Windows32/64` `Solaris32` `Linux32/64`

Do not set cluster system environment settings on a Combined Server.

4.10 Creating the Application Environments

This section explains the following topics:

- 4.10.1 Client Application Environment

- 4.10.2 Server Application Environment

4.10.1 Client Application Environment

When a switch is generated by the shutdown of the operational server system, (for instance, while connection processing is being performed by a client on the operational server) the request processing from the client either generates an error or enters an unresponsive state. If it enters an unresponsive state, handle this by performing time monitoring on the client.

When a switch has occurred, handle the situation in such a way that the client re-connects to the operational server.

When gaining an object reference from a client, it is necessary to set up host information using the *odsethost* command. Set this up using the following IP addresses.

- [For PRIMECLUSTER]

  Use the IP address inheritance.

- [For MSCS]

  Use the IP address inheritance.

When the IP address to be used is "10.34.157.101", and the port number of the CORBA service is 8002, use

```
odsethost -a -h 10.34.157.101 -p 8002
```

Refer to the Reference Manual (Command Edition), for the details of the *odsethost* command.

4.10.2 Server Application Environment

The server application environment has to be created with exactly the same structure on the operational node and the standby node.

4.10.2.1 Server Application Programming

Server application programming is the same as when a cluster system is not used.

4.10.2.2 Skeleton File

The skeleton file output on the operational node should be copied to the standby node.

4.10.2.3 Executable Files for Server Applications

Server applications and other resources used by WorkUnits must be created with the same structure (filename, directory structure etc) on both the operational and the standby node.

4.10.2.4 Creating APM `Windows32/64`

When performing global transaction linkage, use the *tdlinkapm* command to create and operate APM. In this case, create APM so that the configuration such as file names and directory structure must be the same for the operational node and the standby node. Specify the name of the created APM in the WorkUnit definitions. Be sure to create an XA linkage program before creating APM.

4.10.2.5 WorkUnit Definition

The same WorkUnit definition should be used on both the operational and the standby node. You will also need to register the WorkUnit definition individually to both these nodes using the *isaddwudef* command.

Windows32/64　Solaris32　Linux32/64

The *isaddwudef* command cannot be used to register WorkUnit definitions to a server operating as a Managed Sever. Instead, use the Interstage Management Console on the Admin Server to register WorkUnit definitions to both the operation node and standby node.

Note

- Windows32/64　Solaris32　Linux32/64

  If an Independent Managed Server has been rebuilt in a cluster system environment, do not set this server as the deployment destination for any of the IJServer configuration elements.

- Windows32　Solaris32　Linux32

  Component transaction WorkUnit and WRAPPER WorkUnit definitions can also be registered using the *tdadddef* command.

4.10.2.6 Changing a WorkUnit (CORBA application)

To add or delete a WorkUnit after the cluster environment has been set, use the following procedure.

For PRIMECLUSTER

To add a WorkUnit:

1. Create the CORBA application to add on to the operational node and the standby node.

2. Stop the cluster service.

3. Register the WorkUnit definition that is to be added on the operation and standby nodes.

4. Specify the WorkUnit name to be added in the status transition procedure, and register the status transition procedure in the cluster again.

5. Start the cluster service.

To delete a WorkUnit:

1. Stop the cluster service.

2. Delete the WorkUnit definition from the operational and standby nodes.

3. Delete the WorkUnit name from the status transition procedure, and register the status transition procedure in the cluster again.

4. Start the cluster service

For MSCS　Windows32/64

To add a WorkUnit:

1. Create the CORBA application to add on the operational node and the standby node.

2. Stop the cluster service.

3. Register the WorkUnit definition that is to be added on the operation and standby nodes.

4. Describe the start processing of the WorkUnit to be added in the "WorkUnit start batch file" on the operational node and standby node.

5. Start the cluster service.

To delete a WorkUnit:

1. Stop the cluster service.

2. Delete the WorkUnit definition from the operational and standby nodes.

3. Describe the shutdown processing of the WorkUnit to be added in the "WorkUnit start batch file" on the operational node and standby node.

4. Start the cluster service

4.10.2.7 Changing a WorkUnit (Transaction Application)

To add or delete a WorkUnit after the cluster environment has been set, use the following procedure.

For PRIMECLUSTER

To add a WorkUnit:

1. Execute IDL compilation using the *tdc* command on the operational node and the standby node.

2. Create an application using the skeleton output on the operational node and the standby node.

3. Stop the cluster service.

4. Register the WorkUnit definition that is to be added on the operation and standby nodes.

5. Specify the WorkUnit name to be added in the status transition procedure, and register the status transition procedure in the cluster again.

6. Start the cluster service.

To delete a WorkUnit:

1. Delete interface information using the *tdc* command on the operational and standby nodes.

2. Stop the cluster service.

3. Delete the WorkUnit definition from the operational and standby nodes.

4. Delete the WorkUnit name from the status transition procedure, and register the status transition procedure in the cluster again.

5. Start the cluster service

For MSCS `Windows32/64`

To add a WorkUnit:

1. Execute IDL compilation using the *tdc* command on the operational node and the standby node.

2. Create an application using the skeleton output on the operational node and the standby node.

3. Stop the cluster service.

4. Register the WorkUnit definition that is to be added on the operation and standby nodes.

5. Describe the start processing of the WorkUnit to be added in the "WorkUnit start batch file" on the operational node and standby node.

6. Start the cluster service.

To delete a WorkUnit:

1. Delete the interface information using the *tdc* command on the operation node.

2. Stop the cluster service.

3. Delete the WorkUnit definition from the operational and standby nodes.

4. Describe the shutdown processing of the WorkUnit to be added in the "WorkUnit start batch file" on the operational node and standby node.

5. Start the cluster service

4.10.2.8 J2EE Application

To set the J2EE application environment, set up as you would when operation and standby nodes are not clustered. If you are using JDBC resources with J2EE applications, then store resources such as the JDBC datasource set up during JDBC environment setup, on the local disk of each node.

Databases that can be used

There are no restrictions. For details on how to use the database, refer to the manual for that database.

Deployment

To deploy J2EE applications, use the Interstage Admin Console or *ijsdeployment* command and execute the procedure on the active and standby nodes respectively.

Additionally, to configure EJB application operating environment definitions using the Interstage Admin Console, execute the procedure on the active and standby nodes respectively (for details on the J2EE application environment settings, refer to the J2EE User's Guide):

1. Start the cluster service if it has not been started.

2. Stop the IJServer WorkUnit on the active node.

   Use the Interstage Admin Console or *isstopwu* command (for details, refer to the "WorkUnit Management Commands" chapter of the Reference Manual (Command Edition)).

3. Deploy the application to the active node.

   Use the Interstage Admin Console or *ijsdeployment* command (for details, refer to the "J2EE Operation Commands" chapter of the Reference Manual (Command Edition)).

4. Switch the cluster service to the standby node.

5. Stop the IJServer WorkUnit on the standby node (the new active node).

   Use the Interstage Admin Console or *isstopwu* command (for details, refer to the "WorkUnit Management Commands" chapter of the Reference Manual (Command Edition)).

6. Deploy the application to the standby node (the new active node).

   Use the Interstage Admin Console or *ijsdeployment* command (for details, refer to the "J2EE Operation Commands" chapter of the Reference Manual (Command Edition)).

7. Start the IJServer WorkUnit on the standby node (the new active node)

   Use the Interstage Admin Console or *isstartwu* command (for details, refer to the "WorkUnit Management Commands" chapter of the Reference Manual (Command Edition)).

   Note

   This step is not necessary if you are switching the cluster service to the original active node.

8. Switch the cluster service to the original active node, if necessary.

Note

Windows32/64   Solaris32   Linux32/64

Use the Interstage Admin Console to deploy to standalone Managed Servers. The *ijsdeployment* command cannot be used.

4.11 Backup/Restore

For details on how to back up and restore Interstage resources in cluster environments, refer to "Backup/Restore of Resources (Cluster Environments)" in the "Maintenance (Resource Backup)" chapter of the Operator's Guide.

4.12 Miscellaneous Considerations

4.12.1 Synchronizing on the Standby Node after Switching

During a switch due to a server error, processes and IPC resources (for Solaris and Linux) may remain resident in the server.

Therefore, after recovering the resources by terminating and re-starting the server machine, you should synchronize on the standby node.

4.12.2 Precautions When Performing Persistent Channel Operation of Notification Service

For persistent channel operation of the Notification Service, the following procedure is required when deleting a unit and Event Channel after completing the setup described in When Performing Persistent Channel Operations of Notification Service.

1) Switching of the cluster service

Switch the cluster service. Make node 1 the operational node and node 2 a standby node.

2) Resource deletion of the status transition procedure from the cluster service

[For PRIMECLUSTER]

Delete the status transition procedure in which resources were registered for the cluster service from the enabled userApplication, using the "userApplication Configuration wizard".

Then, delete the status transition procedure from the application resources forming the cluster by using the *cldelprocrsc* command.

For the "userApplication Configuration wizard" and *cldelprocrsc* command, refer to the PRIMECLUSTER manual.

3) Termination of Interstage

Stop Interstage using the *isstop* command in the operational node (node 1). It is necessary to set the -f option.

```
isstop -f
```

4) Switching of the cluster service

Switch the cluster service. Make node 2 the operational node, and node 1 a standby node.

5) Startup of Interstage

Start Interstage using the *isstart* command in the operational node (node 2).

```
isstart
```

6) Startup of the unit

Start the extended unit using the *esstartunit* command in the operational node (node 2).

```
esstartunit -unit unit name
```

7) Deletion of Event Channel

Delete all Event Channel using the *esrmchnl* command on the operational node (node 2).

```
esrmchnl -g group name
```

8) Checking Event Channel object references

Check the Event Channel object reference registered in the Naming Service using the *odlistns* command on the operational node (node 2).

Specify the "-l" option for the *odlistns* command, and execute the command to check that 'Interface Repository Name' for 'Name registered in the Naming Service' is "IDL:CosNotifyChannelAdmin/EventChannel:1.0" or "IDL:CosEventChannelAdmin/EventChannel:1.0". 'Name registered in the Naming Service' is specified as "Group Name::Channel Name" in step 9.

```
odlistns -l
[Display Content]
Name(Type) Object information(detail)
              Default object information(detail)
"Name registered in the Naming Service"::"Registered type" "Interface Repository Name",...
```

9) Deleting Event Channel object references

Delete the Event Channel object reference using the *OD_or_adm* command on the operational node (node 2). For 'Group Name::Channel Name', specify 'Name registered in the Naming Service' as shown in step 8.

```
OD_or_adm -d -n group name::channel name
```

10) Termination of unit (only when deleting the unit)

Stop the extended unit using the *esstopunit* command on the operational node (node 2).

```
esstopunit -unit unit name
```

11) Deletion of persistent file from shared disk (only when deleting the unit)

Delete the unit using the *esrmunit* command on the operational node (node 2).

```
esrmunit
```

12) Termination of Interstage

In the operational node (node 2), stop Interstage using the *isstop* command. It is necessary to set the -f option.

```
isstop -f
```

13) Switching of the cluster service

Switch the cluster service. Make node 1 the operational node, and node 2 a standby node.

14) Startup of Interstage

Start Interstage using the *isstart* command in the operational node (node 1).

```
isstart
```

15) Startup of the unit

Start the extended unit using the *esstartunit* command in the operational node (node 1).

```
esstartunit -unit unit name
```

16) Deletion of Event Channel

Delete all Event Channel using the *esrmchnl* command on the operational node (node 1).

```
esrmchnl -g group name
```

17) Termination of unit (only when deleting the unit)

Stop the extended unit using the *esstopunit* command on the operational node (node 1).

```
esstopunit -unit unit name
```

18) Deletion of persistent file from shared disk (only when deleting the unit)

Delete the unit using the *esrmunit* command on the operational node (node 1).

```
esrmunit
```

19) Termination of Interstage

Stop Interstage using the *isstop* command in the operational node (node 1). It is necessary to set the -f option.

```
isstop -f
```

The shared disk unit is necessary for both the default system and the extended system in Mutual-Standby mode.

4.12.3 Precautions When Performing Volatile Channel Operation of Event Service and Notification Service

For volatile channel operation of the Event Service and the Notification Service, the procedure described below is required when deleting a unit and Event Channel.

1) Switching of the cluster service

Switch the cluster service. Make node 1 the operational node and node 2 a standby node.

2) Deleting resource of the status transition procedure from the cluster service

[For PRIMECLUSTER]

Delete the status transition procedure in which resources were registered for the cluster service from the enabled userApplication, using the "userApplication Configuration wizard".

Then, delete the status transition procedure from the application resources forming the cluster by using the *cldelprocrsc* command.

For the "userApplication Configuration wizard" and *cldelprocrsc* command, refer to the PRIMECLUSTER manual.

3) Stopping Interstage

Stop Interstage using the *isstop* command in the operational node (node 1). It is necessary to set the -f option.

```
isstop -f
```

4) Switching the cluster service

Switch the cluster service. Make node 2 the operational node, and node 1 a standby node.

5) Starting Interstage

Start Interstage using the *isstart* command in the operational node (node 2).

```
isstart
```

6) Deleting the Event Channel

Delete all Event Channel using the *esrmchnl* command on the operational node (node 2).

```
esrmchnl -g group name
```

7) Checking Event Channel object references

Check the Event Channel object reference registered in the Naming Service using the *odlistns* command on the operational node (node 2).

Execute the *odlistns* command with the "-l" option to check that 'Interface Repository Name' for 'Name registered in the Naming Service' is "IDL:CosNotifyChannelAdmin/EventChannel:1.0" or "IDL:CosEventChannelAdmin/EventChannel:1.0". 'Name registered in the Naming Service' is specified as "Group Name::Channel Name" in step 8.

```
odlistns -l
[Display Content]
Name(Type) Object information(detail)
              Default object information(detail)
"Name registered in the Naming Service"::"Registered type" "Interface Repository Name",...
```

8) Deleting Event Channel object references

Delete the Event Channel object reference using the *OD_or_adm* command on the operational node (node 2). For 'Group Name::Channel Name', specify 'Name registered in the Naming Service' as shown in step 7.

```
OD_or_adm -d -n group name::channel name
```

9) Stopping Interstage

In the operational node (node 2), stop Interstage using the *isstop* command with the -f option.

```
isstop -f
```

10) Switching the cluster service

Switch the cluster service. Make node 1 the operational node, and node 2 a standby node.

11) Starting Interstage

Start Interstage using the *isstart* command in the operational node (node 1).

```
isstart
```

12) Deleting the Event Channel

Delete all Event Channel using the *esrmchnl* command on the operational node (node 1).

```
esrmchnl -g group name
```

13) Stopping Interstage

Stop Interstage using the *isstop* command with the -f option in the operational node (node 1).

```
isstop -f
```

Solaris32

The shared disk unit is necessary for both the default system and the extended system in Mutual-Standby mode.