Domain Guide

JEUS 9

Copyright

Company Information

TmaxSoft Co., Ltd.

TmaxSoft Tower 10F, 45, Jeongjail-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, South Korea

Website: https://www.tmaxsoft.com/en/

Restricted Rights Legend

This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

Trademarks

Open Source Software Notice

Document History

Product Version	Guide Version	Date	Remarks
JEUS 9	3.1.2	2025-03-24	-
JEUS 9	3.1.1	2024-12-24	-

Contents

1. Introduction to Domain

This chapter describes the concepts and architecture of a domain.

1.1. Basic Concepts

A domain is the basic unit of management. It consists of multiple servers and clusters that are divided according to tasks.

Services, server locations, and security policies can be different for each domain. Each domain has a separate configuration file and does not share settings with other domains. Domains should be created with the knowledge that the security configuration, applications, and resources are shared within each domain.

Consider the following when creating a domain.

- Roles of servers in the domain

  Configure servers with similar roles in the same domain.

- Physical location of servers in the domain

  The physical location of the servers is important for providing stable services.

- Load on servers in the domain

  Use an appropriate number of servers based on the required workload.

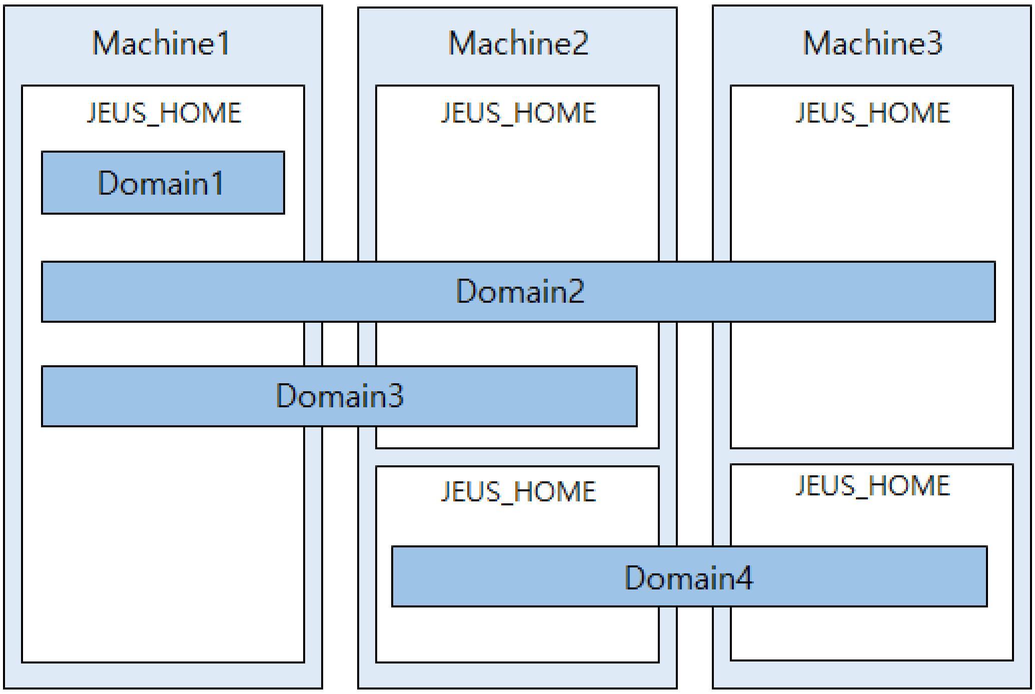
1.2. Relationship between JEUS and Domain

Each machine requires an instance of JEUS (installed in JEUS_HOME), which can contain one or more domains. A domain can be created in one instance or across multiple instances of JEUS.

Usually, a single JEUS instance is installed on each server. However, it is possible for a machine to have multiple instances of JEUS installed.

For instance, a domain for personnel management and another for order management can be installed on three servers according to the applications' requirements. In the following example, Domain1 for personnel management service is installed on a single machine, Machine1. The domain for order management is installed on all three machines so that its services are still available if one of the servers fails.

The following diagram shows the relationship between JEUS and domains.



Relationship between JEUS and Domains

1.3. Components

A domain consists of Master Server (MASTER), Managed Servers (MS), and clusters.

- **Master Server (MASTER)**

  A domain must have a special server called Master Server (MASTER). MASTER configures servers, centrally manages all the applications and resources in the domain, and communicates with the administrator tool (jeusadmin) that are used to control and monitor servers.

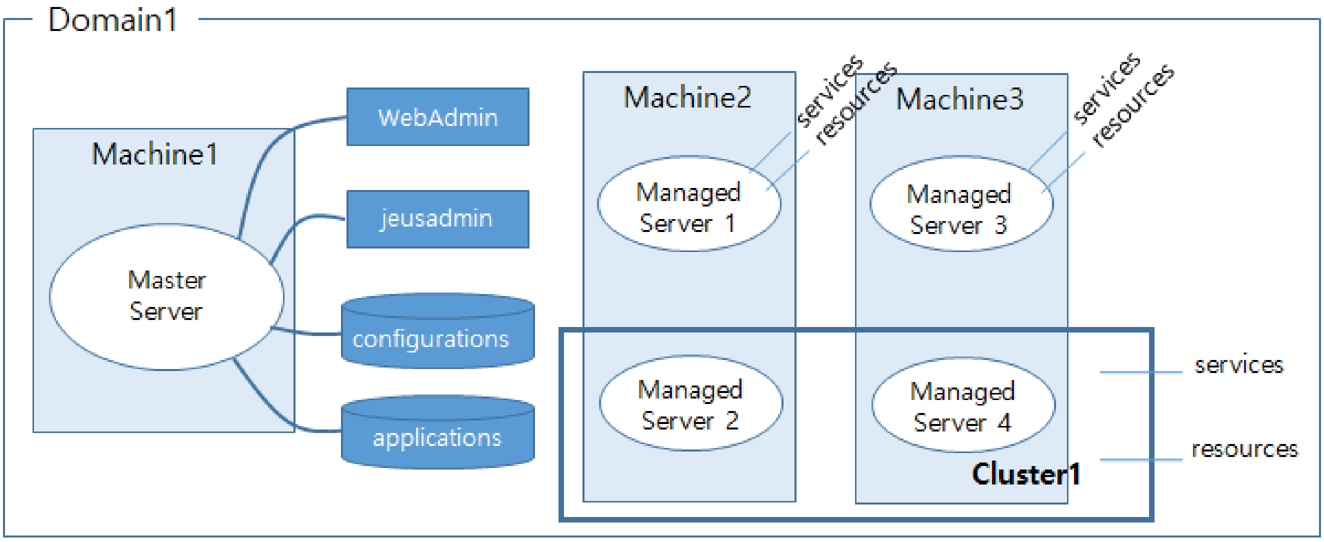
- **Managed Server (MS)**

  A domain consists of one or more Managed Servers (MSs), which are responsible for providing services. A domain can contain a cluster with multiple MSs that provide the same or different services.

- **Cluster**

  A domain can contain zero or more clusters and other servers that do not belong to a cluster.

  A cluster, which is a group of servers that provide the same services, are used to support scalability and reliability. Servers that are in the same cluster use the same resources and run the same applications. Refer to JEUS Clustering for detailed information about a cluster.



Domain Components

MASTER is located on Machine1. MASTER communicates with the administrator tool to receive and process user commands.

Machine 2 and Machine 3 each contain two MSs. Managed Server 2 and Managed Server 4 in Cluster 1 share applications and resources and provide the same services, but Managed Server 1 and Managed Server 3, which are not in Cluster 1, do not.

> A basic domain must contain one server that acts as both MASTER and MS. This domain configuration should only be used for development and testing. In the production environment, separate servers should be used for management and services and MASTER should only be used to manage MSs.

1.3.1. Master Server (MASTER)

Master Server must exist in every domain. It centrally manages and monitors MSs.

The following are the major functions of MASTER.

- **Managing domain configuration**
  - Even if each MS in the domain is on a different server, they can use the same configuration.

    MASTER must start first in order for MS to get the configuration from MASTER to start. If MS starts while MASTER is not running, its configuration will be synchronized when MASTER starts.
  - Configuration changes are applied to all MSs in the domain.

    MASTER verifies any configuration changes and synchronizes them by applying the changes to other servers.

  Refer to Changing Domain Settings for detailed information about managing the configuration for MASTER.

- **Managing all applications in the domain**

    - Manages the status of all the applications in the domain.

      MASTER manages and synchronizes the status of all the applications in the domain, ensuring applications in MSes are maintained in the same states as those in MASTER.

    - Manages all the application files in the domain.

      MASTER synchronizes the files when an MS runs and when applications are deployed to the MS. If application files are not synchronized because MASTER was not running, they are synchronized when MASTER starts.

    - Controls the deployment of application services to servers or clusters in the domain.

      To run applications in the domain, applications should be installed on MASTER or exist in an application store (repository).

      To run applications on an MS, transfer the service objects to MASTER and deploy the applications. All commands associated with application queries and controls as well as deployment commands are performed by MASTER. If MASTER is not running, domain commands cannot be executed.

      If an MS is in a cluster, applications can only be deployed to the target cluster. If a server is added to the cluster, MASTER guarantees that applications can also be serviced from the added server.

  Refer to Application Managment in JEUS Applications & Deployment Guide for detailed information about application management performed by MASTER.

- **Monitoring and controlling the entire domain**

  The administrator tool jeusadmin can be used to monitor and control all servers, services, applications, and resources.

1.3.2. Managed Server(MS)

Managed Servers contain the necessary resources and deployed applications.

MS operation is managed by MASTER in the following ways.

- **Configuration and application synchronization**

  MASTER manages configurations and applications of the entire domain. Hence, any configurations or applications that are deployed to an MS should be deployed using MASTER. Configurations and application files received from MASTER are stored in the local cache as read-only. Upon connecting with MASTER, only the modified files in MASTER are synchronized with those in the cache. Any modifications to individual MS are not applied. All modifications are applied through MASTER.

  - Configuration synchronization

    Configuration synchronization occurs when MS starts, when MS reconnects to MASTER, or when the domain configuration is modified.

  - Application synchronization

    The status of an application is also synchronized with the application status in MASTER. Application synchronization is performed whenever MS starts or reconnects to MASTER. Refer to Application Synchronization in JEUS Deployment Guide for detailed information about application synchronization.

- **Cluster composition**

  To balance the load and handle errors, a cluster can be created with multiple MSs. It is recommended that all servers in the cluster contain the same resources and applications. For detailed information about clusters, refer to JEUS Clustering.

1.4. Production and Development Modes

The domain operation mode can be set to either of the following modes. The mode must be manually set in the configuration file (domain.xml).

- Production mode

  This mode is suitable for the actual production environment. The auto-reload or hot-swap functions of web applications are not provided. Set the production-mode to true in the domain.xml file.

- Development mode

  If the production-mode is set to false in the domain.xml file, the domain runs in the development mode.

1. It is recommended to use the development mode during the development stage where applications are changed frequently, and to use the production mode in the actual production environment.
2. All running servers in the domain must be restarted after modifying the configuration.

1.5. Domain Restrictions

Consider the following when creating a domain.

- Each domain must have one MASTER.
- Configurations and resources cannot be shared between domains.
- All servers in the domain must use the same version.
- The name of each server and cluster in the domain must be unique.
- The name of each domain on a JEUS instance must be unique.

2. Creating a Domain

This chapter describes the requirements for creating a domain, how to create a domain using a tool provided by JEUS (jeusadmin), and the directory structure of a domain.

2.1. Creating a Domain

A domain directory is created when a domain is created. Scripts and basic configuration files that start and stop servers in the domain are created in this directory. The parent directory for each domain is DOMAIN_HOME and its path is as follows:

```
JEUS_HOME/domains/<domain_name>
```

When a domain is created, it only contains MASTER. Managed serves can be added using jeusadmin. After an MS is added and started, it receives the domain information from MASTER. In other words, MASTER creates the domain directory when the domain is created, and managed servers create the domain directory through MASTER when they start.

A domain can be created using Ant task or jeusadmin. The following are the requirements.

- Network configuration

  JEUS requires an IP address and port number because it shares domain server status using multicast. If servers exist outside of the local network and are not able to share information via multicast, TCP is used instead. This is called virtual multicasting in JEUS, and it can be configured when a domain is created.

  Since server information is shared using the default listener IP address and port number on the server, all servers in the domain that exist outside of the local network must specify a listener IP address and port number. They must also set the option to disable multicasting.

- Default MASTER configuration

  Each domain must contain a MASTER. A default configuration is used and can be further configured by the user. Dynamic configuration items can be changed at any time, but static items must be configured when the domain is created. Examples of static configuration items include the default listener IP address, port number, and JVM configuration.

2.1.1. Creating a Default Domain

This section describes how to create the default domain. When the default domain is created, Virtual Multicast is disabled.

The following are the command attributes for Ant task and jeusadmin.

Attribute	Default Value
MASTER name	adminServer
Listen IP address	0.0.0.0 / 0:0:0:0:0:0:0:0
Listen port number	9736
user name	administrator
Multicast IP address	230.30.1.1 / FF01:0:0:0:0:0:0:1
Multicast port number	3030

- **Creating a domain with Ant**

  The default configuration values when using Ant are defined in JEUS_HOME/setup/domain-config-template.properties. The following is an example of configuring the attributes. The default value is used if an attribute is not configured.

  ```
  JEUS_HOME/setup$ ant create-domain -Ddomain=jeus_domain
  -Dservername=adminServer
  -Djeus.address=0.0.0.0
  -Djeus.port=9736
  -Dtransport.address=230.30.1.1
  -Dtransport.port=3030
  ```

- **Creating a domain with jeusadmin**

  The default configuration values when using jeusadmin are defined in JEUS_HOME/setup/domain-config-template.properties. If jeus.password is set in domain-config-template.properties, use the configured value. Otherwise, enter the password as in the following example.

  ```
  offline>create-domain -domain jeus_domain -mastername adminServer -taddress 230.30.1.1 -tport
  3030
  Enter the password for [administrator]: **********
  Confirm the password: **********
  Do you want to encrypt the password? (y/n): y
  =================================================================================
  The domain [jeus_domain] was created successfully.
  +-------------------------------+-----------------------------------------------+
  |         Property              |                        Value                  |
  +-------------------------------+-----------------------------------------------+
  | Domain name                   | jeus_domain                                   |
  | Master Name                   | adminServer                                   |
  | Master Base Listen Port       | 9736                                          |
  | Master Base Listen Address    | 0.0.0.0                                       |
  | HTTP Port                     | 8088                                          |
  | User Name                     | administrator                                 |
  | Node Name                     | node1                                         |
  | Password                      | {base}amV1cw==                                |
  | JVM Option                    | -Xmx256m -XX:MaxMetaspaceSize=128m            |
  | Production Mode               | true                                          |
  | Transport Type                | HYBRID                                        |
  ```

```
+--------------------------------+----------------------------------------------+
================================================================================
```

2.1.2. Creating a Domain that Uses Virtual Multicast

This section describes how to check if the environment supports the IP Multicast functionality, and how to create a domain that uses the virtual multicasting when the environment does not support the IP multitasking or the domain goes beyond the subnet range.

To check if the environment supports IP multitasking, execute the two mcastSender and mcastReceiver test scripts. The two scripts are in JEUS_HOME/bin. Execute each one on a different machine as in the following.

```
JEUS_HOME/bin$ mcastReceiver -addr 224.0.0.1 -port 3030
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:2a37:37ff:fe1e:fc4%5
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /192.168.0.26
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:3e07:54ff:fe6f:95e%4
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /61.77.153.207
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:0:0:0:1%1
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /0:0:0:0:0:0:0:1
Socket == 0.0.0.0/0.0.0.0 : 3030, bound to /127.0.0.1
```

```
JEUS_HOME/bin$ mcastSender -addr 224.0.0.1 -port 3030
1 == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:2a37:37ff:fe1e:fc4%5
2 == 0.0.0.0/0.0.0.0 : 3030, bound to /192.168.0.26
3 == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:3e07:54ff:fe6f:95e%4
4 == 0.0.0.0/0.0.0.0 : 3030, bound to /61.77.153.207
5 == 0.0.0.0/0.0.0.0 : 3030, bound to /fe80:0:0:0:0:0:0:1%1
6 == 0.0.0.0/0.0.0.0 : 3030, bound to /0:0:0:0:0:0:0:1
7 == 0.0.0.0/0.0.0.0 : 3030, bound to /127.0.0.1
```

This test attempts to send and receive IP multicast packets through all network interfaces. The addresses at the top are for all network interfaces.

At the prompt that appears on the Sender side, enter a message that is sent to the Receiver. If the Receiver cannot receive the message, it means that IP multicasting is not working properly. In such a case, set the following virtual multicast configuration. The following is the execution of a test message for the mcastSender example above.

```
> test
>
```

```
test [sender = 192.168.0.26 : 5555], [bind_addr=/127.0.0.1]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:3e07:54ff:fe6f:95e%4]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/61.77.153.207]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/192.168.0.26]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
test [sender = 192.168.0.26 : 5555], [bind_addr=/fe80:0:0:0:2a37:37ff:fe1e:fc4%5]
```

Since the same message is sent/received through all network interfaces of the machine, duplicate messages can be transmitted. To quit the test, enter 'quit' or 'exit' at the Sender’s prompt. At the Receiver’s end, press <Ctrl> + C to quit.

To see help, enter '-help' as a parameter or do not include a parameter.

The following describes how to configure a virtual multicast domain. The default attribute values are the same as those in Creating a Default Domain. The virtual multicast has two types: MESH and Tree.

- **Creating a domain with Ant (MESH type)**

```
JEUS_HOME/setup$ ant create-domain
                    -Ddomain=jeus_domain
                    -Djeus.address=192.168.34.1
                    -Dtransport.type=MESH
```

- **Creating a domain with jeusadmin (MESH type)**

```
offline>create-domain -domain jeus_domain -baseaddress 192.168.34.1 -ttype MESH
Enter the password for [administrator]: **********
Confirm the password: **********
Do you want to encrypt the password? (y/n): y
=================================================================================
The domain [jeus_domain] was created successfully.
+--------------------------------+-----------------------------------------------+
|          Property              |                      Value                    |
+--------------------------------+-----------------------------------------------+
| Domain name                    | jeus_domain                                   |
| Master Name                    | adminServer                                   |
| Master Base Listen Port       | 9736                                          |
| Master Base Listen Address    | 192.168.34.1                                  |
| HTTP Port                     | 8088                                          |
| User Name                     | administrator                                 |
| Node Name                     | node1                                         |
| Password                      | {base}amV1cw==                                |
| JVM Option                    | -Xmx256m -XX:MaxMetaspaceSize=128m            |
| Production Mode               | true                                          |
| Transport Type                | MESH                                          |
+-------------------------------+-----------------------------------------------+
=================================================================================
```

- **Creating a domain with jeusadmin (TREE type)**

```
offline>create-domain -domain jeus_domain -baseaddress 192.168.34.1 -ttype TREE -taddress
192.168.34.1 -tport 9736
Enter the password for [administrator]: **********
Confirm the password: **********
Do you want to encrypt the password? (y/n): y
=================================================================================
The domain [jeus_domain] was created successfully.
+-------------------------------+-----------------------------------------------+
|         Property              |                        Value                  |
+-------------------------------+-----------------------------------------------+
| Domain name                   | jeus_domain                                   |
| Master Name                   | adminServer                                   |
| Master Base Listen Port       | 9736                                          |
| Master Base Listen Address    | 192.168.34.1                                  |
| HTTP Port                     | 8088                                          |
| User Name                     | administrator                                 |
| Node Name                     | node1                                         |
| Password                      | {base}amV1cw==                                |
| JVM Option                    | -Xmx256m -XX:MaxMetaspaceSize=128m            |
| Production Mode               | true                                          |
| Transport Type                | TREE                                          |
| Transport address             | 192.168.34.1                                  |
| Transport port                | 9736                                          |
+-------------------------------+-----------------------------------------------+
=================================================================================
```

2.2. Configuring a Domain

This section describes how to change the composition of a domain by adding servers and clusters to the domain created in Creating a Domain.

The MASTER must be started to add servers and clusters which changes the configuration. Refer to Changing Domain Settings for detailed information about changing the configuration.

```
DOMAIN_HOME/bin$ startMasterServer -u jeus -p jeus
```

2.2.1. Using the Console Tool

The following are the steps for adding a server and a cluster.

1. Add Managed Servers (MS)
2. Add a cluster

Add Managed Servers (MS)

Managed Server configuration cannot be changed dynamically.

Similar to the default configuration of MASTER, when the domain described in Creating a Domain is created, default configuration set when adding an MS is not applied dynamically. Choose a unique IP address and port number that do not conflict with other servers, and configure the JVM options. If a domain does not exist on the local network, specify an IP address to use instead of the default value of 0:0:0:0. Since server names in a domain must be unique, verify that there are no conflicts. Since the servers in the example are in the same cluster, settings that exist at the cluster level do not need to be configured separately for each server.

The following are the steps for adding an MS by using jeusadmin.

Add an MS, server1, using the **add-server** command. Add server2 in the same way. For detailed instructions about using the add-server command, refer to "add-server" in JEUS Reference Guide.

```
[MASTER]domain1.adminServer>add-server server1 -node node1 -addr 192.168.34.1 -port 9836
Successfully performed the ADD operation for server (server1).
 NOTICE : base-addr [192.168.34.1] base-port [9836] http-port [8088]
Check the results using "list-servers or add-server"
```

Add Clusters

When adding a cluster, load balancing, failover, and the servers in the cluster must all be configured. Refer to Configuring a Cluster for detailed information about configuring a cluster.

The following are the steps for adding a cluster.

1. Add a cluster 'cluster1' that consists of server1 and server2 using the **add-cluster** command. Refer to add-cluster for detailed instructions about using the add-cluster command.

   ```
   [MASTER]domain1.adminServer> add-cluster cluster1 -servers server1,server2
   Successfully performed the ADD operation for cluster (cluster1).
   Check the results using "list-clusters or add-cluster"
   ```

2. After adding all items up to the cluster, use the **serverinfo** command to verify the result.

   ```
   [MASTER]domain1.adminServer>serverinfo
   Information about Domain (domain1)
   ```

```
===============================================================================================
+--------+---------+-----+-------+-----+----------------+---------+-------------+-------------+
| Server |  Status | Node|  PID  | Clus|  Latest Start  | Need to | Listen Ports| Running     |
|        |         |Name |       | ter |Time / Shutdown | Restart |             | Engines     |
|        |         |     |       |     |     Time       |         |             |             |
+--------+---------+-----+-------+-----+----------------+---------+-------------+-------------+
| adminSe| RUNNING | N/A | 5360  | N/A | 2016-08-23     | false   | base-192.168| jms,        |
|rver(*) |(00:01:11|     |       |     |(Tue) PM        |         |.34.3:9736   | ejb, web    |
|        |)        |     |       |     |03:53:28 KST    |         | http-server-|             |
|        |         |     |       |     |                |         |0.0.0.0:8088 |             |
|        |         |     |       |     |                |         |jms-internal-|             |
|        |         |     |       |     |                |         |0.0.0.0:9741 |             |
+--------+---------+-----+-------+-----+----------------+---------+-------------+-------------+
| server1| RUNNING |node1| 6424  | clus| 2016-08-23     | false   | base-192.168| jms,        |
|        |(00:00:03|     |       |ter1 |(Tue) PM        |         |.34.1:9836   | ejb, web    |
|        |)        |     |       |     |03:54:37 KST    |         |             |             |
+--------+---------+-----+-------+-----+----------------+---------+-------------+-------------+
| server2| RUNNING |node2| 3012  | clus| 2016-08-23     | false   | base-192.168| jms,        |
|        |(00:00:03|     |       |ter1 |(Tue) PM        |         |.34.2:9936   | ejb, web    |
|        |)        |     |       |     |03:54:36 KST    |         |             |             |
+--------+---------+-----+-------+-----+----------------+---------+-------------+-------------+
===============================================================================================
```

2.3. Directory Structure

When JEUS is installed, JEUS_HOME directory and jeus_domain, which is the sub directory of JEUS_HOME, are created.

This section describes the directory structure with the assumption that the path of the user home is '/home/user1' and the domain is called 'domain1'.

2.3.1. JEUS_HOME Directory

The JEUS directory structure is as follows. JEUS_HOME is the top-level directory of JEUS. The actual directory name and location can be configured during the installation.

```
/home/jeus/JEUS_HOME
```

The following is the directory structure of JEUS_HOME.

```
{JEUS_HOME}
   |--bin
   |    |--[01]startMasterServer
   |    |--[01]startManagedServer
   |    |--[01]stopserver
   |    |--[01]jeusadmin
   |--derby
   |--docs
   |--domains
   |    |--domain1
```

```
    |--lib
    |--license
    |--setup
    |--templates

* Legend
- [01]: binary or executable file
- [X] : XML document
- [J] : JAR file
- [T] : Text file
- [C] : Class file
- [V] : java source file
- [DD] : deployment descriptor
```

bin

Contains executable files, including startMasterServer, startManagedServer, and stopServer, which are scripts that start and stop servers. It also contains jeusadmin, JEUS console tool.

derby

Contains Apache Derby so that sample applications and tests can be easily used.

docs

Contains Javadoc documents for the API provided by JEUS.

domains

Contains a DOMAIN_HOME subdirectory for each domain.

lib

Contains the libraries that are needed to run JEUS. A users does not need to access other directories except the shared directory.

Subdirectory	Description
shared	Contains libraries that are used by applications. To use the libraries, add the information about the libraries to libraries.xml. Also, the reference information of the libraries must be configured in the JEUS Deployment Descriptor (DD) of the application that will reference them. Refer to Shared Libraries for detailed information about shared libraries.

license

Contains JEUS license files that are required to run JEUS.

setup

Contains the files that are required for implementing JEUS environment.

templates

Contains template files for various configurations and environments.

2.3.2. Domain Directory

JEUS_HOME/domains/domain1 is the DOMAIN_HOME directory for domain1.

The following is the directory structure of domain1.

```
domain1
   |--.applications
   |--.deploymentplans
   |--.libraries
   |--bin
   |--config
   |--lib
   |   |--application
   |--servers
```

.applications

Contains application files managed by the domain. This directory has restricted access. Files can only be added or deleted using the install-application or uninstall-application commands. For details about each command, refer to install-application, uninstall-application.

.deploymentplans

Contains deployment plans managed by the domain. This directory has restricted access. Files can only be added or deleted using the install-deployment-plan and uninstall-deployment-plan commands. For details about each command, refer to install-deployment-plan and uninstall-application.

.libraries

Contains the library files installed in the domain. User access is limited to the directory that JEUS uses. The install-library and uninstall-library can be used to add and delete libraries. For details about each command, refer to install-library and uninstall-library.

bin

Contains the MASTER and MS start and stop scripts.

These have the same function as the startMasterServer, startManagedServer, and stopServer scripts in 'JEUS_HOME/bin' except that they do not need to specify the name of the domain.

config

Contains the backup copies of the domain.xml file. Refer to Changing Domain Settings for detailed information about configuring the domain.

- **security**

Subdirectory	Description
SYSTEM_DOMAIN	Directory that contains the accounts.xml and policies.xml domain security files. These files contain settings that can be changed dynamically using jeusadmin. Refer to "Configuring the Security System Domain" in JEUS Security Guide for a detailed description of domain security settings.
security.key	File that contains the encryption keys that is created when JEUS_HOME/bin/encryption is created. Refer to "Configuring Password Security" in JEUS Security Guide for a detailed description of the security.key file.
policy	Java permissions configuration file. This is unrelated to the JEUS security system; it is used by the Java SE Security Manager.

- **servlet**

Subdirectory	Description
webcommon.xml	The common configuration file which applies to all web modules on the web engine of a server in the domain. For details about this configuration, refer to "Directory Structure" in JEUS Web Engine Guide.

lib/application

Contains application libraries that are applied to the entire domain.

If there is a duplicate copy of the library in the SERVER_HOME directory, the copy in the SERVER_HOME/lib/application directory has priority, and a warning message is also displayed. For a detailed description of the lib/application directory, refer to "lib/application Directory" in JEUS Applications & Deployment Guide.

servers

The SERVER_HOME directory is a subdirectory that uses the server name. For a detailed description of the SERVER_HOME directory, refer to "Server Directory Structure" in JEUS Server Guide.

3. Changing Domain Settings

This chapter provides background information on managing domain settings and explains how to change the settings.

3.1. Overview

The domain configuration file is in the config directory (DOMAIN_HOME/config) of each domain. The configuration file, domain.xml, contains the configuration for servers and clusters as well as resources and services in the domain. The file also contains information about the deployed applications.

Dynamically applying settings refers to making changes in real-time to a running server without restarting the server. Since changes are verified by MASTER and then distributed to the MSs, the changes that are made manually to the XML file are not applied dynamically. Only the changes made by using jeusadmin are applied dynamically.

To change a dynamic configuration item, the user must have permission for the resource name jeus.domain.<*domain-name*> and the DynamicConfiguration resource action. When changing security configuration, users cannot delete their own permission to edit the configuration. This is to allow the users to be able to always undo their actions.

For information about how to give resource permissions to a user, refer to "Configuring Security System Policies" in JEUS Security Guide.

3.2. Modifying the Configuration

The configuration that is required to operate a server is inside a file in the domain to which the server belongs.

The configuration file is managed by MASTER, distributed to each server, and stored in a local read-only cache for better performance. JEUS supports dynamic configuration through MASTER.

3.3. Applying the Modified Configuration

Once the configuration has been modified, the changes are applied to the servers. Some settings can be applied dynamically, and some settings require restarting the server.

3.3.1. Dynamic Settings

Settings that can be applied dynamically are applied without restarting servers.

Using the Console Tool

In jeusadmin, dynamically applied command parameters have the word **[Dynamic]** in their description.

Commands used for jeusadmin automatically perform **Lock**, **Save**, and **Apply Changes**, and the changes for items that are labeled as **[Dynamic]** are applied immediately.

```
[MASTER]domain1.adminServer>help modify-server
...
OPTIONS
[-removelogdir,--removeLogHomeDirectory]
   remove the log directory which has all the log files created by this
   server

[-node,--nodeName <node-name>]
   name of the node this server is located

<server-name>
   the name of server you want to modify configuration of

[-jvm,--jvmOptions <jvm-options>]
   jvm configurations applied to this server jvm

[-logdir,--logHomeDirectory <server-log-home-directory>]
   the log directory which has all the log files created by this server

[-a,--actionOnResourceLeak <action-on-resource-leak>]
   strategy when resource leak is detected. it must be one of
   AutoClose, NoAction, Warning

[-l,--logStdoutToRawFormat <print-stdout-to-raw-format>]
[Dynamic] whether stdout is printed in row format or in JEUS Logger
   format.

[-m,--mejb <enable-MEJB>]
[Dynamic] whether using MEJB or not .true, false

[-c,--classFtp <enable-class-ftp-service>]
[Dynamic] whether using class ftp service or not .true, false

[-f,--forceLock]
    Acquires the configuration lock forcibly.

[-detail]
    show detail results
```

Refer to "jeusadmin" in JEUS Reference Guide for more information about dynamic changes.

3.3.2. Static Settings

If both dynamic and static settings are changed, only the dynamic settings are applied immediately. The static changes will be applied when the server restarts. This section describes how to identify the modified settings that have not been applied.

Using the Console Tool

Run the server-info command in the console tool (jeusadmin) to show the following server list and status information. Servers with 'true' in the **'Need to Restart'** option must be restarted to apply the changes.

```
[MASTER]domain1.adminServer>server-info

Information about Domain (domain1)
===============================================================================================
+--------+---------+-----+-------+-----+----------------+---------+-------------+-------------+
| Server |  Status | Node|  PID  | Clus|  Latest Start  | Need to | Listen Ports| Running     |
|        |         |Name |       | ter |Time / Shutdown | Restart |             | Engines     |
|        |         |     |       |     |     Time       |         |             |             |
+--------+---------+-----+-------+-----+----------------+---------+-------------+-------------+
| adminSe| RUNNING | N/A | 5360  | N/A | 2016-08-23     | false   | base-192.168| jms,        |
|rver(*) |(00:25:52|     |       |     |(Tue) PM        |         |.34.3:9736   | ejb, web    |
|        |)        |     |       |     |03:53:28 KST    |         | http-server-|             |
|        |         |     |       |     |                |         |0.0.0.0:8088 |             |
|        |         |     |       |     |                |         |jms-internal-|             |
|        |         |     |       |     |                |         |0.0.0.0:9741 |             |
+--------+---------+-----+-------+-----+----------------+---------+-------------+-------------+
| server1| RUNNING |node1| 6424  | N/A | 2016-08-23     | true    | base-192.168| jms,        |
|        |(00:16:15|     |       |     |(Tue) PM         |         |.34.1:9836   | ejb, web    |
|        |)        |     |       |     |04:03:05 KST    |         |             |             |
+--------+---------+-----+-------+-----+----------------+---------+-------------+-------------+
| server2| RUNNING |node2| 3012  | N/A | 2016-08-23     | true    | base-192.168| jms,        |
|        |(00:16:15|     |       |     |(Tue) PM         |         |.34.2:9936   | ejb, web    |
|        |)        |     |       |     |04:03:05 KST    |         |             |             |
+--------+---------+-----+-------+-----+----------------+---------+-------------+-------------+
===============================================================================================
```

3.4. Settings Rotation

When settings are changed, the original settings are backed up. This is called **settings rotation**.

The current settings are always in the following files.

```
DOMAIN_HOME/config/domain.xml
DOMAIN_HOME/config/security/SECURITY_DOMAIN_NAME/policies.xml
DOMAIN_HOME/config/security/SECURITY_DOMAIN_NAME/accounts.xml
```

When there are any changes in the domain.xml file, whether they are dynamic or not, the previous domain.xml file is backed up as a file named 'domain_yyyymmdd_hhmmss.xml' in the

DOMAIN_HOME/config/.history directory. The number of backed up files can be specified using the jeus.server.configfile.backup.size option (default value: 50).

3.5. Changing Common Domain Settings

The following is an example of domain common settings.

```
<domain>
    ...
    <production-mode>true</production-mode>
    <id>427655071</id>
    <master-server-name>adminServer</master-server-name>
    <domain-log-home>${JEUS_HOME}/domain1/logs</domain-log-home>
    <enable-json-command>false</enable-json-command>
    <enable-to-resynchronize-applications>false</enable-to-resynchronize-applications>
    <application-repositories>
        ...
    </application-repositories>
    <system-clustering-framework>
        <transport-type>HYBRID</transport-type>
        <trsnsport-address>224.0.0.1</trsnsport-address>
        <transport-port>12488</transport-port>
        <leader-discovery-timeout>3000</leader-discovery-timeout>
        <failure-detection-timeout>3000</failure-detection-timeout>
        <max-thread-pool-size>10</max-thread-pool-size>
        <min-thread-pool-size>0</min-thread-pool-size>
    </system-clustering-framework>
    <domain-backup>
        <backup-on-boot>false</backup-on-boot>
    </domain-backup>
    ...
</domain>
```

The following describes each item.

Item	Description
Jeus Master Server Name	MASTER ID. This cannot be changed after the domain is created.
Id	Domain ID. This cannot be changed after the domain is created.
Domain Log Home	Directory name to be shared by servers within a domain. It cannot be changed after a domain is configured
Production Mode	Actual production mode. Auto-reload or hot-swap function is not available.
Enable Json Command	Option to use Json Command. This is off by default. This can be selectively enabled/disabled.
Enable To Resynchronize Applications	Option for MS to synchronize applications with MASTER when MS is changed from the INDEPENDENT mode to the DEPENDENT mode. (Default: false, do not synchronize)

Item	Description
Application Repositories	Location of the Applications. This can be changed dynamically. Refer to "Adding, Deleting, and Searching Application Repositories" in JEUS Applications & Deployment Guide for detailed information.
System Clustering Framework	Network configuration used to check internal communication and status of servers. This is applied to the entire domain. If this item is updated, the entire domain must be restarted. By default, JEUS uses multicast to allow servers to share their status. For more information, refer to Creating a Domain. ◦ transport: Specifies a communication method (DUMMY, HYBRID, MESH, and TREE) for communicating with other servers. ◦ DUMMY: Does not involve any communication with other servers. ◦ HYBRID: Uses IP multicast. ◦ MESH, TREE: Operates similarly to multicast by using TCP/IP in an environment where the domain configuration extends beyond the sebnet or IP multicast is not supported. MESH connects all servers in a full-mesh topology, while TREE designates the Master Server as the root and Managed Server as leaf nodes like a tree structure. ◦ transport-address, transport-port: Configures the required address or port for the selected transport type. The following items are not required but they must be properly adjusted according to the domain size. ◦ 'failure-detection-timeout': Time required to detect a server failure. Too low of a timeout setting may result in an incorrect detection. In HYBRID mode, it may take longer for a failure to be detected since IP multicast is used. ◦ 'leader-discovery-timeout': Time required for a starting server to find a group of servers that have been running. ◦ 'max-thread-pool-size': Maximum size of the thread pool used by SCF. ◦ 'min-thread-pool-size': Minimum size of the thread pool used by SCF.
Domain Backup	Option to create backup files when starting the JEUS Master Server. Backup files and directories can be created. By default, a domain backup is not created. For more information, refer to Failover through Domain Backup.

All settings except for **'Enable Json Command'**, **'Enable To Resynchronize Application'**, **'Application Repositories'**, and **'Domain Backup'** require the restart of the entire domain to apply the settings. Unlike individual server settings, since domain settings affect the domain configuration, **'Jeus Master Server Name'**, **'Id'**, **'Domain Log Home'**, **'Production Mode'**, and **'System**

> **Clustering Framework'** should be configured with care when a domain is created.

3.6. Example: Changing Settings

This section describes how to change the system thread pool count and JVM settings of the server by using the `jeusadmin` console tool.

3.6.1. Changing the System Thread Pool Count

This section describes how to change the system thread pool count by using the console tool `jeusadmin`.

Using the Console Tool

You can change the number of system thread pools by executing the **modify-system-thread-pool** command of the console tool (jeusadmin). For more information about the modify-system-thread-pool command, refer to "modify-system-thread-pool" in JEUS Reference Guide.

```
[MASTER]domain1.adminServer>modify-system-thread-pool server1 -max 120
Successfully performed the MODIFY operation for the system thread pool of the server (server1), but
all changes were non-dynamic. They will be applied after restarting.
Check the results using "modify-system-thread-pool server1 or show-system-thread-pool server1".
```

3.6.2. Changing a Server’s JVM Settings

This section describes how to change a server’s JVM settings using the console tool jeusadmin.

Using the Console Tool

To add the server’s JVM settings using jeusadmin, run the **add-jvm-option** command as in the following example. The add-jvm-option command compares the user-entered settings to those in the jvm-option list and add the settings that are not in the list already.

```
[MASTER]domain1.adminServer>add-jvm-option -server server1 -opt "-Xmx256m -XX:MaxPermSize=128m"
Successfully performed the ADD operation for JVM configuration for the server(server1)..
Check the results using "list-jvm-options or add-jvm-option".
```

To change the server’s JVM settings using jeusadmin, run the **modify-jvm-option** command as in the following example. The modify-jvm-option command compares user-entered options with the jvm-option list and modifies the corresponding items.

```
[MASTER]domain1.adminServer>modify-jvm-option -server server1 -old "-Xmx256m -XX:MaxPermSize=128m"
-new "-Xmx512m -XX:MaxPermSize=128m"
Successfully performed the MODIFY operation for JVM configuration for the server(server1)..
Check the results using "list-jvm-options or modify-jvm-option".
```

To delete the server's JVM settings using jeusadmin, run the **remove-jvm-option** command as in the following example. The remove-jvm-option command compares user-entered options with the jvm-option list and deletes the corresponding items.

```
[MASTER]domain1.adminServer>remove-jvm-option -server server1 -opt "-Xmx256m -XX:MaxPermSize=128m"
Successfully performed the REMOVE operation for JVM configuration for the server(server1)..
Check the results using "list-jvm-options or remove-jvm-option".
```

It is recommended to add one option item at a time to the jvm-option list, and to modify or delete an option by using the **list-jvm-option** command.

4. Server Lifecycle

This chapter describes how to start and close a server and check the server lifecycle.

4.1. Preparing to Start the Server

Complete the following tasks to prepare for server startup.

- JEUS must be installed and domains and servers must be created. A user name and password for starting and stopping the server must also be created in advance. If this task is not complete, set up the environment by following the steps in JEUS Installation and Getting Started Guide.
- Confirm that the JVM configuration for the server is configured in the xml file.

  Changing JVM settings requires a server restart. If it is not configured when the domain is created, start MASTER to change the settings and then apply them by restarting MASTER.

  If JVM settings are missing for an MS, start the server after adding JVM settings. Refer to Changing a Server's JVM Settings for the modification method.
- Confirm that a server has an appropriate license.

4.2. Starting Servers

This section describes how to start the Master Server (MASTER) and a Managed Server (MS).

4.2.1. Starting the Master Server (MASTER)

The **startMasterServer** script is used to start MASTER. It requires the domain name of MASTER as well as a user name and password with the authority to start JEUS.

All three options are required if it is run from JEUS_HOME/bin. The domain name can be omitted if it is run from the DOMAIN_HOME path on MASTER.

Run the script depending on the execution path.

- JEUS_HOME/bin

```
JEUS_HOME/bin$ startMasterServer -domain <domain_name> -u <user_name> -p <password>
```

- DOMAIN_HOME/bin

```
DOMAIN_HOME/bin$ startMasterServer -u <user_name> -p <password>
```

- SERVER_HOME/bin

```
SERVER_HOME/bin$ startMasterServer -u <user_name> -p <password>
```

Once the MASTER’s state becomes RUNNING, server log is recorded in the file, SERVER_HOME/logs/JeusServer.log. To monitor the server log in console tool, use the -verbose option. The server process and the launcher process both run on MASTER. The launcher process runs the server process and outputs logs to the console tool.

Use the **jps** command to check LauncherBootStrapper and MasterServerBootstrapper.

```
jps -l
399092 jeus.server.LauncherBootstrapper
399352 jeus.server.admin.MasterServerBootstrapper
409880 sun.tools.jps.Jps
```

If the server fails to start, check the cause in SERVER_HOME/logs/JeusLauncher.log.

1. Jps is a tool, which is in the JAVA_HOME/bin directory, that lists the Java processes.
2. For detailed information about the Launcher, refer to Launcher in JEUS Server Guide.

4.2.2. Starting a Managed Server (MS)

An MS can be started using a script installed in a machine where the MS is located. If multiple servers are distributed on multiple machines in the domain, run the script on each machine that hosts the MS. The name of the script is **startManagedServer**, and **masterurl** setting must be configured in order to synchronize the settings with MASTER.

Using a Script

An MS can be started using a script instead of MASTER.

If multiple servers are distributed on multiple machines in the domain, run the script on each machine that hosts the MS. The name of the script is **startManagedServer**, and **masterurl** setting must be configured in order to synchronize the settings with MASTER.

If masterurl is not configured, MS will start in the INDEPENDENT mode. After MASTER finds the MS, it is managed by MASTER. If MASTER has not started, masterurl can be omitted so that the MS can start in the INDEPENDENT mode. Otherwise, it is recommended to configure masterurl. For detailed information about INDEPENDENT mode, refer to Managed Server (MS) in INDEPENDENT

Mode.

The following shows how to execute the startManagedServer script from different locations.

- JEUS_HOME/bin

```
JEUS_HOME/bin$ startManagedServer -masterurl <master_ip:master_baseport>
-domain <domain_name> -server <server_name> -u <user_name> -p <password>
```

- DOMAIN_HOME/bin

```
DOMAIN_HOME/bin$ startManagedServer -masterurl <master_ip:master_baseport>
-server <server_name> -u <user_name> -p <password>
```

- SERVER_HOME/bin

```
SERVER_HOME/bin$ startManagedServer -masterurl <master_ip:master_baseport> -u <user_name> -p
<password>
```

4.3. Shutting Down Servers

This section describes how to shut down the Master Server (MASTER) and a Managed Server (MS).

4.3.1. Shutting Down the Master Server (MASTER)

To shut down the MASTER, use either the console tool (jeusadmin) or a script.

Using the Console Tool

MASTER can be shut down after stopping all managed servers (refer to Shutting Down a Managed Server (MS)). Run the following **local-shutdown** command.

```
[MASTER]domain1.adminServer>local-shutdown
```

Using a Script

Use a script installed in a machine where the MASTER is located. Run the following **stopServer** script depending on the location.

- JEUS_HOME/bin

```
JEUS_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```

- DOMAIN_HOME/bin

```
DOMAIN_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```

- SERVER_HOME/bin

```
SERVER_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```

4.3.2. Shutting Down a Managed Server (MS)

An MS can be shut down using either MASTER or a script.

Using MASTER

An MS can be shut down through MASTER by using the console tool (jeusadmin).

Managed servers can be shut down by MASTER using the **stop-server** command in jeusadmin, as shown in the following example. For detailed information about the stop-server command, refer to stop-server in JEUS Reference Guide.

```
[MASTER]domain1.adminServer>stop-server <server_list>
```

Using a Script

An MS can be shut down using a script which is installed on the server. If multiple servers are distributed on multiple machines in the domain, run the script on each machine that hosts the MS.

Run the following **stopServer** script depending on the location.

- JEUS_HOME/bin

```
JEUS_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```

- DOMAIN_HOME/bin

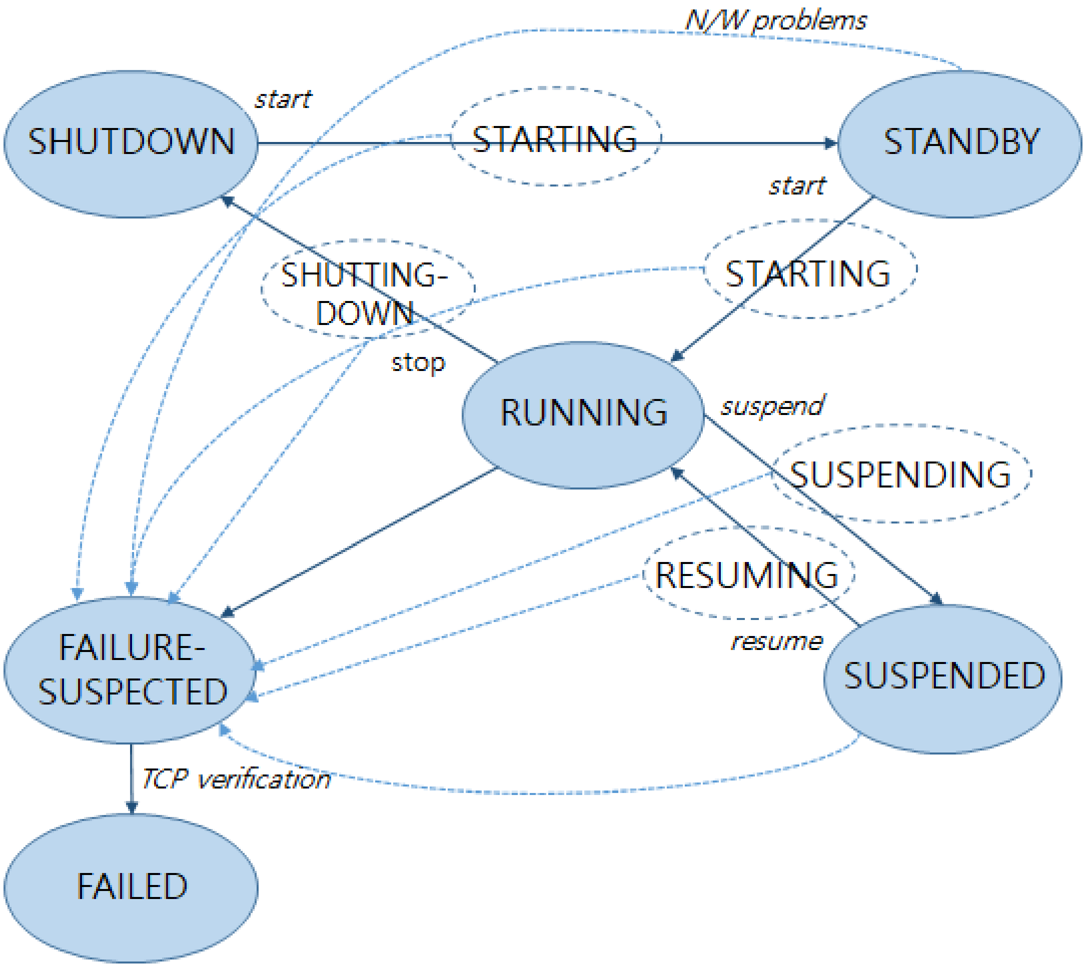
```
DOMAIN_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```

- SERVER_HOME/bin

```
SERVER_HOME/bin$ stopServer -host <server_ip:server_baseport> -u <user_name> -p <password>
```

4.4. Checking Server Lifecycle

MASTER determines the lifecycle of the managed servers.



Server Lifecycle

The following describes each state.

State	Description
SHUTDOWN	Server has not been started or has been shut down normally.
STARTING	Server is in the startup process, but is not ready to run services.
STANDBY	The server is not in the RUNNING state yet because some applications failed to be distributed during startup. Modify the problematic application or use the force option with the start command to ignore the error and start the server.
RUNNING	Server is running and is ready to provide services.

State	Description
SHUTTING_DOWN	Server is shutting down.
SUSPENDING	Server, in the running state, is suspending all running applications.
RESUMING	Server is resuming previously suspended applications.
SUSPENDED	All applications have been suspended. This state allows new applications to be distributed, but they cannot run.
FAILURE_SUSPECTED	MASTER suspects that the MS may have shut down abnormally because it cannot connect to the MS.
FAILED	MASTER has determined that the MS has shut down abnormally because it has not been connected for a specific period of time.

MASTER checks MS state by using System Clustering Framework (SCF).

SCF is a module that allows servers in a domain to check other servers' state and to send/receive messages to/from other servers. Since it uses a network, it can recognize communication delay caused by a network issue (such as a remote server’s load) as an error. If the issue is resolved, it recognizes the state as normal.

SCF uses multicast by default, but it can use virtual multicast if the environment does not support multicast or there is a specific request. Virtual multicast allows socket communication between servers in a domain.

The following are the three ways to check for the server state.

- **Using an MBean**

  Server states can be monitored using J2EEDomain#getServerState (String serverName). Refer to JEUS JMX Guide for detailed information about MBeans.

- **Using the console tool (jeusadmin)**

  Server statuses can be checked using the following **serverinfo** command in the console tool.

```
[MASTER]domain1.adminServer>serverinfo
Information about Domain (domain1)
================================================================================
+--------+--------+-----+-----+-----+------------+--------+-----------+--------+
| Server | Status |Node | PID | Clu |   Latest   |  Need  |   Listen  |Running |
|        |        |Name |     |ster | Start Time |   to   |   Ports   |Engines |
|        |        |     |     |     | / Shutdown |Restart |           |        |
|        |        |     |     |     |    Time    |        |           |        |
+--------+--------+-----+-----+-----+------------+--------+-----------+--------+
| adminS | RUNNIN | nod | 902 | N/A | 2022-07-19 | false  | base-0.0. | jms,   |
|erver   |G(00:42 |e1   |88   |     | (Tue) PM   |        |0.0:9736   |web, ejb|
|(*)     |:15)    |     |     |     |12:55:20 KST|        | http-serv |        |
|        |        |     |     |     |            |        |er-0.0.0.0 |        |
|        |        |     |     |     |            |        |:8088      |        |
+--------+--------+-----+-----+-----+------------+--------+-----------+--------+
| server1| RUNNIN | N/A | 928 | clu | 2022-07-19 | false  | base-0.0. | jms,   |
|        |G(00:00 |     |05   |ster1| (Tue) PM   |        |0.0:9836   |web, ejb|
|        |:08)    |     |     |     |01:37:27 KST|        | http-serv |        |
|        |        |     |     |     |            |        |er-0.0.0.0 |        |
|        |        |     |     |     |            |        |:8188      |        |
+--------+--------+-----+-----+-----+------------+--------+-----------+--------+
| server2| SHUTDO | N/A | N/A | clu | 2022-07-19 | N/A    | N/A       | N/A    |
|        |WN      |     |     |ster1| (Tue) PM   |        |           |        |
|        |        |     |     |     |12:55:20 KST|        |           |        |
+--------+--------+-----+-----+-----+------------+--------+-----------+--------+
| server3| SHUTDO | nod | N/A | N/A | 2022-07-19 | N/A    | N/A       | N/A    |
|        |WN      |e1   |     |     | (Tue) PM   |        |           |        |
|        |        |     |     |     |12:55:20 KST|        |           |        |
+--------+--------+-----+-----+-----+------------+--------+-----------+--------+
================================================================================

===============================================================================================
```

For more information about the serverinfo command, refer to "server-info" in JEUS Reference Guide.

5. JEUS Clustering

This chapter defines a cluster as a part of a domain and describes the composition of a cluster and its relationship to the domain. In addition, it describes how to create, configure, change, and delete a cluster.

5.1. Overview

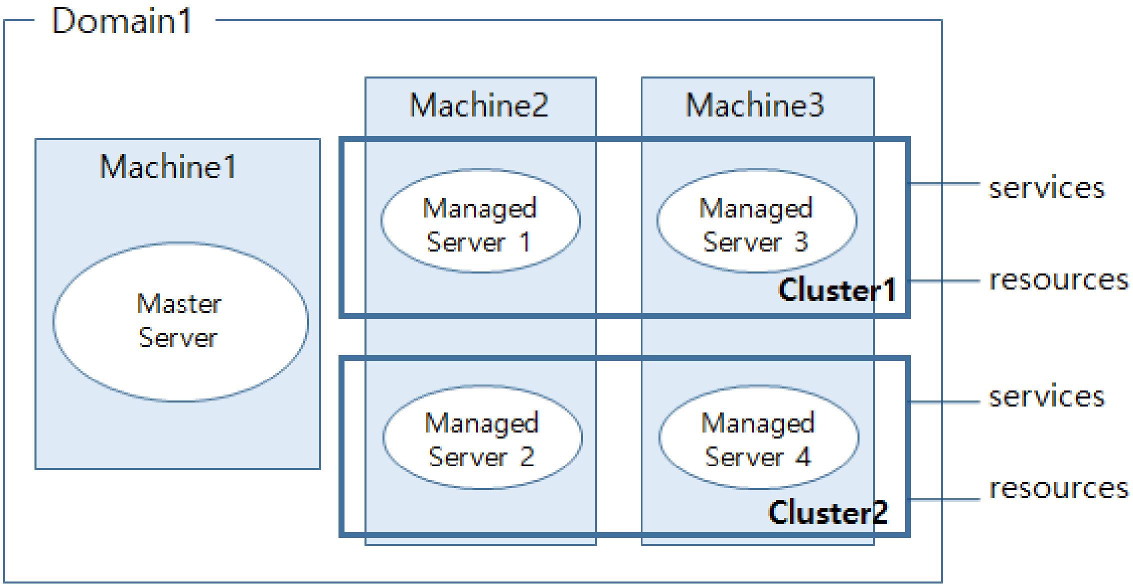
A cluster is a group of servers that run the same services in order to support load balancing for scalability and failover for stability.

Performance can be improved because multiple servers share the load and dynamic load balancing can be performed by adding or deleting servers to the cluster. Even if some servers fail, the remaining servers continue to function, so users are not aware of the failures. A Cluster can easily handle large loads and provide uninterrupted services.

5.2. Relationship between Clusters and Domain

A domain is a group of servers or clusters that provide services. A cluster is a group of servers that provide the same services. A domain can have multiple clusters. Servers in a cluster can be on the same machine or different machines. It is recommend to use servers on different machines, so that if one machine fails the other servers on other machines can continue to provide the services.

A Master server manages a single domain. If there are multiple clusters in a domain, those clusters are managed by the same Master server.



Relationship Between Clusters and Domain

5.3. Cluster Features

A cluster consists of MSs that run the same services in a domain. Clusters usually contain multiple machines and support the following functions.

- **Load balancing**

  Load balancing distributes services across multiple servers to prevent a service from stopping due to excessive load and to improve the service response speed.

  An environment where same services can be executed must be configured, and each server must know the location and status of the other servers.

- **Failover**

  If a server is shut down abnormally, another server takes over the services that the failed server was executing.

  The server that takes over the services must be able to execute the same services and monitor the failed server. The server should also know about the failed server’s service progress status.

To make this possible, it is recommended that all servers have the same applications deployed and registered on them and have the same resources. Servers in a cluster share the location and state of each server using multicast, and also share service progress in the event that a server fails. To improve resource efficiency, servers can be added to or removed from the cluster according to the load.

Some services cannot be clustered. The following are the services that can be clustered.

Service	Description
Servlets/Jsps	Web engines support web application clustering using the web server and HTTP session clustering. For detailed explanations about web application clustering by a web server, refer to "Configuring Web Server Load Balancing" in "JEUS Web Engine Guide", and for more information about HTTP serssion clustering, see "Distributed Session Servers" in "JEUS Session Guide".
EJB	EJB engines support session beans, message-driven beans, entity beans, and timer service clustering. For more information about clustering in EJB engines, refer to "EJB Clustering" in "JEUS EJB Guide".
JMS	JMS supports connection factory, destination, and durable subscriber clustering. For detailed information about clustering in JMS, refer to "JEUS MQ Clustering" in "JEUS MQ Guide".

The following constraints should be considered when creating a cluster.

- All the servers in a cluster must use the same version of JEUS.
- Each server must belong to only one cluster.

- Because resources cannot be shared across domains, all servers in a cluster must be in the same domain.

5.4. Creating a Cluster

When configuring a domain, create a cluster by considering the types and priorities of services. Refer to Adding a Cluster in Configuring a Domain for information about creating a cluster.

5.5. Cluster Settings

Cluster configurations can be divided into common server settings and cluster settings.

5.5.1. Common Server Settings

Common server settings allow all servers in a cluster to run the same services and are needed to create both clusters as well as individual servers.

For more information about common server configurations, refer to "JEUS Configuration" in "JEUS Server Guide".

5.5.2. Cluster Settings

The cluster settings are for load balancing, failover, configuring cluster communication, session servers, timer services, JMS resources, and for registering and deleting data sources.

If the configuration changes, restarting the entire cluster is recommended in case any of the settings requires a server restart.

- **Setting a session server cluster**

  A distributed session server in a clustered environment can use both load balancing and failover.

  There is no configuration to disable a distributed session server. When servers are clustered, a distributed session server is automatically configured across the servers.

  Refer to "Distributed Session Servers" for detailed information about distributed session servers and "Distributed Session Server Configuration" in JEUS Session Guide for detailed description and configuration of each item.

- **Setting the cluster timer service**

  This is used to set up a timer service in a clustered environment. Refer to "EJB Timer Service" in

"JEUS EJB Guide" for detailed information about setting up timer services.

- **Setting clustered JMS resources**

  This is used to set up destination and durable subscriber in the cluster. Refer to "Server Configuration" in "JEUS MQ Guide" for detailed information about setting up JMS resources.

- **Registering and deleting data sources**

  Data sources that are registered in a cluster can be used by all servers in the cluster and the registration is performed dynamically. Deleting data sources that are registered in a cluster can also be performed dynamically. Data sources that are deleted from a cluster are not available to servers in the cluster.

  Refer to "Dynamic Data Source Configuration" in "JEUS Server Guide" for detailed information about using registered data sources and how to register and delete them.

5.6. Changing a Cluster

The number of servers in a cluster can be adjusted by adding or deleting servers based on the load.

5.6.1. Adding a Server to the Cluster

As the load increases, a cluster needs to scale by adding more servers. Adding a server with the same configuration as the existing servers is recommended because it will execute the same services.

This section describes how to add a server to the cluster.

Using the Console Tool

Servers can be added to the cluster using jeusadmin, a console tool.

The following are the steps for creating servers and adding them to a cluster.

1. Create a server (server2) to add to the cluster by using the **add-server** command. Refer to "add-server" in "JEUS Reference Guide" for detailed information about the add-server command.

   ```
   [MASTER]domain1.adminServer>add-server server2 -node node2 -addr 0.0.0.0 -port 9936
   Successfully performed the ADD operation for server (server2).
    NOTICE : base-addr [0.0.0.0] base-port [9936] http-port [8088]
   Check the results using "list-servers or add-server".
   ```

2. Add server2 to the cluster (cluster1) by using the **add-servers-to-cluster** command.

   Refer to "add-servers-to-cluster" in JEUS Reference Guide for detailed information about the add-servers-to-cluster command.

```
[MASTER]domain1.adminServer>add-servers-to-cluster cluster1 -servers server2
Successfully performed the ADD operation for The server list for cluster(cluster1)..
Check the results using "list-clusters cluster1 or add-servers-to-cluster cluster1".
```

3. Start the newly added server.
4. Check that server1 and server2 are running properly in 'cluster1' by using the **serverinfo** command.

```
[MASTER]domain1.adminServer>serverinfo
Information about Domain (domain1)
================================================================================
+-------+--------+-----+-----+-----+--------------+-------+----------+-------+
| Server| Status |Node | PID | Clu |  Latest Start |  Need |   Listen  | Runni |
|       |        |Name |     |ster |    Time /     |  to   |   Ports   |  ng   |
|       |        |     |     |     | Shutdown Time |Restart|           |Engines|
+-------+--------+-----+-----+-----+--------------+-------+----------+-------+
| admin | RUNNIN | nod | 902 | N/A | 2022-07-19    | false | base-0.0. | jms,  |
|Server |G(02:56 |e1   |88   |     |(Tue) PM       |       |0.0:9736   |web,   |
| (*)   |:51)    |     |     |     |12:55:20 KST   |       | http-serv |ejb    |
|       |        |     |     |     |               |       |er-0.0.0.0 |       |
|       |        |     |     |     |               |       |:8088      |       |
+-------+--------+-----+-----+-----+--------------+-------+----------+-------+
| serve | RUNNIN | nod | 100 | clu | 2022-07-19    | false | base-0.0. | jms,  |
|r1     |G(00:00 |e1   |083  |ster1|(Tue) PM       |       |0.0:9836   |web,   |
|       |:17)    |     |     |     |03:51:54 KST   |       | http-serv |ejb    |
|       |        |     |     |     |               |       |er-0.0.0.0 |       |
|       |        |     |     |     |               |       |:8188      |       |
+-------+--------+-----+-----+-----+--------------+-------+----------+-------+
| serve | RUNNIN | nod | 100 | clu | 2022-07-19    | false | base-0.0. | jms,  |
|r2     |G(00:00 |e2   |240  |ster1|(Tue) PM       |       |0.0:9936   |web,   |
|       |:09)    |     |     |     |03:52:02 KST   |       | http-serv |ejb    |
|       |        |     |     |     |               |       |er-0.0.0.0 |       |
|       |        |     |     |     |               |       |:8288      |       |
+-------+--------+-----+-----+-----+--------------+-------+----------+-------+
================================================================================
```

5.6.2. Removing a Server from the Cluster

Servers in a cluster can be removed to reduce the size of a cluster. Shut down the server to remove a server from the cluster. Check and ensure that all running services are complete, and then remove the server from the cluster. This section describes how to remove a server from the cluster.

Using the Console Tool

The following are the steps for removing a server from the cluster using jeusadmin, a console tool.

The following are the steps for removing a server from a cluster when it has been configured by selecting servers.

1. Shut down server2 by using the **stop-server** command. Refer to "stop-server" in JEUS Reference

Guide for detailed information about the stop-server command.

```
[MASTER]domain1.adminServer>stop-server server2
Stop server message to server [server2] was successfully sent.
```

2. Remove server2 from cluster1 by using the **remove-servers-from-cluster** command. Refer to "remove-servers-from-cluster" in JEUS Reference Guide for detailed information about the remove-servers-from-cluster command.

```
[MASTER]domain1.adminServer>remove-servers-from-cluster cluster1 -servers server2
Successfully performed the REMOVE operation for The server list for cluster(cluster1)..
Check the results using "list-clusters cluster1 or remove-servers-from-cluster cluster1".
```

3. Remove server2 by using the **remove-server** command if it is not to be used as an independent server. Refer to "remove-server" in JEUS Reference Guide for detailed information about the remove-server command.

```
[MASTER]domain1.adminServer>remove-server server2
Successfully performed the REMOVE operation for server (server2).
Check the results using "list-servers or remove-server"
```

5.7. Removing a Cluster

A cluster can be removed when a service does not need to run. Remove the cluster after shutting it down and then verifying that all the services have completed. This section describes how to remove a cluster.

Using the Console Tool

The following are the steps for removing a cluster using jeusadmin, a console tool.

1. Shut down the cluster using the **stop-cluster** command. In this example, cluster1, which consists of server1 and server2 will be deleted. Refer to "stop-cluster" in JEUS Reference Guide for detailed information about the stop-cluster command.

```
[MASTER]domain1.adminServer>stop-cluster cluster1
Stopping servers [server1, server2].
Stop server message to the cluster [cluster1] was successfully sent.
```

Run the **serverinfo** command to verify that the state of server1 and server2, which are in cluster1, is SHUTDOWN.

```
[MASTER]domain1.adminServer>serverinfo
================================================================================
+-------+---------+-----+-----+-----+--------------+-------+-----------+-------+
| Server|  Status |Node | PID | Clu | Latest Start |  Need |   Listen  | Runni |
|       |         |Name |     |ster |    Time /    |  to   |   Ports   |  ng   |
|       |         |     |     |     |Shutdown Time |Restart|           |Engines|
+-------+---------+-----+-----+-----+--------------+-------+-----------+-------+
| admin | RUNNING | nod | 102 | N/A | 2022-07-19   | false | base-0.0. | jms,  |
|Server |(00:02:4 |e1   |314  |     |(Tue) PM      |       |0.0:9736   |web,   |
| (*)   |3)       |     |     |     |04:16:59 KST  |       | http-serv |ejb    |
|       |         |     |     |     |              |       |er-0.0.0.0 |       |
|       |         |     |     |     |              |       |:8088      |       |
+-------+---------+-----+-----+-----+--------------+-------+-----------+-------+
| serve | SHUTDOW | N/A | N/A | clu | 2022-07-19   | N/A   | N/A       | N/A   |
|r1     |N(00:01: |     |     |ster1|(Tue) PM      |       |           |       |
|       |03)      |     |     |     |04:18:39 KST  |       |           |       |
+-------+---------+-----+-----+-----+--------------+-------+-----------+-------+
| serve | SHUTDOW | N/A | N/A | clu | 2022-07-19   | N/A   | N/A       | N/A   |
|r2     |N(00:01: |     |     |ster1|(Tue) PM      |       |           |       |
|       |03)      |     |     |     |04:18:39 KST  |       |           |       |
+-------+---------+-----+-----+-----+--------------+-------+-----------+-------+
================================================================================
```

2. Remove the cluster by using the **remove-cluster** command. Refer to "remove-cluster" in JEUS Reference Guide for detailed information about the remove-cluster command.

```
[MASTER]domain1.adminServer>remove-cluster cluster1
Successfully performed the REMOVE operation for cluster (cluster1).
Check the results using "list-clusters or remove-cluster".
```

3. After the cluster has been removed, delete the servers that belonged to the removed cluster unless they are to be used as independent servers. Remove server1 and server2 using the **remove-server** command. Refer to "remove-server" in JEUS Referene Guide for detailed information about the remove-server command.

```
[MASTER]domain1.adminServer>remove-server server1
Successfully performed the REMOVE operation for server (server1).
Check the results using "list-servers or remove-server"

[MASTER]domain1.adminServer>remove-server server2
Successfully performed the REMOVE operation for server (server2).
Check the results using "list-servers or remove-server"
```

6. Server Failure

Abnormal termination of a server that causes service issues can occur for many reasons. These issues can be minimized by creating clusters. This chapter describes the issues that may occur if a server terminates abnormally and how to handle them.

6.1. Abnormal Termination of Master Server (MASTER)

Even if MASTER terminates abnormally, managed servers (MS) can still run their services. If MASTER stops abnormally due to a hardware error, MSs on the faulty machine are likely to fail as well. However, abnormal termination of MASTER itself does not affect the services of MSs in the domain.

6.1.1. Lost Functionality When MASTER Terminates Abnormally

If MASTER terminates abnormally, managing the configuration and adding and modifying applications cannot be performed until MASTER restarts. After MASTER restarts and notifies the MSs, the MSs are managed by MASTER again.

Without being connected to MASTER, an MS cannot start using the configuration in MASTER. Instead, it must use the file in its local cache if it was able to connect to MASTER at least once to obtain the configuration file. It will use the copy in the local cache and then synchronize with MASTER when MASTER starts. If the MS and MASTER share the domain directory because they are on the same machine, the MS can use the configuration in MASTER even if MASTER is not running.

6.1.2. Failover through Domain Backup

Even if MASTER terminates abnormally, the services can still run but some functions cannot be used. MASTER should be restarted as soon as possible, but there are situations where MASTER cannot restart due to such problems as a hardware failure.

To prevent this, MASTER settings and applications must be backed up. MASTER can start in another machine by using the backed up settings and applications. However, to use the applications in the new machine, the JEUS_HOME path of the new machine must be the same as that of the existing machine.

The following are the methods for backing up MASTER settings and applications.

- Automatically backing up MASTER settings and applications whenever MASTER starts
- Specifying to back up MASTER settings and applications using the pack-domain command

When automatic backup is specified using backup configuration in WebAdmin, backup files will be automatically created. Also, the pack-domain command is used to create a backup file when necessary. The created file should be managed by the user.

Automatically backing up MASTER settings and applications whenever MASTER starts

MASTER settings and applications are automatically backed up whenever MASTER is started. Domain backup configuration can be specified using jeusadmin.

- **Using the console tool**

  The **set-domain-backup** command can be used to create the MASTER backup file whenever MASTER starts.

  ```
  [MASTER]domain1.adminServer>set-domain-backup -backupOnBoot true
  Successfully performed the MODIFY operation for Domain Backup Policy.
  Check the results using "set-domain-backup"
  ```

  By default, the backup file is created in the directory where **pack-domain** command is executed. The backup file can be decompressed on a different machine using the **unpack-domain** command.

Specifying to back up MASTER settings and applications using the pack-domain command

- **Backing up MASTER settings and applications**

  Use the **pack-domain** command to back up the settings and applications of MASTER.

  ```
  offline>pack-domain domain1
  Packing the domain [domain1] configuration completed successfully at the path
  [JEUS_HOME/domains/domain1_packed.zip].
  ```

- **Applying backed up MASTER settings and applications to a new machine**

  Use the **unpack-domain** command to decompress the backup files. Users can change MASTER information such as IP, Port, and Node when running the **unpack-domain** command.

  ```
  offline>unpack-domain domain1
  The JEUS Master Server listener address is already set to [0.0.0.0]. Do you want to change it?
  (y/n): y
  Enter the JEUS Master Server base listener address: 0.0.0.0
  The JEUS Master Server listener port is already set to [9736]. Do you want to change it? (y/n): n
  The JEUS Master Server nodename is already set to [node1]. Do you want to change it? (y/n): n
  Unpacking the domain [domain1] configuration completed successfully.
  ```

  Once MASTER starts with the recovered settings, MSs that were in the INDEPENDENT mode synchronize their settings and applications with MASTER. Afterwards, they leave the INDEPENDENT mode and are managed by MASTER again.

  For detailed usage of the pack-domain and unpack-domain commands, refer to

> "pack-domain" and "unpack-domain" in JEUS Reference Guide.

6.1.3. Managed Servers (MSs) in INDEPENDENT Mode

Since Managed Servers run services regardless of the presence of MASTER, the services are not affected when MASTER terminates abnormally.

MSs have a local cache that stores settings and applications, which are synchronized when a server starts and when changes are made. If an MS is not connected to MASTER when it starts, it uses the local cache to start in the **INDEPENDENT mode**. When this happens, the MS is not controlled by MASTER and MASTER cannot monitor the MS. This can occur when MASTER terminates abnormally, when the MASTER address is incorrectly configured, or when a network error occurs.

After MASTER restarts and notifies MS, the MS leaves the INDEPENDENT mode and is managed by MASTER.

6.2. Abnormal Termination of Managed Server (MS)

Managed servers in a domain can detect the abnormal termination of another server because they exchange state information through multicast.

7. Security Management

This chapter briefly explains some approaches to security, including encrypting the password that is required to use JEUS server and manage accounts. The security information is shared by all servers in the domain and is configured using MASTER.

For detailed information about JEUS security, refer to JEUS Security Guide.

7.1. Managing the Account

JEUS has both user and group accounts.

In JEUS, roles are granted rights to the resources. These rights are granted to the user by assigning the user to the role.

There are three ways to assign a role to a user.

- Assign the role to a user name
- Assign the role to a group that the user belongs to
- Assign the role to the parent group of the group the user belongs to

Since the account hierarchy consists of users, groups, and subgroups, users with similar roles in JEUS can be granted similar permissions at once. Refer to Configuring Security System Policies for detailed information about security policies.

7.2. Encrypting and Storing the Password

Passwords for system management can be encrypted and stored.

The encrypted information is stored in the following format.

```
{algorithm}password
```

The following encryption algorithms are available for use.

- AES
- DES
- DESede
- blowfish
- SEED
- base64

- SHA
- SSHA

The password can be encrypted and stored using the console tool (jeusadmin), or manually by using the encryption tool. This section describes each method.

7.2.1. Using the Console Tool

The following is an example of encrypting and storing the password "password' for the account "administrator" using the AES encryption algorithm through the console tool jeusadmin.

```
[MASTER]domain1.adminServer>set-password administrator password -algorithm AES
The password is set for [administrator].

[MASTER]domain1.adminServer>exit

JEUS_HOME/bin$ jeusadmin -u administrator -p password

Attempting to connect to 127.0.0.1:9736.
The connection has been established to JEUS Master Server [adminServer]
in the domain domain1.
JEUS 9 Administration Tool
To view help, use the 'help' command.
```

7.2.2. Using Encryption Tools

The password can be manually encrypted using an encryption tool. For example, the DB password and the passwords in accounts.xml can be encrypted instead of using plain text.

The encrypted text is configured in the following format anywhere a password is required.

```
{algorithm}password
```

Use the encryption tool provided by JEUS to encrypt the string. Enter the string to encrypt and the algorithm, and the tools will output the encrypted text.

1. For more information about the encryption tool, refer to "encryption" in JEUS Reference Guide.

Both base64 are hashing algorithms that always produce the same output for a given string. Other algorithms require a secret key. Refer to Managing the Secret Key File for detailed information about how to manage this secret key.

The following is an example of configuring an encrypted password in accounts.xml.

Using Encryption Tool - <accounts.xml>

```
<accounts xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
    <users>
        <user>
            <name>administrator</name>
            <password>{base64}amV1czEyMw==</password>
        </user>
        <user>
            <name>user1</name>
            <password>{AES}i06wYRz3Gqun2sKtXHIq+Tw3vUcc=</password>
        </user>
        . . .
    </users>
    . . .
</accounts>
```

XML can be directly edited when a domain is first created, but afterward it is recommended to use the console tool, jeusadmin.

7.3. Managing the Secret Key File

This section describes how to create, manage, and protect the secret key file.

7.3.1. Creating and Managing the Secret Key File

The AES, DES, DESede, SEED, and BlowFish algorithms that are provided by the encryption tool in 'JEUS_HOME/bin' require a secret key to encrypt or decrypt. JEUS stores the secret key in a file. This file is automatically created when the encryption tool is used for the first time.

The file is named **security.key** and can be found in the following path.

```
JEUS_HOME/domains/<domain-name>/config/security
```

The path can be configured with the system property jeus.security.keypath which can be an absolute or relative path. For a relative path, the base path is the path to JVM. If the path points to a directory, the security.key file in that directory is used. If the path points to a file, that file is used.

7.3.2. Encrypting the Secret Key File

JEUS can encrypt the secret key file. A password called the master password is required to encrypt the secret key file. Use the -protectkey option with the encryption tool to protect the secret key file with this master password.

The master password is required to start a server.

There are two ways to enter the master password.

- If a server is started by a background process, the master password can be set in the jeus.security.master system property.
- If a server is started by JEUS script, the master password can be entered in the console tool.

Using a shell script to set the master password in jeus.security.master is not secure. It is recommended to directly enter the master password in the console tool.

1. For detailed information on server startup, refer to "Script Mode Usage and Script Writing Method" in JEUS Reference Guide.
2. Refer to "encryption" in JEUS Reference Guide for detailed information about secret keys.

7.4. Managing the Keystore and Truststore

This section briefly describes how to manage the keystore and truststore that are used in SSL communication.

Both keystore and truststore are a Java Key Store (JKS). They can be created and managed by using key tools provided by the JDK.

For detailed information about JSSE, JKS, and keytools, refer to the Java documentation. For detailed information about keystores, truststores, and SSL, refer to Oracle Java Technical Documentation.

By default, JEUS uses the following files as the keystore and truststore. To use an alternate path or file name, it must be configured manually.

- Keystore

  The repository where private keys are saved.

  ```
  JEUS_HOME/domains/<domain-name>/config/security/keystore files
  ```

- Truststore

  The repository where public keys are saved.

  ```
  JEUS_HOME/domains/<domain-name>/config/security/truststore files
  ```

The properties can be configured at the system level or separately wherever a keystore and truststore are needed.

The following describes each system property.

System Property	Description
jeus.ssl.keystore	Path to the keystore file. (Default value: JEUS_HOME/domains/<*domain-name*>/config/security/keystore)
jeus.ssl.truststore	Path to the truststore file. (Default value: JEUS_HOME/domains/<*domain-name*>/config/security/truststore)
jeus.ssl.keypass	Password for the keystore. (Default value: jeuskeypass)
jeus.ssl.trustpass	Password for the truststore. (Default value: jeustrustpass)

A password is required to use a keystore or truststore. As previously mentioned, the passwords for the keystore and truststore can be configured in JEUS. Similar to the file path setting, passwords can be configured at the system level or separately as needed.

A keystore requires a password for the keys in the keystore, and a password for the keystore itself. In general, the key password is same as the keystore password.

If the key password is different from the repository's password, set the key password in <keystore-keypassword>. Also, in order to use the keystore all the keys in the keystore must use the same key password.

Although a keystore and truststore are created when JEUS is installed, it is recommended to create and configure them manually.

7.5. Managing Executable Scripts in a SSL-Enabled Domain

This section describes the configurations that must be added to various executable scripts when the Base Listener of each server is set to SSL in the domain.

When setting each server in the domain to SSL, the server's SSL configuration is used for inter-server communication. However, additional SSL configuration is required for the processes that run on the client side such as the consol tool or executable scripts.

SSL settings are required for the following.

Subdirectory	Description
jeusadmin	Script to start the Console Admin.

Subdirectory	Description
startManagedServer	Script to directly start an MS.
stopServer	Script to stop the server.
Other Standalone Client	Standalone Client that is not a script, but must communicate with a server using SSL.

In the following example, the bolded texts are the configurations that must be added to the scripts.

jeusadmin script that specifies SSL-related client configurations

```
. . .
# execute jeusadmin
"${JAVA_HOME}/bin/java" -Xmx128m -classpath "${BOOTSTRAP_CLASSPATH}" ${TOOL_OPTION}
-Djmx.remote.x.request.timeout=600000
-Djeus.home="${JEUS_HOME}"
-Djeus.baseport=${JEUS_BASEPORT}
-Djeus.tool.console.useJLine="false"
-Djava.naming.factory.initial=jeus.jndi.JEUSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url
-Djava.util.logging.config.file="${JEUS_HOME}/bin/logging.properties"
-Djeus.net.client.use-ssl=true
-Djavax.net.ssl.trustStore=${JEUS_HOME}/domains/domain1/config/truststore
-Djavax.net.ssl.trustStorePassword=changeit
-Djavax.net.ssl.trustStoreType=JKS
-Dssl.TrustManagerFactory.algorithm=SunX509
-Djavax.net.ssl.keyStore=${JEUS_HOME}/domains/domain1/config/keystore
-Djavax.net.ssl.keyStorePassword=changeit
-Djavax.net.ssl.keyStoreType=JKS
-Dssl.KeyManagerFactory.algorithm=SunX509
-Djeus.security.keypath=${JEUS_HOME}/domains/domain1/config/security/security.key
${JAVA_ARGS}
jeus.tool.console.ConsoleBootstrapper ${BOOT_PARAMETER}
```

The bolded text is a separate file that contains the SSL-related client configurations.

jeusadmin script that specifies a separate file containing SSL-releated client configurations

```
. . .
# execute jeusadmin
"${JAVA_HOME}/bin/java" -Xmx128m -classpath "${BOOTSTRAP_CLASSPATH}" ${TOOL_OPTION}
-Djmx.remote.x.request.timeout=600000
-Djeus.home="${JEUS_HOME}"
-Djeus.baseport=${JEUS_BASEPORT}
-Djeus.tool.console.useJLine="false"
-Djava.naming.factory.initial=jeus.jndi.JEUSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url
-Djava.util.logging.config.file="${JEUS_HOME}/bin/logging.properties"
-Djeus.net.client.ssl-properties-file=ssl.properties
${JAVA_ARGS}
jeus.tool.console.ConsoleBootstrapper ${BOOT_PARAMETER}
```

The following is an example file that defines SSL-releated client configurations. Note that a relative path beginning with ${JEUS_HOME} cannot be used in a file.

SSL-related client configuration - <ssl.properties>

```
javax.net.ssl.keyStore=/Users/tmax/JEUS9/domains/domain1/config/keystore
javax.net.ssl.trustStore=/Users/tmax/JEUS9/domains/domain1/config/truststore
javax.net.ssl.keyStorePassword=jeus
javax.net.ssl.trustStorePassword={AES}i06wYRz3Gqun2sKtXHIq+Tw3vUcc=
jeus.ssl.keystorekeypass={base64}amV1czEyMw==
jeus.security.keypath=/Users/tmax/JEUS9/domains/domain1/config/security/security.key
```

The password property can be set to an encoded or encrypted character string in the same way that they are used in accounts.xml. Use the encryption tool provided by JEUS.

1. For more information about the encryption tool, refer to "encryption" in JEUS Reference Guide.