



# WebSphere Application Server V7: Administration of WebSphere Processes

In this chapter, we provide information about basic administration tasks. The focus of this chapter is on managing WebSphere® processes, including the deployment manager, nodes and node agents, application servers, and application server clusters.

We cover the following topics:

- ▶ “Working with the deployment manager” on page 2
- ▶ “Starting and stopping an administrative agent” on page 8
- ▶ “Starting and stopping the job manager” on page 9
- ▶ “Working with application servers” on page 9
- ▶ “Working with nodes in a distributed environment” on page 37
- ▶ “Working with clusters” on page 49
- ▶ “Working with virtual hosts” on page 62
- ▶ “Managing your configuration files” on page 65
- ▶ “Managing applications” on page 69

# Working with the deployment manager

In this section we provide information about how to manage the deployment manager and introduce you to the configuration settings associated with it.

## Deployment manager configuration settings

A deployment manager is built by creating a deployment manager profile. After it is built, there is usually not much that you need to do. However, it is good to note that there are settings that you can modify from the administration tools.

To view the deployment manager from the administrative console, select **System Administration** → **Deployment manager**. You have two pages available, the Runtime page and the Configuration page. Figure 1 shows the Configuration page.

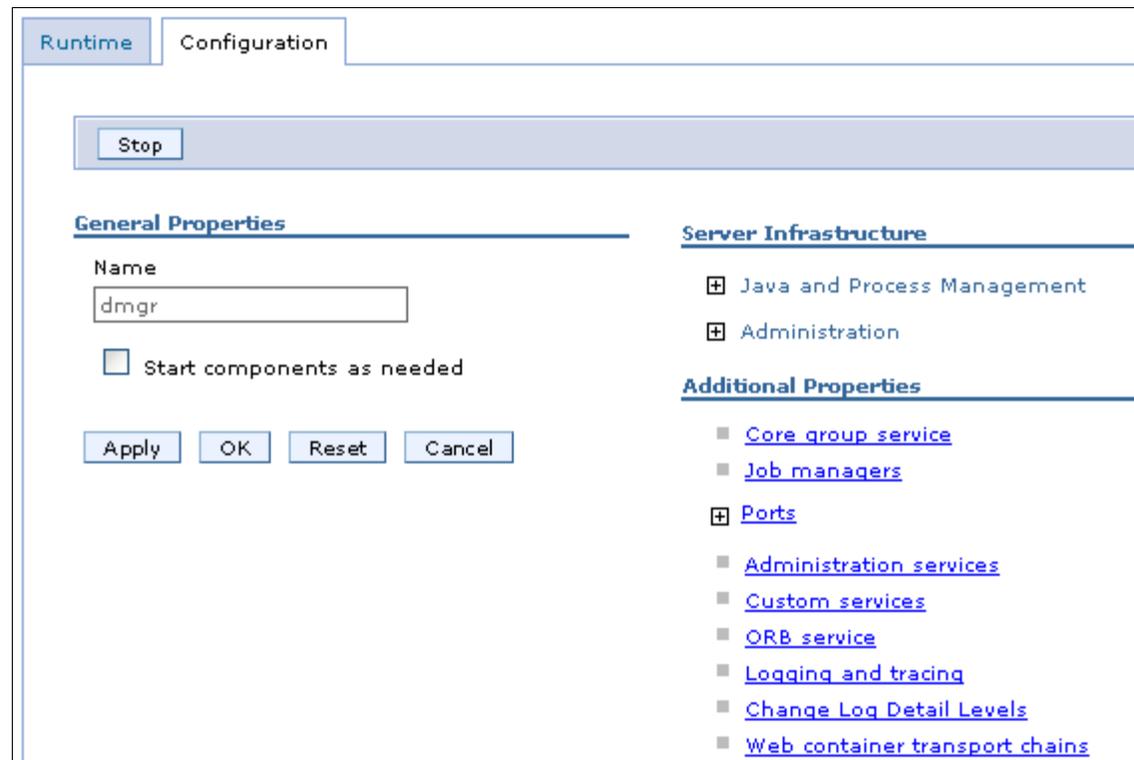


Figure 1 Deployment manager configuration

## Configuration tab

Two options on this panel to note are:

- ▶ **Job managers** (*New in v7*)

Click this link to view work with job managers. You can view the job managers this deployment manager is registered to and you can register or unregister the deployment manager with a job manager.

- ▶ **Ports**

Select this link to view and manage the ports used by the deployment manager. This is useful for diagnostics, for example, when the deployment manager does not start because of a port conflict with another process. It is also useful for finding the SOAP connector port required when federating a custom profile.

## Deployment manager Runtime tab

In addition to the Configuration page, the administrative console contains a Runtime page for the deployment manager. To view the Runtime page, select **System Administration** → **Deployment manager** and click the **Runtime** tab at the top of the page. Figure 2 shows the Runtime tab.

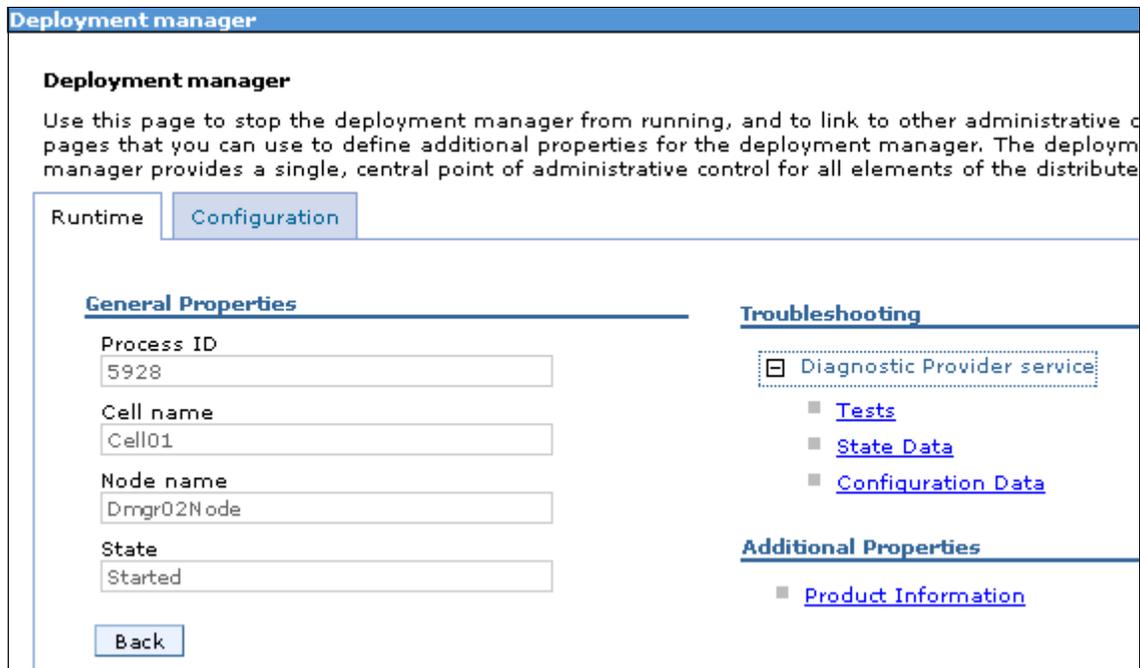


Figure 2 Deployment manager runtime page

The fact that the state is Started does not mean much, because you would not be able to access the administrative console otherwise. Items on this panel to note are:

► **Diagnostic Provider service**

These options allow you to query components for current configuration data, state data, and to run a self-diagnostic test routine. You would most likely use these options at the request of IBM® support.

► **Product information (New in v7)**

Selecting this link opens a new page (Figure 3) that provides information about the level of code running on the deployment manager system and provides links for more detailed information, including the installation history for the product and maintenance.

The product information is stored as XML files in *install\_root/properties/version/dtd* folder and can be viewed without the administrative console.

**Product Information**  
Specifies the product information for this installation of WebSphere(R) Application Server.

Runtime

**General Properties**

| Product Name                          | ID | Version | Build Date | Build Level |
|---------------------------------------|----|---------|------------|-------------|
| IBM WebSphere Application Server - ND | ND | 7.0.0.1 | 11/12/08   | cf010845.08 |

OK

**Additional Properties**

- [Components](#)
- [Extensions](#)
- [History Report](#)
- [Product Report](#)
- [PTFs](#)

Figure 3 Product information

## Starting and stopping the deployment manager

The deployment manager must be started and stopped with commands. The administrative console is not available unless it is running.

### Starting the deployment manager with startManager

The **startManager** command is used to start the deployment manager on distributed systems, as shown in Example 1.

### Example 1 *startManager* command

---

```
C:\WebSphereV7\AppServer\profiles\dmgr01\bin>startManager
ADMU7701I: Because dmgr is registered to run as a Windows Service, the
request to start this server will be completed by starting the
associated Windows Service.
ADMU0116I: Tool information is being logged in file

C:\WebSphereV7\AppServer\profiles\dmgr01\logs\dmgr\startServer.log
ADMU0128I: Starting tool with the Dmgr01 profile
ADMU3100I: Reading configuration for server: dmgr
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server dmgr open for e-business; process id is 1536
```

---

Run this command from the deployment manager *profile\_root/bin* directory. If you run it from the *install\_root/bin* directory, use the `-profileName` parameter to ensure that the command is run against the deployment manager profile.

### **Syntax of *startManager***

The syntax of the `startManager` command is:

```
startManager.bat(sh) [options]
```

The options are shown in Example 2.

### Example 2 *startManager* options

---

```
Usage: startManager [options]
      options: -nowait
              -quiet
              -logfile <filename>
              -replacelog
              -trace
              -script [<script filename>] [-background]
              -timeout <seconds>
              -statusport <portnumber>
              -profileName <profile>
              -recovery
              -help
```

---

All arguments are optional. For more information about these options, see:

► *startManager* command

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml\\_startmanager.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml_startmanager.html)

## Starting the deployment manager on z/OS (START command)

On z/OS®, the deployment manager can be started using a JCL start procedure. The exact command can be found in the BBOCCINS instruction member of the JCL generated to create the profile.

For example:

```
START WPDCR,JOBNAME=WPDMMGR,ENV=WPCELL.WPDMNODE.WPDMMGR
```

Where:

- ▶ WPDCR is the JCL start procedure.
- ▶ WPDMMGR is the Job name.
- ▶ ENV is the concatenation of the cell short name, node short name, and server short name.

Starting the deployment manager will start the following components:

- ▶ A daemon. In our example, named WPDEMND. There will be one daemon per cell per MVS image. One of the functions of the daemon server is to provide the “location name service” for the cell. All daemons in the cell are fully aware of all the objects in the cell and use the same port values.
- ▶ A controller region. In our example, named WPDMMGR. The controller region serves many functions, including acting as the endpoint for communications.
- ▶ A servant region. In our example, named WPDMMGRS. The servant region contains the JVM™ where the applications are run.
- ▶ If you are using messaging, you will also see a control region adjunct server start.

## Stopping the deployment manager

The deployment manager is stopped with the **stopManager** command, as shown in Example 3.

*Example 3 stopManager command*

---

```
C:\WebSphereV7\AppServer\profiles\dmgr01\bin>stopManager
```

```
ADMU7702I: Because dmgr is registered to run as a Windows Service, the request to stop this server will be completed by stopping the associated Windows Service.
```

```
ADMU0116I: Tool information is being logged in file
```

```
C:\WebSphere\AppServer\profiles\Dmgr01\logs\dmgr\stopServer.log
```

```
ADMU0128I: Starting tool with the Dmgr01 profile
```

```
ADMU3100I: Reading configuration for server: dmgr
```

ADMU3201I: Server stop request issued. Waiting for stop status.  
ADMU4000I: Server dmgr stop completed.

---

### ***Syntax of stopManager***

The syntax of the **stopManager** command is:

stopManager.bat(sh) [options]

The options are shown in Example 4.

#### ***Example 4 startManager options***

---

Usage: stopManager [options]  
options: -nowait  
          -quiet  
          -logfile <filename>  
          -replacelog  
          -trace  
          -timeout <seconds>  
          -statusport <portnumber>  
          -conntype <connector type>  
          -port <portnumber>  
          -username <username>  
          -password <password>  
          -profileName <profile>  
          -help

---

All arguments are optional. See:

► ***stopManager command***

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml\\_stopmanager.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml_stopmanager.html)

### **Stopping the deployment manager on z/OS (STOP command)**

To stop the deployment manager with a STOP command, use the following format:

STOP *dmgr\_job*

For example:

STOP WPDAGR

Stopping the daemon server will stop all servers for that cell, and all the servers on that daemon instance's MVS image will be stopped in an order fashion.

For example:

```
STOP WPDEM
```

## Windows start menu and services

On a Windows® system, you have the option of starting and stopping the deployment manager using the Start menu. For example:

- ▶ **Start** → **All Programs** → **IBM WebSphere** → **Application Server Network Deployment V7.0** → **Profiles** → *profile\_root* → **Start the deployment manager**

Also, on a Windows system, you have the option of registering the deployment manager as a Windows service. In order to have it registered, you must select this option when you create the deployment manager profile or register it later using the **WASService** command.

For more information about the WASService command, see:

- ▶ *WASService command*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rins\\_wasservice.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rins_wasservice.html)

If the deployment manager is registered as a Windows service, all other options for starting the dmgr process are unchanged from the administrator's point of view, however the command used will start or stop the process through the service. In addition, you have the option to allow the service to be started automatically when the operating system starts.

## Starting and stopping an administrative agent

An administrative agent process is managed in the same manner as an application server process. The process name is **adminagent**.

To view the status of the administrative agent process, use the following command:

```
serverStatus -adminagent
```

To start an administrative agent:

```
startServer adminagent
```

To stop an administrative agent:

```
stopServer adminagent
```

## Starting and stopping the job manager

A job manager process is managed in the same manner as an application server process. The process name is **jobmgr**.

To view the status of the jobmgr, use the following command:

```
serverStatus -jobmgr
```

To start a job manager:

```
startServer jobmgr
```

To stop a job manager:

```
stopServer jobmgr
```

## Working with application servers

This section covers the following topics:

- ▶ Creating an application server
- ▶ Viewing the status of an application server
- ▶ Starting an application server
- ▶ Stopping an application server
- ▶ Viewing runtime attributes of an application server
- ▶ Customizing application servers

### Terminology—application server types:

- ▶ A *standalone application server* is created with an application server profile and is not federated to a cell or registered with an administrative agent. A standalone application server hosts its own administrative services and operates independently from other WebSphere processes. It cannot participate in application server clusters. Each standalone application server has its own node.

This option is available on all WebSphere Application Server packages, but is the only option in the Base and Express environments.

- ▶ An *unfederated application server* (**New in V7**) is an application server that resides on a node managed from an administrative agent. Unfederated application servers have the characteristics of a standalone application server, in that they cannot be used in a cluster. However, multiple application servers can exist on one node.

This option is available on all WebSphere Application Server packages.

- ▶ A *managed application server* is one that resides on a node that is managed from a deployment manager. A managed server can either be an application server that was created using an application server profile and subsequently federated to the cell, or it can be created directly from the deployment manager's administrative console.

Managed application servers can be clustered for high-availability and workload balancing.

This is only possible with the Network Deployment package.

## Creating an application server

The process to create an application server depends on your WebSphere Application Server package.

### Standalone application servers

Standalone application servers are created by creating an application server profile. This results in a profile that defines one standalone application server. This application server hosts the sample applications and the administrative console application.

**(New in V7)** In previous versions, a standalone server was always named `server1`. With V7 you have the opportunity to give the server a different name during profile creation.

For information about creating an application server profile, see Chapter 2, *Working with profiles on distributed systems*.

## Unfederated application server

**(New in V7)** An administrative agent can monitor and control multiple unfederated application servers on one or more nodes.

Unfederated application servers can be created in multiple ways:

- ▶ The first server on a node to be managed by an administrative agent must be created with a standalone profile and then registered with the administrative agent.

The registration process disables the administrative console on the server and makes a console for the application server node available on the administrative agent process.

- ▶ Additional unfederated application servers on that node are created from the administrative agent. See “Creating an application server from the administrative console” on page 13.
- ▶ When you use the administrative agent console to register the application server node to a job manager, additional application servers can be created on the node by submitting a job from the job manager console.

### What about wsadmin?

The administrative service remains active in unfederated application servers that are registered to an administrative agent. You can connect to either the application server or the administrative agent to run wsadmin commands, but all admin operations performed by connecting to the application server are forwarded to the agent. So you should ideally connect to the agent to avoid that extra hop.

## Managed application servers

In a distributed server environment, you create an application server from the deployment manager administrative console. See “Creating an application server from the administrative console” on page 13.

If you are creating an application server with the intention of adding it to a cluster, use the **Servers** → **Clusters** → **WebSphere application server cluster** option. See “Working with clusters” on page 49.

## Application server options

You need to consider certain options as you create an application server. The method by which you select these options varies depending on how you are creating the server, but the values are the same.

### Templates

An application server is created based on a template that defines the configuration settings. Four template options are provided:

- ▶ default: Standard production server. You get this option if you do not specify a template for a server on a distributed system.
- ▶ defaultZOS: This is only available on z/OS platforms and is the only option until you create new templates.

DeveloperServer: The DeveloperServer template is used when setting up a server for development use. This template will configure a JVM for a quick start-up by disabling bytecode verification, and performing JIT compilations with a lower optimization level. This option should not be used on a production server, where long run throughput is more important than early server startup.

- ▶ Custom template: You can create templates based on existing application servers. (see “(Optional) Creating an application server template” on page 18).

### Ports

Each server process uses a set of ports that must be unique on the system. When you create an application server, you have the following options:

- ▶ Use the default ports:  
Use this selection if you will only have one application server on the system or if this is the first application server created and port selection is not an issue.
- ▶ Have a set of ports selected that are unique to the WebSphere system installation:

This ensures that no two WebSphere processes in the installation will have unique ports assigned. It does not guarantee that ports will be selected that are not in use by non-WebSphere processes, or by WebSphere processes installed as a separate installation.

- ▶ Specify the ports:  
This option is best if you have a convention for port assignment on your system that ensures unique ports are used by all processes, both WebSphere and non-WebSphere.

This option is only available when you create a new profile using the Advanced option or the **manageprofiles** command.

If you find that you have a port conflict, you can change the ports after the application server is created.

### ***z/OS settings***

Next we describe the various z/OS settings:

▶ Long name:

The long name of an application server is the name used in scripting and the administrative console. Long names can be up to 50 characters long and include mixed-case alphabetic characters, numeric characters, and the following special characters: ! ^ ( ) \_ - . { } [ ]

▶ Short name:

Short names are specific to the z/OS implementation of WebSphere Application Server and are the principal names by which cells, nodes, servers, and clusters are known to z/OS.

▶ Specific short name (z/OS):

The short name is also used as the JOBNAME for the server. If you do not specify a value for the short name field, the short name defaults to BBOSnnn, where nnn is the first free number in the cell that can be used to create a unique short name. Make sure that you set up a RACF® SERVER class profile that includes this short name.

▶ Generic short name (z/OS):

Nodes that have not been clustered have a server generic short name, also called a *cluster transition name*. When a cluster is created from an existing application server, the server's generic short name becomes the cluster name.

No two servers on the same z/OS system should have the same server generic short name unless they are in the same cluster.

If you do not specify a value for the generic short name field, the generic short name defaults to BBOCnnn, where nnn is the first free number in the cell that can be used to create a unique generic short name.

▶ Bit mode (z/OS):

The default setting is that the application server runs in 64-bit mode, however, you can elect to run in 31-bit mode. Note that 31-bit mode is deprecated.

### **Creating an application server from the administrative console**

To create an application server from the administrative console:

1. Open the deployment manager administrative console.
2. Select **Servers** → **Server types** → **WebSphere application server**.

3. Click **New**.
4. Select the node for the new server and enter a name for the new server (Figure 4).

**Step 1: Select a node**

Step 2: Select a server template

Step 3: Specify server specific properties

Step 4: Confirm new server

**Select a node**

Select the node that corresponds to the server you wish to create.

Select node  
node40a (ND 7.0.0.1) ▼

\* Server name  
server40a1

Next Cancel

Figure 4 Create an application server: Step 2

Click **Next**.

5. Select a template to use by clicking the appropriate radio button (Figure 5).

Step 1: Select a node

**Step 2: Select a server template**

Step 3: Specify server specific properties

Step 4: Confirm new server

**Select a server template**

| Select                           | Name            | Type   | Description  |
|----------------------------------|-----------------|--------|--|
| <input checked="" type="radio"/> | default         | System | The WebSphere Default Server Template                            |
| <input type="radio"/>            | DeveloperServer | System | This template is optimized to perform well for development uses. |

Previous Next Cancel

Figure 5 Create an application server: Step 2

On z/OS systems, there is one system defined template called defaultZOS.  
Click **Next**.

6. The options you see on the next window vary depending on the platform.  
For distributed platforms, you see a panel as shown in Figure 6.

Check the **Generate Unique Ports** box to have unique ports generated for this server. Deselecting this option will generate the default set of ports.

If you have multiple core groups defined, you have the option to select the core group.

The screenshot shows a dialog box titled "Specify server specific properties". On the left, a vertical sidebar lists four steps: "Step 1: Select a node", "Step 2: Select a server template", "Step 3: Specify server specific properties" (highlighted with a yellow arrow), and "Step 4: Confirm new server". The main area of the dialog contains a checked checkbox for "Generate Unique Ports" and a "Core Group" dropdown menu with "DefaultCoreGroup" selected. At the bottom, there are three buttons: "Previous", "Next", and "Cancel".

Figure 6 Create an application server: Step 3 for distributed systems

For z/OS systems, you see a panel as shown in Figure 7.

The screenshot shows a dialog box titled "Specify server specific properties". On the left, a vertical sidebar lists four steps: "Step 1: Select a node", "Step 2: Select a server template", "Step 3: Specify server specific properties" (highlighted with a yellow arrow), and "Step 4: Confirm new server". The main area of the dialog contains a checked checkbox for "Generate Unique Ports", two text input fields labeled "Server Specific Short Name" and "Server Generic Short Name", and a checked checkbox for "Run in 64 bit JVM mode". At the bottom, there are three buttons: "Previous", "Next", and "Cancel".

Figure 7 Create an application server: Step 3 for z/OS

The server specific short name specifies the short name for the server. This is also used as the job name (for example, BBOS002). The generic short name is the short name that is converted to a cluster short name if the server is later used in a cluster.

Click **Next**.

7. A summary window is presented with the options you chose. Click **Finish** to create the server.
8. In the messages box, click **Save** to save the changes to the master repository.
9. Review and update the virtual host settings (see “Update the virtual host settings” on page 18)
10. Regenerate the Web server plug-in and propagate it to the Web server.

Managing Web server plug-ins is discussed in the Information Center. See the topic, *Communicating with Web server*, at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/twsv\\_plugin.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/twsv_plugin.html)

**Note:** If you are creating an application server on a Windows operating system, this process does not give you the option of registering the new server as a Windows service. You can do this later with the **WASService** command (see “Enabling process restart on failure” on page 87).

## Creating an application server from the job manager

To create an application server from the job manager, make the following selections as you step through the process to submit the job:

1. Step 1: Select the **Create application server** job type.
2. Step 2: Select the job target:
  - If you are creating an application server on an unfederated node, select the application server node.
  - If you are creating a new managed server, select the deployment manager node.

Enter the user ID and password with administrative authority on the target node.

3. Step 3: Specify the job parameters (Figure 8 on page 17).

At minimum:

- Specify the name of the new application server.
- If the target is a deployment manager, enter the name of the node that the server will be created on.

Step 1: Choose a job type

Step 2: Choose job targets

→ **Step 3: Specify job parameters**

Step 4: Schedule the job

Step 5: Review the summary and submit the job

### Specify job parameters

**Job type: Create application server**

\* Server name

Node name

Additional job parameters.

---

**Server template**

Template name

Template location

---

**Port control**

Generate unique ports

---

**Platform specific**

Specific short name

Generic short name

Bit mode

Figure 8 Specify the options for the new server

The optional settings allow you to add platform specific settings, specify a different template, and specify the setting that determines if ports unique to the installation are generated.

- The template name field will default to the production server template for the operating system on which the application server will run. You only need to specify this setting if you want to use a custom template or the DeveloperServer template.
- Generate unique ports:
  - Specify true to generate ports unique to that installation (the default)
  - Specify false to use the default port numbers.

4. Review and update the virtual host settings (see “Update the virtual host settings” on page 18).
5. Regenerate the Web server plug-in and propagate it to the Web server.

Managing Web server plug-ins is discussed in the Information Center. See the topic, *Communicating with Web servers*, at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/twsv\\_plugin.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/twsv_plugin.html)

## Update the virtual host settings

When you install applications, you associate a virtual host with each Web module.

When you create a new application server, the default\_host virtual host is set as the default virtual host for Web modules installed on the server. You can change this default in the Web container settings for the application server or simply select a new virtual host when you install the applications.

If the application will only be accessed via a Web server, and the virtual host that you will use has been set up with the Web server port in the list of host aliases, then no action is necessary.

However, if application clients will access the Web container directly, or if you will be installing SIP applications on this server, you need to ensure that the relevant ports generated for this application server are added in the host alias list. See “Working with virtual hosts” on page 62 for more information.

## (Optional) Creating an application server template

To create an application server template based on an existing server:

1. Select **Servers** → **Server Types** → **WebSphere application servers**.
2. Click **Templates...** at the top of the server list.
3. Click **New**.
4. Select a server from the list to build the template from and click **OK**.
5. Enter a name and description for the template and click **OK**.
6. Save your configuration.

The new template will be in the list of templates and will be available to select, the next time you create an application server.

## Viewing the status of an application server

There are multiple ways to check the status of an application server:

- ▶ Use the `serverStatus` command on the system where the application server is running.
- ▶ In a distributed environment, you can view the status from the administrative console. The node for the application server must be active for the deployment manager to know the status of a server on that node.
- ▶ From the job manager
- ▶ If the server is registered as a Windows service, you can check the status of the service.

### Using the administrative console

To check the status of a managed server using the deployment manager's administrative console, the node agent must be started. To use the administrative console, do the following steps:

1. Select **Servers** → **Server Types** → **WebSphere application servers**.
2. The servers are listed. The last column to the right contains an icon to indicate the status of each server. Figure 9 shows the icons and the corresponding status.

|   |                          |   |
|---|--------------------------|---|
|    | <b>Started</b>           | The server is running.  |
|    | <b>Partially stopped</b> | The server is in the process of changing from a started state to a stopped state. |
|  | <b>Stopped</b>           | The server is not running.  |
|  | <b>Unknown</b>           | The server status cannot be determined.   |

Figure 9 Status icons

**Note:** If the server status is Unknown, the node agent on the node in which the application server is installed is not active. The server cannot be managed from the administrative console unless its node agent is active.

### Using the `serverStatus` command

The syntax of the `serverStatus` command in Example 5.

*Example 5 serverStatus options*

```
Usage: serverStatus <server name | -all>
[-logfile <filename>]
```

```
[-replaceLog]
[-trace]
[-username <username>]
[-password <password>]
[-profileName <profile>]
[-help]
```

---

The first argument is mandatory. The argument is either the name of the server for which status is desired, or the `-all` keyword, which requests status for all servers defined on the node.

If you have administrative security enabled, you will need to enter a user ID and password of an administrator ID. If you do not include it in the command, you will be prompted for it.

For example, to view the status of a server, type:

```
cd profile_home/bin
serverStatus.sh server_name -username adminID -password adminpw
```

To check the status of all servers on the node, type:

```
cd profile_home/bin
serverStatus.sh -all -username adminID -password adminpw
```

For more information about this command, see:

► *serverStatus command*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml\\_serverstatus.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml_serverstatus.html)

Example 6 shows an example of using the `serverStatus` command.

*Example 6 serverStatus example - Windows operating system*

---

```
C:\WebSphereV7\AppServer\profiles\node40a\bin>serverstatus -all
-username admin -password adminpw
ADMU0116I: Tool information is being logged in file

C:\WebSphereV7\AppServer\profiles\node40a\logs\serverStatus.log
ADMU0128I: Starting tool with the node40a profile
ADMU0503I: Retrieving server status for all servers
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: nodeagent
ADMU0506I: Server name: server40a1
ADMU0508I: The Node Agent "nodeagent" is STARTED
```

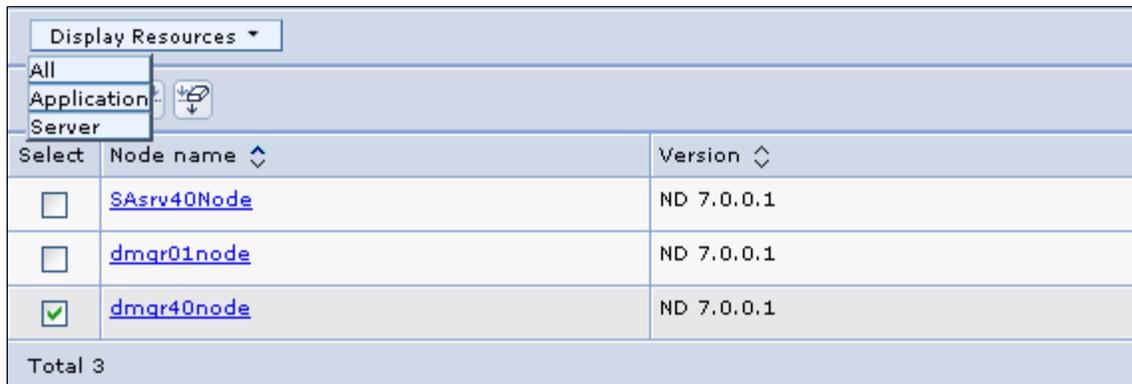
ADMU0509I: The Application Server "server40a1" cannot be reached. It appears to be stopped.

---

### From the job manager console

To display the servers and their status from the job manager console:

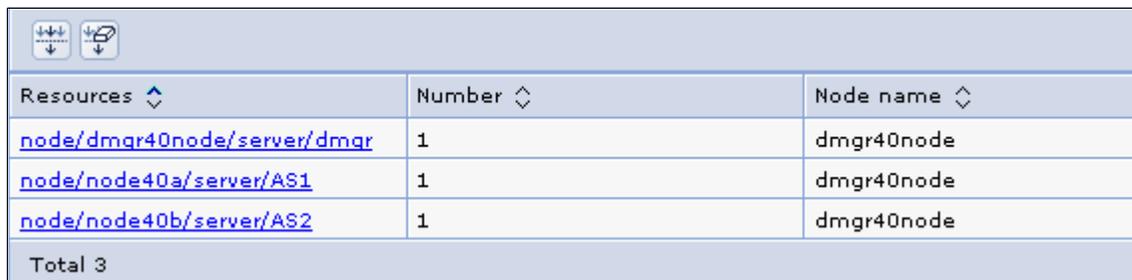
1. Select **Jobs** → **nodes**.
2. Click the box to the left of the node name. In the **Display resources** drop-down box, select **Server** (Figure 10).



| Display Resources ▾                 |                             |            |
|-------------------------------------|-----------------------------|------------|
| All                                 |                             |            |
| Application                         |                             |            |
| Server                              |                             |            |
| Select                              | Node name                   | Version    |
| <input type="checkbox"/>            | <a href="#">SAsrv40Node</a> | ND 7.0.0.1 |
| <input type="checkbox"/>            | <a href="#">dmqr01node</a>  | ND 7.0.0.1 |
| <input checked="" type="checkbox"/> | <a href="#">dmqr40node</a>  | ND 7.0.0.1 |
| Total 3                             |                             |            |

Figure 10 Display the servers on a node

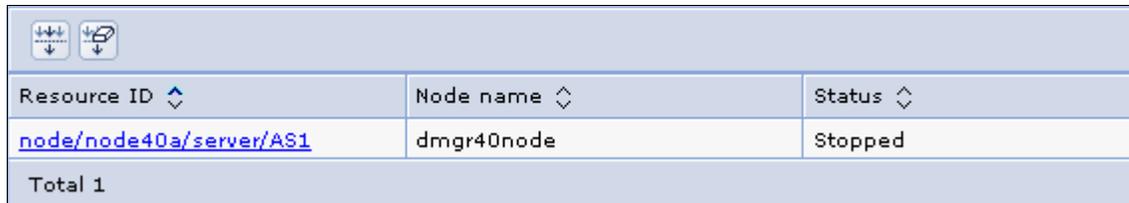
3. This action displays the list of servers in the node. Click the name of the server (Figure 11).



| Resources                                   | Number | Node name  |
|---|--------|------------|
| <a href="#">node/dmqr40node/server/dmqr</a> | 1      | dmqr40node |
| <a href="#">node/node40a/server/AS1</a>     | 1      | dmqr40node |
| <a href="#">node/node40b/server/AS2</a>     | 1      | dmqr40node |
| Total 3                                     |        |            |

Figure 11 Display the servers on a node

The status of the server is displayed (Figure 12).



| Resource ID                             | Node name  | Status  |
|---|------------|---------|
| <a href="#">node/node40a/server/AS1</a> | dmgr40node | Stopped |
| Total 1                                 |            |         |

Figure 12 Display the servers on a node

## Starting an application server

How you start an application server depends largely on personal preference and on whether the application server is standalone or managed. This section discusses how to start individual application servers. You can also start all application servers in a cluster by starting the cluster (see “Managing clusters” on page 61).

### Using the administrative console to start a managed server

**Tip:** Before managing a server in a distributed server environment using the administrative console, the node agent for the server’s node must be running. To check the status of the node:

1. Select **System Administration** → **Node Agents**.
2. The status of the node agent is in the far right column. If it is not started, you must start it (see “Starting and stopping nodes” on page 38).

From the administrative console, do the following steps:

1. Select **Servers** → **Server Types** → **WebSphere application servers**.
2. Check the box to the left of each server you want to start.
3. Click **Start**.
4. Verify the results in the Server status feedback window.

If there are any errors, check the log files for the application server process. The default location for the logs is:

- ▶ `profile_home/logs/server_name/SystemOut.log`
- ▶ `profile_home/logs/server_name/startServer.log`

On z/OS, check the output in application server job log.

**Tip:** By default, all the applications on a server start when the application server starts. To prevent an application from starting, see “Preventing an enterprise application from starting on a server” on page 80.

## Using the startServer command

The syntax of the **startServer** command is shown in Example 7.

### Example 7 startServer options

---

```
Usage: startServer <server> [options]
      options: -nowait
               -quiet
               -logfile <filename>
               -replacelog
               -trace
               -script [<script filename>] [-background]
               -timeout <seconds>
               -statusport <portnumber>
               -profileName <profile>
               -recovery
               -help
```

---

<server> is the name of the server to be started. The first argument is mandatory and case sensitive.

See the following article for more information about this command.

► *startServer command*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml\\_startserver.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml_startserver.html)

### **startServer example**

Example 8 shows an example of using the **startServer** command. Note that the user ID and password are not required to start the server.

### Example 8 startServer example

---

```
C:\WebSphereV7\AppServer\profiles\node40a\bin>startserver server40a1
ADMU0116I: Tool information is being logged in file

C:\WebSphereV7\AppServer\profiles\node40a\logs\server40a1\startServer
.log
ADMU0128I: Starting tool with the node40a profile
ADMU3100I: Reading configuration for server: server40a1
```

```
ADMU3200I: Server launched. Waiting for initialization status.  
ADMU3000I: Server server40a1 open for e-business; process id is 3928
```

---

## Starting a server from the job manager

To start an application server from the job manager, make the following selections to submit the job:

1. Step 1: Select the **Start server** job type.
2. Step2: Select the job target.
  - If you are starting an application server on an unfederated node, select the application server node.
  - If you are starting a new managed server, select the deployment manager node.

Enter the user ID and password with administrative authority on the target node.
3. Step 3: Specify the job parameters.
  - Specify the name of the new application server.
  - If the target is a deployment manager, enter the name of the node that the server will be created on.

## Stopping an application server

This section shows multiple methods for stopping a server.

### Using the administrative console to stop a managed server

**Note:** These directions assume that the node agent for the application server is running.

From the administrative console, you have the following options to stop an application server:

- ▶ The **Stop** button quiesces the application server and stops it. In-flight requests are allowed to complete.
- ▶ The **Immediate Stop** button stops the server, but bypasses the normal server quiesce process that supports in-flight requests to complete before shutting down the entire server process. This shutdown mode is faster than the normal server stop processing, but some application clients can receive exceptions.

- ▶ The **Terminate** button deletes the application server process. Use this if immediate stop fails to stop the server.
- ▶ The **Restart** button stops, then starts the server.

From the administrative console, do these steps to stop an application server:

1. Select **Servers** → **Server Types** → **WebSphere application servers**.
2. Check the box to the left of each server you want to stop.
3. Click the appropriate stop option.

If there are any errors, check the log files for the application server process:

- ▶ *profile\_home/logs/server\_name/SystemOut.log*
- ▶ *profile\_home/logs/server\_name/stopServer.log*

On z/OS, check the output in application server job log.

### Restarting all servers on a node

If you want to stop, and then restart, all the application servers on a node, you can do the following from the administrative console:

1. Select **System Administration** → **Node Agents**.
2. Check the box to the left of the node agent.
3. Click **Restart all Servers on the Node**.

### Stopping all servers in a cluster

If you want to stop all the servers in a cluster, you can do the following from the administrative console:

1. Select **Servers** → **Clusters** → **WebSphere application server clusters**.
2. Check the box to the left of the cluster.
3. Click **Stop** or **Immediate Stop**.

### Restarting all servers in a cluster

If you want to stop, and then restart, all the servers in a cluster, you can do the following from the administrative console:

1. Select **Servers** → **Clusters** → **WebSphere application server clusters**.
2. Check the box to the left of the cluster.
3. Click **Ripplestart**.

### Using the stopServer command

The syntax of the **stopServer** command is shown in Example 9 on page 26.

### Example 9 stopServer command

---

```
Usage: stopServer <server> [options]
      options: -nowait
              -quiet
              -logfile <filename>
              -replacelog
              -trace
              -timeout <seconds>
              -statusport <portnumber>
              -conntype <connector type>
              -port <portnumber>
              -username <username>
              -password <password>
              -profileName <profile>
              -help
```

---

<server> is the name of the server to be started. The first argument is mandatory and is case sensitive.

See the following article for a list of available options.

► *stopServer command*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml\\_stopserver.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rxml_stopserver.html)

Example 10 shows an example of the **stopServer** command.

### Example 10 stopServer command example

---

```
C:\WebSphereV7\AppServer\profiles\node40a\bin>stopserver server40a1
-username admin -password adminpw
ADMU0116I: Tool information is being logged in file

C:\WebSphereV7\AppServer\profiles\node40a\logs\server40a1\stopServer.
log
ADMU0128I: Starting tool with the node40a profile
ADMU3100I: Reading configuration for server: server40a1
ADMU3201I: Server stop request issued. Waiting for stop status.
ADMU4000I: Server server40a1 stop completed.
```

---

**Note:** If you attempt to stop a server and for some reason it does not stop, you can use the operating system commands to stop the Java™ process for the server.

## Viewing runtime attributes of an application server

To view runtime attributes, do the following steps:

1. Select **Servers** → **Server Types** → **WebSphere application servers** to display the list of servers.
2. Click the server name to access the detail page.
3. If the server is running, you will see both a Configuration tab and Runtime tab. If it is not running, you will see only a Configuration tab. Click the **Runtime** tab. Figure 13 shows the Runtime tab and the information that it provides.

**Application servers** > **server40a1**

Use this page to configure an application server. An application server is a server that provides services required to run enterprise applications.

Runtime **Configuration**

---

**General Properties**

Process ID  
4836

Cell name  
Cell40

Node name  
node40a

State  
Started

[Back](#)

**Server messaging**

- [Messaging engines](#)

**Troubleshooting**

- ⊕ [Diagnostic Provider service](#)

**Additional Properties**

- [Transaction Service](#)
- [Performance Monitoring Infrastructure \(PMI\)](#)
- [Product Information](#)

Figure 13 Application server Runtime tab

4. From the Runtime tab, you have:
  - Access to a list of messaging engines that run on this application server. There will be one messaging engine for each bus that the server is a member of. You can start and stop the messaging engine from this panel.
  - Access to the Diagnostic Provider service, allowing you to query current configuration data, state, and to initiate diagnostic tests.
  - Access to the Transaction Service properties settings. You can change the timeout settings while the server is running, but not the transaction log directory setting. Changes made in this panel are active until the server is stopped.

To make these settings permanent and to configure additional Transaction Service settings, click the Configuration tab while you have this page open. This takes you directly to the Transaction Service settings.

You can also view or act on transactions in the following states by clicking **Review** to the right of the state. This action is not normally necessary, but in an exceptional situation, it might be useful:

- Manual transactions:

These transactions await administrative completion. For each transaction, the local or global ID is displayed. You can display each transaction resource and its associated resource manager. You can choose also to commit or rollback transactions in this state.

- Retry transactions:

These are transactions with some resources being retried. For each transaction, the local or global ID is displayed, and whether the transaction is committing or rolling back. You can display each transaction resource and its associated resource manager. You can choose also to finish, or abandon retrying, transactions in this state.

- Heuristic transactions:

These are transactions that have completed heuristically. For each transaction, the local or global ID and the heuristic outcome is displayed. You can display each transaction resource and its associated resource manager. You can also choose to clear the transaction from the list.

- Imported prepared transactions:

Transactions that have been imported and prepared but not yet committed. For each transaction, the local or global ID is displayed. You can display each transaction resource and its associated resource manager. You can also choose to commit or rollback transactions in this state.

- Performance Monitoring Service settings allow you to change the instrumentation levels while the server is running or make permanent changes to the configuration for that server.
- Product Information gives you access to extensive information about the product installation and Fix Pack information.

## Customizing application servers

When you create a new application server, it inherits most of its configuration settings from the specified template server. To view or modify these settings,

select **Servers** → **Server Types** → **WebSphere application servers**. A list of application servers defined in the cell appears in the workspace. Click the name of the application server to make a modification.

This section gives you a quick overview of the types of settings that you can customize. See Figure 14 (not all settings are shown due to the size of the configuration window).

The screenshot shows a configuration window with two main sections: **General Properties** and **Container Settings**.  
**General Properties** includes:

- Name: server40a1
- Node name: node40a
- Run in development mode:
- Parallel start:
- Start components as needed:
- Access to internal server classes: Allow (dropdown)
- Server-specific Application Settings:
  - Classloader policy: Multiple (dropdown)
  - Class loading mode: Classes loaded with parent class loader first (dropdown)
- Buttons: Apply, OK, Reset, Cancel

**Container Settings** includes:

- Session management
- SIP Container Settings
- Web Container Settings
- Portlet Container Settings
- EJB Container Settings
- Container Services
- Business Process Services

**Applications** includes:

- Installed applications

**Server messaging** includes:

- Messaging engines
- Messaging engine inbound transports
- WebSphere MQ link inbound transports
- SIB service

**Server Infrastructure** is listed at the bottom.

Figure 14 Application server configuration

## General properties

The general properties consist of a few items that you can see immediately:

- ▶ Run in development mode: Enable this option to streamline the startup time of an application server. Do not enable this setting on production servers.
- ▶ Parallel start: Select this field to start the server components, services, and applications on multiple threads. This might shorten the startup time.

The order in which the applications start depends on the weights you assigned to each of them. Applications that have the same weight are started in parallel.

To set the weight of an application, in the administrative console, click **Applications** → **Application Types** → **WebSphere enterprise applications** → *application\_name* → **Startup behavior**, and then specify an appropriate value in the **Startup order** field.

- ▶ Start components as needed: (**New in V7**) Select this field to start the application server components as they are needed. This might shorten the start time.
- ▶ Access to internal server classes: Specifies whether the applications can access many of the server implementation classes.
- ▶ Application classloader policy and class loading mode: These settings allow you to define an application server-specific classloader policy and class loading mode. Class loaders are discussed in Chapter 13, *Understanding class loaders*.

## Container settings

Each application server has containers that run specific application components. This section in the configuration page for the server provides links to pages where you can modify the settings for the containers.

**Tip:** Modifying container settings is not something you would normally do on a daily basis. The most commonly used settings in these sections are discussed throughout this book in the appropriate topics. For now, it is enough to know that each container has settings that allow you to modify its configuration and how to find those settings.

### ***SIP container settings***

Session Initiation Protocol (SIP) support extends the application server to allow it to run SIP applications written to the JSR 116 specification. SIP is used to establish, modify, and terminate multimedia IP sessions including IP telephony, presence, and instant messaging. If you have SIP applications, review these settings.

### ***Web container settings***

The Web container serves application requests for servlets and JSPs. The Web container settings allow you to specify the default virtual host, enable servlet caching, specify session manager settings such as persistence and tuning parameters, and HTTP transport properties. See Figure 15.

**(New in V7)** The Asynchronous Request Dispatcher (ARD) enables servlets and JSPs to make standard include calls concurrently on separate threads. Selecting this link allows you to enable ARD and configure related settings.

**General Properties**

Default virtual host:  
default\_host ▼

Enable servlet caching

Disable servlet request and response pooling

Apply OK Reset Cancel

**Additional Properties**

- [Asynchronous Request Dispatching](#)
- [Custom properties](#)
- [Web container transport chains](#)
- [Session management](#)

Figure 15 Web container settings

### ***Portlet container services***

The portlet container is the runtime environment for portlets using the JSR 168 Portlet Specification. Portlets based on this JSR 168 Portlet Specification are referred to as standard portlets. You can use these settings to enable portlet fragment caching to save the output of portlets to the dynamic cache.

### ***EJB container properties***

These properties allow you configure the services provided by the EJB™ container. This includes setting the passivation directory path, EJB cache and timer service settings, pool cleanup interval, a default data source JNDI name, and to enable stateful session bean failover using memory-to-memory replication. See Figure 16.

| General Properties  | Additional Properties  |
|---|--|
| * Passivation directory<br><input type="text" value="\${USER_INSTALL_ROOT}/temp"/>  | <ul style="list-style-type: none"> <li>■ <a href="#">EJB cache settings</a></li> <li>■ <a href="#">EJB timer service settings</a></li> </ul> |
| Inactive pool cleanup interval<br><input type="text" value="30000"/> milliseconds   |  |
| Default data source JNDI name<br><input type="text" value="(none)"/>  |  |
| <input type="checkbox"/> Enable stateful session bean failover using <a href="#">memory-to-memory replication</a> (No replication domains have been defined.) |  |
| <input type="button" value="Apply"/> <input type="button" value="OK"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>             |  |

Figure 16 EJB container settings

### **Container services**

The following settings are available under the container services section:

- ▶ **Application profiling service:** Application profiling is a WebSphere extension that used along with access intents allows you to define strategies to dynamically control concurrency, prefetch, and read-ahead. The container services settings allows you to enable this service and to set the compatibility mode for J2EE™ 1.3 applications.
- ▶ **Transaction service:** The transaction service properties allow you to specify settings for the transaction service, as well as manage active transaction locks. The settings include the directory location for the transaction service on the application server to store log files for recovery, the total transaction lifetime timeout, and client inactivity timeout.

When the application server is running, a Runtime tab is available in the Transaction Service properties workspace. From here, you can manage running transactions and modify timeout settings at runtime.

- ▶ **Dynamic cache service:** This page allows you to specify settings for the dynamic cache service of this server.
- ▶ **(New in V7)** The Java Persistence API (JPA) default settings: JPA provides a mechanism for managing persistence and object-relational mapping and functions for the EJB 3.0 specifications. This page allows you to configure default settings for JPA. JPA settings in an application will override these settings.
- ▶ **Compensation service:** The compensation service supports server-level configuration for compensation enablement and logging. This service is not

started automatically. If you will be running applications that require this service you must enable it here.

- ▶ Internationalization service: This service enables you to configure and manage an internationalization context for an application for which components are distributed across the enterprise. This section of the configuration panel allows you to enable this service. It is not enabled by default.
- ▶ Object pool service: The object pool service manages object pool resources that are used by the application server. This section of the configuration panel allows you to disable this service (it is enabled by default).
- ▶ Startup beans service: Startup beans are session beans that run business logic through the invocation of start and stop methods when applications start and stop. This section of the configuration panel allows you to enable this service (it is disabled by default).
- ▶ ORB service settings: These settings allow you to specify settings for the Object Request Broker service. These include request timeout, thread settings, and connection cache minimum and maximum.

### ***Business process services***

The business process settings allow you to manage the following features

- ▶ Activity session service
- ▶ Work area partition service
- ▶ Work area service

### **Applications**

Use the **Installed Applications** link to view the applications installed on this server. This will display the collection of applications as links to the configuration page for each application.

### **Server messaging**

The server messaging settings provide configuration settings and information for the messaging services.

### **Server infrastructure**

The server infrastructure settings include settings for Java and process management and administration services.

- ▶ Java and Process Management:
  - Class loader: Create and configure class loader instances. Class loaders are discussed in Chapter 13, *Understanding class loaders*.

- Process definition: These settings define runtime properties, such as the program to run, arguments to run the program, and the working directory. Within the process definitions, you will find the JVM definitions, such as the initial and maximum heap sizes, debug options, the process classpath, or different runtime options, such as profiler support and heap size.
- Process execution: These include settings such as the process priority, or the user and group that should be used to run the process. These settings are not applicable on the Windows platform.
- Monitoring policy: These properties determine how the node agent will monitor the application server. It includes ping intervals, timeouts, and an initial state setting. These can be used to ensure that the server is started when the node starts and will be restarted in the event of a failure.
- ▶ Administration:
  - Custom properties: Specifies additional custom properties for this component.
  - Administration services: This group of settings allows you to specify various settings for administration facility for this server, such as administrative communication protocol settings and timeouts. (These settings are not something you would normally be concerned with.)
  - Server components: Creates an additional runtime components that are configurable.
  - Custom Services: Creates a custom service classes that run within this server and their configuration properties.

If you plan to extend the administration services by adding custom MBeans, see the topic, *Extending WebSphere Application Server Administrative System with custom MBeans*, in the Information Center.

## Performance

These settings allow you to specify settings for the Performance Monitoring Infrastructure (PMI) and the Performance and Diagnostic Advisor framework.

## Communications

Here we describe the communications settings:

- ▶ Ports:
  - These settings contain the basic port definitions for the server. See Figure 17 on page 36.

**Tip:** You might not ever need to manually change these ports. It is likely, however, that you will want to view these.

For example, if you use the `dumpNameSpace` command, you can specify the bootstrap port of the process to dump the name space from. When you federate a node, you will need to know the SOAP connector port of the node or deployment manager. And the inbound communications ports are essential for accessing applications and the administrative console.

Clicking the **Details** button takes you to a page that has links for the configurable port settings.

Some port settings will be defined to use the channel framework. These will have an associated transport chain that represents the network protocol stack. A transport chain consists of one or more types of channels, each of which supports a different type of I/O protocol, such as TCP or HTTP. Network ports can be shared among all of the channels within a chain. The Channel Framework function automatically distributes a request arriving on that port to the correct I/O protocol channel for processing.

Ports

| Port Name                             | Port | Details |
|---------------------------------------|------|---------|
| BOOTSTRAP_ADDRESS                     | 9813 |         |
| SOAP_CONNECTOR_ADDRESS                | 8880 |         |
| ORB_LISTENER_ADDRESS                  | 9100 |         |
| SAS_SSL_SERVERAUTH_LISTENER_ADDRESS   | 9401 |         |
| CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS | 9403 |         |
| CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS | 9402 |         |
| WC_adminhost                          | 9060 |         |
| WC_defaulthost                        | 9080 |         |
| DCS_UNICAST_ADDRESS                   | 9353 |         |
| WC_adminhost_secure                   | 9043 |         |
| WC_defaulthost_secure                 | 9443 |         |
| SIP_DEFAULTHOST                       | 5060 |         |
| SIP_DEFAULTHOST_SECURE                | 5061 |         |
| SIB_ENDPOINT_ADDRESS                  | 7276 |         |
| SIB_ENDPOINT_SECURE_ADDRESS           | 7286 |         |
| SIB_MQ_ENDPOINT_ADDRESS               | 5558 |         |
| SIB_MQ_ENDPOINT_SECURE_ADDRESS        | 5578 |         |
| IPC_CONNECTOR_ADDRESS                 | 9637 |         |

Figure 17 Viewing application server ports

► Message listener service:

The message listener service provides support for WebSphere Application Server V5 message-driven beans applications.

### Security

Security settings for the application server allow you to set specific settings at the server level. Security settings are covered in *WebSphere Application Security V7 Security Handbook*, SG24-7660.

### Troubleshooting

These settings include those for logging and tracing. For information about troubleshooting and using these settings, see *WebSphere Application Server V6: Diagnostic Data*, REDP-4085.

## Additional properties

The following settings are defined under the additional properties section:

- ▶ Class loader viewer service: This is used to enable or disable service to keep track of classes loaded.
- ▶ Core group service: These settings are related to high availability.
- ▶ Endpoint listeners: An endpoint listener receives requests from service requester applications within a specific application server or cluster.
- ▶ Debugging service: On this page, you can specify settings for the debugging service, to be used in conjunction with a workspace debugging client application, for example, the Application Server Toolkit.
- ▶ Thread pool: The thread pool specifies the possible maximum number of concurrently running threads in the Web container. As one thread is needed for every client request, this directly relates to the number of active clients that can possibly access the Web container on this application server at any given time. A timeout value can be specified for the application server to remove threads from the pool based on a timed period of inactivity.

Finally, an option for creating threads beyond the maximum pool size is available. Be careful when using this option. It can have the unexpected effect of allowing the Web container to create more threads than the JVM might be able to process, creating a resource shortage and bringing the application server to a halt.

- ▶ **(New in V7)** Reliable messaging state: This link allows you to view and manage the WS-ReliableMessaging runtime state.
- ▶ Web server plug-in properties: This is used to change the HTTP plug-in configuration without having to stop the server and start it again.

## Working with nodes in a distributed environment

Managing nodes is a concept specific to a Network Deployment environment. Nodes are managed by the deployment manager through a process known as a *node agent* that resides on each node. In order to manage a node in a Network Deployment environment, the node must be defined and the node agent on each WebSphere Application Server node must be started.

Nodes are created when you create a profile. Nodes are added to a cell through federation (see Chapter 2, *Working with profiles on distributed systems*).

## Starting and stopping nodes

A node consists of the node agent and the servers. There are several ways to start and stop a node and node agent, or stop them individually. Before using any of these methods, be sure to note whether it affects the entire node, including servers, or just the node agent.

### Starting a node agent

When a node agent is stopped, the deployment manager has no way to communicate with it. Therefore, the node agent has to be started with the **startNode** command run from on the profile node system.

#### ***startNode*** command

The command syntax is shown in Example 11.

##### *Example 11 startNode command*

---

```
Usage: startNode [options]
      options: -nowait
               -quiet
               -logfile <filename>
               -replacelog
               -trace
               -script [<script filename>] [-background]
               -timeout <seconds>
               -statusport <portnumber>
               -profileName <profile>
               -recovery
               -help
```

---

For information about the use of the startNode command, see:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml\\_startnode.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml_startnode.html)

See Example 12 for an example of the **startNode** command. Note that a user ID and password is not required.

##### *Example 12 startNode command*

---

```
C:\WebSphereV7\AppServer\profiles\node40c\bin>startnode
ADMU0116I: Tool information is being logged in file

C:\WebSphereV7\AppServer\profiles\node40c\logs\nodeagent\startServer.
log
ADMU0128I: Starting tool with the node40c profile
```

```
ADMU3100I: Reading configuration for server: nodeagent
ADMU3200I: Server launched. Waiting for initialization status.
ADMU3000I: Server nodeagent open for e-business; process id is 7480
```

---

## Starting a node on z/OS using the START command

To start a node agent on z/OS using the START command, use the following format:

```
START nodeagent_procname,JOBNAME=server_shortname,
ENV=cell_shortname.node_shortname.server_shortname
```

For example:

```
START WPACRA,JOBNAME=WPAGNTA,ENV=WPCELL.WPNODEA.WPAGNTA
```

## Stopping a node agent

To stop the node agent and leave the servers running, do the following actions, depending on your preferred method.

From the administrative console, do the following steps:

1. From the administrative console, select **System Administration** → **Node Agents**.
2. Check the box beside the node agent for the server and click **Stop**.

Or, from a command prompt, use the **stopNode** command.

**Note:** After you stop the node agent, the deployment manager has no way to communicate with the servers on that node. The servers might be up and running, but the administrative console is not able to determine their status.

### ***stopNode command***

The command syntax is shown in Example 13.

#### *Example 13 stopNode command*

---

```
Usage: stopNode [options]
      options: -nowait
               -stopservers [-saveNodeState]
               -quiet
               -logfile <filename>
               -replacelog
               -trace
               -timeout <seconds>
               -statusport <portnumber>
```

```
-conntype <connector type>
-port <portnumber>
-username <username>
-password <password>
-profileName <profile>
-help
```

---

For information about the stopNode command and its options, see:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml\\_stopnode.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml_stopnode.html)

See Example 14 for an example and sample output of the **stopNode** command.

*Example 14 stopNode command*

---

```
C:\WebSphereV7\AppServer\profiles\node40c\bin>stopnode -username admin
-password adminpw
```

```
ADMU0116I: Tool information is being logged in file
```

```
C:\WebSphereV7\AppServer\profiles\node40c\logs\nodeagent\stopServer.log
```

```
ADMU0128I: Starting tool with the node40c profile
```

```
ADMU3100I: Reading configuration for server: nodeagent
```

```
ADMU3201I: Server stop request issued. Waiting for stop status.
```

```
ADMU4000I: Server nodeagent stop
completed.
```

---

## Stopping a node on z/OS using the STOP command

To stop a node agent on z/OS, you can use the following command:

```
STOP nodeagent_JOBNAME
```

For example:

```
STOP WPAGNTA
```

## Stopping a node (the node agent and servers)

You can use the administrative console to stop a node and its servers with one action. Follow these steps:

1. From the administrative console, select **System Administration** → **Nodes**.
2. Check the box beside the node and click **Stop**.

## Restarting a node agent

You can restart a running node agent from the administrative console by doing the following steps from the administrative console:

1. Select **System Administration** → **Node Agents**.
2. Check the box beside the node agent for the server and click **Restart**.

## Node agent synchronization

During a synchronization operation, a node agent checks with the deployment manager to see if any configuration documents that apply to the node have been updated. New or updated documents are copied to the node repository, and deleted documents are removed from the node repository.

### Automatic synchronization

Automatic configuration synchronization between the node and the deployment manager is enabled by default. You can configure the interval between synchronizations in the administrative console by doing the following steps:

1. Expand **System Administration** → **Node Agents** in the administrative console.
2. Select the node agent process on the appropriate server to open the Properties page.
3. In the Additional Properties section, click **File Synchronization Service**.
4. Configure the synchronization interval. By default, the synchronization interval is set to one minute.

**Tip:** Increase the synchronization interval in a production environment to reduce the overhead.

Note that a separate setting exists as an administrative console preference. The Synchronize changes with Nodes option, when selected, indicates that any time a change is saved to the console, it is automatically synchronized out to the running nodes. To set this preference, select **System administration** → **Console Preferences**.

### Forced synchronization

Synchronization can be forced by selecting **System Administration** → **Nodes**. Select a node and click **Synchronize** or **Full Synchronization**.

- ▶ Synchronize performs an immediate synchronization on the selected node. This type of synchronization is optimized for performance and only

synchronizes changed files. If there are issues with manually edited files, this might not result in a complete synchronization.

- ▶ The Full Synchronization option disregards optimization and ensures that the node and cell configuration are identical.

### **Using the `syncNode` command**

The `syncNode` command can be used from the node to force the synchronization of a node's local configuration repository with the master repository on the deployment manager node.

**Tip:** The `syncNode` command is normally only used in exception situations. To use the `syncNode` command, the node agent must be stopped. You can use the `-stopservers` and `-restart` options on the `syncNode` command to stop the node agent and application servers, and then restart the node agent.

The syntax of the `syncNode` command is shown in Example 15.

#### *Example 15 syncNode command*

---

```
Usage: syncNode dmgr_host [dmgr_port] [-conntype <type>] [-stopservers]
[-restart] [-quiet] [-nowait] [-logfile <filename>] [-replacelog]
[-trace] [-username <username>] [-password <password>]
[-localusername <localusername>] [-localpassword <localpassword>]
[-profileName <profile>] [-help]
```

---

You can find information about these options at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml\\_syncnode.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml_syncnode.html)

Example 16 shows an example of using the `syncNode` command. The command is executed from the node. The `-stopservers` and `-restart` options are used to stop all the servers on the node, including the node agent, then restart the node agent after the synchronization.

#### *Example 16 syncNode usage examples*

---

```
C:\WebSphereV7\AppServer\profiles\node40c\bin>syncNode T60 8882
-stopsservers -re start -username admin -password adminpw
ADMU0116I: Tool information is being logged in file
          C:\WebSphereV7\AppServer\profiles\node40c\logs\syncNode.log
ADMU0128I: Starting tool with the node40c profile
ADMU0401I: Begin syncNode operation for node node40c with Deployment
Manager T60: 8882
ADMU0505I: Servers found in configuration:
```

```
ADMU0506I: Server name: nodeagent
ADMU0506I: Server name: server40ca
ADMU2010I: Stopping all server processes for node node40c
ADMU0512I: Server server40ca cannot be reached. It appears to be
stopped.
ADMU0512I: Server nodeagent cannot be reached. It appears to be
stopped.
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0018I: Launching Node Agent process for node: node40c
ADMU0020I: Reading configuration for Node Agent process: nodeagent
ADMU0022I: Node Agent launched. Waiting for initialization status.
ADMU0030I: Node Agent initialization completed successfully. Process id
is: 4924
ADMU0402I: The configuration for node node40c has been synchronized
with Deployment Manager T60: 8882
```

---

## Removing a node from a cell

There are two ways of removing a node from a network distributed administration cell.

**Note:** When a node is removed, it is restored to its original configuration.

### Using the administrative console

From the administrative console, do the following steps:

1. Select **System Administration** → **Nodes**.
2. Place a check mark in the check box beside the node you want to remove and click **Remove Node**.

This method runs the **removeNode** command in the background.

### Using the removeNode command

The **removeNode** command detaches a node from a cell and returns it to a standalone configuration.

The **-removeNode** command syntax is shown in Example 17.

*Example 17 removeNode command*

---

```
Usage: removeNode [-force] [-quiet] [-nowait] [-logfile <filename>]
[-replaceLog] [-trace] [-username <username>] [-password <password>]
```

`[-profileName <profile>][-help]`

---

For more information, see:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml\\_removeNode.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml_removeNode.html)

To use the command, do the following steps:

1. Change the directory to the *profile\_root/bin* directory.
2. Run **removeNode**. All parameters are optional for this command.

In a distributed environment on z/OS, the **removeNode.sh** command is in the *install\_root/bin* directory. You will need to specify the `-profileName` parameter to specify the profile for the node you want to remove.

The command performs the following operations:

1. Connects to the deployment manager process to read the configuration data.
2. Stops all of the running server processes of the node, including the node agent process.
3. Removes servers in the node from clusters.
4. Restores the original standalone node configuration. This original configuration was backed up when the node was originally added to the cell.
5. Removes the node's configuration from the master repository of the cell. The local copy of the repository held on each node will get updated at the next synchronization point for each node agent. Although the complete set of configuration files are not pushed out to other nodes, some directories and files are pushed out to all nodes.
6. Removes installed applications from application servers in the cell that are part of the node being removed.
7. Copies the original application server cell configuration into the active configuration.

The command provides the `-force` option to force the local node's configuration to be decoupled from the cell even if the deployment manager cannot be contacted. However, if this situation occurs, the cell's master repository will then have to be separately updated to reflect the node's removal, for example, through manual editing of the master repository configuration files.

## Example

Example 18 shows an example of using the **removeNode** command.

### *Example 18 removeNode example*

---

```
C:\WebSphereV7\AppServer\bin>removeNode -profileName node40c -username  
admin -password admin
```

```
ADMU0116I: Tool information is being logged in file
```

```
C:\WebSphereV7\AppServer\profiles\node40c\logs\removeNode.log
```

```
ADMU0128I: Starting tool with the node40c profile
```

```
ADMU2001I: Begin removal of node: node40c
```

```
ADMU0009I: Successfully connected to Deployment Manager Server:  
t60:8882
```

```
ADMU0505I: Servers found in configuration:
```

```
ADMU0506I: Server name: nodeagent
```

```
ADMU0506I: Server name: server40ca
```

```
ADMU2010I: Stopping all server processes for node node40c
```

```
ADMU0512I: Server server40ca cannot be reached. It appears to be  
stopped.
```

```
ADMU0510I: Server nodeagent is now STOPPED
```

```
ADMU2021I: Removing all servers on this node from all clusters in the  
cell.
```

```
ADMU2014I: Restoring original configuration.
```

```
ADMU2017I: The local original configuration has been restored.
```

```
ADMU0306I: Note:
```

```
ADMU2031I: Any applications that were uploaded to the Cell40 cell  
configuration during addNode using the -includeapps option are not  
uninstalled by removeNode.
```

```
ADMU0307I: You might want to:
```

```
ADMU2032I: Use wsadmin or the Administrative Console to uninstall any  
such applications from the Deployment Manager.
```

```
ADMU0306I: Note:
```

```
ADMU2033I: Any buses that were uploaded to the Cell40 cell  
configuration during addNode using the -includebuses option are not  
uninstalled by removeNode.
```

```
ADMU0307I: You might want to:
```

```
ADMU2034I: Use wsadmin or the Administrative Console to uninstall any  
such buses from the Deployment Manager.
```

```
ADMU2024I: Removal of node node40c is complete.
```

---

## Renaming a node

The **renameNode** command allows you to modify the node name of a federated server.

### renameNode command

The command syntax is shown in Example 19.

*Example 19 renameNode command syntax*

---

```
Usage: renameNode dmgr_host dmgr_port node_name [-nodeshortname <name>]
[-conntype <type>] [-logfile <filename>] [-trace]
[-username <username>] [-password <password>] [-help]
```

---

For more information about the **renameNode** command, see:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml\\_renamenode.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml_renamenode.html)

To run the command, do the following steps:

1. Change to the *profile\_root/bin* directory of the deployment manager.
2. Run the **renameNode** command.

The command:

1. Connects to the deployment manager.
2. Stops all servers.
3. Changes the node configuration on the deployment manager.
4. Synchronizes the node.

## Node groups

In a distributed environment, you can have nodes in a cell with different capabilities. However, there are restrictions on how the nodes can coexist.

Node groups are created to group nodes of similar capability together to allow validation during system administration processes. Effectively, this means that a node group establishes a boundary from which servers can be selected for a cluster. Nodes on distributed platforms and nodes on the IBM i platform can be members of the same node group, however, they cannot be members of a node group that contains a node on a z/OS platform.

**Note:** Do not confuse node groups with “groups of nodes” in the job manager. These are two different concepts.

A default node group called DefaultNodeGroup is automatically created for you when the deployment manager is created, based on the deployment manager platform. New nodes on similar platforms are automatically added to the default group. A node must belong to at least one node group, but can belong to more than one.

As long as you have nodes in a cell with similar platforms, you do not need to do anything with node groups. New nodes are automatically added to the node group. However, before adding a node on a platform that does not have the same capabilities as the deployment manager platform, you will need to create the new node group.

### Working with node groups

You can display the default node group and its members by selecting **System Administration** → **Node Groups**. See Figure 18.



| Select                   | Name                             | Members | Description                   |
|--------------------------|----------------------------------|---------|-------------------------------|
| <input type="checkbox"/> | <a href="#">DefaultNodeGroup</a> | 4       | WebSphere Default Node Group. |
| Total 1                  |                                  |         |                               |

Figure 18 Display a list of node groups

- ▶ To create a new node group, click **New**. The only thing that you need to enter is the name of the new node group. Click **OK**.
- ▶ To delete a node group, check the box to the left of the node group name and select **Delete**.
- ▶ To display a node group, click the node group name. For example, in Figure 19, we have displayed the DefaultNodeGroup.

**General Properties**

Name  
DefaultNodeGroup

Members  
2

Description  
WebSphere Default Node Group.

**Additional Properties**

- [Custom properties](#)
- [Node group members](#)

Apply OK Reset Cancel

Figure 19 Node group properties

- ▶ To add a node to a node group, display the node group and click **Node group members** in the Additional Properties section (Figure 19). When the list appears, click **Add**. You will be able to select from a list of nodes (Figure 20).

[Node groups](#) > [DefaultNodeGroup](#) > **Node group members**

Use this page to view and configure the nodes in this node group.

⊕ Preferences

Add Remove

⊞ ⊞ ⊞ ⊞

| Select                                      | Name                       |
|---|----------------------------|
| You can administer the following resources: |                            |
| <input type="checkbox"/>                    | <a href="#">dmqr40node</a> |
| <input type="checkbox"/>                    | <a href="#">node40a</a>    |
| <input type="checkbox"/>                    | <a href="#">node40b</a>    |
| <input type="checkbox"/>                    | <a href="#">node40c</a>    |

Figure 20 Displaying node group members

## Working with clusters

This section discusses creating, configuring, and managing application server clusters using the administrative console.

Clusters consist of one or more application servers that run the same applications. The configuration of each server can be unique.

Before creating a cluster, consider the number of servers you want to add to the cluster, the nodes that they will be created on, and how the workload should be distributed across the servers.

Work is distributed across the servers in the cluster based on weights assigned to each application server. If all cluster members have identical weights, work is distributed among the cluster members equally. Servers with higher weight values are given more work. A rule of thumb formula for determining routing preference would be:

$$\% \text{ routed to Server1} = \text{weight1} / (\text{weight}_1 + \text{weight}_2 + \dots + \text{weight}_n)$$

In the formula, n represents the number of cluster members in the cluster. Factors to consider are the capacity of the system that hosts the application server.

## Creating application server clusters

When you create a cluster, you have the option to create an empty cluster (no servers) or to create the cluster with one or more servers. The first application server added to the cluster acts as a template for subsequent servers. You can create the first server during the cluster creation process or you can convert an existing application server. The rest of the servers must be new and can be created when you create the cluster or added later.

**Tip:** When creating a cluster, it is possible to select the template of an existing application server for the cluster without adding that application server into the new cluster. If you need to change the attributes of the servers in your cluster after the cluster has been created, you must change each server individually. For this reason, consider creating an application server with the server properties that you want as a standard in the cluster first, then use that server as a template or as the first server in the cluster.

## Cluster and cluster member options

When you create a new cluster, you have the following options to consider:

- ▶ Prefer local:

This setting indicates that a request to an EJB should be routed to an EJB on the local node if available. This is the default setting and generally will result in better performance.

- ▶ Configure HTTP session memory-to-memory replication (create a replication domain):

WebSphere Application Server supports session replication to another WebSphere Application Server instance. In this mode, sessions can replicate to one or more WebSphere Application Server instances to address HTTP Session single point of failure.

When you create a cluster, you can elect whether to create a replication domain for the cluster. The replication domain is given the same name as the cluster and is configured with the default settings for a replication domain. When the default settings are in effect, a single replica is created for each piece of data and encryption is disabled. Also, the Web container for each cluster member is configured for memory-to-memory replication.

For more information about replication domains, see Chapter 12, *Session management*.

When you create a new cluster member, you have the following options to consider:

- ▶ Basis for first cluster member:

You can add application servers to the cluster when you create the cluster or later.

The first cluster member can be a new application server or you can convert an existing application server so that it becomes the first cluster member.

Subsequent application servers in the cluster must be created new. The first application server in the cluster acts as a template for the subsequent servers.

The options you have depend on the how you create the cluster.

When you use the job manager, you have the option to convert an existing server to use as the first cluster member, or create an empty cluster and run additional jobs to add cluster members.

When you use the deployment manager, you can convert an existing server, create one or more new servers, or create an empty cluster.

**Note:** The option to use an existing application server does not appear in the deployment manager administrative console if you create an empty cluster, then add a member later. If you want to convert an existing application server as the first server, specify that option when you create the cluster or use the job manager to create the cluster member.

**Tip:** To remove a server from a cluster, you must delete the server. Take this into consideration when you are determining whether to convert an existing server to a cluster.

► Server weight for each cluster member:

The weight value controls the amount of work that is directed to the application server. If the weight value for this server is greater than the weight values that are assigned to other servers in the cluster, then this server receives a larger share of the workload. The weight value represents a relative proportion of the workload that is assigned to a particular application server. The value can range from 0 to 20.

Member weight: Specify the relative weight of this server in the cluster. Values are from 0 to 20. 0 indicates that work is to be routed to this server only in the event that no other servers are available.

On z/OS, weight is used to balance some of the workload types, but others are balanced by the z/OS system:

- For HTTP requests, weights are used to distribute HTTP traffic between the Web server plug-in and the controller handling the clustered application server. Assign a higher weight value to the application server that should receive the HTTP traffic.
- For Web services calls, information is transferred from a servant in one application server to a controller in another application server. The application server that receives the call has the highest weight value.
- Weight has no effect on Internet Inter-ORB Protocol (IIOP) requests. IIOP requests are distributed to the correct application server using the sysplex distributor.

## Using the deployment manager administrative console

To create a new cluster:

1. Select **Servers** → **Clusters** → **WebSphere application server clusters**.
2. Click **New**.
3. Enter the information for the new cluster (see Figure 21):
  - Enter a cluster name of your choice.
  - On z/OS, you will also be asked for the short name for the cluster.

The screenshot shows a dialog box titled "Enter basic cluster information". On the left, a vertical sidebar lists four steps: "Step 1: Enter basic cluster information" (highlighted with a yellow arrow), "Step 2: Create first cluster member", "Step 3: Create additional cluster members", and "Step 4: Summary". The main area of the dialog contains the following fields and options:

- A text input field for "Cluster name" containing the text "ClusterB".
- A checked checkbox labeled "Prefer local. Specifies whether enterprise bean requests will be routed to the node on which the client resides when possible."
- An unchecked checkbox labeled "Configure HTTP session memory-to-memory replication".

At the bottom of the dialog, there are two buttons: "Next" and "Cancel".

Figure 21 Creating a new cluster

4. Create first cluster member: The first cluster member determines the server settings for the cluster members (Figure 22).

Step 1: Enter basic cluster information

→ Step 2: Create first cluster member

Step 3: Create additional cluster members

Step 4: Summary

### Create first cluster member

The first cluster member determines the server settings for the cluster members. A server configuration template is created from the first member and stored as part of the cluster data. Additional cluster members are copied from this template.

\* Member name  
server40b1

Select node  
node40a(ND 7.0.0.1) ▼

\* Weight  
2 (0..20)

Generate unique HTTP ports

**Select basis for first cluster member:**

Create the member using an application server template.  
default ▼

Create the member using an existing application server as a template.  
Cell40/node40c(ND 7.0.0.1)/server40ca ▼

Create the member by converting an existing application server.  
Cell40/node40a(ND 7.0.0.1)/server40a1 ▼

None. Create an empty cluster.

Previous Next Cancel

Figure 22 First cluster member

- Member Name: Type a name of the new server to be added to the cluster. On z/OS, you will also be asked for the short name for the server.
- Select Node: Specifies the node on which this new cluster member is created.

- Server weight: Assign the weight for this server.
- Generate unique HTTP ports: Generates unique port numbers for every transport that is defined in the source server, so that the resulting server that is created will not have transports that conflict with the original server or any other servers defined on the same node.
- Select basis for first cluster member:
  - If you select **Create the member using an application server template**, the settings for the new application server are identical to the settings of the application server template you select from the list of available templates.
  - If you select **Create the member using an existing application server as a template**, the settings for the new application server are identical to the settings of the application server you select from the list of existing application servers. However, applications that are installed on the template server are not installed on the new servers.
  - If you select **Create the member by converting an existing application server**, the application server you select from the list of available application servers becomes a member of this cluster.  
 Applications that are installed on the existing server are automatically installed on new members of the cluster.  
 Note that the only way to remove a server from a cluster is to delete it, and when you delete the cluster, all servers in the cluster are deleted. Consider this before selecting this option.
  - If you select **None. Create an empty cluster**, a new cluster is created, but it does not contain any cluster members.

Click **Next**.

5. Create additional cluster members: Use this page to create additional members for a cluster. You can add a member to a cluster when you create the cluster or after you create the cluster. A copy of the first cluster member that you create is stored as part of the cluster data and becomes the template for all additional cluster members that you create.

To add a member, enter a new server name, select the node, and click **Add Member**. The new member will be added to the list. See Figure 23.

Step 1: Enter basic cluster information

Step 2: Create first cluster member

→ **Step 3: Create additional cluster members**

Step 4: Summary

### Create additional cluster members

Enter information about this new cluster member, and click Add Member to add this cluster member to the member list. A server configuration template is created from the first member, and stored as part of the cluster data. Additional cluster members are copied from this template.

\* Member name

Select node

\* Weight  
 (0..20)

Generate unique HTTP ports

Use the Edit function to edit the properties of a cluster member that is already included in this list. Use the Delete function to remove a cluster member from this list. You are not allowed to edit or remove the first cluster member or an already existing cluster member.

| Select                   | Member name | Nodes   | Version    | Weight |
|--------------------------|-------------|---------|------------|--------|
| <input type="checkbox"/> | server40b1  | node40a | ND 7.0.0.1 | 2      |
| Total 1                  |             |         |            |        |

Figure 23 Additional cluster members

6. When all the servers have been entered, click **Next**.
7. A summary page shows you what will be created.
8. Click **Finish** to create the cluster and new servers.
9. Save the configuration.

### ***Adding additional servers to the cluster***

To add a server using the administrative console:

1. Select **Servers** → **Clusters** → **WebSphere application server clusters**.
2. Click the cluster name.
3. Under the Additional Properties sections, click **Cluster members**.
4. Click **New**. This will open the same configuration panel presented when you created the cluster (Figure 23 on page 55).
5. Enter the name of the new server to create, select the node, and select the options to use.
6. Click **Next**, then click **Finish**.

### ***Using cluster member templates***

When you created your cluster's servers, a server template was created for the cluster by copying the first cluster member's configuration. This template is then used when you create additional servers for that cluster. This is important to understand, because you might not get the results you expect when working with clusters.

To view a cluster's member templates using the administrative console:

1. Select **Servers** → **Clusters** → **WebSphere application server clusters**.
2. Click the cluster name.
3. Under the Additional Properties sections, click **Cluster members**.
4. Click **Templates...**

This will open a configuration panel that lists the templates in this cluster. Typically, you will only have one template, but you will have additional templates if the cluster includes servers that are at different versions of WebSphere Application Server (for example, versions 7 and 6.1).

From this configuration panel, you can view and modify the server attributes of the template. If you modify the attributes, it is important to understand existing cluster members will not be affected. The template is only used for creating new cluster members.

To modify the attributes of a cluster's member using the administrative console:

1. Select **Servers** → **Clusters** → **WebSphere application server clusters**.
2. Click the cluster name.
3. Under the Additional Properties sections, click **Cluster members**.
4. Click the cluster member. This will open the server configuration panel where you can make your change.

If you want to make the same change to multiple cluster members, you must repeat these steps. You should also modify the same attributes in the cluster member template, because new cluster members will be created based on the template. If you do not change the cluster's templates, additional cluster members will not match the existing members.

If you want to modify the same server attributes across all of a cluster's members and you have several cluster members, one way to accomplish this using the administrative console:

1. Navigate to the cluster's templates (see before).
2. Click the cluster template and make your desired changes.
3. Save your changes.
4. Delete all of the members in the cluster.
5. Recreate the members.

The new members will be created from the updated cluster member template, and all of the cluster members will have the same configuration.

## **Using the job manager**

When you create an application server cluster from a job manager, you can either create an empty cluster and run subsequent jobs to add cluster members, or you can convert an existing application server as your first cluster member. If you want to create a cluster with one or more new application servers in one step, use the administrative console instead of the job manager.

Building a cluster using the job manager is done in two major steps:

1. Create the cluster.
2. Create the cluster members.

### ***Create the cluster***

To create an application server cluster from the job manager, make the following selections as you step through the process to submit the job:

1. Step 1: Select the **Create cluster** job type.
2. Step 2: Select the deployment manager as the job target.

Enter the user ID and password with administrative authority on the deployment manager.

3. Step 3: Specify the job parameters (see Figure 24 on page 58):
  - Specify the name of the new cluster.

Step 1: Choose a job type

Step 2: Choose job targets

→ **Step 3: Specify job parameters**

Step 4: Schedule the job

Step 5: Review the summary and submit the job

### Specify job parameters

**Job type: Create cluster**

\* Cluster name

Prefer local

Cluster type

Short name

Additional job parameters.

---

**Replication domain**

Create domain

---

**Convert server**

Server node

Server name

Member weight

Node group

Replication entry

Previous
Next
Cancel

Figure 24 Specify the options for the new server

Optionally:

- Prefer local: true (the default) or false
- Short name: cluster short name on z/OS platforms
- Cluster type: The options are:
  - APPLICATION\_SERVER (the default)
  - PROXY\_SERVER
  - ONDEMAND\_ROUTER

Leave this field blank to create an application server cluster.

- Create replication domain: true or false (the default)
- Convert server settings: If you want to use an existing server as the first member of the cluster, complete the server node and server name fields. The other fields are optional.

If you specify the cluster name, and take all other defaults, you will create an empty cluster. When you create an empty cluster, it does not appear in the deployment manager console until you submit a job to add a member to it.

### ***Create the cluster members***

To create new application server cluster members from the job manager, make the following selections as you step through the process to submit the job:

1. Step 1: Select the **Create cluster member** job type.
2. Step 2: Select the deployment manager as the job target.

Enter the user ID and password with administrative authority on the deployment manager.
3. Step 3: Specify the job parameters (see Figure 25 on page 60).
  - Specify the name of the cluster.
  - Specify the node where the cluster member will be created.
  - Specify the name for the new cluster member.

The screenshot shows a web-based configuration interface. On the left, a vertical sidebar lists five steps: 'Step 1: Choose a job type', 'Step 2: Choose job targets', 'Step 3: Specify job parameters' (highlighted with a yellow arrow), 'Step 4: Schedule the job', and 'Step 5: Review the summary and submit the job'. The main panel is titled 'Specify job parameters' and contains the following fields:

- Job type:** Create cluster member
- Cluster name:** Cluster40A (with a 'Find' button to the right)
- Member node:** node40a
- Member name:** Cl40ASrv1
- Member weight:** (empty text box)
- Member UUID:** (empty text box)
- Generate unique ports:** (empty text box)
- Replicator entry:** (empty text box)
- Short name:** (empty text box)
- Additional job parameters:** (expandable section, currently collapsed)

At the bottom of the dialog are three buttons: 'Previous', 'Next', and 'Cancel'.

Figure 25 Specify the options for the new server

Optionally:

- Member weight: Specify the relative weight of this server in the cluster. Values are from 0 to 20. 0 indicates that work is to be routed to this server only in the event that no other servers are available.
- Member UUID
- Generate unique ports (true or false):
- Replicator entry (true or false): Specifying true will add a replicator entry for this server in the cluster replication domain. The default is false.
- Short name: The short name for the server on z/OS.

If this is the first server in the cluster, or if you want to specify a different node or core group, expand the additional job parameters section. It contains settings to specify the template information and the option to specify a node and core group other than the default.

## Viewing cluster topology

The administrative console provides a graphical view of the existing clusters and their members. To see the view, do the following steps:

1. Select **Servers** → **Cluster Topology**.
2. Expand each category. See Figure 26.

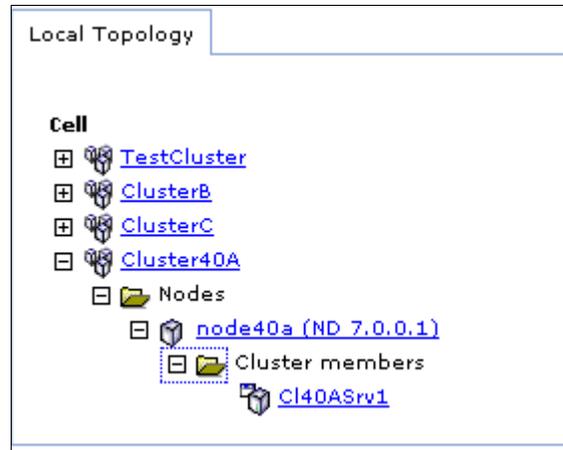


Figure 26 Cluster topology view

3. Selecting a server will take you to the configuration window for the application server.

## Managing clusters

Application servers within a cluster can be managed as independent servers. A second option is to manage all the servers in the cluster as a single entity.

### Using the administrative console

To display and manage the application server clusters:

1. Select **Servers** → **Clusters** → **WebSphere application server clusters**.
2. Check each cluster you want to work with and select one of the following options:
  - Start: Use this option to start all servers in the cluster.
  - Stop: Use this option to stop all servers in the cluster. This allows the server to finish existing requests and allows failover to another member of the cluster.

- Ripplestart: Use this option to stop, then start all servers in the cluster.
- ImmediateStop: Stop all servers immediately.
- Delete: Deletes the cluster and all servers in the cluster.

## Using the job manager

The job manager provides several job types that help you manage clusters:

- ▶ Delete cluster
- ▶ Delete cluster member
- ▶ Start cluster: Use this option to start all servers in a cluster:
  - You can specify that a ripplestart be used (specify true or false). The default is that a ripplestart is not used. A ripplestart will stop, then restart each server.
  - You can also specify a timeout value. If the timeout expires and all servers have not started, the state of the cluster will be reported without waiting any longer for the servers to start.
- ▶ Stop cluster:
  - You can also specify a timeout value. If the timeout expires and all servers have not started, the state of the cluster will be reported without waiting any longer for the servers to start.

## Working with virtual hosts

**Note:** For many users, creating virtual hosts is unnecessary because the default\_host that is provided is sufficient.

For an example of defining and using a new virtual host, see Chapter 15, *Deploying applications*.

A *virtual host* is a configuration enabling a single host machine to resemble multiple host machines. It consists of a host alias or aliases, which consist of a host name and a port number. If you specify an asterisk (\*) as a host name, all host names and IP addresses that the Web server can receive will be mapped to that virtual host.

There are three virtual hosts defined during installation:

- ▶ The *default\_host* virtual host is intended for access to user applications, either through the HTTP transport or through a Web server.

Host aliases in this virtual host generally include the ports required to access applications from the Web server and directly to the application server. For example, the `wc_defaulthost`, `wc_defaulthost_secure`, `sip_defaulthost`, and `sip_defaulthost_secure` ports for application servers and ports 80 and 443 for requests via the Web server.

- ▶ The *admin\_host* virtual host is used for access to the WebSphere administrative console.

At installation time, it is configured to match requests on the `wc_adminhost` and `wc_adminhost_secure` ports for the standalone server or deployment manager.

- ▶ The *proxy\_host* virtual host includes default port definitions, port 80 and 443, which are typically initialized as part of the proxy server initialization. Use this proxy host as appropriate with routing rules associated with the proxy server.

When you install an application, you associate a virtual host with each Web module in the application. By associating a virtual host with a Web module, requests that match the host aliases for the virtual host should be processed by servlets in this Web module. The Web server plug-in also checks the URI of the request against the URIs for the Web module to determine whether the Web module can handle them or not. You can view or modify the virtual host a Web module is assigned to by selecting **Applications** → **WebSphere enterprise applications** → *app\_name* → **[Web Module Properties] Virtual hosts**.

A single virtual host can be associated with multiple Web modules as long as each application has unique URIs. If there are duplicate URIs among applications, different virtual hosts must be created and associated with each of the applications.

A default virtual host is associated with a Web container when you create the application server. To find the default virtual host select **Servers** → **Server Types** → **WebSphere application servers** and click the server name to open the configuration page. In the Container settings section, expand Web container settings and select **Web container**.

## Creating and updating virtual hosts

By default, `default_host` is associated with all user application requests. There are some cases in which multiple virtual hosts should be created, for example:

- ▶ Applications with conflicting URIs
- ▶ Special support for extra ports
- ▶ Provide independence of each virtual host for applications and servers

To create a new virtual host, do the following steps:

1. Select **Environment** → **Virtual hosts** and then click **New**.
2. Enter a name for the virtual host and click **Apply**.  
Note that two links become active: Host Aliases and MIME Types.
3. Click **Host Aliases** in the Additional Properties pane.
4. Click **New**.
5. Enter values for the Host Name and Port fields and click **OK**.

The host aliases are not necessarily the same as the host name and port number of the WebSphere Application Servers. They are the host names and port numbers that the Web server plug-in is expecting to receive from the browser. The Web server plug-in will send the request to the application server using the host name and port number in the transport setting for that server. If the Web server is running on a separate machine from WebSphere, then the host aliases are for Web server machines.

Mapping HTTP requests to host aliases is case sensitive and the match must be alphabetically exact. Also, different port numbers are treated as different aliases.

For example, the request `http://www.myhost.com/myservlet` does *not* map to any of the following steps:

```
http://myhost/myservlet
http://www.myhost.com/MyServlet
http://www.myhost.com:9876/myservlet
```

If the Web server plug-in receives a request that does not match one of the virtual hosts, then an HTTP error will be returned to the user.

Simple wild cards can be used on the host aliases. An asterisk character (\*) can be used for the host name, the port, or both. It means that any request will match this rule.

**Note:** If the virtual host is used in a cluster environment, all host aliases used by servers in the cluster should be registered in the virtual host.

6. Save your changes.

Host aliases can also be updated for virtual hosts through the administrative console.

1. Select **Environment** → **Virtual hosts**.
2. Click the virtual host name to open the configuration page.
3. Click **Host Aliases** in the Additional Properties pane.

4. Click **New**.
5. Enter values for the Host Name and Port fields and click **OK**.

**Important:** If you create, delete, or update virtual hosts, you need to regenerate the Web server plug-in.

## Managing your configuration files

This section summarizes some of the most common system management tasks dealing with configuration files:

- ▶ Backing up a profile
- ▶ Restoring a profile
- ▶ Exporting and importing profiles

### Backing up a profile

Use the **backupConfig** command to back up a profile. You can run this command from the *was\_home/bin* directory and use the **-profileName** option to specify the profile to back up. Or you can execute the command from the *profile\_home/bin* directory.

The command will zip the configuration file and store it in the current directory or a specified file name. The zip file can be restored using the **restoreConfig** command. By default, **backupConfig** will stop all servers in the configuration before performing the backup.

The syntax is shown in Example 20.

*Example 20 backupConfig command*

---

```
Usage: backupConfig [backup_file] [-nostop] [-quiet] [-logfile  
<filename>] [-replaceLog] [-trace] [-username <username>] [-password  
<password>] [-profileName <profile>] [-help]
```

---

For information about the command options and usage, see:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml\\_backupconfig.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml_backupconfig.html)

The **backup\_file** parameter specifies the file where the backup is to be written. If you do not specify a backup file name, a unique name is generated and the file is

stored in the current directory. If you specify a backup file name in a directory other than the current directory, the specified directory must exist.

## Example

Example 21 shows an example of backing up a deployment manager.

### *Example 21 backupConfig example*

---

```
C:\WebSphereV7\AppServer\bin>backupConfig
c:\WebSphereV7\Backups\Dmgr01Mar1509 -profileName Dmgr01 -logfile
c:\WebSphereV7\Backups\logs\Dmgr01Mar1509
ADMU0116I: Tool information is being logged in file
           c:\WebSphereV7\Backups\logs\Dmgr01Mar1509
ADMU0128I: Starting tool with the Dmgr01 profile
ADMU5001I: Backing up config directory
           C:\WebSphereV7\AppServer\profiles\Dmgr01\config to file
           C:\WebSphereV7\Backups\Dmgr01Mar1509
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: dmgr
ADMU2010I: Stopping all server processes for node Dmgr02Node
ADMU0512I: Server dmgr cannot be reached. It appears to be stopped.
.....
.....
.....
ADMU5002I: 908 files successfully backed up
```

---

## Restoring a profile

Use the **restoreConfig** command to restore a profile configuration using an archive previously generated using **backupConfig**. If the configuration to be restored exists, the config directory is renamed to config.old (then config.old\_1, and so on) before the restore begins. The command then restores the entire contents of the *profile\_root/config* directory. By default, all servers on the node stop before the configuration restores so that a node synchronization does not occur during the restoration.

The syntax is shown in Example 22.

### *Example 22 restoreConfig command*

---

```
Usage: restoreConfig backup_file [-location restore_location] [-quiet]
[-nostop] [-nowait] [-logfile <filename>] [-replaceLog] [-trace]
[-username <username>] [-password <password>] [-profileName <profile>]
[-help]
```

---

For information about the command, see:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml\\_restoreconfig.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multipatform.doc/info/ae/ae/rxml_restoreconfig.html)

## Example

Example 23 shows an example of restoring an application server profile.

*Example 23 restoreConfig example*

---

```
C:\WebSphereV7\AppServer\bin>restoreConfig
c:\WebSphereV7\Backups\Dmgr01Mar1509
-profileName Dmgr01 -logfile
c:\WebSphereV7\Backups\logs\Dmgr01Mar1509_r
ADMU0116I: Tool information is being logged in file
           c:\WebSphereV7\Backups\logs\Dmgr01Mar1509_r
ADMU0128I: Starting tool with the Dmgr01 profile
ADMU0505I: Servers found in configuration:
ADMU0506I: Server name: dmgr
ADMU2010I: Stopping all server processes for node Dmgr02Node
ADMU0512I: Server dmgr cannot be reached. It appears to be stopped.
ADMU5502I: The directory
C:\WebSphereV7\AppServer\profiles\Dmgr01\config
           already exists; renaming to
           C:\WebSphereV7\AppServer\profiles\Dmgr01\config.old
ADMU5504I: Restore location successfully renamed
ADMU5505I: Restoring file c:\WebSphereV7\Backups\Dmgr01Mar1509 to
location C:\WebSphereV7\AppServer\profiles\Dmgr01\config
.....
.....
.....
ADMU5506I: 908 files successfully restored
ADMU6001I: Begin App Preparation -
ADMU6009I: Processing complete.
ADMU6002I: Begin Asset Preparation -
ADMU6009I: Processing complete.
```

---

## Exporting and importing profiles

The target configuration of an archive export / import can be a specific server or an entire profile.

To use an archive:

1. Export a WebSphere configuration. This creates a zip file with the configuration.
2. Unzip the files for browsing or update for use on other systems. For example, you might need to update resource references.
3. Send the configuration to the new system. An import can work with the zip file or with the expanded format.
4. Import the archive. The import process requires that you identify the object in the configuration you want to import and the target object in the existing configuration. The target can be the same object type as the archive or its parent:
  - If you import a server archive to a server configuration, the configurations are merged.
  - If you import a server archive to a node, the server is added to the node.

## Server archives

The following command in **wsadmin** can be used to create an archive of a server:

```
$AdminTask exportServer {-archive <archive_location> -nodeName <node>
-serverName <server>}
```

This process removes applications from the server that you specify, and breaks the relationship between the server that you specify and the core group of the server, cluster, or bus membership. If you export a single server of a cluster, the relation to the cluster is eliminated.

To import a server archive, use the following command:

```
$AdminTask importServer {-archive <archive_location> [-nodeInArchive
<node>] [-serverInArchive <server>] [-nodeName <node>] [-serverName
<server>]}
```

When you use the **importServer** command, you select a configuration object in the archive as the source and select a configuration object on the system as the target. The target object can match the source object or be its parent. If the source and target are the same, the configurations are merged.

## Profile archives

You can create a configuration archive (CAR) file containing the configuration of a standalone application server profile for later restoration. A CAR file can be used to clone the original profile to another machine or system. CAR files can be bundled in a customized installation package for use with the Installation Factory

feature. For more information about using the Installation Factory, refer to the Information Center.

You can only create an archive of an unfederated profile (standalone application server).

The following command in **wsadmin** can be used to create an archive of a profile:

```
$AdminTask exportWasprofile {-archive <archive_location>}
```

## Managing applications

WebSphere Application Server V7 supports J2EE 1.3, J2EE 1.3, and Java EE 5, which we refer to as *enterprise applications*.

A new concept has been introduced with V7 that allows you to manage applications at a business level. A *business-level application* does not contain binary application files, but rather, identifies WebSphere and non-WebSphere artifacts that logically belong together to form a business-level application. Deployable EAR files and modules are assets for the business-level application. After being deployed, an application becomes a composition unit to the business-level application. Administration of the artifacts that belong to the business-level application is separate from the administration of the business level definition.

Both enterprise and business-level applications can be configured and managed using the following methods:

► Using the **wsadmin** tool:

The wsadmin scripting tool can be used to manage both business level and enterprise applications.

Using scripts to manage applications is more complicated than using the other methods. It requires skill in at least one of the supported scripting languages and a complete understanding of the WebSphere Application Server configuration. However, scripting can offer a greater degree of control and can be quite useful in situations where you are performing the same administrative tasks multiple times, or when the tasks are to be done by multiple administrators. With introduction of script libraries, many of the scripting tasks have become easier to perform.

Information about using **wsadmin** scripts is found in Chapter 8, *Administration with scripting*.

- ▶ Using the administrative console:  
Using the administrative console is an easy way to install or update business level and enterprise applications. Wizards take you through the process and provide help information at each step. This is the method discussed in this section at a high level.
- ▶ Using the job manager:  
The job manager can be used to install, update, uninstall, stop, and start enterprise applications. Currently, business-level applications must be managed by submitting wsadmin jobs.
- ▶ Shortcuts:  
WebSphere Rapid Deployment tools provide a shortcut to installing, uninstalling, and updating applications. However, these tools do not support JEE 5 nor J2EE 1.2 specification levels. For information about using this feature, see the topic, *Rapid deployment of J2EE applications*, in the Information Center.

## Managing enterprise applications: Administrative console

To view and manage applications using the administrative console, select **Applications** → **Enterprise Applications**.

In the window, you see the list of installed applications and options for performing application management tasks. Select one or more applications by checking the box to the left of the application name, and then click an action to perform. The exception to this is the Install option, which installs a new application, and requires no existing application to be selected. See Figure 27.

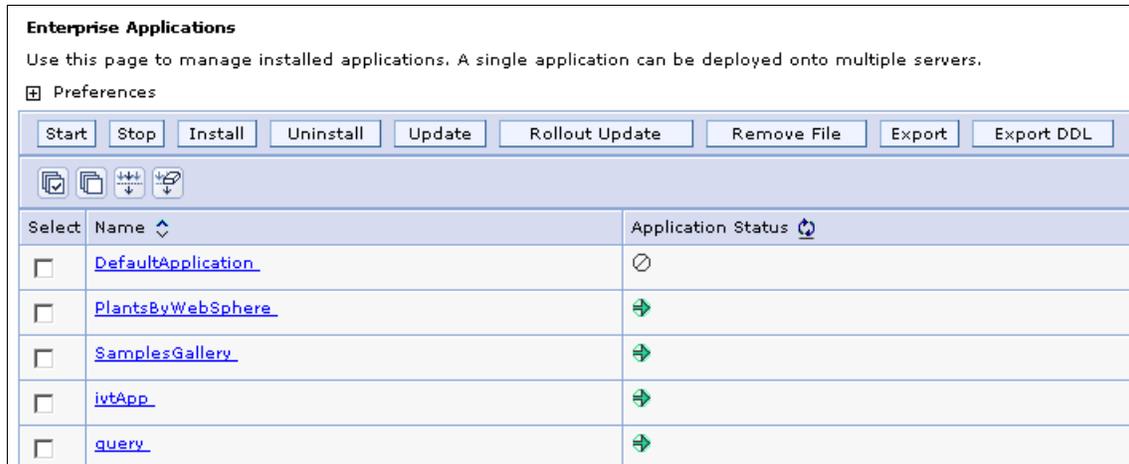


Figure 27 Working with enterprise applications

The following list describes the actions you can choose on this window:

- ▶ **Start:**  
Applications normally start when the server to which they are mapped starts. Exceptions to this include when the application has just been installed, and when the application has been stopped manually.
- ▶ **Stop:**  
You can stop an application manually without affecting the rest of the application server processes. This is common when you are updating an application or want to make it unavailable to users.
- ▶ **Install:**  
The install option takes you through the process of installing a new enterprise application EAR file.
- ▶ **Uninstall:**  
Use this option to uninstall an application. This removes it from the application servers and from the configuration repository.
- ▶ **Update or Rollout Update:**  
Applications can be updated in several ways. The update options include full application, single module, single file, and partial application.
- ▶ **Remove file:**  
With this option, you can remove a single file from an application.

- ▶ **Export:**  
Use this option to export an EAR file of the application.
- ▶ **Export DDL:**  
Use this option to export DDL files found in the application.
- ▶ **Export File:**  
Use this option to export individual files found in the application.

## Installing an enterprise application

To install an enterprise application into a WebSphere configuration, you must install its modules onto one or more application servers.

**Business-level applications (BLAs):** Enterprise applications are considered a composite unit of a business-level application (BLA) and will be can either be added to the BLA at installation time, or can be installed when the undeployed application assets are added to the BLA.

**Adding a new cluster member:** When an application server is added as a member to a server cluster, the modules installed on other members are also installed on the new member. You do not need to re-install or upgrade the application.

In this section, we give you an overview of application installation. Later in this book, we discuss application installation in detail.

- ▶ An example of installing an enterprise application is given in Chapter 15, *Deploying applications*.
- ▶ An example of installing a business-level application can be found in Chapter 14, *Packaging applications for deployment*.

### Using business-level applications

When you install an application, it is added to a business-level application (BLA). This can be an existing BLA, or you can create a new BLA as part of the application install process. If you do not plan to use BLAs to manage your applications, allowing the BLA to be created during the application install process is fine.

However, if you plan to manage your application as part of a BLA, you should start the installation process by importing the application EAR file as an asset,

creating a business-level application to hold the assets, and then installing the assets.

The steps that this entails are:

1. Import the application EAR file as an asset:
  - a. Open the WebSphere administrative console and select **Applications** → **Application Types**. Click the **Assets** link.
  - b. Click the **Import** button and complete the steps in the wizard.

The asset will be imported to the asset repository. The default location is *profile\_root/installedAssets/asset\_name/BASE/*.

Repeat this step for any additional assets that will belong to the BLA.

**Save to master configuration** when done.

2. Create a business-level application:
  - a. Select **Applications** → **Application Types** → **Business-level applications**. Click the **New** button.
  - b. Enter the name for the BLA and click **Apply**.
  - c. On the Business-level applications page, click the **Add** button under the Deployed assets section. You can use this option to add an asset (the EAR file) or a shared library.  
Select the **Add Asset** option.
  - d. On the next panel select EAR file and click **Continue**. You will now configure (install) this asset with the necessary deployment options. This follows the normal steps for installing an application to WebSphere Application Server (see “Installing an application using the administrative console” on page 74).
  - e. Proceed through the installation panels until the Summary page is shown, and click **Finish**. WebSphere now installs the application and it is now a composition unit, an asset which has been configured.

**Note:** At Step 1: Select installation options WebSphere generates a unique application name such as app1143018803114601914 for the application. You might want to change this to something more descriptive.

3. **Save** to the master configuration when done.

See Chapter 14, *Packaging applications for deployment* for an example.

## Installing an application using the administrative console

There are two approaches to installing an enterprise application. The first approach is to install the application as part of the process of adding assets to a BLA (as described in the previous section). The second approach is to install the application and select a BLA, or create a new BLA as part of the process. This section assumes that you are taking the latter approach.

In either case, the process is basically the same after you get into the application installation wizard.

Follow these steps for this task:

1. Select **Applications** → **New Application** → **New Enterprise Application**.
2. Specify the location of the EAR file to install, as shown in Figure 30 on page 77.

The EAR file that you are installing can be either on the client machine running the Web browser, or on any of the nodes in the cell (Figure 28).

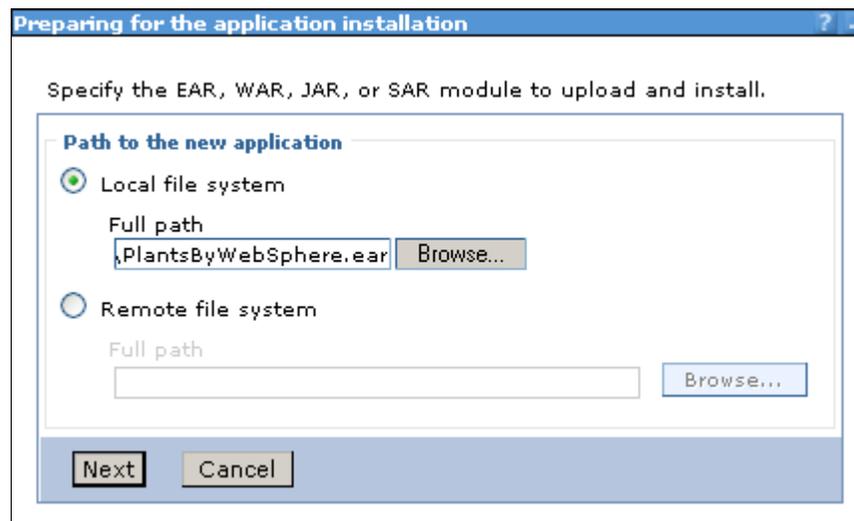


Figure 28 Installing an enterprise application

Click **Next**.

3. Select whether to use the fast path through the wizard or to display all the installation steps in the wizard.

The fast path will only display the steps in the wizard where decisions cannot be made through defaults.

If you expand the **Choose to generate default bindings and mapping** you can alter the bindings for the application you are deploying. If you select the

**Generate Default Bindings** option, WebSphere Application Server completes any incomplete bindings in the application with default values, but it does not alter any existing bindings. Checking the **Override existing bindings** allows you to specify a bindings file which contains new bindings (Figure 29).

Preparing for the application installation

How do you want to install the application?

Fast Path - Prompt only when additional information is required.

Detailed - Show all installation options and parameters.

Choose to generate default bindings and mappings

Generate Default Bindings

Override existing bindings

Specific bindings file

Specify unique prefix for beans:

Prefix

Use default virtual host name for Web and SIP modules:

Host name

Figure 29 Select the path through the wizard and default binding options

Click **Next**.

4. The rest of the installation process is done in steps. The steps can vary, depending on the contents of the EAR file and whether you selected the fast path or detailed path through the wizard. The following steps are a typical sequence for the detailed path:
  - a. Provide options to perform the installation.

**(New in V7)** This panel also provides you the option of selecting a BLA or creating a new one. If you are installing the application as a result of adding an asset to a BLA, the correct BLA should be selected.

If you create a new BLA, it will have the same name as the application.

- b. Map modules to servers.
  - c. Provide JSP™ reloading options for Web modules
  - d. Map shared libraries
  - e. Map shared library relationships
  - f. Initialize parameters for servlets
  - g. Provide JNDI names for beans
  - h. **(New in V7)** Bind EJB business interfaces to a JNDI name.
  - i. Map EJB references to beans
  - j. Map resource references to resources
  - k. Map virtual hosts for Web modules
  - l. Map context roots for Web modules
  - m. Map security roles to users or groups
  - n. **(New in V7)** Specify metadata options. The settings on this page can be used to instruct a Java EE 5 enterprise bean (EJB) or Web module deployment descriptor to ignore annotations that specify deployment information.
  - o. Summary
5. Click **Finish** to install the application.
  6. Save the configuration and synchronize the changes to the nodes.
  7. Update the Web server plug-in and propagate it to the Web server.

## Using the job manager

You can install enterprise applications from the job manager by selecting the following options to submit a job.

**Note:** When you use the job manager to install an application, the only options you can specify from the job manager console is the location to install. All other the options that you would see in the administrative console will default.

The first step is to make the application available to the system where the installation job will run:

1. Before installing an application, you must transfer the application binaries (EAR file) from the job manager to the node where the job will run using the `distributeFile` job (see Chapter 5, *Administration consoles and commands* to define the directories required for this task).

In Rational® Application Developer, export the EAR file to:

`jmgr_profile_root/config/temp/JobManager` directory

2. In the Job manager console, select the **Job** → **Submit** menu. This will launch the Job properties wizard.
  - a. Select **Distribute file** as the job type and click **Next**.
  - b. Select the deployment manager or administrative agent as the target node.
  - c. Enter the EAR file location on the job manager and the location to store the EAR file on the node. The arguments are entered as shown in Figure 30.

Step 1: Choose a job type

Step 2: Choose job targets

→ Step 3: Specify job parameters

Step 4: Schedule the job

Step 5: Review the summary and submit the job

**Specify job parameters**

**Job type: Distribute file**

\* Source  
file:/PlantsByWebSphere.ear

\* Destination  
/PlantsByWebSphere.ear

Distribution provider

Previous Next Cancel

Figure 30 Specify the location of the source and target file

Using these arguments, the script file will be distributed to the following location:

`dmgr_profile_root/downloadedContent/app_name`

Click **Next**.

- d. Take the defaults for the job schedule. The defaults will execute the distribute file job once. Click **Next**.
- e. Click **Finish**. Monitor the status of the job and ensure it completes successfully.

To install the application from the job manager, make the following selections as you step through the process to submit the job:

1. Step 1: Select the **Install application** job type.
2. Step 2: Select the node where the job will run.

Enter the user ID and password with administrative authority on the target node.

3. Step 3: Specify the job parameters.

At minimum:

- Specify the name of the new application.
- If the target is a deployment manager, enter the name of the node and server or cluster that the application will be installed on (Figure 31).

The screenshot shows a dialog box titled "Specify job parameters" for installing an application. On the left, a vertical sidebar lists five steps: "Step 1: Choose a job type", "Step 2: Choose job targets", "Step 3: Specify job parameters" (highlighted with a yellow arrow), "Step 4: Schedule the job", and "Step 5: Review the summary and submit the job". The main area of the dialog is titled "Specify job parameters" and contains the following fields and controls:

- Job type:** Install application
- \* Application name:** PlantsByWebSphere
- Application location:** (empty text box)
- Server name:** (empty text box) with a "Find..." button to its right.
- Node name:** (empty text box)
- Cluster name:** ClusterB with a "Find..." button to its right.
- At the bottom, there are three buttons: "Previous", "Next", and "Cancel".

Figure 31 Specify the options for application install

## Uninstalling an enterprise application

To uninstall a no longer needed enterprise application, use one of the following methods:

From the administrative console:

1. Select **Applications** → **Application Types** → **WebSphere enterprise applications**.
2. Check the application you want to uninstall and click **Uninstall**.

From the job manager, use the **Uninstall application** job type.

**Note:** When you uninstall an application and it is the only composite unit in a business-level application, the BLA is also deleted.

## Starting an enterprise application

An application starts automatically when the application server to which it is mapped starts. You only need to start an application immediately after installing it, or if you have manually stopped it.

**Application startup:** Starting an application server starts the applications mapped to that server. The order in which the applications start depends on the startup order you assigned to each them. The application with the lowest startup order value is started first. Applications that have the same order designation are started in no particular order. Enabling the parallel start option for the application server means to start applications with the same weight in parallel.

To view or change the application startup order, select **Applications** → **Application Types** → **WebSphere enterprise applications**. Open the configuration page for the application by clicking the application name and click **Startup behavior**.

An application can be started by following these steps from the administrative console:

1. Select **Applications** → **Application Types** → **WebSphere enterprise applications**.
2. Check the application you want and click **Start**.

An application can be started from the job manager console by selecting the **Start application** job type.

**Note:** In order to start an application, the application server that contains the application has to be started. If not, the application displays in the administrative console as unavailable and you are not able to start it.

## Stopping an enterprise application

An application can be stopped using the administrative console.

From the administrative console, do the following steps:

1. Select **Applications** → **Application Types** → **WebSphere enterprise applications**.
2. Check the application you want to stop and click **Stop**.

Or, from the job manager console, use the **Stop application** job type.

## Preventing an enterprise application from starting on a server

By default, an application will start when the server starts. The only way to prevent this occurrence is to disable the application from running on the server.

From the administrative console:

1. Select **Applications** → **Application Types** → **WebSphere enterprise applications**.
2. Click the application to open the configuration.
3. Select **Target specific application status** in the Detail Properties table.
4. Select the server for which you want to disable the application.
5. Click the **Disable Auto Start** button.
6. Save the configuration.

## Viewing application details

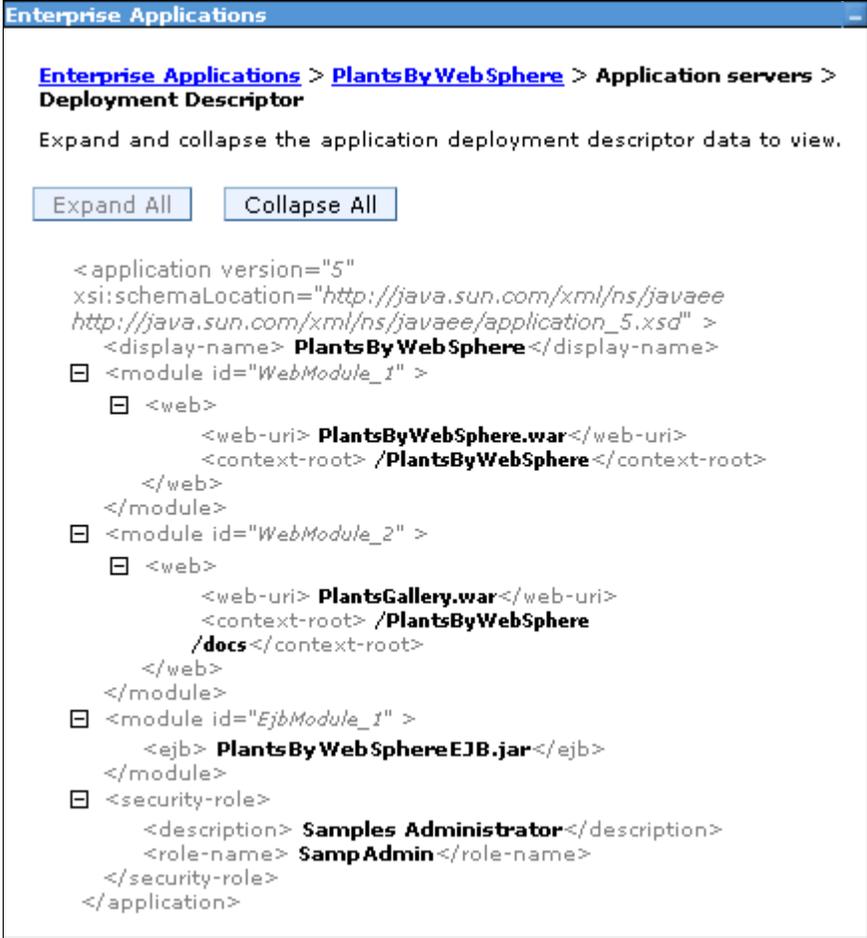
The administrative console does not display the deployed servlets, JSPs, or EJBs directly on the console. However, you can use the console to display XML deployment descriptors for the enterprise application, Web modules, and EJB modules.

To view the application deployment descriptor for an application, do the following steps:

1. Select **Applications** → **Application Types** → **WebSphere enterprise applications**.
2. Click the application that you are interested in.
3. Under the Configuration tab, select **View Deployment Descriptor** under Detail Properties.

Figure 32 shows the deployment descriptor window for the PlantsByWebSphere enterprise application. The Configuration tab shows you the structure defined by the deployment descriptor:

- ▶ The name of the enterprise application
- ▶ The Web modules and their context roots
- ▶ The EJB modules and their associated JAR files
- ▶ The security roles associated with the enterprise application



The screenshot shows a window titled "Enterprise Applications" with a breadcrumb path: "Enterprise Applications > PlantsByWebSphere > Application servers > Deployment Descriptor". Below the breadcrumb, there is a text instruction: "Expand and collapse the application deployment descriptor data to view." and two buttons: "Expand All" and "Collapse All". The main content area displays XML code for the deployment descriptor, with several elements expanded to show their details. The XML code is as follows:

```
<application version="5"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/application_5.xsd" >
  <display-name> PlantsByWebSphere </display-name>
  <module id="WebModule_1" >
    <web>
      <web-uri> PlantsByWebSphere.war</web-uri>
      <context-root> /PlantsByWebSphere</context-root>
    </web>
  </module>
  <module id="WebModule_2" >
    <web>
      <web-uri> PlantsGallery.war</web-uri>
      <context-root> /PlantsByWebSphere
/docs</context-root>
    </web>
  </module>
  <module id="EjbModule_1" >
    <ejb> PlantsByWebSphereEJB.jar</ejb>
  </module>
  <security-role>
    <description> Samples Administrator</description>
    <role-name> SampAdmin</role-name>
  </security-role>
</application>
```

Figure 32 Enterprise application deployment descriptor

## Viewing EJB modules

To see the EJBs that are part of an enterprise application:

1. Select **Applications** → **Application Types** → **WebSphere enterprise applications**.

2. Click the application name.
3. Select **Manage Modules** under Modules Items.
4. Click the EJB module that you want to view (Figure 33).

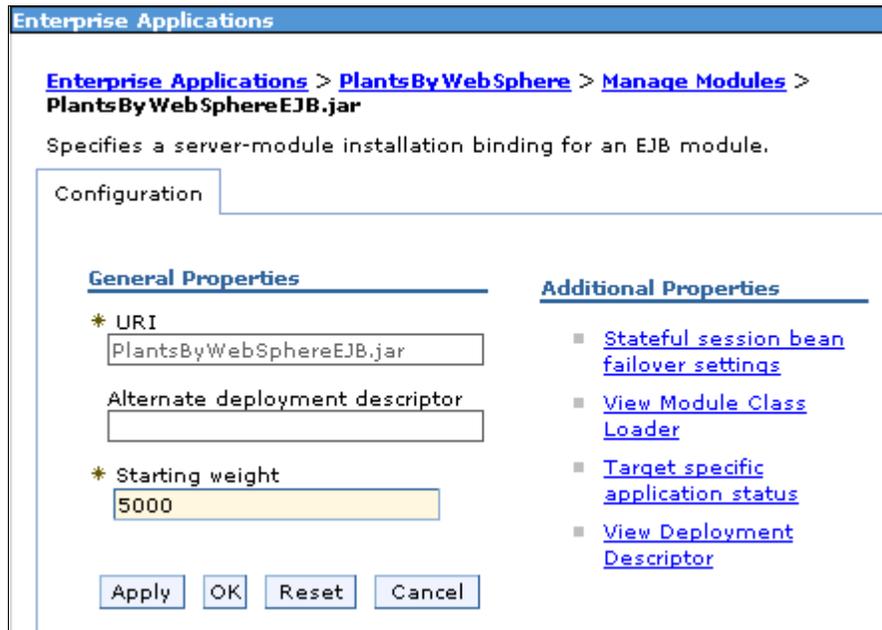


Figure 33 Viewing an EJB module configuration

5. Click **View Deployment Descriptor** under Additional Properties to see the EJB deployment descriptor.

## Viewing Web modules

To see the servlets and JSPs that are part of an enterprise application:

1. Select **Applications** → **Application Types** → **WebSphere enterprise applications**
2. Click the application name.
3. Select **Manage Modules** under Modules.

4. Click the Web module you want to view (Figure 34).

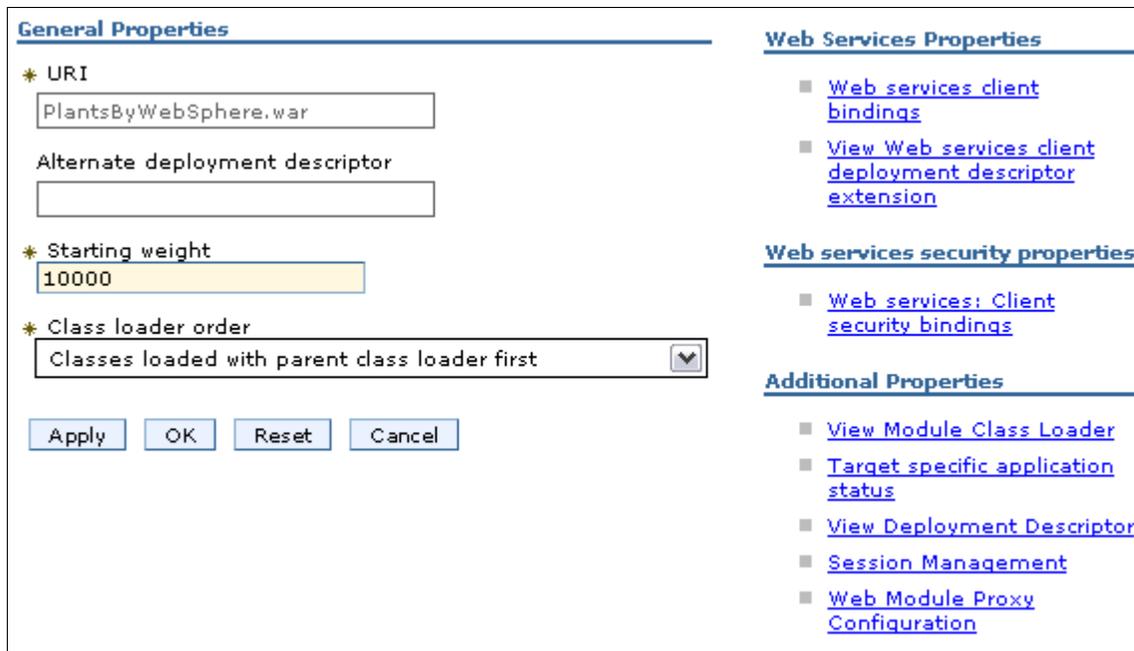


Figure 34 View a Web module

5. Click **View Deployment Descriptor** to see the details of the Web module content.

## Finding a URL for a servlet or JSP

The URL for a servlet or JSP is the path used to access it from a browser. The URL is partly defined in the deployment descriptor provided in the EAR file and partly defined in the deployment descriptor for the Web module containing the servlet or JSP.

To find the URL for a servlet or JSP:

1. Find the context root of the Web module containing the servlet.
2. Find the URL for the servlet.
3. Find the virtual host where the Web module is installed.
4. Find the aliases by which the virtual host is known.
5. Combine the virtual host alias, context root, and URL pattern to form the URL request of the servlet/JSP.

For example, to look up the URL for the snoop servlet:

1. Find the context root of the Web module DefaultWebApplication of the DefaultApplication enterprise application. This Web module contains the snoop servlet.
  - a. Select **Applications** → **Application Types** → **WebSphere enterprise applications**.
  - b. Click the application that you are interested in, in our case, **DefaultApplication.ear**.
  - c. On the Configuration tab, select **Context Root for Web Modules** (Figure 36). You can see that:
    - i. There is only one Web module in this application, Default Web Application.
    - ii. The context root is `/`. We will use this later.

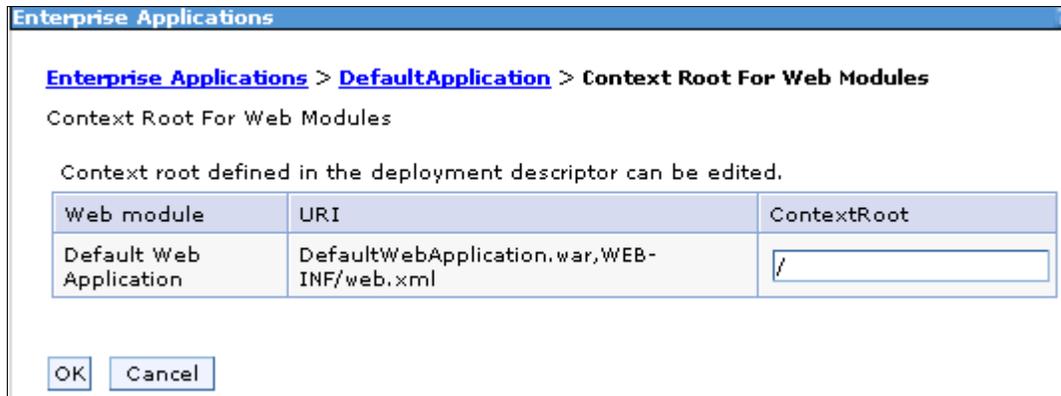


Figure 35 Context root for the Web modules in DefaultApplication

- d. Click **OK** to return to the DefaultApplication configuration.
2. Find the URL for the snoop servlet:
  - a. In the DefaultApplication configuration page, select **Manage Modules**.
  - b. Click the **Default Web Application** Web module to see the general properties.
  - c. Click **View Deployment Descriptor**.

This displays the Web module properties window, as shown in Figure 36 on page 85. Note that the URL pattern for the snoop servlet starting from the Web module context root is `/snoop/*`. The Web module context root was `/`.

[Enterprise Applications](#) > [DefaultApplication](#) > [Manage Modules](#) > [DefaultWebApplication.war](#) > **Deployment Descriptor**

Expand and collapse the application deployment descriptor data to view.

Expand All

Collapse All

```
<web-app id="WebApp_ID" >
  <display-name> Default Web Application</display-name>
  <description> This is the IBM WebSphere Application Sever Default Web
  Application.</description>
  ⊕ <servlet id="Servlet_1" >
  ⊕ <servlet id="Servlet_2" >
  ⊕ <servlet id="Servlet_3" >
  ⊖ <servlet-mapping id="ServletMapping_1" >
    <servlet-name> Snoop Servlet</servlet-name>
    <url-pattern> /snoop/*</url-pattern>
  </servlet-mapping>
  ⊖ <servlet-mapping id="ServletMapping_2" >
    <servlet-name> Hello Pervasive Servlet</servlet-name>
    <url-pattern> /hello</url-pattern>
  </servlet-mapping>
  ⊖ <servlet-mapping id="ServletMapping_3" >
    <servlet-name> Hit Count Servlet</servlet-name>
    <url-pattern> /hitcount</url-pattern>
  </servlet-mapping>
  ⊕ <welcome-file-list id="WelcomeFileList_1" >
  ⊕ <error-page id="ErrorPage_1" >
  ⊕ <security-constraint id="SecurityConstraint_1" >
  ⊕ <security-role id="SecurityRole_1" >
  ⊕ <ejb-ref id="EjbRef_1" >
  </web-app>
```

Figure 36 DefaultWebApplication Web module deployment descriptor

- d. Note that as you navigate through the windows, a navigation path is displayed below the Messages area. Click **DefaultApplication.ear** to return to the application configuration page.
3. Find the virtual host where the Web module is installed:
  - a. In the DefaultApplication configuration page, select **Virtual hosts** under Web Module Properties.

This will display all of the Web modules contained in the enterprise application, and the virtual hosts in which they have been installed. See Figure 37. Note that the Default Web Application Web module has been installed on the default\_host virtual host.

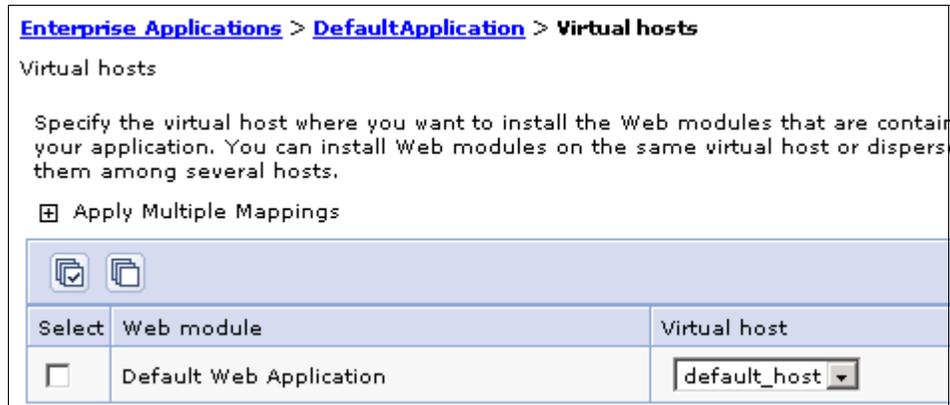


Figure 37 List of virtual hosts

4. Find the host aliases for the default\_host virtual host:
  - a. From the console navigation tree, select **Environment** → **Virtual Hosts**.
  - b. Click **default\_host**.
5. Select **Host Aliases** under Additional Properties:
 

This shows the list of aliases by which the default\_host virtual host is known. See Figure 38.

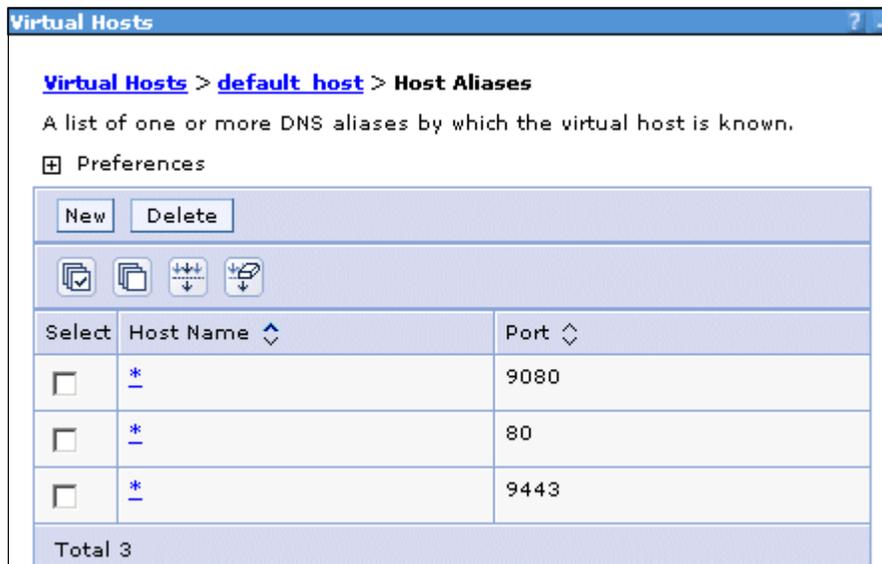


Figure 38 Default\_host virtual host aliases

Note that the aliases are composed of a DNS host name and a port number. The host aliases for the default\_host virtual host are \*:80, \*:9080 and \*:9443, "\*" meaning any host name.

6. Combine the virtual host alias, context root and URL pattern to form the URL request of the snoop servlet. Requests for the servlet with any of the following URLs will map to the default\_host virtual host:
  - `http://hostname:80/snoop`
  - `http://hostname:9080/snoop`
  - `https://hostname:9443/snoop`

## Enabling process restart on failure

WebSphere Application Server does not have either:

- ▶ A nanny process to monitor whether the AdminServer process is running, and restart it if the process has failed
- ▶ An AdminServer process to monitor whether each application server process is running, and restart it if the process has failed

Instead, WebSphere Application Server uses the native operating system functionality to restart a failed process. Refer to the following sections that discuss your operating system.

### Windows

The administrator can choose to register one or more of the WebSphere Application Server processes on a machine as a Windows service during profile creation, or after by using the **WASService** command. With this command, Windows then automatically attempt to restart the service if it fails.

### Syntax

Enter **WASService.exe** with no arguments to get a list the valid formats. See Example 24.

#### *Example 24 WASService command format*

---

```
Usage: WASService.exe (with no arguments displays this help)
|| -add <service name>
   -serverName <Server>
   -profilePath <Server's Profile Directory>
   [-wasHome <Websphere Install Directory>]
   [-configRoot <Config Repository Directory>]
   [-startArgs <additional start arguments>]
```

```

        [-stopArgs <additional stop arguments>]
        [-userid <execution id> -password <password>]
        [-logFile <service log file>]
        [-logRoot <server's log directory>]
        [-encodeParams]
        [-restart <true | false>]
        [-startType <automatic | manual | disabled>]
parameters] || -remove <service name>
parameters] || -start <service name> [optional startServer.bat
parameters] || -stop <service name> [optional stopServer.bat
parameters] || -status <service name>
parameters] || -encodeParams <service name>

```

---

Be aware of the following considerations when using the **WASService** command:

- ▶ When adding a new service, the **-serverName** argument is mandatory. The **serverName** is the process name. If in doubt, use the **serverstatus -all** command to display the processes. For a deployment manager, the **serverName** is **dmgr**, for a node agent it is **nodeagent**, and for a server, it is the server name.
- ▶ The **-profilePath** argument is mandatory. It specifies the home directory for the profile.
- ▶ Use unique service names. The services are listed in the Windows Services control window as:

```
IBM WebSphere Application Server V7.0 - <service name>
```

The convention used by the Profile Management Tool is to use the node name as the service name for a node agent. For a deployment manager, it uses the node name of the deployment manager node concatenated with **dmgr** as the service name.

### **Examples**

Example 25 shows using the **WASService** command to add a node agent as a Windows service.

*Example 25 Registering a deployment manager as a Windows service*

---

```
C:\WebSphereV7\AppServer\bin>WASService -add "node40b" -serverName
nodeagent -profilePath "C:\webspherev7\appserver\profiles\node40b"
-restart true
```

```
Adding Service: node40b
```

```
Config Root: C:\webspherev7\appserver\profiles\node40b\config
```

```
Server Name: nodeagent
```

```
Profile Path: C:\webspherev7\appserver\profiles\node40b
```

```
Was Home: C:\WebSphereV7\AppServer\  
Start Args:  
Restart: 1
```

IBM WebSphere Application Server V7.0 - node40b service successfully added.

---

Note that the service name added in Figure 39 will be IBM WebSphere Application Server V7.0, concatenated with the name you specified for the service name.

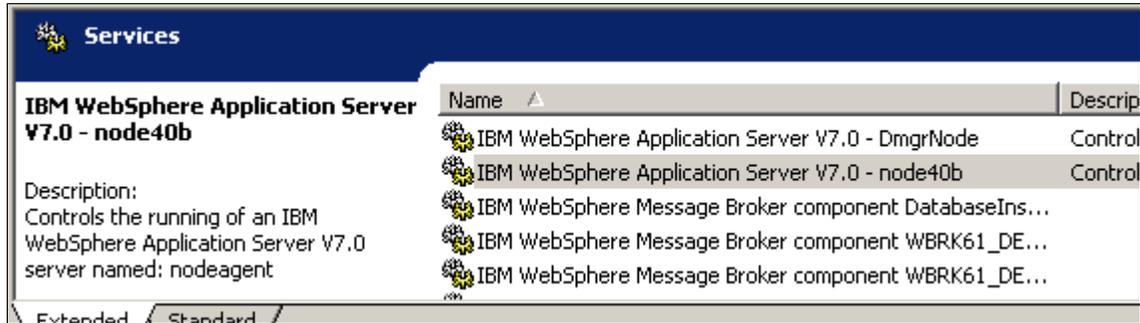


Figure 39 New service

If you remove the service using the **WASService -remove** command, specify only the latter portion of the name, as in Example 26.

#### Example 26 Removing a service

---

```
C:\WebSphereV7\AppServer\bin>WASService -remove "node40b"  
Remove Service: node40b  
Successfully removed service
```

---

## UNIX and Linux

The administrator can choose to include entries in `inittab` for one or more of the WebSphere Application Server processes on a machine, as shown in Example 27. Each such process will then be automatically restarted if it has failed.

#### Example 27 Inittab contents for process restart

---

```
On deployment manager machine:  
ws1:23:respawn:/usr/WebSphere/DeploymentManager/bin/startManager.sh  
  
On node machine:  
ws1:23:respawn:/usr/WebSphere/AppServer/bin/startNode.sh
```

```
ws2:23:respawn:/usr/WebSphere/AppServer/bin/startServer.sh nodename_server1
ws3:23:respawn:/usr/WebSphere/AppServer/bin/startServer.sh nodename_server2
ws4:23:respawn:/usr/WebSphere/AppServer/bin/startServer.sh nodename_server2
```

---

**Note:** When setting the action for startServer.sh to respawn in /etc/inittab, be aware that init will always restart the process, even if you intended for it to remain stopped. As an alternative, you can use the rc.was script located in \${WAS\_HOME}/bin, which allows you to limit the number of retries.

The best solution is to use a monitoring product that implements notification of outages and logic for automatic restart.

## z/OS

WebSphere for z/OS takes advantage of the z/OS Automatic Restart Management (ARM) to recover application servers. Each application server running on a z/OS system (including servers you create for your business applications) are automatically registered with an ARM group. Each registration uses a special element type called SYSCB, which ARM treats as restart level 3, ensuring that RRS (It is a z/OS facility that provides two-phase sync point support across participating resource managers) restarts before any application server.

**Note:** If you have an application that is critical for your business, you need facilities to manage failures. z/OS provides rich automation interfaces, such as automatic restart management, that you can use to detect and recover from failures. The automatic restart management handles the restarting of servers when failures occur.

Some important things to consider when using automatic restart management:

- ▶ If you have automatic restart management (ARM) enabled on your system, you might want to disable ARM for the WebSphere Application Server for z/OS address spaces before you install and customize WebSphere Application Server for z/OS. During customization, job errors might cause unnecessary restarts of the WebSphere Application Server for z/OS address spaces. After installation and customization, consider enabling ARM.
- ▶ If you are ARM-enabled and you cancel or stop a server, it will restart in place using the **armrestart** command.

- ▶ It is a good idea to set up an ARM policy for your deployment manager and node agents. For more information about how to change the ARM policies, refer to:  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.w.ebsphere.installation.zseries.doc/info/zseries/ae/cins\\_changearm.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.w.ebsphere.installation.zseries.doc/info/zseries/ae/cins_changearm.html)
- ▶ If you start the location service daemon on a system that already has one, it will terminate.
- ▶ Every other server will come up on a dynamic port unless the configuration has a fixed port. Therefore, the fixed ports must be unique in a sysplex.
- ▶ If you issue STOP, CANCEL, or MODIFY commands against server instances, be aware of how automatic restart management behaves regarding WebSphere Application Server for z/OS server instances; Table 1 depicts the behavior of ARM regarding WebSphere Application Server for z/OS server instances.

*Table 1 Behavior of ARM and WebSphere Application Server for z/OS server instances*

| <b>When you issue</b>                    | <b>ARM behavior</b>                    |
|--|--|
| STOP address_space                       | It will not restart the address space. |
| CANCEL address_space                     | It will not restart the address space. |
| CANCEL address_space, ARMRESTART         | It will restart the address space.     |
| MODIFY address_space, CANCEL             | It will not restart the address space. |
| MODIFY address_space, CANCEL, ARMRESTART | It will restart the address space.     |

For more information about how to activate the ARM, refer to:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.web.sphere.installation.zseries.doc/info/zseries/ae/tins\\_activearm.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.web.sphere.installation.zseries.doc/info/zseries/ae/tins_activearm.html)

Let us say you have activated ARM and want to check the status of address spaces registered for automatic restart management; in order to get this information, you need to:

1. Initialize all servers.
2. Issue one or both of the commands shown in Example 28.

*Example 28 Displaying the status of address spaces registered for automatic restart management*

---

To display all registered address spaces (including the address spaces of server instances), issue the command:

```
d xcf,armstatus,detail
```

To display the status of a particular server instance, use the display command and identify the job name. For example, to display the status of the Daemon server instance (job BBODMN), issue the following command:

```
d xcf,armstatus,jobname=bbodmn,detail
```

---

For more information about how to use the **display** command, refer to:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multiplatform.doc/info/ae/ae/rxml\\_mvdisplay.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.webSphere.nd.multiplatform.doc/info/ae/ae/rxml_mvdisplay.html)

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

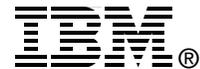
#### **COPYRIGHT LICENSE:**

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

**© Copyright International Business Machines Corporation 2009. All rights reserved.**

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This document REDP-4574-00 was created or updated on October 13, 2009.



Send us your comments in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:  
[ibm.com/redbooks](http://ibm.com/redbooks)
- ▶ Send your comments in an email to:  
[redbook@us.ibm.com](mailto:redbook@us.ibm.com)
- ▶ Mail your comments to:  
IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099, 2455 South Road  
Poughkeepsie, NY 12601-5400 U.S.A.



## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

IBM®  
RACF®

Rational®  
Redbooks (logo) ®

WebSphere®  
z/OS®

The following terms are trademarks of other companies:

EJB, J2EE, Java, JSP, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.