BEA MSCS Extensions

Configuration Design Guide

BEA MSCS Extensions
Document Edition 1.0
November 1998

Copyright

Restricted Rights Legend

Trademarks or Service Marks

BEA MSCS Extensions Configuration Design Guide

Document Edition	Date	Software Version
1.0	November 1998	BEA MSCS Extensions 1.0

Contents

About the BEA Online Guide

1. Cluster Technology Overview

2. Configuration Scenarios

3. Administration Scenarios

About the BEA Online Guide

You can access the online guide directly from the BEA MSCS Extensions graphical user interface (GUI) as long as you have Netscape Communicator 4.0 or above installed. To access the Online Guide, you simply right-mouse click anywhere on the Parameters window of the BEA MSCS Extensions GUI and then select Go to BEA MSCS Configuration Design Guide option.

While the BEA MSCS Extensions installation process automatically installs the BEA online guide, you have the option to install a Netscape Communicator 4.0 browser if the browser is not present on your system. If you chose not to install the browser during the original installation, you can install it later from the BEA MSCS Extensions CD by repeating the installation process and selecting Yes to the prompt to install the browser.

This section includes the following topics:

- How to Use the Online Guide
- What if the Online Guide Does Not Work?
- Important Considerations About the BEA Installed Browser
- Using Your Favorite Web Browser
- How to Print
- Documentation Conventions
- Where to Find Related Information
- Contact Information

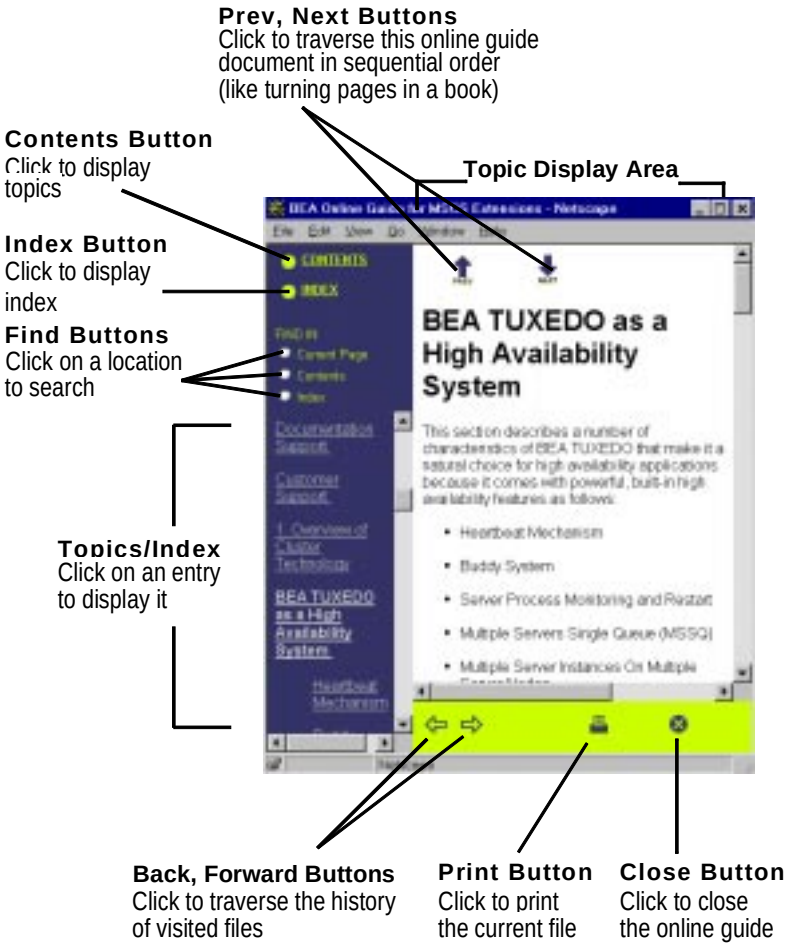
How to Use the Online Guide

To display a topic in the online guide, choose any of the following:

- Cluster Technology Overview—an overview section that puts the BEA MSCS Extensions product into the context of an integration of the BEA TUXEDO system and Windows NT MSCS. The chapter also describes the MSCS tools you use in configuring and administering a BEA Application resource.
- Configuration Scenarios—describes ways in which a BEA TUXEDO application may be configured in the MSCS environment.
- Administration Scenarios—maps MSCS online, offline, and move operations to their BEA TUXEDO equivalents.
- Configuration Examples—contains sample BEA TUXEDO configuration file entries for the Active/Hot Spare and Active/Active architectures described in Configuration Scenarios.
- Glossary—defines terms that are unique to the MSCS environment or that have an unfamiliar meaning in this case.

Figure 1 shows the online guide as it appears in your browser.

Figure 1 BEA MSCS Extensions Online Guide



What if the Online Guide Does Not Work?

The online guide relies on the Netscape Communicator browser for its functionality. Therefore, you may encounter problems that are generally related to the version of the Netscape browser that is active on your system, and the font size preference settings on that browser. To help troubleshoot these problems, this section provides the following topics:

- Making Sure You Are Using an Up-to-Date Browser
- Customizing the Font Size so the Online Guide Is Easy to Read

Making Sure You Are Using an Up-to-Date Browser

The online guide requires that Netscape Communicator version 4.0 or above be present on the local system and in use. If you are using an earlier version of the Netscape browser, you will get an error message when you try to use the Find or Print buttons. (See Figure 2.)

Figure 2 Error Message if Communicator is Not the "Active" Netscape Browser

Note that even if you have Communicator 4.0 installed, you can still get this error if you also have *earlier versions* of the Netscape browser and one of these down-level versions was the last browser used. The remedy for this problem is to close out of the current application, close any earlier versions of the Netscape browser (if you have some open) and open Communicator 4.0. (You can close Communicator as soon as you have opened it.)

When you re-start the application, the online guide should work properly.

Customizing the Font Size so the Online Guide Is Easy to Read

The online guide relies on your Netscape browser font preference settings. If the information shown in the online guide is difficult to read because the print is too small (or too large), you can change the font size. To do this, simply re-set your font preferences in the Netscape Communicator browser. The fonts sizes and styles you set in the browser also will show up in the online guide.

If you have more than one version of the Netscape browser on your system, make sure you set the font preferences in the active browser (which is preferably the most up-to-date browser). The online guide uses the last active browser. If you might be using more than one browser version to view files, set preferences in all browsers for optimal readability.

For more about why it is important to use an up-to-date browser, refer to the section Making Sure You Are Using an Up-to-Date Browser.

Important Considerations About the BEA Installed Browser

If you did not have the Netscape Communicator on your system when you installed BEA MSCS Extensions, it is likely that you have a BEA installed version of this browser.

The online guide requires that Netscape Communicator version 4.0 or above be present on the local system. So, the BEA MSCS Extensions product installation checks to see if the Netscape Communicator 4.0 browser is already present on the target system. If the appropriate version of the browser is not found; the install script gives you the option of installing it as a part of the BEA MSCS Extensions product installation to support the online guide.

The Netscape Communicator 4.0 that gets installed during the BEA MSCS Extensions product installation contains a level of encryption that is allowed to be exported from the United States. If you use this browser for anything other than the online guide, please note that this is not the most secure version of the Netscape Communicator.

Using Your Favorite Web Browser

The online guide is designed to use Netscape Nethelp.

However, you can also view the online guide with the Microsoft Internet Explorer browser or any other Web browser that supports HTML 3.0 and above. If you choose to use the Internet Explorer (or some other browser) to view the online guide, the primary difference is that you cannot access the online guide directly from the BEA MSCS Extensions GUIs. You may find some discrepancies in display since browsers other than the Netscape Nethelp viewer are not officially supported for this documentation set.

To view the online guide with your Internet Explorer browser, open the following file in the browser:

YourDrive:TUXDIR\doc\html\index.htm

How to Print

You can print a single topic directly from the online guide or you can print all the topics in book form (with page numbers, table of contents, and so on). The following sections explain how to print.

Printing the Current Topic

To print the topic file showing in the online guide display, click on the print button (see Figure 1).

Printing the Complete Book

A PDF version of this guide is made available on your system in the following location:

YourDrive:TUXDIR\doc\pdf\mcxtitl.pdf

To print the documentation, open a PDF file in an Adobe Acrobat Reader and choose the file print option.

If you do not have the reader, you can download one from the Adobe web site at http://www.adobe.com/.

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys sequentially.
italics	Indicates emphasis or book titles.

monospace text	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

Convention	Item
`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

`|`	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

`...`	Indicates one of the following in a command line:
	♦ That an argument can be repeated several times in a command line
	♦ That the statement omits additional optional arguments
	♦ That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Where to Find Related Information

- *BEA MSCS Extensions Release Notes*—document distributed with the BEA MSCS Extensions product.
- *Administrator's Guide for Microsoft Windows NT Cluster Server*—familiarity with that guide will make it much easier for you to understand and use the content of this document.
- Equally important information is contained in the guides and reference manuals shipped with the BEA TUXEDO product.

Contact Information

The following sections provide information about how to obtain support for the documentation and software.

Documentation Support

If you have questions or comments on the documentation, you can contact the BEA Information Engineering Group by e-mail at **docsupport@beasys.com**. (For information about how to contact Customer Support, refer to the following section.)

Customer Support

If you have any questions about this version of BEA MSCS Extensions, or if you have problems installing and running BEA MSCS Extensions, contact BEA Customer Support through BEA WebSupport at `www.beasys.com`. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

1 Cluster Technology Overview

High availability is a key goal of many mission-critical business systems. One of the major attributes of cluster technology is its ability to provide high availability for mission-critical applications.

Clustering enables you to connect a group of server machines in order to enhance data availability, server manageability and performance; in short, to achieve high availability.

Regardless of how many servers are connected in a cluster, a client machine sees the cluster as a single server. If one server in a cluster fails, the software moves the work to another server. This automatic move is referred to as a failover.

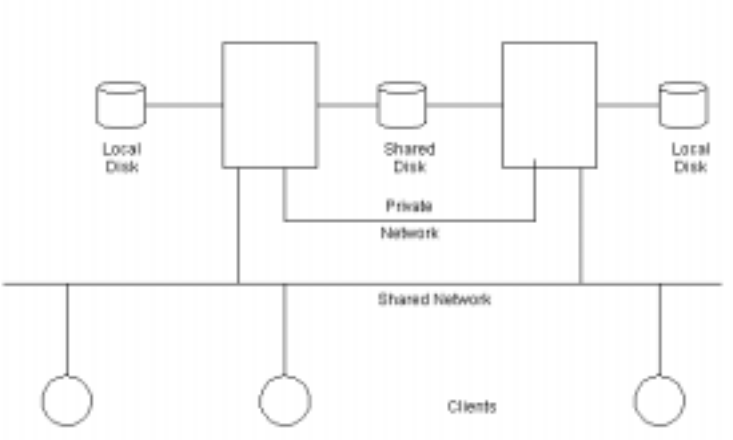
Cluster technology also makes it easy to move applications manually from one server node to another (for load balancing or for planned maintenance) or to add servers to a cluster.

A cluster is similar to a general distributed system, except that it provides the following additional capabilities:

- Every node in the cluster has full connectivity and communication with other nodes in the cluster.
- Cluster nodes are aware when a node joins or leaves the cluster.
- Each node knows what components it has as well as what components are available on every other node in the cluster.
- It ensures reliable delivery of messages between the nodes in the cluster.

Figure 1-1 shows a typical cluster arrangement.

Figure 1-1 Typical Cluster Arrangement



The BEA TUXEDO System High Availability Characteristics

This section describes a number of BEA TUXEDO system characteristics that make it a natural choice for high availability applications because it comes with powerful, built-in high availability features as follows:

- Heartbeat Mechanism
- Buddy System
- Server Process Monitoring and Restart
- Multiple Servers Single Queue
- Multiple Server Instances on Multiple Server Nodes

- Remote Client Automatic Reconnection
- Network Redundancy for Communications Between Nodes and Between Domains

Heartbeat Mechanism

The BEA TUXEDO system has a heartbeat mechanism for monitoring server nodes and the network links connecting server nodes. Periodically, BBLs on each server node send an I'm OK message to the DBBL on the master node. The DBBL checks to make sure that all BBLs have reported in within a cycle. If a BBL has not been heard from, the DBBL sends a message to that BBL asking for status and waits for a reply. If no reply is received in the waiting period, the BBL is partitioned and all the services provided by the servers associated with the LMID of that BBL are unadvertised. The BBL is unpartitioned when an I'm OK message is received again by the DBBL.

The BEA TUXEDO system parameters that control the timing of the heartbeat cycle are documented in ubbconfig(5) in the *BEA TUXEDO Reference Manual*.

Buddy System

The administrative servers (BBL and BRIDGE) of a configured BEA TUXEDO system LMID implement a buddy system by monitoring each other for liveness. If one server fails, the other restarts it.

Server Process Monitoring and Restart

Every BBL of a BEA TUXEDO application monitors the server processes associated with its LMID for viability. A server process is considered to be not viable if it has terminated abnormally or if it is looping (such that the server has not returned in SVCTIMEOUT seconds since it started to process a service request). Server processes deemed not viable are either cleaned up or restarted depending on their configuration options.

Multiple Servers Single Queue

MSSQ is a powerful server process configuration model. By associating multiple server processes with a single queue for service requests, almost continuous service availability is assured. A failing server process affects only the current request. The remaining service requests are served by the remaining available server processes. The result is not only an availability feature but also a performance enhancer as it allows concurrent service processing.

Multiple Server Instances on Multiple Server Nodes

The MSSQ model addresses server process failure on a given server node but does not address server node or network link failure. By providing the same services through servers on multiple nodes, almost continuous service availability is assured even when a server node becomes unavailable. Failure conditions are detected by the BEA TUXEDO system heartbeat mechanism and the server node is partitioned from the rest of the domain. Subsequent service requests are routed to the surviving server nodes offering the services.

Remote Client Automatic Reconnection

The `WSNADDR` environment variable can contain a series of IP addresses to be used to listen for remote clients trying to connect to a BEA TUXEDO system application. When a server node and/or a network link fails, the remote client receives a disconnect error. In such a case, the remote client terminates the current session by calling `tpterm()`, then reconnects by calling `tpinit()`. The `tpinit()` function scans the sequence of IP addresses and reconnects to the first one available.

Network Redundancy for Communications Between Nodes and Between Domains

Inter-BRIDGE communications support network redundancy between nodes and between Domains. Up to two physical network links between server nodes and between Domains are supported for data transfer. When one link goes down, the failure condition is detected and the network load is failed over to the alternate link automatically. When the failed link is restored, the failback of the data flow to its original link occurs automatically.

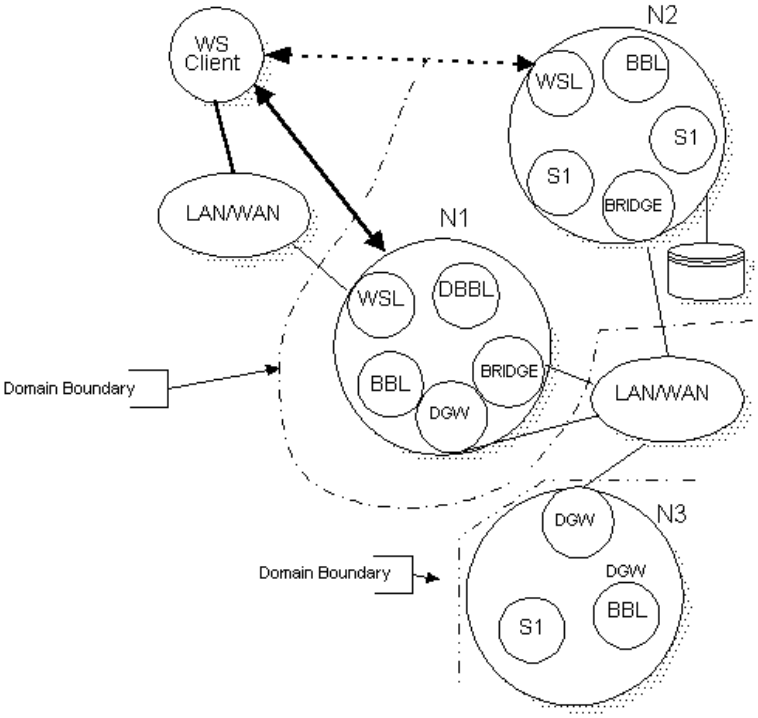
Summary

The characteristics mentioned support the assertion that even on its own (without the benefit of cluster technology) the BEA TUXEDO system has many built-in characteristics that enable users to develop mission-critical applications.

The BEA TUXEDO system parameters that control the features mentioned in this section are documented in ubbconfig(5) of the *BEA TUXEDO Reference Manual*. A typical BEA TUXEDO system distributed application that includes Workstation clients and Domains gateways is illustrated in Figure 1-2.

In Figure 1-2, S1 represents servers, which can be either application servers of administrative servers.

Figure 1-2 Typical Distributed BEA TUXEDO System Configuration



Microsoft Windows NT Cluster Server

Microsoft Windows NT Cluster Server (MSCS) is Microsoft's clustering technology for Windows NT Servers. The MSCS software is written by Microsoft, but incorporates algorithms and expertise provided by Microsoft's core partners in the MSCS initiative: Compaq, Digital, IBM, Intel, HP, NCR, Tandem.

A Windows NT cluster is a configuration of two independent Windows NT Server computer systems, which appear to network clients and administrators to be a single, highly available Windows NT Server system.

A cluster has its own identity distinct from the identities of its constituent nodes. This identity is used by the network clients to access server applications running on the cluster and by administrators to manage the cluster. Clusters provide high availability by detecting hardware and software failures and automatically failing over data and processing to healthy nodes. Because a cluster projects its own unique identity, services can be moved between nodes, failed nodes can be replaced or more nodes can be added to a cluster while users continue to see a single computer system.

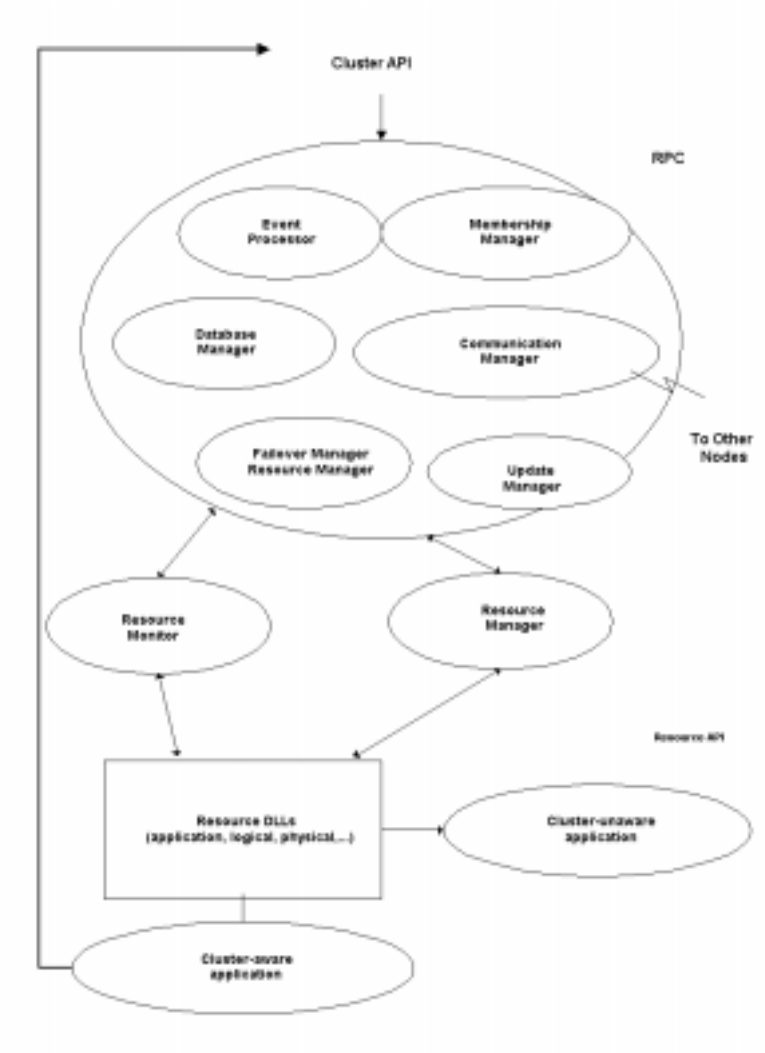
MSCS Elements

An MSCS cluster consists of various physical and logical components as follows:

- Cluster Service
- Cluster Resource
- Resource Monitor
- Resource Group (Virtual Servers)
- MSCS Communication
- MSCS Security
- MSCS Management

These components are described in the sections that follow. The relationships among them are shown in Figure 1-3.

Figure 1-3 Cluster Node Components



Cluster Service

The Cluster Service is the core component of an MSCS cluster. It is implemented as a Windows NT service and controls cluster services like coordinating event notification, facilitating communication between cluster components, handling failover operations, and managing configuration.

The Cluster Service runs on every node in the cluster. It is made up of several closely related subcomponents as listed in Table 1-1.

Table 1-1 Cluster Service Subcomponents

Communication Manager	Maintains communication between cluster members; blocks communication when a node fails
Database Manager	Maintains the cluster configuration database
Event Processor	Initializes the Cluster Service and supports event notification between cluster components
Failover Manager Resource Manager	Work together to manage resource dependencies, start and stop resources and initiate the transfer of cluster groups to alternate nodes
Global Update Manager	Provides atomic update services to cluster members
Membership Manager	Maintains cluster membership. Keeps track of the nodes that are active in a cluster as well as those that are configured to participate but are currently inactive

Cluster Resource

A cluster resource is any physical or logical entity used to provide a service to cluster clients. Examples of resources are disks, network names and addresses (that is, DNS names and IP addresses), databases, web sites, print queues, network file shares, and application programs. A resource is hosted on a single cluster node at any point in time.

Resources are categorized by type, which include physical hardware such as SCSI disks and logical items such as TCP/IP addresses, file shares, and generic applications. Each resource type has a collection of properties that describe resources of the type. For example, a physical disk includes the name of its resource DLL and two polling values. Other resource types may define properties specific to their corresponding types. Also associated with each resource is a list of nodes in the cluster that can host the resource, and a local restart policy. The restart policy specifies how many times the Cluster Service should attempt to restart a resource on the same node when it fails.

Resources and resource types publish configuration information in the cluster-wide configuration database that is accessed through the functions in the cluster API.

Resources can be dependent on any number of other resources. Dependency implies a two-way association that affects the timing and order of failover.

Each cluster has a special resource known as the quorum resource. The quorum resource is used as a tie-breaker when the nodes in a cluster are running, but cannot communicate. The quorum resource is any resource which can be physically arbitrated for and controlled by only one node, for example, a SCSI disk.

MSCS supports several of the most common types of resources by providing DLL files that communicate with the Resource Monitors and the Cluster Service. Support for other resource types can be added by third-party vendors using the API provided in the MSCS SDK.

- The MSCS Clustering SDK includes resource DLLs for the following resource types: Fault Tolerant Disk
- Generic Application
- File Share
- Generic Service
- Internet Information Server Virtual Root
- IP Address
- Network Name
- Physical Disk
- Print Spooler
- Time Service

All resources can have the following states:

- Offline
- Online
- Offline Pending
- Online Pending
- Failed

When a resource is offline, it is unavailable for use by a client or another resource. When a resource is online, it is available for use. When a resource is in one of the pending states, it is in the process of either being brought online or offline. If the pending resource cannot be brought online or taken offline after a specified amount of time, the resource is set to the failed state.

Resource Monitor

The Resource Monitor tracks the state of cluster resources, notifying the Cluster Service of resource state transitions. Each node in a cluster can have one or more Resource Monitors assigned to one or more resources.

Resource Monitors contain no decision-making algorithms or policy; they simply carry out the commands of the Cluster Service. To determine if a resource has failed, a Resource Monitor calls functions of the MSCS resource API.

Resource Monitor runs in a process separate from the Cluster Service process to prevent resource DLLs from hanging the Cluster Service or interfering with its operations. Cluster service failure can be detected by the Resource Monitor, which responds by initiating an immediate system shutdown.

Resource Group

A resource group is a collection of resources that are related and/or dependent on each other and that must be hosted together on one node. For example, a database resource depends on the disk resource that contains the database files. These two resources would be in a group that offers the database service. A resource upon which another resource depends must be brought online before the dependent resource and taken offline after it.

Unrelated and/or independent resources may be placed in the same group for administrative convenience.

If a hardware or software failure occurs that affects a resource group, the group fails over from the node experiencing the failure to another node which is capable of hosting the group. A resource group may also be moved from one node to another under administrative control for load-balancing or maintenance purposes.

The resource group is the unit of failover in MSCS.

Associated with each group is a cluster-wide policy that specifies on which node the group prefers to run and to which node the group should move in case of a failure.

Resource groups that contain an IP address resource and a network name resource are referred to as virtual servers. Clients requesting the services provided by a virtual server connect to the virtual server instead of connecting to the node that hosts the virtual server. The underlying physical platform becomes totally transparent to them.

Most groups contain the following types of resources:

- IP address
- Network name
- Physical disk
- Generic application or service

MSCS Communication

MSCS relies on the Windows NT network services for handling cluster communication.

Nodes in a cluster communicate using their Cluster Services. The Cluster Service tracks the current state of the nodes within a cluster and determines when a resource group and its resources should fail over to an alternate node. One Cluster Service polls the other Cluster Service and then waits for an acknowledgment. This polling takes place using User Datagram Protocol (UDP).

The Cluster Service communicates with resources through the clustering software component known as the Resource Monitor. The Resource Monitor controls and monitors cluster resources and reports any changes in the resources to the Cluster Service via local interprocess communications (IPC). The Resource Monitor accepts requests from the Cluster Service and forwards those requests to the relevant resource DLLs. When a resource detects a change in its state, it reports that change to the Resource Monitor, which in turn forwards it to the Cluster Service.

Cluster-aware applications, such as the Cluster Administrator, communicate with the Cluster Service on a node using the Cluster API and RPC over TCP/IP.

MSCS Security

MSCS relies on Windows NT security (WNT network security) and NTFS (file/share level security) at all levels for controlling authentication and access.

MSCS Management

The Cluster Administrator is the tool you use to interact with all the cluster operations. It is a GUI based tool. An illustration is shown in Figure 1-4. With this tool, you can:

- Create resources and groups
- Set properties for resources and groups
- Change the online status of groups and resources
- Pause and restore a cluster node
- Delete a resource or group
- Rename a group or resource
- Initiate a resource failure
- Move a resource from one group to another through various wizards and dialog boxes

Figure 1-4 A Cluster Administrator Parameters Properties Window

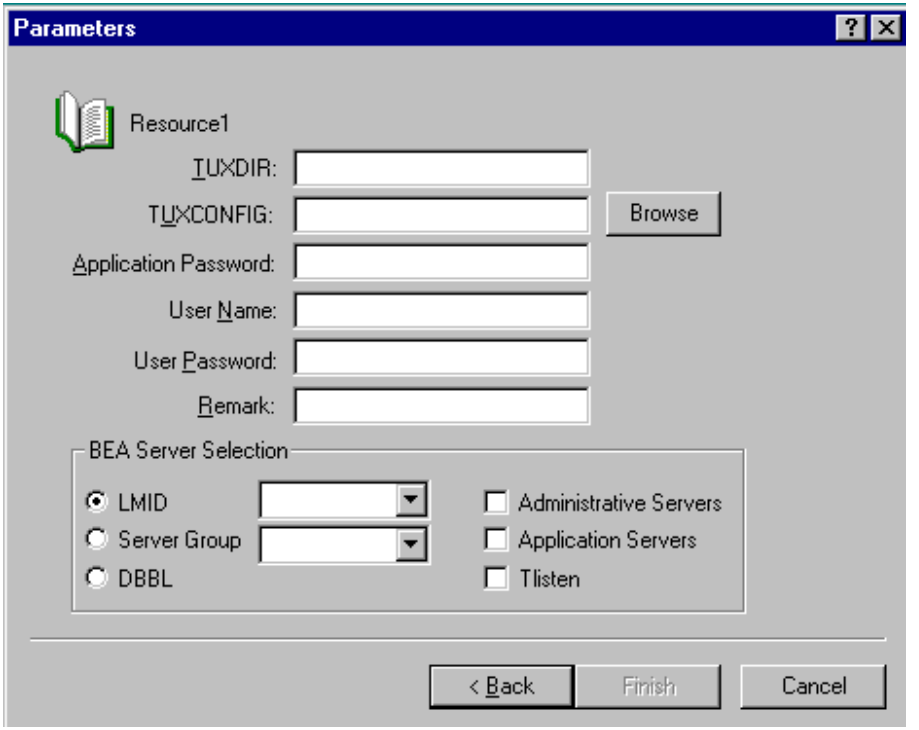


Figure 1-4 is actually the Parameters properties window for the BEA Application resource. This window is shown again in Administration Scenarios. The use of the individual fields is described there.

Administering MSCS largely involves managing groups and resources.

An administrator can set the following properties of a resource group:

- Group name
- Group description
- Preferred owner node
- Failover policy
- Failback policy

An administrator can set the following properties/parameters of a resource:

- Resource description
- Owner nodes
- If a resource is restartable and if so, then how many attempts and in what period
- How often to do liveliness check
- Resource dependencies
- Maximum amount of time in pending state
- Executable image (generic application resource type)
- Command-line arguments for executables (generic application resource type)
- Directory for image files (generic application resource type)

BEA MSCS Extensions

The BEA MSCS Extensions product integrates the inherent high-availability characteristics of the BEA TUXEDO system with those of Microsoft's MSCS.

To implement BEA MSCS Extensions, a new MSCS resource type, known as a BEA Application resource, was developed. Two DLLs are provided for the implementation:

- A resource DLL
- A Cluster Administrator extension DLL

With the MSCS Extensions product you can identify a previously developed BEA TUXEDO application as an MSCS-aware resource.

The BEA Application Resource Type

Once the new BEA Application resource is defined, you can

- Monitor BEA Application resource servers for liveness from MSCS or from the BEA TUXEDO system itself
- Bring a BEA Application resource online or offline, or terminate it.

This approach effectively makes a BEA MSCS Extensions application an MSCS-aware application. The resource DLL implements the resource API and the corresponding functionality as specified by MSCS.

2 Configuration Scenarios

This chapter covers the following topics:

- BEA TUXEDO Configurations in MSCS
- Active/Hot Spare Configuration
- Active/Active Configuration
- BEA Applications Involved in Transactions
- BEA Applications with Remote Clients
- Special Configuration Considerations

BEA TUXEDO Configurations in MSCS

In the MSCS environment, a BEA Application resource is the same as any other cluster-based application providing services to its clients over the connected network. It is viewed as a resource running on cluster nodes.

A typical non-MSCS BEA TUXEDO application consists of multiple server groups running on virtual machines dispersed on the physical nodes of an underlying network. These server groups have both administrative and application server processes, associated resources like IP addresses (at which the `BRIDGE` and `tlisten` processes listen), and network names (DNS names for the associated virtual machines). They also need access to some persistent storage resources (that is, physical disks) for

TUXCONFIG, TLOGs, queue spaces, ACLs, Event Broker databases, and application-specific databases. In this respect, a BEA TUXEDO instance can be viewed as an MSCS resource group (virtual server) that has the following resources:

- Application server processes
- Administrative processes
- Physical disks
- IP addresses
- A DNS name for the virtual machine

An MSCS cluster-based BEA Application resource can be configured in two basic ways depending on your enterprise's performance and availability needs:

- Active/Active Configuration
- Active/Hot Spare Configuration

Active/Hot Spare Configuration

In the Active/Hot Spare configuration, the application runs on one node with a second node available as a stand-by.

Single Server BEA Application (SHM Mode)

The BEA TUXEDO shared memory (SHM) mode is mainly used in early development stages because it assumes that the entire application runs on one processor. This is the antithesis of cluster technology, but it does serve as an illustration of the Active/Hot Spare configuration. An example of the SHM mode is shown in Figure 2-1. In this configuration, node B of the MSCS cluster is a hot spare.

Figure 2-1 Active/Hot Spare Configuration

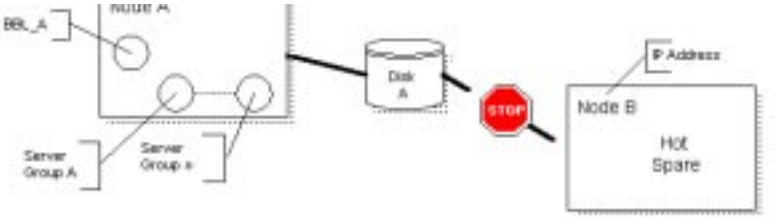


Table 2-1 lists the application resources diagrammed in Figure 2-1. Note that all the resources are in Group A.

Table 2-1 Single Server Resources

MSCS Resource	Resource Group	MSCS Resource Type	Description	Resource Dependency
LMID_A	A	BEA Application	Administrative and Application Servers of LMID_A, that is, the BBL and Server Group A to *n* in Figure 2-1	DISK_A IP_A
DISK_A	A	Physical disk	Shared disk drive	
IP_A	A	IP address	IP address at which the WSL process waits for client requests	

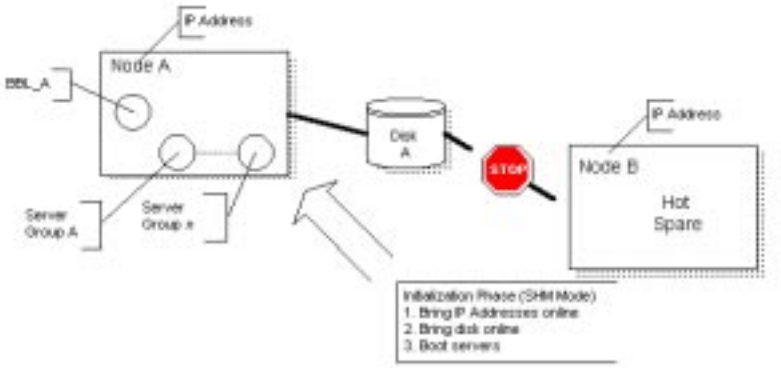
Before the application is booted, the PMID environment variable on Node A is set to PMID_A.

Configuration files, transaction logs, queue spaces, and databases all reside on the shared disk drive.

Initialization Phase (SHM Mode)

During initialization, the administrator brings the resources online starting with the physical disk and the IP address resources. Once those resources are online, the servers can be brought online. The process is illustrated in Figure 2-2.

Figure 2-2 Initialization Phase (SHM Mode)



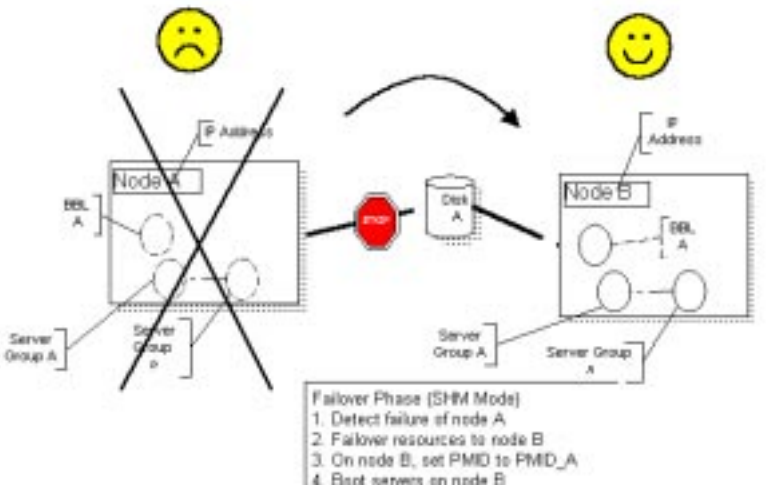
Failover Phase (SHM Mode)

If node A fails, MSCS takes the following actions automatically:

1. Detects failure of node A.
2. Fails over resources in this order:
   a. Fails over resource `DISK_A` onto node B.
   b. Fails over resource `IP_A` onto node B.
   c. Fails over resource `LMID_A` onto node B.
3. Sets environment variable `PMID` to `PMID_A` on node B.
4. Boots `LMID_A` on node B.

This process is illustrated in Figure 2-3.

Figure 2-3 Failover Phase (SHM Mode)



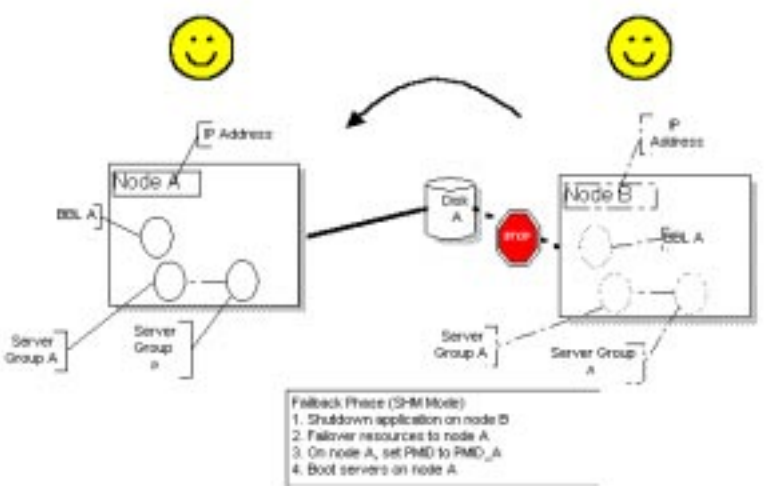
Failback Phase (SHM Mode)

When node A is restored, the administrator manually returns the BEA Application resource to its original node (node A) by moving MSCS resource group A from node B to node A. To accomplish this the administrator does the following:

1. Shuts down BEA Application resource on node B.
   a. Fails back resource `DISK_A` onto node A.
   b. Fails back resource `IP_A` onto node A.
   c. Fails back resource `LMID_A` onto node A.
2. Sets environment variable `PMID` to `PMID_A` on node A.
3. Boots `LMID_A` on node A.

The failback phase is illustrated in Figure 2-4.

Figure 2-4 Failback Phase (SHM Mode)



Active/Active Configuration

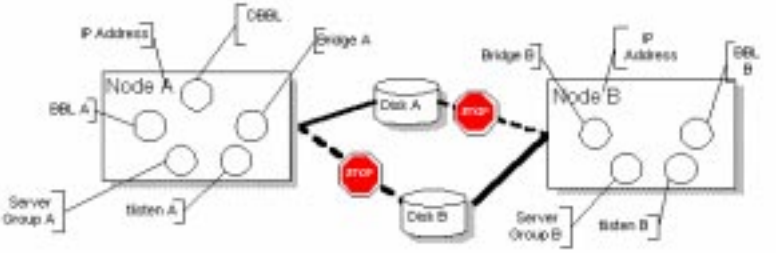
In the Active/Active configuration the application resources are allocated to more than a single node.

Multi-Server Domain BEA Application (MP Mode)

In an Active/Active Configuration the resources are typically split between the nodes of the cluster so the resources on each node run at optimum performance. This approach is illustrated in Figure 2-5. When one node fails, MSCS moves the BEA Application resource group from the failed node to the other node.

A configuration of this kind offers acceptable performance plus high availability when one node is offline and high performance when both nodes are online.

Figure 2-5 Active/Active Configuration



This example is based on the BEA TUXEDO application that has resources as listed in Table 2-2.

Table 2-2 Multi-Server Resources

MSCS Resource	Resource Group	MSCS Resource Type	Possible Owners	Description	Resource Dependency
DBBL	FAILOVER GROUP_A	BEA Application resource	Node A Node B	DBBL server	
DISK_A	FAILOVER GROUP_A	Physical disk	Node A Node B	Disk A	
GROUP_A	FAILOVER GROUP_A	BEA Application resource	Node A Node B	Servers of server group GROUP_A	DISK_A DBBL
BBL_LMID_A	BBL LMID_A	BEA Application resource	Node A	Administrative servers of LMID_A	
BBL_LMID_B	BBL LMID_A	BEA Application resource	Node A	Administrative servers of LMID_B	
DISK_B	FAILOVER GROUP_B	Physical disk	Node A Node B	Disk B	

Table 2-2 Multi-Server Resources

MSCS Resource	Resource Group	MSCS Resource Type	Possible Owners	Description	Resource Dependency
GROUP_B	FAILOVER GROUP_B	BEA Application resource	Node A Node B	Servers of server group GROUP_B	DISK_B
TLISTEN_A	TLISTEN_A	Generic application or BEA Application resource	Node A	tlisten process associated with LMID_A	
TLISTEN_B	TLISTEN_B	Generic application or BEA Application resource	Node B	tlisten process associated with LMID_B	

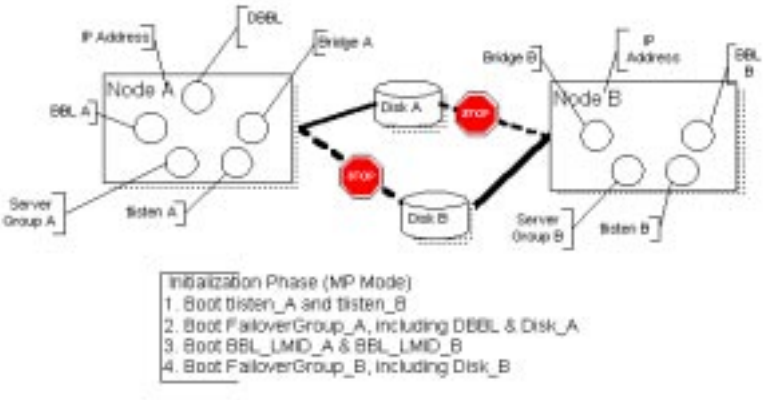
In our sample configuration, the following assumptions hold:

- LMID_A is associated with node A (PMID_A) and LMID_B is associated with node B (PMID_B).
- LMID_A is the primary LMID and LMID_B is the backup LMID for server group GROUP_A.
- LMID_B is the primary LMID and LMID_A is the backup LMID for server group GROUP_B.
- Both BBL_LMID_A and BBL_LMID_B are configured to be owned only by node A (the move operation does not apply).
- TLISTEN_A is configured to be owned only by node A (the move operation does not apply).
- TLISTEN_B is configured to be owned only by node B (the move operation does not apply).
- TLOGs, queue spaces and other BEA TUXEDO system-specific or application-specific databases are configured to reside on the shared disks.

Initialization Phase (MP Mode)

During initialization, the administrator brings the resources online starting with the physical disk and the IP address resources. Once those resources are online, the servers can be brought online. The sequence is illustrated in Figure 2-6.

Figure 2-6 Initialization Phase (MP Mode)



During the initialization phase the administrator can bring the application resources online in either of the following two ways:

- By taking advantage of Resource Group definitions to bring resources online in parallel.
- By bringing each resource online separately, observing the dependencies shown in Table 2-2.

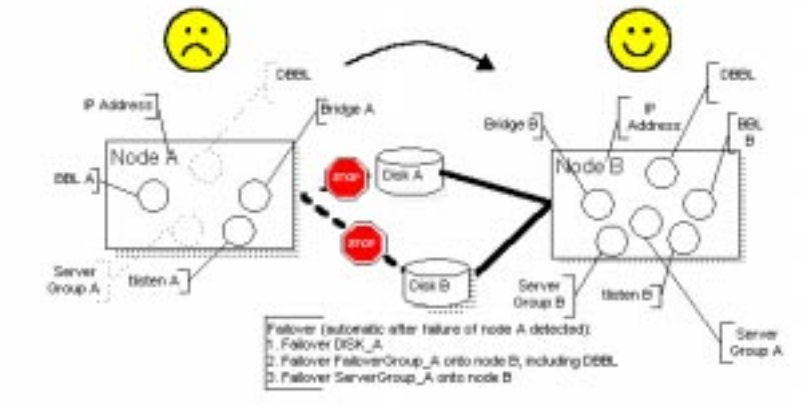
Failover Phase (MP Mode)

If node A fails, MSCS takes the following actions:

1. Detects failure of node A.
2. Fails over `DISK_A`.
3. Fails over `FAILOVERGROUP_A`, which includes the `DBBL`.
4. Fails over `SERVERGROUP_A`.

The process is shown in Figure 2-7.

Figure 2-7 Failover Phase (MP Mode)



Failback Phase (MP Mode)

When node A is restored, the administrator boots the administrative servers (the BBL and BRIDGE processes) of LMID_A from node B (current MASTER node) before moving resource group FAILOVERGROUP_A back to its original host (node A). Bringing LMID_A online is also possible once the resource is received by node B. However, after bringing the resource online, node B should be removed from the list of possible hosts for this resource. Not removing it causes MSCS to attempt to failover LMID_A onto node B when node A fails again and this operation will certainly fail as node A has already crashed.

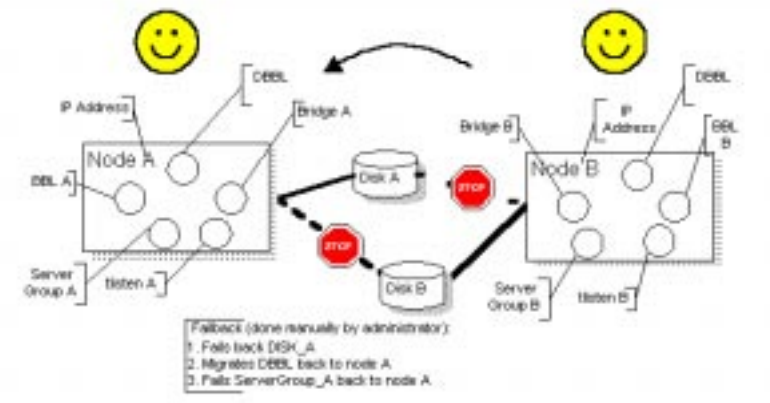
Moving resource group FAILOVERGROUP_A back to node A triggers the following actions. The administrator manually:

1. Fails back resource DISK_A to node A.
2. Migrates DBBL back to node A.
3. Fails back SERVERGROUP_A to node A.

TLISTEN_A is brought online automatically by MSCS when node A is restored.

The process is shown in Figure 2-8.

Figure 2-8 Failback Phase (MP Mode)



BEA Applications Involved in Transactions

The BEA MSCS Extensions product supports transparent and automatic failover of transactional BEA TUXEDO server groups, provided the relevant TLOGs and application databases are configured to reside on the shared disks and the shared disks are failed over before the transactional server groups are failed over.

The BEA MSCS Extensions product expects the disks containing the TLOG of the LMID associated with the failed node to be available locally on the surviving node following a failover. However, in a planned failover of transactional server groups, there is no need to move the shared disk containing the TLOG prior to the move because the TLOG remains available through its source LMID.

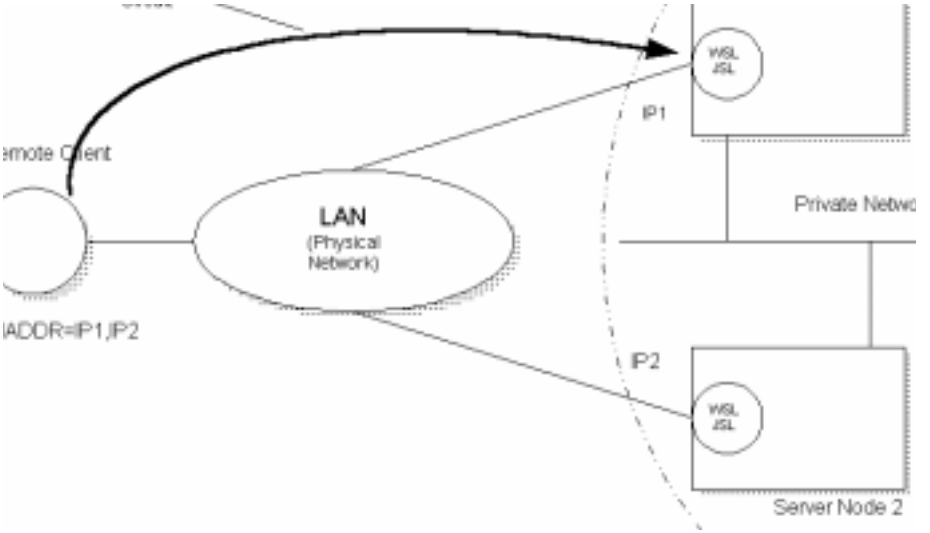
Finally, note that a transactional LMID attempts to open the associated TLOG file during boot and fails if it cannot. This means that a booting transactional LMID expects the disk containing the associated TLOG to be available locally.

BEA Applications with Remote Clients

The BEA TUXEDO system supports multiple IP addresses for remote client access to server nodes via the `WSNADDR` environment variable. This allows you to configure multiple workstation listener processes (WSLs) residing on different server nodes, listening at different IP addresses for incoming connection requests from remote clients. If a remote client connection attempt fails, the BEA TUXEDO Workstation component tries subsequent IP addresses (as specified in `WSNADDR`) until it succeeds.

The only requirement on the remote client (once the outstanding request results in a disconnect error) is to terminate the current session by invoking `tpterm()` and reconnect to the BEA TUXEDO application by invoking `tpinit()`. Figure 2-9 shows the arrangement with remote clients before a failure occurs.

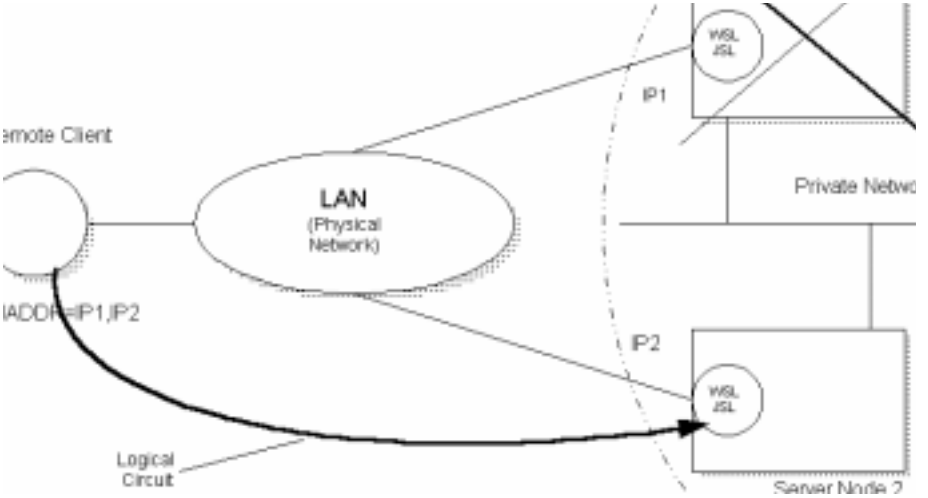
Figure 2-9 Remote Client before Failure



Remote Client Failover

When Server Node 1 fails, the reconnection attempt by a remote client via `tpinit()` connects the remote client to the BEA TUXEDO application at IP2. This is shown in Figure 2-10.

Figure 2-10 Remote Client after Failover

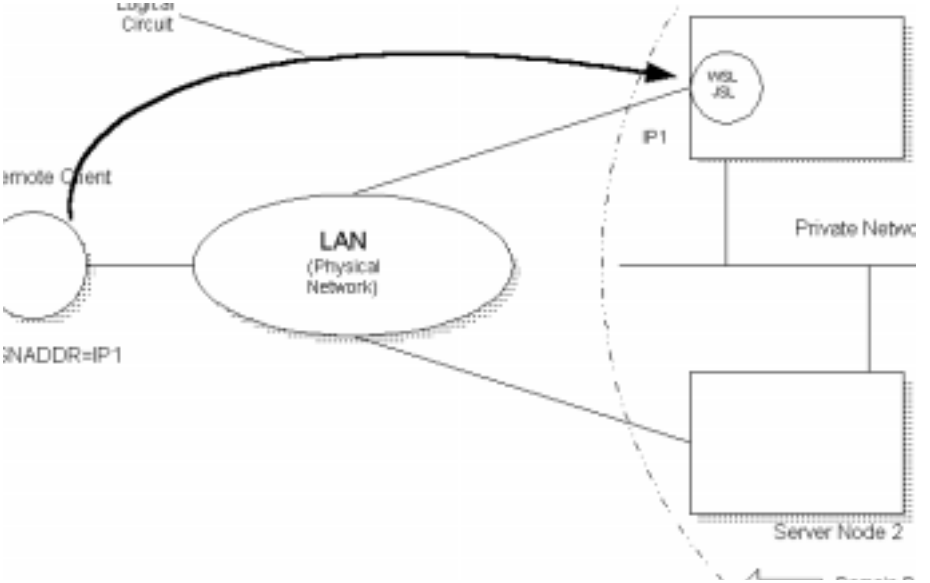


Alternative Way of Configuring Remote Clients

Another variation of high-availability BEA TUXEDO configurations involving remote clients is diagrammed in Figure 2-11 and Figure 2-12.

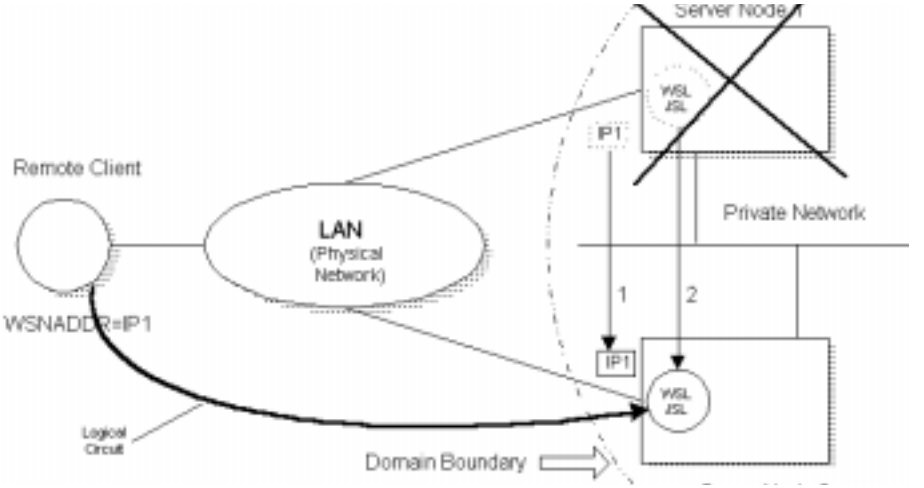
In this method, remote client access to a BEA TUXEDO application is via only one server node just as though the application was configured as a `SHM` mode BEA TUXEDO application. Only one IP address is specified in `WSNADDR`; the address is the IP address of Server Node 1.

Figure 2-11 Remote Client Alternative before Failover



When Server Node 1 fails, MSCS fails over all the listening IP addresses to Server Node 2 followed by the server groups with which the WSLs are associated (the failover sequence is dictated by the dependencies). To implement this, configure all the listening IP addresses as MSCS resources of type IP Address, and configure the BEA TUXEDO server groups containing WSLs as resources of type BEA Application resource. Finally these resources are configured in the same MSCS resource group with WSLs having a dependency on IP address resources.

Figure 2-12 Remote Client Alternative after Failover



Special Configuration Considerations

During the configuration of a BEA TUXEDO application for MSCS, the administrator should apply the following changes to the BEA TUXEDO configuration in order to support the additional workload moved over (planned or unplanned) from one MSCS cluster node to the other.

- Identify all BEA TUXEDO domain nodes that are also members of the MSCS clusters configured.
- Determine the maximum `MAXACCESSERS` value (`max(MAXACCESSERS)`) configured with any `LMID` associated with the cluster nodes.
- Increase the value of `MAXSERVERS` (in the `RESOURCES` section of configuration file) by `max(MAXACCESSERS) - MAXSERVERS`.

Both primary and secondary `LMID`s associated with a BEA Application resource have to be hosted on cluster nodes for planned or unplanned failover and failback.

Because of the current BEA TUXEDO limitation of one TLOG per LMID rather than one TLOG per server group and the BBL's dependency on TLOG for certain operations, for example, warm start, it is necessary to failover and failback all the transactional servers associated with a LMID in the same resource group together with the physical disk hosting the TLOG and other shared dependent resources, for example, disks hosting the databases or IP addresses.

The tlisten process should be configured as a resource of Generic Service type. It can also be configured as a BEA Application resource. However, in the latter case, bringing it online requires the existence of TUXCONFIG on the local node and it can only be brought online from the local node. When tlisten is configured as a BEA Application resource, it runs as a non-service process and can be monitored for liveness.

BEA Application resources should always be configured to run in a separate Resource Monitor. Doing so provides two advantages:

- It prevents potential conflict caused by the process environment being shared by all the worker threads within a Windows NT process.
- It facilitates the setting of certain environment variables by the BEA Application resources during online/offline operations.

We recommend that TLOGs be configured on shared disks other than those used by other application shared resources.

CHAPTER

3 Administration Scenarios

This chapter covers the following topics:

- Creating an MSCS-Aware Application
  - Adding a BEA Application as a Resource
  - Parameters Properties Window: Common Area
  - Parameters Properties Window: Server Selection Area
  - Adding a BEA Application Resource from the Command Line
  - Error Codes
  - How Your Choices Affect the Way the Application Works
- How MSCS Operations Map to BEA TUXEDO Operations
  - Offline to Online
  - Online to Offline
  - Move

Creating an MSCS-Aware Application

In Cluster Technology Overview, we described some of the basic attributes of the Microsoft Cluster Server (MSCS).

In Configuration Scenarios, we described how some typical BEA TUXEDO applications can be mapped to MSCS configurations.

This section describes how a user of MSCS and the BEA MSCS Extensions can use the MSCS interface to turn a BEA TUXEDO application into an MSCS BEA Application resource.

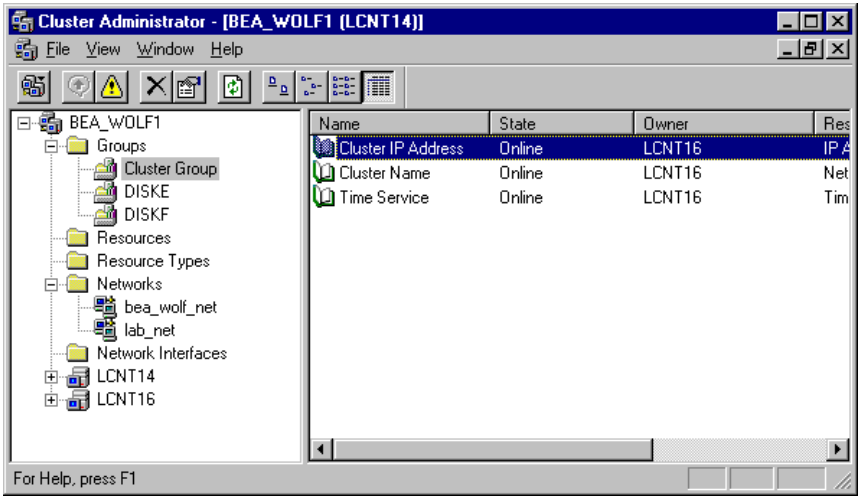
Adding a BEA Application as a Resource

The configuration windows shown in Figure 3-2 to Figure 3-7 are under the control of the NT Cluster Administrator.

To reach the Cluster Administrator utility from the NT desktop do the following:

1. From the Start menu, select Programs→Administrative Tools (Common)→Cluster Administrator. You see a window like Figure 3-1.

Figure 3-1 Cluster Administrator Window



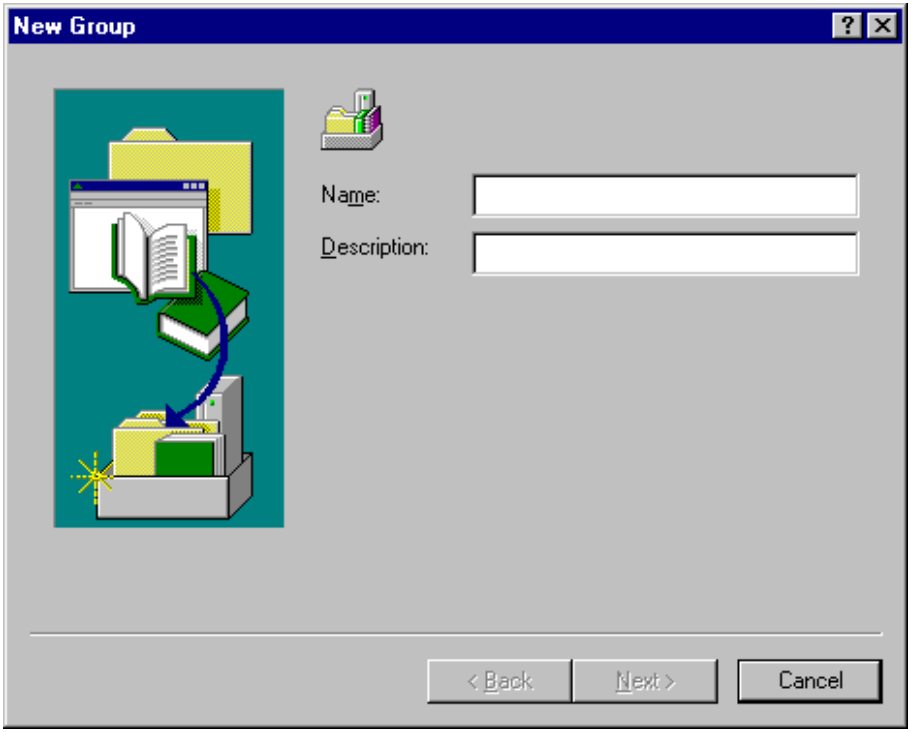
2. Use the New Group window to add a new group to your cluster. The New Group window is shown in Figure 3-2.

3. Do the following:

a. Select File→New→Group.

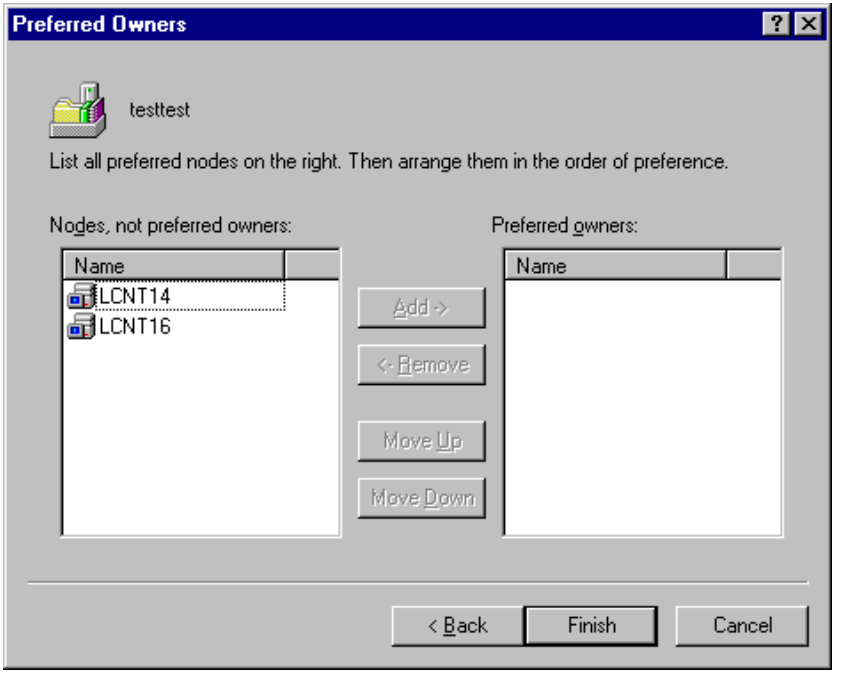
b. Enter a group name and description.

Figure 3-2 New Group Window



c. Click Next. That takes you to the Preferred Owners window shown in Figure 3-3.

Figure 3-3 Preferred Owners Window



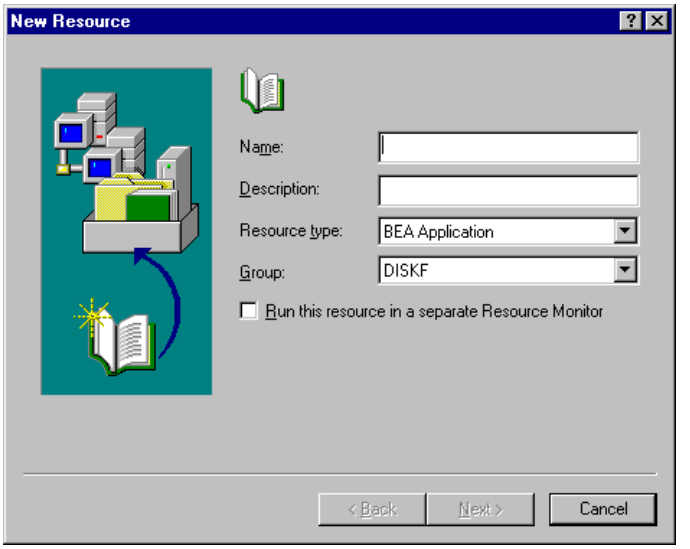
d. Designate preferred owners. (In this case, owners means machines, not persons.)

e. Click Finish.

The BEA MSCS Extensions product enhances MSCS's configuration GUI so that you can define resources of the BEA Application resource type. The enhanced GUI provides a pair of BEA Application resource Parameter properties windows. One of these windows is used by the New Resource creation wizard and the other is used for modifying attributes. Both versions look like the Parameter properties window shown in Figure 3-7.

To add a BEA Application resource as a resource, while in the Cluster Administrator window, select File→New→Resource. This opens the New Resource window like the one shown in Figure 3-4.

Figure 3-4 New Resource Window



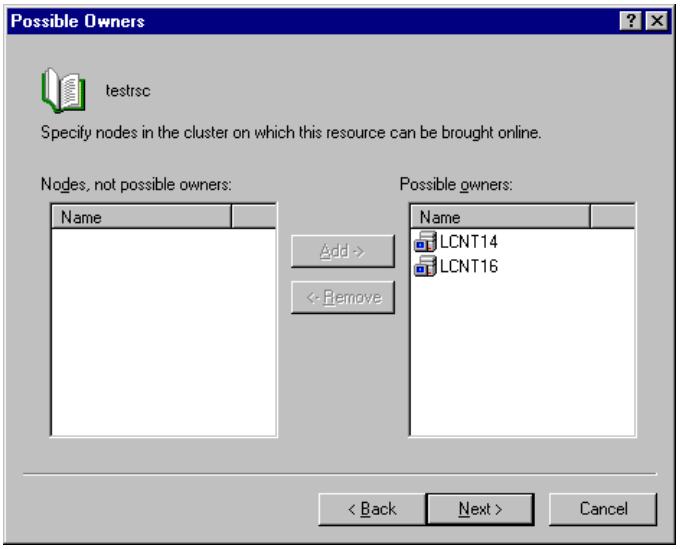
Specify the name and description of the resource, then from the pull-down lists, elect the following attributes relating to the resource:

- Resource type (BEA Application from the list of available resource types).
- Name of the resource group to which the resource belongs—this provides a unique identity for each resource in the cluster.

The New Resource window also offers an opportunity to specify a preference for a separate Resource Monitor. Specify this by checking the Run this resource in a separate Resource Monitor check box.

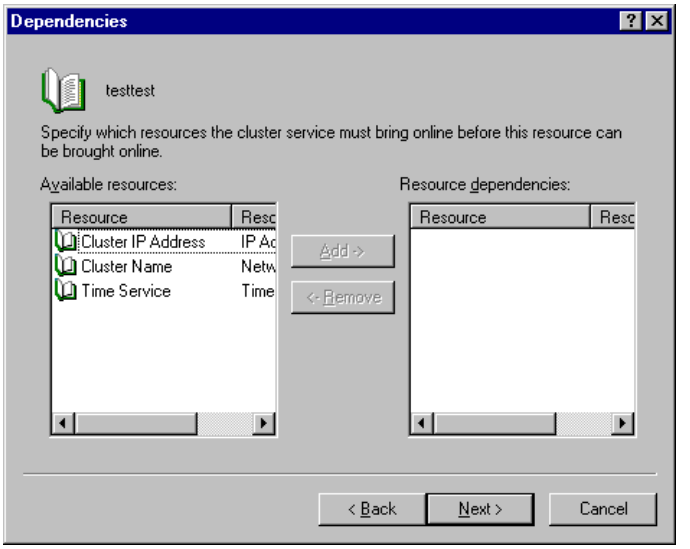
Click Next and you move on to the Possible Owners window shown in Figure 3-5.

Figure 3-5 Possible Owners Window



- List of possible owners, that is, machines in the cluster that can host this resource—defines machines in a cluster capable of running the resource.
- After identifying possible owners of the resource, click Next and you move on to the Dependencies window as shown in Figure 3-6.

Figure 3-6 Dependencies Window



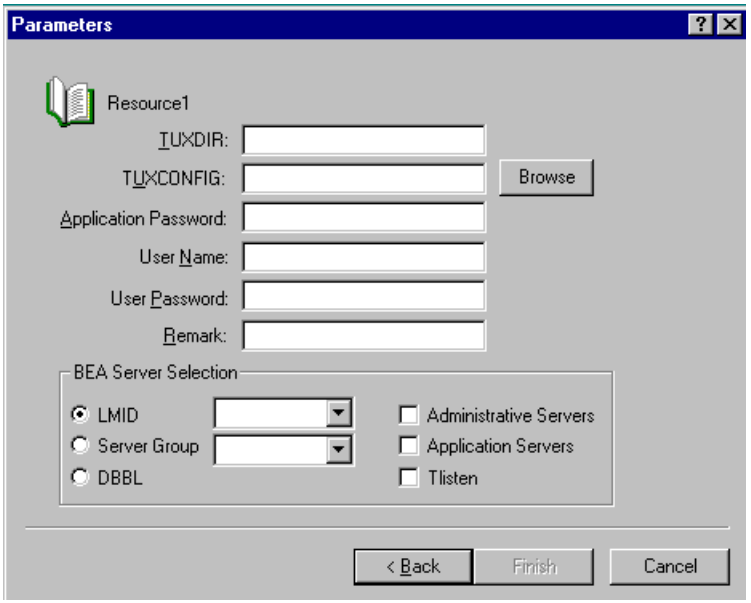
- Resource dependencies—names other resources on which this resource is dependent.

Once these attributes are specified, Click Next in the Dependencies window and you move on to the Parameters properties window shown in Figure 3-7.

Adding A Resource from the Command Line

Please see the MSCS Administrator's Guide for information on creating groups and resources via the command line interface.

Figure 3-7 Parameters Properties Window



Parameters Properties Window: Common Area

The parameter fields for a BEA Application resource common area have the following meaning:

TUXDIR
: The root directory of a BEA TUXEDO installation. The default value is the root directory of the most recently installed version of the BEA TUXEDO product for the node associated with the current Resource Group. This value is retrieved from the node's Registry database. This is a required field.

TUXCONFIG
: The full path name of the configuration file of this BEA TUXEDO application. This is a required field. There is a File Browse control so you can look for the correct path if you do not remember it. (For remote administration, remember to select the drive and folder.)

Application Password
: The password for this application (if it has one). This is a required field if the application is configured with `SECURITY` set to `APP_PW`, `USER_AUTH`, `ACL`, or `MANDATORY_ACL`.

User Name
: Your login ID. This is a required field if the application is configured with `SECURITY` set to `USER_AUTH`, `ACL`, or `MANDATORY_ACL`.

User Password
: Your user password. This is a required field if the application is configured with `SECURITY` set to `USER_AUTH`, `ACL`, or `MANDATORY_ACL`.

Remark
: Any brief helpful comment can be entered. The field is always optional.

Parameters Properties Window: Server Selection Area

In the common area the values entered may apply to more than a single instance of the BEA Application resource type. There is another section of the page, the Server Selection group box, for entering values that apply to a single instance of a BEA Application resource. Parameter specification is not complete until a valid combination of fields from the common area and the Server Selection group box is entered.

Mutually exclusive or irrelevant controls are automatically disabled during configuration; the configuration process is complete when all required fields are filled in. (The Finish button is disabled until the selection is complete.)

Choose one of the following valid combinations of controls in the BEA Application resource Server Selection group box:

- DBBL
- LMID + Administrative Servers
- LMID + Application Servers
- LMID + Administrative Servers + Application Servers
- Server Group (meaning all the servers in a group, both administrative and application servers)
- LMID + `tlisten`

Accessing Online Help or Online Documentation

From the Parameters properties window you can get help in the following two ways:

- You can get brief reminders of what you should enter in each field. To get that help, press the F1 key or click on the Question Mark icon in the upper right corner of the page, drag the Question Mark to any field, and release it.
- You can launch a browser and jump directly to this guide; simply right click on any static field on the page (that is, on any of the labels within the dialog).

Adding a BEA Application Resource from the Command Line

The following table lists valid values for BEA Application resource properties.

Table 3-1 BEA Application Resource Properties

Property Name	Value
TUXDIR	The full pathname of the root directory of your BEA TUXEDO installation. This is a required field. Enclose the pathname in double quotes.
TUXCONFIG	The full pathname of the binary configuration file for the BEA TUXEDO application. This is a required field. Enclose the pathname in double quotes.
Application Password	The application password for this BEA Application resource. Required if `SECURITY=APP_PW`, `USER_AUTH`, `ACL`, or `MANDATORY_ACL` is specified in `TUXCONFIG`. Enclose the value in double quotes.
User Name	The login name of an authorized user. Required if `SECURITY=USER_AUTH`, `ACL`, or `MANDATORY_ACL` is specified in `TUXCONFIG`. Enclose the value in double quotes.
User Password	The password of the authorized user. Required if `SECURITY=USER_AUTH`, `ACL`, or `MANDATORY_ACL` is specified in `TUXCONFIG`. Enclose the value in double quotes.
Remark	This field is optional. Enclose any value in double quotes.
Server Selection	0=`LMID`, 1=Server Group, 2=`DBBL`
`LMID`	The logical machine identifier of machine being specified.
Server Group Name	The name of the server group being specified.
Administrative Servers	0=Not selected, 1=Selected
Application Servers	0=Not selected, 1=Selected
`tlisten`	0=Not selected, 1=Selected

Here is an example of command line specification of properties:

```
tuxdir="C:\tuxedo" tuxconfig="C:\users\apps\simpapp\tuxconfig"
App_Password="" Username="" User_Password="" Server_Selection="0"
```

```
LMID="mylmid" Server_Group_Name="" Admin_Servers="1" tlisten="0"
```

Error Codes

The following table lists error codes that may be returned from BEA Application resource properties specification.

Table 3-2 Error Codes

Error No.	Meaning
31103	The file specified in the TUXCONFIG field cannot be found.
31104	A valid configuration file must be entered in the TUXCONFIG field.
31105	A valid LMID contained in TUXCONFIG must be entered in the LMID field.
31106	A valid Server Group contained in TUXCONFIG must be entered in the Server Group field.
31107	At least one checkbox (Admin Servers/App Servers/Tlisten) must be selected if LMID valued.
31108	A valid directory containing your BEA Product must be entered in the TUXDIR field.
31109	A value must be entered in the Application Password field based on the security level of your BEA product configuration.
31110	A value must be entered in the User Name field based on the security level of your BEA product configuration.
31111	A value must be entered in the User Password field based on the security level of your BEA product configuration.
31112	A valid license for the BEA/MSCS integrated product was not found. Associated services will not function without proper licensing.
31115	A value must be entered in the TUXDIR field.
31116	A valid Server Selection must be entered (LMID=0, Server Group=1, DBBL=2).
31117	Server Selection DBBL must not be valued in conjunction with a SHM mode configuration.
31118	Admin Servers/App Servers/tlisten properties must be valued 0 or 1.

Table 3-2 Error Codes

Error No.	Meaning
31119	The maximum number of characters entered in the TUXDIR field must not exceed 78.
31120	The maximum number of characters entered in the TUXCONFIG field must not exceed 64.
31121	The maximum number of characters entered in the Application Password field must not exceed 8.
31122	The maximum number of characters entered in the User Name field must not exceed 30.
31123	The maximum number of characters entered in the User Password field must not exceed 8.
31124	The maximum number of characters entered in the Remark field must not exceed 128.
31125	The maximum number of characters entered in the LMID field must not exceed 30.
31126	The maximum number of characters entered in the Server Group field must not exceed 30.

How Your Choices Affect the Way the Application Works

The values you enter on the Parameters properties window have a direct bearing on how the BEA Application resource functions. Table 3-1 and Table 3-2 show how the values in the Server Selection Group Box affect the behavior of the resulting BEA Application resource resource.

Table 3-3 BEA Application Resource Behavior (MP Mode)

Selection	Operation		
	Offline→Online (on Node A)	**Online→Offline (on Node A)**	**Move Online on A to Online on B**
DBBL	`tmboot -M`	`tmshutdown -M`	Migrate DBBL

[1] `TLOG` move is handled automatically if the physical disk is a dependent resource to this group.

Table 3-3 BEA Application Resource Behavior (MP Mode)

Selection	Operation		
	Offline→Online (on Node A)	**Online→Offline (on Node A)**	**Move Online on A to Online on B**
LMID + Administrative Servers	`tmboot -B lmid`	`tmshutdown -B lmid`	Not allowed
LMID + Application Servers	`tmboot -l lmid`	`tmshutdown -l lmid`	Migrate all application servers groups on LMID [1]
LMID + Administrative Servers + Application Servers	`tmboot -l lmid -B lmid`	`tmshutdown -B lmid -l lmid`	Migrate all administrative servers and application server groups on LMID [1]
Server Group	`tmboot -g grpname`	`tmshutdown -g grpname`	Migrate *grpname* [2]

[1] `TLOG` move is handled automatically if the physical disk is a dependent resource to this group.

[2] `TLOG` entries moved automatically (as in Note 1). The difference is that *grpname* is the only transactional server group associated with the primary `LMID`.

Table 3-4 BEA Application Resource Behavior (SHM Mode)

Selection	Operation		
	Offline→Online (on Node A)	**Online→Offline (on Node A)**	**Move Online on A to Online on B**
LMIDs	`tmboot -y`	`tmshutdown -y`	1. `tmshutdown -y` on Node A 2. IP and physical disk failover to Node B 3. Set `PMID` to A on Node B 4. `tmboot -y` on Node B

How MSCS Operations Map to BEA TUXEDO Operations

The BEA MSCS Extensions product makes use of the Microsoft Cluster Administrator interface to create and control a BEA Application resource. The *Administrator's Guide for Microsoft Widows NT Cluster Server* describes the Cluster Administrator in greater detail than is appropriate for our purposes here. Operations available in the Cluster Administrator are listed in Cluster Technology Overview. In this section, we focus on three operations to emphasize how they map to ordinary BEA TUXEDO operations.

Offline to Online

To bring an MSCS resource (or group) online, highlight it and click File→Bring Online.

For a BEA Application resource, this is equivalent to the BEA TUXEDO `tmboot`(1) command. The `tmboot` command has a long list of options. As shown in the following table, the choices you make on the BEA Application resource Parameters properties window translate to some of the `tmboot` options.

You select	The system does
DBBL	`tmboot -M` This starts administrative servers on the `MASTER` machine. If `MODEL` is `MP`, a `DBBL` is started on the machine indicated by the `MASTER` parameter in the `RESOURCES` section. A `BBL` is started; a `BRIDGE` process is started if the `LAN` option and a `NETWORK` entry are specified in the configuration file.
LMID + Administrative Servers	`tmboot -B` *`lmid`* A `BBL` is started on a processor with logical name *`lmid`*.
LMID + Application Servers	`tmboot -l` *`lmid`* For each group whose associated `LMID` parameter is *`lmid`*, all `TMS` and gateway servers associated with the group are booted and all servers in the `SERVERS` section associated with those groups are started.
LMID + Administrative Servers + Application Servers	`tmboot -B` *`lmid`* `-l` *`lmid`* This combines the two previous operations.
Server Group + Administrative Servers + Application Servers	`tmboot -g` *`grpname`* All `TMS` and gateway servers for the group whose `SRVGRP` parameter is *`grpname`* are started followed by all servers in the `SERVERS` section associated with that group. `TMS` servers are started based on the `TMSNAME` and `TMSCOUNT` parameters for the group entry.

Online to Offline

To bring an MSCS resource (or group) offline, highlight it and click File→Bring Offline.

For a BEA Application resource, this is equivalent to the BEA TUXEDO `tmshutdown`(1) command. The `tmshutdown` command has a long list of options. As shown in the following table, the choices you make on the BEA Application resource Parameters properties window translate to some of the `tmshutdown` options.

You select	The system does
DBBL	`tmshutdown -M` This command shuts down administrative servers on the `MASTER` machine. The `BBL` is shut down on the `MASTER` machine, and the `BRIDGE` is shut down if the `LAN` option and a `NETWORK` entry are specified in the configuration file. If the `MODEL` is `MP`, the `DBBL` administrative server is shut down.
LMID + Administrative Servers	`tmshutdown -B` *`lmid`* The `BBL` is shut down on a processor with logical name *`lmid`*.
LMID + Application Servers	`tmshutdown -l` *`lmid`* For each group whose associated `LMID` parameter is *`lmid`*, all `TMS` and gateway servers associated with the group are shut down and all servers in the `SERVERS` section associated with those groups are shut down.
LMID + Administrative Servers + Application Servers	`tmshutdown -B` *`lmid`* `-l` *`lmid`* This combines the two previous operations.
Server Group + Administrative Servers + Application Servers	`tmshutdown -g` *`grpname`* All `TMS` and gateway servers for the group whose `SRVGRP` parameter is *`grpname`* are shut down followed by all servers in the `SERVERS` section associated with that group. `TMS` servers are shut down based on the `TMSNAME` and `TMSCOUNT` parameters for the group entry.

Bringing the BEA Application resource online or offline uses the dependency rules of the BEA TUXEDO system. For example, the DBBL offline operation does not bring the DBBL offline if there are multiple active LMIDs.

Move

MSCS executes a move of an active resource group as follows.

First, all the resources in the group are taken offline on the primary node and then brought online on the alternate node following the dependency rules in both cases. (See the preceding sections to learn the results of online and offline.)

In the case of the DBBL, changing the DBBL from online to offline does not necessarily mean that the DBBL has been immediately deactivated; the DBBL cannot be deactivated as long as there are active LMIDs. So taking the DBBL offline deactivates the associated BEA TUXEDO domain only when there are no active application and administrative servers and no active client processes except the administrative servers for the current master LMID.

Logging Errors

State changes caused by failure conditions during online, offline and move of BEA Application resource resources are logged in the BEA TUXEDO userlog (ULOG).

A Configuration Examples

This appendix contains two examples of configuration files:

- An Active/Hot Spare (that is, SHM mode) configuration
- An Active/Active (that is, MP mode) configuration

SHM Mode

Listing 3-1 An Active/Hot Spare Configuration

```
#Skeleton UBBCONFIG file for BEA TUXEDO SHM Mode application

#Replace the <bracketed> items with the appropriate values.

     *RESOURCES
      IPCKEY        <Replace with valid IPC Key greater than 32,768>

     #Example:     #IPCKEY         62345

     MASTER          simple
     UID             0
     GID             0
     MAXACCESSERS    5
     MAXSERVERS      5
     MAXSERVICES     10
     MODEL           SHM
     LDBAL           N
```

```
*MACHINES
DEFAULT:

                APPDIR="<Replace with the current pathname>"

                TUXCONFIG="<Replace with TUXCONFIG Pathname>"

                TUXDIR="<Root directory of TUXEDO (not \)>"

#Example:
#               APPDIR="C:\usr\me\shmdir"
#               TUXCONFIG="C:\usr\me\shmdir\tuxconfig"
#               TUXDIR="C:\usr\tuxedo"

<Machine-name>  LMID=Node_A

#Example:
#NodeA                LMID=LMID_A

*GROUPS
A                     LMID=LMID_A     GRPNO=1 OPENINFO=NONE

*SERVERS
DEFAULT:                      CLOPT="-A"

serverA               SRVGRP=A SRVID=1

*SERVICES
srvAservices
```

MP Mode

Listing 3-2 An Active/Active Configuration

```
# The following configuration file defines a 2-site
# configuration with two machine types.  Data dependent
# routing is used.

*RESOURCES

IPCKEY  80952   # key for well known address
DOMAINID        My_Domain_Name
UID     0       #
GID     0       # group id for ipc structures
PERM    0660    # permissions for ipc access

MAXSERVERS      20      # at most 20 simultaneous servers
MAXSERVICES     40      # offering at most 40 services
MAXGTT  20      # at most 20 simultaneous global transactions
MASTER  SITE1
SCANUNIT        10
SANITYSCAN      12
BBLQUERY        180
BLOCKTIME       30
DBBLWAIT        6
NOTIFY          DIPIN
OPTIONS         LAN,MIGRATE
SECURITY        USER_AUTH
AUTHSVC        AUTHSVC
MODEL   MP      # a multiprocessor based bulletin board
LDBAL   Y       # perform load balancing

#

*MACHINES
NodeA   LMID=LMID_A TUXDIR="C:\usr4\tuxbin"
        MAXACCESSERS=25
        APPDIR="C:\usr2\apps\mpdir"
        ENVFILE="C:\usr2\apps\mpdir\ENVFILE"
        TLOGDEVICE="C:\usr2\apps\mpdir\TLOG" TLOGNAME=TLOG
        TUXCONFIG="C:\usr2\apps\mpdir\tuxconfig"
        ULOGPFX="C:\usr2\apps\mpdir\ULOG"
        SPINCOUNT=5

NodeB LMID=SITE2 TUXDIR="C:\usr5\tuxbin"
        MAXACCESSERS=100
```

```
        MAXWSCLIENTS=50
        APPDIR="C:\usr4\apps\mpdir"
        ENVFILE="C:\usr4\apps\mpdir\ENVFILE"
        TLOGDEVICE="C:\usr4\apps\mpdir\TLOG" TLOGNAME=TLOG
        TUXCONFIG="C:\usr4\apps\mpdir\tuxconfig"
        ULOGPFX="C:\usr4\apps\mpdir\ULOG"

#

*GROUPS
DEFAULT:        TMSNAME=TMS_SQL TMSCOUNT=2

A             LMID=LMID_A      GRPNO=1


OPENINFO="TUXEDO/SQL:C”\usr2\apps\bank\mp1;mympdb;readwrite"

B             LMID=LMID_B      GRPNO=2

OPENINFO="TUXEDO/SQL:C:\usr4\apps\bank\mp2;mympdb;readwrite"
DEFAULT:    AUTHGRP LMID=LMID_A      GRPNO=3

#

*NETWORK
NodeA   NADDR="mach1.80952" BRIDGE="C:\dev\starlan"
        NLSADDR="mach1.serve"

#

NodeB   NADDR="mach386.80952" BRIDGE="C:\dev\starlan"
        NLSADDR="mach386.serve"

*SERVERS

#

DEFAULT: RESTART=Y MAXGEN=5 REPLYQ=Y CLOPT="-A"

TLR     SRVGRP=A   SRVID=1 RQADDR=tlr1
        CLOPT="-A -- -T 100"

TLR     SRVGRP=A   SRVID=2 RQADDR=tlr1
        CLOPT="-A -- -T 200"

TLR     SRVGRP=B   SRVID=3 RQADDR=tlr2
        CLOPT="-A -- -T 600"

TLR     SRVGRP=B   SRVID=4 RQADDR=tlr2
        CLOPT="-A -- -T 700"
```

```
XFER    SRVGRP=A   SRVID=5
XFER    SRVGRP=B   SRVID=6
ACCT    SRVGRP=A   SRVID=7
ACCT    SRVGRP=B   SRVID=8
BAL     SRVGRP=A   SRVID=9
BAL     SRVGRP=B   SRVID=10
BTADD   SRVGRP=A   SRVID=11
BTADD   SRVGRP=B   SRVID=12
AUTHSVR SRVGRP=AUTHGRP  SRVID=20
#

*SERVICES

DEFAULT:        LOAD=50 AUTOTRAN=N

SVCA      PRIO=50 ROUTING=ACCOUNT_ID
SVCB      PRIO=50 ROUTING=ACCOUNT_ID
SVCC      PRIO=50 ROUTING=ACCOUNT_ID
SVCD      PRIO=50 ROUTING=ACCOUNT_ID
SVCE      PRIO=40 ROUTING=ACCOUNT_ID
SVCF      PRIO=40 ROUTING=BRANCH_ID
SVCG      PRIO=20 ROUTING=BRANCH_ID
SVCH      PRIO=20 ROUTING=BRANCH_ID
SVCI      PRIO=30 ROUTING=b_id
SVCJ      PRIO=30 ROUTING=b_id
SVCK      PRIO=30 ROUTING=b_id
SVCL      PRIO=30 ROUTING=b_id SVCTIMEOUT=300
#

*ROUTING

ACCOUNT_ID      FIELD=ACCOUNT_ID        BUFTYPE="FML"
RANGES="MIN-9999:*,10000-59999:BANKB1,60000-109999:BANKB2,*:*"

BRANCH_ID       FIELD=BRANCH_ID         BUFTYPE="FML"
RANGES="MIN - 0:*,1-5:BANKB1,6-10:BANKB2,*:*"

b_id    FIELD=b_id      BUFTYPE="VIEW:aud"
RANGES="MIN - 0:*,1-5:BANKB1,6-10:BANKB2,*:*"
```

Glossary

The following terms are used in this guide:

Cluster Administrator

This is not a person, but rather a GUI based, cluster-aware utility for configuring clusters and their components; that is to say, resources or resource groups.

Cluster Administrator Extension API

MSCS facility for extending a cluster administrator's GUI interface; for example, the Cluster Administrator Extension API can be used to create wizards, property sheets, context menus with controls specific to one particular resource type.

Cluster Administrator Extension DLL

A Windows NT dynamic link library implementing a Cluster Administrator Extension API for a particular resource type. It is implemented as a COM in-proc server, utilizing ActiveX Template libraries.

Cluster API

A pool of API specifications for managing and communicating with clusters and their components. It includes specifications for such things as cluster configuration, node status, cluster event subscription and notification. It is implemented by core NT cluster software.

Cluster-aware application

An application that uses the Cluster API. An example is the Cluster Administrator utility Cluster-enabled application

An application whose resources are monitored by MSCS. Only a cluster-enabled application can be failedover/failedback.

Cluster Service

A core component of MSCS, implemented as a Windows NT service. It performs services like coordinating event notification, facilitating communication between cluster components, handling failover operations, and managing configuration.

HA

High Availability.

High Availability

An essential characteristic for mission-critical applications.

PMID

Physical Machine Identifier, that is, the name the machine is known by to the system.

Resource

Any physical or logical entity that can be used by clients of a cluster. Example of resources are disks, network names, network addresses, databases, web sites, print queues, network file shares, and application programs. A resource is housed on a single cluster node at any point in time.

Resources can be in various states: `offline`, `online`, `failed`, `online pending`, `offline pending`.

Resource API

MSCS programming interface for controlling and monitoring cluster resources. The following function calls make up the Resource API: `Close`, `IsAlive`, `LookAlive`, `Offline`, `Online`, `Open`, `Startup`, `Terminate`, `Arbitrate`, `Release`, `LogEvent` and `SetResourceStatus`.

Resource DLL

A Windows NT dynamic link library implementing the Resource API for a particular Resource Type.

Resource Group

A collection of MSCS resources that are related in some way and/or are dependent on each other. For example, a database resource is likely to depend on the disk resource that contains the database files. These two resources, the disk and the database files, belong in a group that offers the database service.

A resource upon which another resource depends must be brought online before the dependent resource and taken offline after it. For example, to bring the database service online, the disk must be online before the database files. Similarly; to take the database service offline, the files must be taken offline ahead of the disk. Note that the database files must be managed as a separate resource even though they reside on the disk.

The Resource Group is the unit of failover/failback in MSCS.

Resource Monitor

A process that tracks the state of cluster resources and notifies the Cluster Service of state transitions. Each node in a cluster can have one or more Resource Monitors assigned to one or more resources.

Resource Type

MSCS resources are categorized by type. Resource types include physical hardware such as SCSI disks and logical items such as TCP/IP addresses. Each resource type has a set of properties that apply to all resources of that type. For example, the properties of a physical disk include the name of its resource DLL and two polling values. Other resource types define other properties specific to their types.

(Windows NT) Cluster

A configuration of two independent Windows NT Server computer systems, which appear to network clients and administrators to be a single, highly available Windows NT Server system.

Index