WebSphere® Application Server Network Deployment for Distributed Platforms, Version 7.0

Setting up the application serving environment

Note

Before using this information, be sure to read the general information under “Notices” on page 357.

Compilation date: September 8, 2008

Contents

How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information.

- To send comments on articles in the WebSphere Application Server Information Center
  1. Display the article in your Web browser and scroll to the end of the article.
  2. Click on the **Feedback** link at the bottom of the article, and a separate window containing an e-mail form appears.
  3. Fill out the e-mail form as instructed, and click on **Submit feedback** .
- To send comments on PDF books, you can e-mail your comments to: **wasdoc@us.ibm.com** or fax them to 919-254-5250.

  Be sure to include the document name and number, the WebSphere Application Server version you are using, and, if applicable, the specific page, table, or figure number on which you are commenting.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Changes to serve you more quickly

Print sections directly from the information center navigation

PDF books are provided as a convenience format for easy printing, reading, and offline use. The information center is the official delivery format for IBM WebSphere Application Server documentation. If you use the PDF books primarily for convenient printing, it is now easier to print various parts of the information center as needed, quickly and directly from the information center navigation tree.

To print a section of the information center navigation:

1. Hover your cursor over an entry in the information center navigation until the **Open Quick Menu** icon is displayed beside the entry.
2. Right-click the icon to display a menu for printing or searching your selected section of the navigation tree.
3. If you select **Print this topic and subtopics** from the menu, the selected section is launched in a separate browser window as one HTML file. The HTML file includes each of the topics in the section, with a table of contents at the top.
4. Print the HTML file.

For performance reasons, the number of topics you can print at one time is limited. You are notified if your selection contains too many topics. If the current limit is too restrictive, use the feedback link to suggest a preferable limit. The feedback link is available at the end of most information center pages.

Under construction!

The Information Development Team for IBM WebSphere Application Server is changing its PDF book delivery strategy to respond better to user needs. The intention is to deliver the content to you in PDF format more frequently. During a temporary transition phase, you might experience broken links. During the transition phase, expect the following link behavior:

- Links to Web addresses beginning with http:// work
- Links that refer to specific page numbers within the same PDF book work
- The remaining links will *not* work. You receive an error message when you click them

Thanks for your patience, in the short term, to facilitate the transition to more frequent PDF book updates.

Chapter 1. Configuring ports

When you configure WebSphere® Application Server resources or assign port numbers to other applications, you must avoid conflicts with other assigned ports. In addition, you must explicitly enable access to particular port numbers when you configure a firewall.

1. Review the port number settings, especially when you are planning to coexist.
2. Optional: Change the port number settings.

   During installation, you can use the Installation wizard as described in the ″Installing the product and additional software″ article in the information center.

   You can set port numbers when configuring the product after installation.

   - During profile creation using the manageprofiles command, you can accept the default port values or you can specify your port settings. If you want to specify ports, you can do so in any of the following ways:
     – Specify the use of a port file that contains the port values.
     – Specify the use of a starting port value.
     – Specify the use of the default port values.

     Read the ″manageprofiles command″ article in the information center for more information.
   - During profile creation using the Profile Management tool, you can accept the port settings recommended by the tool or you can specify your port settings.

     Read the ″Creating profiles using the graphical user interface″ article in the information center for more information.

   You can perform one of the following actions to change port settings after installation:

   - Use the updatePorts tool to change port settings.

     Read the ″Updating ports in an existing profile″ article in the information center for more information.
   - Edit the *profile_root*/config/cells/*cell_name*/nodes/*node_name*/serverindex.xml file to change the port settings, or use scripting to change the values.

Port number settings in WebSphere Application Server versions

You should be able to identify the default port numbers used in the various versions of WebSphere Application Server so that you can avoid port conflicts if you plan for an earlier version to coexist or interoperate with Version 7.0.

When you configure WebSphere Application Server resources or assign port numbers to other applications, you must avoid conflicts with other assigned ports. In addition, when you configure a firewall, you must explicitly enable access to particular port numbers.

If ports are already defined in a configuration being migrated, the migration tools fix the port conflicts in the Version 7.0 configuration and log the changes for your verification .

Version 7.0 port numbers

Table 1. Port definitions for WebSphere Application Server Version 7.0

Port Name	Default Value							Files
	Standalone Application Server	Federated Application Server	Deployment Manager	Administrative Agent	Job Manager	Secure Proxy Server	Administrative Subsystem	
Administrative Console Port (WC_ adminhost)	9060	----	9060	9060	9960	----	----	serverindex.xml and virtualhosts.xml
Administrative Console Secure Port (WC_ adminhost_ secure)	9043	----	9043	9043	9943	----	----	
HTTP Transport Port (WC_ defaulthost)	9080	9080	----	----	----	80	----	
HTTPS Transport Secure Port (WC_ defaulthost_ secure)	9443	9443	----	----	----	443	----	

Table 1. Port definitions for WebSphere Application Server Version 7.0 (continued)

Port Name	Default Value							Files
	Standalone Application Server	Federated Application Server	Deployment Manager	Administrative Agent	Job Manager	Secure Proxy Server	Administrative Subsystem	
Bootstrap Port (BOOTSTRAP_ ADDRESS)	2809	2809	9809	9807	9808	----	----	serverindex.xml
Cell Discovery Address (CELL_ DISCOVERY_ ADDRESS)	----	----	7277	----	----	----	----	
CSIV2 Client Authentication Listener Port (CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS)	9402	9405	9402	9402	9402	----	----	
CSIV2 Server Authentication Listener Port (CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS)	9403	9406	9403	9403	9403	----	----	
High Availability Manager Communication Port (DCS_ UNICAST_ ADDRESS)	9353	9353	9352	----	----	----	----	
Internal JMS Server Port (JMSSERVER_ SECURITY_ PORT)	5557	----	----	----	----	----	----	
IPC Connector Port (IPC_ CONNECTOR_ ADDRESS)	9633	9633	9632	9630	9631	9633	9634	
MQ Transport Port (SIB_ MQ_ ENDPOINT_ ADDRESS)	5558	5558	----	----	----	----	----	
MQ Transport Secure Port (SIB_ MQ_ ENDPOINT_ SECURE_ ADDRESS)	5578	5578	----	----	----	----	----	

Table 1. Port definitions for WebSphere Application Server Version 7.0 (continued)

Port Name	Default Value							Files
	Standalone Application Server	Federated Application Server	Deployment Manager	Administrative Agent	Job Manager	Secure Proxy Server	Administrative Subsystem	
ORB Listener Port (ORB_ LISTENER_ ADDRESS)	9100	0	9100	9098	9099	----	----	serverindex.xml
RMI Connector Port (RMI_ CONNECTOR_ ADDRESS)	----	----	----	----	----	----	9810	
JSR 160 RMI Connector Port (JSR160RMI_ CONNECTOR_ ADDRESS)	----	----	----	----	----	----	9811	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9401	9404	9401	9401	9401	----	----	
Service Integration Port (SIB_ ENDPOINT_ ADDRESS)	7276	7276	----	----	----	----	----	
Service Integration Secure Port (SIB_ ENDPOINT_ SECURE_ ADDRESS)	7286	7286	----	----	----	----	----	
SIP Container Port (SIP_ DEFAULTHOST)	5060	5060	----	----	----	5060	----	
SIP Container Secure Port (SIP_ DEFAULTHOST_ SECURE)	5061	5061	----	----	----	5061	----	
SOAP Connector Port (SOAP_ CONNECTOR_ ADDRESS)	8880	8880	8879	8877	8876	----	8881	
IBM® HTTP Server Port	80	----	----	----	----	----	----	virtualhosts.xml, plugin-cfg.xml, and *web_ server_ root*/conf/ httpd.conf
IBM HTTPS Server Administration Port	8008	----	----	----	----	----	----	*web_ server_ root*/conf/ admin.conf

When you federate an application server node into a deployment-manager cell, the deployment manager instantiates the node agent server process on the application server node. The node agent server uses these port assignments by default.

Table 2. Port definitions for the Version 7.0 node agent server process

Port Name	Default Value		File
	Cell Node Agent	Node Agent for Job Management	
Bootstrap Port (BOOTSTRAP_ ADDRESS)	2810	2810	serverindex.xml
CSIV2 Server Authentication Port (CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS)	9201	9201	
CSIV2 Client Authentication Listener Port (CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS)	9202	9202	
High Availability Manager Communication Port (DCS_ UNICAST_ ADDRESS)	9354	9354	
IPC Connector Port (IPC_ CONNECTOR_ ADDRESS)	9626	9626	
Node Discovery Address (NODE_ DISCOVERY_ ADDRESS)	7272	7272	
Node IPV6 Discovery Address (NODE_ IPV6_ MULTICAST_ DISCOVERY_ ADDRESS)	5001	5001	
Node Multicast Discovery Address (NODE_ MULTICAST_ DISCOVERY_ ADDRESS)	5000	5000	
ORB Listener Port (ORB_ LISTENER_ ADDRESS)	9101	9101	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9901	9901	
SOAP Connector Port (SOAP_ CONNECTOR_ ADDRESS)	8878	8878	

Version 6.1 port numbers

Table 3. Port definitions for WebSphere Application Server Version 6.1

Port Name	Default Value		Files
	Application Server	Deployment Manager	
Administrative Console Port (WC_ adminhost)	9060	9060	serverindex.xml and virtualhosts.xml
Administrative Console Secure Port (WC_ adminhost_ secure)	9043	9043	
HTTP Transport Port (WC_ defaulthost)	9080	9080	
HTTPS Transport Secure Port (WC_ defaulthost_ secure)	9443	9443	

Table 3. Port definitions for WebSphere Application Server Version 6.1 (continued)

Port Name	Default Value		Files
	Application Server	Deployment Manager	
Bootstrap Port (BOOTSTRAP_ ADDRESS)	2809	9809	serverindex.xml
Cell Discovery Address (CELL_ DISCOVERY_ ADDRESS)	----	7277	
CSIV2 Server Authentication Listener Port (CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS)	9403	9403	
CSIV2 Client Authentication Listener Port (CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS)	9402	9402	
DRS_ CLIENT_ ADDRESS **Note:** This port is deprecated and is no longer used in the current version of WebSphere Application Server.	7873	7989	
High Availability Manager Communication Port (DCS_ UNICAST_ ADDRESS)	9353	9352	
Internal JMS Server Port (JMSSERVER_ SECURITY_ PORT)	5557	----	
MQ Transport Port (SIB_ MQ_ ENDPOINT_ ADDRESS)	5558	----	
MQ Transport Secure Port (SIB_ MQ_ ENDPOINT_ SECURE_ ADDRESS)	5578	----	
ORB Listener Port (ORB_ LISTENER_ ADDRESS)	9100	9100	
Proxy Server Port (PROXY_ HTTP_ ADDRESS)	80	80	
Proxy Server Secure Port (PROXY_ HTTPS_ ADDRESS)	443	443	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9401	9401	
Service Integration Port (SIB_ ENDPOINT_ ADDRESS)	7276	7276	
Service Integration Secure Port (SIB_ ENDPOINT_ SECURE_ ADDRESS)	7286	7286	
SIP Container Port (SIP_ DEFAULTHOST)	5060	----	
SIP Container Secure Port (SIP_ DEFAULTHOST_ SECURE)	5061	----	
SOAP Connector Port (SOAP_ CONNECTOR_ ADDRESS)	8880	8879	
IBM HTTP Server Port	80	----	virtualhosts.xml, plugin-cfg.xml, and *web_ server_ root*/conf/ httpd.conf
IBM HTTPS Server Administration Port	8008	----	*web_ server_ root*/conf/ admin.conf

When you federate an application server node into a deployment manager cell, the deployment manager instantiates the node agent server process on the application server node. The node agent server uses these port assignments by default.

Table 4. Port definitions for the Version 6.1 node agent server process

Port Name	Default Value	File
Bootstrap Port (BOOTSTRAP_ ADDRESS)	2809	serverindex.xml
CSIV2 Server Authentication Port (CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS)	9201	
CSIV2 Client Authentication Listener Port (CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS)	9202	
High Availability Manager Communication Port (DCS_ UNICAST_ ADDRESS)	9354	
Node Discovery Address (NODE_ DISCOVERY_ ADDRESS)	7272	
Node IPV6 Discovery Address (NODE_ IPV6_ MULTICAST_ DISCOVERY_ ADDRESS)	5001	
Node Multicast Discovery Address (NODE_ MULTICAST_ DISCOVERY_ ADDRESS)	5000	
ORB Listener Port (ORB_ LISTENER_ ADDRESS)	9100	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9901	
SOAP Connector Port (SOAP_ CONNECTOR_ ADDRESS)	8879	

Version 6.0.x port numbers

Table 5. Port definitions for WebSphere Application Server Version 6.0.x

Port Name	Default Value		Files
	Application Server	Deployment Manager	
HTTP_ TRANSPORT	9080	----	serverindex.xml and virtualhosts.xml
HTTP_ TRANSPORT_ ADMIN	9060	9060	
HTTPS_ TRANSPORT	9443	----	
HTTPS_ TRANSPORT_ ADMIN	9043	9043	
BOOTSTRAP_ ADDRESS	2809	9809	serverindex.xml
SOAP_ CONNECTOR_ ADDRESS	8880	8879	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9401	9401	
CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9403	9403	
CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS	9402	9402	
ORB_ LISTENER_ ADDRESS	9100	9100	
DCS_ UNICAST_ ADDRESS	9353	9352	
SIB_ ENDPOINT_ ADDRESS	7276	7276	
SIB_ ENDPOINT_ SECURE_ ADDRESS	7286	7286	
SIB_ MQ_ ENDPOINT_ ADDRESS	5558	5558	
SIB_ MQ_ ENDPOINT_ SECURE_ ADDRESS	5578	5578	
Internal JMS Server (JMSSERVER_ SECURITY_ PORT)	5557	----	
DRS_ CLIENT_ ADDRESS **Note:** This port is deprecated and is no longer used in the current version of WebSphere Application Server.	7873	7989	
IBM HTTP Server Port	80	----	virtualhosts.xml, plugin-cfg.xml, and *web_ server_ root*/conf/ httpd.conf
IBM HTTPS Server Administration Port	8008	----	*web_ server_ root*/conf/ admin.conf

Table 5. Port definitions for WebSphere Application Server Version 6.0.x (continued)

Port Name	Default Value		Files
	Application Server	Deployment Manager	
CELL_ DISCOVERY_ ADDRESS	----	7277	serverindex.xml
CELL_ MULTICAST_ DISCOVERY_ ADDRESS	----	7272	
NODE_ MULTICAST_ IPV6_ DISCOVERY_ ADDRESS	5001	5001	

When you federate an application server node into a deployment manager cell, the deployment manager instantiates the node agent server process on the application server node. The node agent server uses these port assignments by default.

Table 6. Port definitions for the Version 6.0.x node agent server process

Port Name	Default Value	File
BOOTSTRAP_ ADDRESS	2809	serverindex.xml
ORB_ LISTENER_ ADDRESS	9900	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9901	
CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS	9202	
CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9201	
NODE_ DISCOVERY_ ADDRESS	7272	
NODE_ MULTICAST_ DISCOVERY_ ADDRESS	5000	
NODE_ IPV6_ MULTICAST_ DISCOVERY_ ADDRESS	5001	
DCS_ UNICAST_ ADDRESS	9354	
DRS_ CLIENT_ ADDRESS	7888	
SOAP_ CONNECTOR_ ADDRESS	8878	

Version 5.1.x port numbers

Table 7. Port definitions for WebSphere Application Server Version 5.1.x

Port Name	Default Value		Files
	WebSphere Application Server	Network Deployment	
HTTP_ TRANSPORT	9080	----	server.xml and virtualhosts.xml
HTTPS_ TRANSPORT	9443	----	
HTTP_ TRANSPORT_ ADMIN	9090	9090	
HTTPS_ TRANSPORT_ ADMIN	9043	9043	
JMSSERVER_ SECURITY_ PORT	5557	----	server.xml
JMSSERVER_ QUEUED_ ADDRESS	5558	----	serverindex.xml
JMSSERVER_ DIRECT_ ADDRESS	5559	----	
BOOTSTRAP_ ADDRESS	2809	9809	
SOAP_ CONNECTOR_ ADDRESS	8880	8879	
DRS_ CLIENT_ ADDRESS	7873	7989	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	0	9401	
CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	0	9403	
CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS	0	9402	
IBM HTTP Server Port	80	----	virtualhosts.xml, plugin-cfg.xml, and *web_ server_ root*/conf/ httpd.conf

Table 7. Port definitions for WebSphere Application Server Version 5.1.x (continued)

Port Name	Default Value		Files
	WebSphere Application Server	Network Deployment	
IBM HTTPS Server Administration Port	8008	----	*web_ server_ root*/conf/ admin.conf
CELL_ DISCOVERY_ ADDRESS	----	7277	serverindex.xml
ORB_ LISTENER_ ADDRESS	9100	9100	
CELL_ MULTICAST_ DISCOVERY_ ADDRESS	----	7272	

When you federate an application server node into a deployment manager cell, the deployment manager instantiates the node agent server process on the application server node. The node agent server uses these port assignments by default.

Table 8. Port definitions for the Version 5.1.x node agent server process

Port Name	Default Value	File
BOOTSTRAP_ ADDRESS	2809	serverindex.xml
ORB_ LISTENER_ ADDRESS	9900	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9901	
CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS	9101	
CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9201	
NODE_ DISCOVERY_ ADDRESS	7272	
NODE_ MULTICAST_ DISCOVERY_ ADDRESS	5000	
DRS_ CLIENT_ ADDRESS	7888	
SOAP_ CONNECTOR_ ADDRESS	8878	

Chapter 2. Creating and deleting profiles

You can create and delete profiles, which are sets of files that define the runtime environment. At least one profile must exist to run the product.

Before you begin

This task assumes a basic familiarity with the **manageprofiles** command, the Profile Management Tool, system commands, and profile concepts.

About this task

Typically, you create a profile when you install the product. Depending on which WebSphere Application Server product you have, you might create additional profiles.

You can create profiles through the Profile Management Tool or the **manageprofiles** command.

You cannot use the Profile Management Tool to create profiles on the following platforms:

- 64-bit platforms
- Linux The Linux® for zSeries® platform for 64-bit or 31-bit

For the Network Deployment product, you can create any combination of profiles.

Linux HP-UX Solaris AIX Non-root users can create their own profiles so that they can manage their own application servers. Typically, non-root users manage application servers for development purposes.

You can delete profiles through the **manageprofiles** command or by other means if necessary. You might delete a profile if the configuration that you specified in the profile is not what you want.

Perform any of the following tasks to create or delete profiles.

- Create profiles through the Profile Management Tool.

  Read about how to create profiles for WebSphere Application Server through the Profile Management Tool.
- Create profiles through commands.

  Read about how to set up and use the profile environment through commands.
- Create profiles for non-root users.

  Read about how to give a non-root user permissions for files and directories so that the non-root user can create a profile.
- Delete a profile.

  Read about how to delete a profile with and without the **manageprofiles** command.

Results

You might have created or deleted a profile depending on the tasks you completed.

What to do next

What do next depends on the tasks you completed. You could create or delete profiles, start a server, or proceed to other tasks, such as deploying an application.

Profile concepts

A profile defines the runtime environment. The profile includes all of the files that the server processes in the runtime environment and that you can change.

You can create a runtime environment either through the **manageprofiles** command or the Profile Management Tool graphical user interface. You can use the Profile Management Tool to enter most of the parameters that are described in this topic. Some parameters, however, require you to use the **manageprofiles** command. You must use the **manageprofiles** command to delete a profile, for instance, because the Profile Management Tool does not provide a deletion function. You can use either the Profile Management Tool or the **manageprofiles** command to create a cell profile. The Profile Management Tool creates the cell in a single step, whereas the **manageprofiles** command requires two separate invocations.

Core product files

The core product files are the shared product binaries, which are shared by all profiles.

The directory structure for the product has the following two major divisions of files in the installation root directory for the product:

- The core product files are shared product binary files that do not change unless you install a refresh pack, a fix pack, or an interim fix. Some log information is also updated.

  The following list shows default installation locations for root users on supported platforms:

  – AIX /usr/IBM/WebSphere/AppServer

  – Linux HP-UX Solaris /opt/IBM/WebSphere/AppServer

  – Windows C:\Program Files\IBM\WebSphere\AppServer
- The *app_server_root*/profiles directory is the default directory for creating profiles.

When you want binaries at different service levels, you must use a separate installation of the product for each service level.

The configuration for every defined application server process is within the profiles directory unless you specify a new directory when you create a profile. These files change as often as you create a new profile, reconfigure an existing profile, or delete a profile.

Each of the folders except for the profiles directory and a few others such as the logs directory and the properties directory do not change, unless you install service fixes. The profiles directory, however, changes each time you add, change, or delete a profile. The profiles directory is the default repository for profiles. However, you can put a profile anywhere on the machine or system, provided enough disk space is available.

If you create a profile in another existing folder in the installation root directory, then a risk exists that the profile might be affected by the installation of a service fix that applies maintenance to the folder. Use a directory outside of the installation root directory when using a directory other than the profiles directory for creating profiles.

Why and when to create a profile

The **manageprofiles** command-line tool defines each profile for the product.

Run the Profile Management Tool or the **manageprofiles** command each time that you want to create a profile. A need for more than one profile on a machine is common.

Administration is greatly enhanced when using profiles instead of multiple product installations. Not only is disk space saved, but updating the product is simplified when you maintain a single set of product core files. Also, creating new profiles is more efficient and less prone to error than full product installations, allowing a developer to create separate profiles of the product for development and testing.

You can run the Profile Management Tool or the command-line tool to create a new profile on the same machine as an existing profile. Define unique characteristics, such as profile name and node name, for the new profile. Each profile shares all runtime scripts, libraries, the Java™ SE Runtime Environment 6 (JRE 6) environment, and other core product files.

Profile types

Templates for each profile are located in the *app_server_root*/profileTemplates directory.

Multiple directories exist within this directory, which correspond to different profile types and vary with the type of product that is installed. The directories are the paths that you indicate while using the **manageprofiles** command with the -templatePath option. You can also specify profile templates that exist outside the profileTemplates directory, if you have any.

See the -templatePath parameter description in the Manageprofiles command topic in the *Using the administrative clients* PDF for more information.

The **manageprofiles** command in the Network Deployment product can create the following types of profiles:

Management profile with a deployment manager server
The basic function of the deployment manager is to deploy applications to a cell of application servers, which it manages. Each application server that belongs to the cell is a *managed node*.

You can create the management profile with a deployment manager server using the Profile Management Tool or the **manageprofiles** command. If you create the profile with the **manageprofiles** command, specify *app_server_root*/profileTemplates/management for the -templatePath parameter and DEPLOYMENT_MANAGER for the -serverType parameter.

Management profile with an administrative agent server
The basic function of the administrative agent is to provide a single interface to administer multiple unfederated application servers.

You can create the profile using the Profile Management Tool or the **manageprofiles** command. If you create the profile with the **manageprofiles** command, specify *app_server_root*/profileTemplates/management for the -templatePath parameter and ADMIN_AGENT for the -serverType parameter to create this type of management profile.

Management profile with a job manager server
The basic function of the job manager is to provide a single console to administer multiple base servers, multiple deployment managers, and do asynchronous job submission.

You can create the profile using the Profile Management Tool or the **manageprofiles** command. If you create the profile with the **manageprofiles** command, specify *app_server_root*/profileTemplates/management for the -templatePath parameter and JOB_MANAGER for the -serverType parameter to create this type of management profile.

Application server profile
Use the application server to make applications available to the Internet or to an intranet.

An important product feature is the ability to scale up a stand-alone application server profile by adding the application server node into a deployment manager cell. Multiple application server processes in a cell can deploy an application that is in demand. You can also remove an application server node from a cell to return the node to the status of a stand-alone application server.

Each stand-alone application server can optionally have its own administrative console application, which you use to manage the application server. You can also use the wsadmin scripting facility to perform every function that is available in the administrative console application.

No node agent process is available for a stand-alone application server node unless you decide to add the application server node to a deployment manager cell. Adding the application server node to a cell is known as *federation*. Federation changes the stand-alone application server node into a managed node. You use the administrative console of the deployment manager to manage the node. If you remove the node from the deployment manager cell, then use the administrative console and the scripting interface of the stand-alone application server node to manage the process.

You can create the profile using the Profile Management Tool or the **manageprofiles** command. If you create the profile with the **manageprofiles** command, specify *app_server_root*/profileTemplates/default for the -templatePath parameter to create this type of profile.

Cell profile

Use the cell profile to make applications available to the Internet or to an intranet under the management of the deployment manager.

Creation of a cell profile generates a deployment manager and a federated node in one iteration through the Profile Management Tool. The result is a fully functional cell on a given system.

To create a cell profile using the **manageprofiles** command, you must create two portions of the profile: the cell deployment manager portion and the cell node portion. Additionally, you can have only one cell deployment manager and one cell node associated with each other when you create a cell. The initial cell profile that you create with the **manageprofiles** command is equivalent to the cell profile you create with the Profile Management Tool. After you create the initial cell profile, you can create custom profiles or stand-alone profiles and federate the profiles into the deployment manager.

On the **manageprofiles** command, specify *app_server_root*/profileTemplates/cell/dmgr for the -templatePath parameter for the deployment manager and *app_server_root*/profileTemplates/cell/default for the -templatePath parameter for the cell node.

After you create the two portions that make up the cell profile, you have a deployment manager and federated node. The federated node contains an application server and the default application, which contains the snoop servlet, the HitCount application, and the HelloHTML servlet.

Custom profile

Use the custom profile which belongs to a deployment manager cell, to make applications available to the Internet or to an intranet under the management of the deployment manager.

The deployment manager converts a custom profile to a managed node by adding the node into the cell. The deployment manager also converts an application server node into a managed node when you add an application server node into a cell. When either node is added to a cell, the node becomes a managed node. The node agent process is then instantiated on the managed node. The node agent acts on behalf of the deployment manager to control application server processes on the managed node. The node agent can start or stop application servers, for example.

A deployment manager can create multiple application servers on a managed node so long as the node agent process is running. Processes on the managed node can include cluster members that the deployment manager uses to balance the workload for heavily used applications.

Use the administrative console of the deployment manager to control all of the nodes that the deployment manager manages. You can also use the wsadmin scripting facility of the deployment manager to control any of the managed nodes. A custom profile does not have its own administrative console or scripting interface. You cannot manage the node directly with the wsadmin scripting facility.

A custom profile does not include default applications or a default server like the application server profile includes. A custom profile is an empty node. Add the node to the deployment manager cell. Then, you can use the administrative interface of the deployment manager to customize the managed node by creating clusters and application servers.

You can create the profile using the Profile Management Tool or the **manageprofiles** command. If you create the profile with the **manageprofiles** command, specify *app_server_root*/profileTemplates/managed for the -templatePath parameter to create this type of profile.

Secure proxy profile

Use the secure proxy server to take requests from the Internet and forward them to application servers. The secure proxy server resides in the DMZ.

You can create the profile using the Profile Management Tool or the **manageprofiles** command. If you create the profile with the **manageprofiles** command, specify *app_server_root*/profileTemplates/secureproxy for the -templatePath parameter to create this type of profile.

Default profiles

Profiles use the concept of a default profile when more than one profile exists. The default profile is set to be the default target for scripts that do not specify a profile. You can use the -profileName parameter with most of the scripts to enable the scripts to act on a profile other than the default profile.

The default profile name is *profileTypeProfileName*:

- *profileType* is a value of AppSrv, Dmgr, Custom, AdminAgent, JobMgr, or SecureProxySrv.
- *ProfileName* is a sequential number that is used to create a unique profile name.

Security policy for application server profiles

In environments where you plan to have multiple stand-alone application servers, the security policy of each application server profile is independent of the others. Changes to the security policy in one application server profile are not synchronized with the other profiles.

Installed file set

You decide where to install the files that define a profile.

The default location is in the profiles directory in the installation root directory. You can change the location on the Profile Management Tool or in a parameter when using the command line tool. For example, assume that you create two profiles on a Linux platform with host name devhost1. The profile directories resemble the following example if you do not relocate them:

```
/opt/IBM/WebSphere/AppServer/profiles/AppSrv01
/opt/IBM/WebSphere/AppServer/profiles/AppSrv02
```

You can specify a different directory, such as /opt/profiles for the profile directory using the**manageprofiles** command. For example:

```
manageprofiles.sh
   -profileName AppSrv01
   -profilePath /opt/profiles

manageprofiles.sh
   -profileName AppSrv02
   -profilePath /opt/profiles
```

Then the profile directories resemble the directories shown in the following example:

```
/opt/profiles/AppSrv01
/opt/profiles/AppSrv02
```

The following directories exist within a typical profile. This example assumes that the profile, AppSrv01, exists:

- *app_server_root*/profiles/AppSrv01/bin
- *app_server_root*/profiles/AppSrv01/config
- *app_server_root*/profiles/AppSrv01/configuration
- *app_server_root*/profiles/AppSrv01/etc
- *app_server_root*/profiles/AppSrv01/firststeps
- *app_server_root*/profiles/AppSrv01/installableApps
- *app_server_root*/profiles/AppSrv01/installedApps
- *app_server_root*/profiles/AppSrv01/installedConnectors
- *app_server_root*/profiles/AppSrv01/installedFilters
- *app_server_root*/profiles/AppSrv01/logs
- *app_server_root*/profiles/AppSrv01/properties
- *app_server_root*/profiles/AppSrv01/samples
- *app_server_root*/profiles/AppSrv01/temp
- *app_server_root*/profiles/AppSrv01/wstemp

Profiles: file system requirements

A minimum amount of space must be available in the directory where you create a profile.

An error can occur when you do not provide enough space to create a profile. Verify that you have, in addition to the minimum space required for a particular profile, an additional 40 MB of space. The 40 MB of space is used for log files and temporary files.

Table 9. Space requirements

Profile or server type	Space required
Application server	200 MB
Deployment manager	30 MB
Administrative agent	30 MB
Job manager	30 MB
Custom	10 MB
Cell	230 MB
Secure proxy	5 MB

Situations in which you could have insufficient file system space

The Profile Management Tool and the **manageprofiles** command check that the amount of file system space needed to create the profile is available right before profile creation begins. However, a slight chance exists that the profile creation can fail due to a lack of file system space. This failure can occasionally occur in the following situations:

- Another user performs an action, such as copying files, that occupies file system space at the same time that either the Profile Management Tool or the **manageprofiles** command writes to the file system.
- Another program writes to the disk at the same time that either the Profile Management Tool or the **manageprofiles** command writes to it to create a profile.
- The Profile Management Tool writes its logs and the profile that it creates to the same file system at the same time.

- The **manageprofiles** command writes its logs and the profile that it creates to the same file system at the same time.

Use the following recommendations to avoid profile creation failure:

- Ensure that enough temporary space is allocated for profile creation. Some temporary space is needed for the profile creation logs. These logs can be on a different file system than the file system on which the profile is created.
- Ensure no other program writes to the file system space when either the Profile Management Tool or the **manageprofiles** command creates the profile.
- Ensure no user performs actions that occupy the file system space when either the Profile Management Tool or the **manageprofiles** command creates the profile.

Differences between the manageprofiles command and the Profile Management tool when creating cell profiles

Both the **manageprofiles** command and the Profile Management tool can create a cell profile that has both a federated application server profile and a deployment manager profile. However, the Profile Management tool and the **manageprofiles** command create cell profiles differently. The differences are important to understand in terms of the available file system space needed to create the cell profiles. You can create a cell profile in one pass through the Profile Management tool. In this case, you need 230 MB of available file system space to create the cell profile. However, to create a cell profile using the **manageprofiles** command that is equivalent to the cell profile that the Profile Management tool creates, you must create two individual profiles, the cell deployment manager profile and the cell node profile. The cell deployment manager profile requires 30 MB of available file system space, while the cell node profile requires 200 MB of available file system space.

Setting up and using the profile environment through commands

Use commands to create a profile, start the server of the profile, display the profile ports, and open the administrative console.

Before you begin

This task assumes a basic familiarity with the **manageprofiles** command, other application server commands, and system commands.

Before you can create and use a profile, you must install the product.

About this task

Perform the following steps to create a profile, start the server of the profile, display ports for your profile, and open the administrative console for your server.

This example deals with the profile environment of a stand-alone application server.

1. Create the server profile from the original installation:

   Linux HP-UX Solaris AIX *app_server_root*/bin/manageprofiles.sh

   Windows *app_server_root*\bin\manageprofiles.bat

   Assume that you create the profile by using the defaults. The following script is an example for creating an application server profile:

   Windows

   ```
   app_server_root\bin\manageprofiles.bat -create
   -templatePath app_server_root\profileTemplates\default
   ```

(The script is displayed on multiple lines for printing purposes.)

Linux HP-UX Solaris AIX

```
app_server_root/bin/manageprofiles.sh -create
-templatePath app_server_root/profileTemplates/default
```

(The script is displayed on multiple lines for printing purposes.)

2. Windows Change directories to the bin directory of the new server profile.

   For example, issue the following command:

   ```
   cd profile_root\bin
   ```

3. Start the server.

   Windows Change directories to the *app_server_root*\bin directory of the original installation.

   Issue the startServer command.

   Windows

   ```
   startServer.bat server1 -profileName profile_name
   ```

   Linux HP-UX Solaris AIX

   ```
   startServer.sh server1 -profileName profile_name
   ```

4. Display the ports.

   These are the ports assigned during profile creation.

   Windows Open the portdef.props file in the *profile_root*\properties directory.

   Linux HP-UX Solaris AIX Open the portdef.props file in the *profile_root*/properties directory.

5. Open the administrative console.

   The server1 administrative console is defined on the WC_adminhost setting for the non-secure administrative console port, or the WC_adminhost_secure setting for the secure administrative console port. If the value of the setting is 20003, then specify the following Web address in your browser:

   ```
   http://hostname_or_IP_address:20003/ibm/console/
   ```

Results

You created an application server profile, started an application server, and accessed the administrative console through commands.

What to do next

Deploy an application.

Creating profiles using the graphical user interface

You can create profiles, which define runtime environments, using the Profile Management Tool. Using profiles instead of multiple product installations saves disk space and simplifies updating the product because a single set of core product files is maintained.

Before you begin

Before you use the Profile Management Tool, install the product files.

The Profile Management Tool is the graphical user interface for the **manageprofiles** command. See the description of the **manageprofiles** command for more information.

You must provide enough system temporary space to create a profile. For information, read about the file system requirements for profiles.

You cannot use the Profile Management Tool to create profiles on the following platforms:

- 64-bit platforms
- Linux The Linux for zSeries platform for 64-bit or 31-bit

About this task

You can have the installation procedure create a default profile. After installing the core product files for the Network Deployment product, use the Profile Management Tool or the manageprofiles command to create additional profiles.

- Create a cell profile.

  With a cell profile, you can create a deployment manager profile and a profile for a federated application server node in a single pass through the Profile Management tool. Use the cell profile creation option to create the deployment manager profile and the federated application server node profile, unless you have a specific reason to create them separately.
- Create a management profile with a deployment manager server.

  With a deployment manager you can create the administrative node for a multinode, multi-machine group of application server nodes that you create later. This logical group of application server processes is known as a *cell*.
- Create a management profile with an administrative agent server.

  You can create a management profile for the administrative agent to administer multiple application servers that run customer applications only. The administrative agent provides a single administrative console to administer the application servers.
- Create a management profile with a job manager server.

  You can create a management profile for the job manager to coordinate administrative actions among multiple deployment managers, administer multiple unfederated application servers, asynchronously submit jobs to start servers, and a variety of other tasks.
- Create an application server profile.

  Create an application server profile so that you can make applications available to the Internet or to an intranet, typically using Java technology.
- Create a custom profile.

  A custom profile is an empty node that you can customize through the deployment manager to include application servers, clusters, or other Java processes, such as a messaging server. Create a custom profile on a distributed machine and add the node into the deployment manager cell to get started customizing the node.
- Create a secure proxy profile.

  You can create a secure proxy profile to serve as the initial point of entry into your enterprise environment. Typically, a secure proxy server exists in the DMZ, accepts requests from clients on the Internet, and forwards the requests to servers in your enterprise environment.

Results

You have created one or more profiles using the Profile Management Tool.

What to do next

See the description of the **manageprofiles** command to learn more about the command-line alternative method of creating a profile and to see examples of using the command.

Read about planning for installation for examples of configurations that you can create by creating profiles.

Creating a management profile with a deployment manager server

You can create a management profile for the deployment manager to administer servers within the deployment manager cell. Use the Profile Management Tool to create the profile.

Before you begin

Before you use the Profile Management Tool, install the product files.

The Profile Management Tool is the graphical user interface for the **manageprofiles** command. See the description of the **manageprofiles** command for more information.

You must provide enough system temporary space to create a profile. For information, read about the file system requirements for profiles.

Note: When you launch the Profile Management Tool, the tool could lock up in the following situation for a non-root user: Log into a machine as root, use the SetPermissions utility to change the user from *x* to *y*. Assume that you are user *x* and log back into the machine. Launch the Profile Management Tool, click **Profile Management Tool**, and click **Create**. The next click after the click on **Create** could lock up the tool.

Note: When you use the Profile Management Tool with the Motif graphical user interface on the Solaris operating system, the default size of the Profile Management Tool might be too small to view all the messages and buttons of the Profile Management Tool. To fix the problem, add the following lines to the *app_server_root*/.Xdefaults file:

```
Eclipse*spacing:0
Eclipse*fontList:-misc-fixed-medium-r-normal-*-10-100-75-75-c-60-iso8859-1
```

After adding the lines, run the following command before launching the Profile Management Tool:

```
xrdb -load user_home/.Xdefaults
```

About this task

After installing the core product files for the Network Deployment product, you must create a profile. This procedure describes creating a management profile with a deployment manager using the graphical user interface that is provided by the Profile Management Tool. You can also use the **manageprofiles** command to create a management profile with a deployment manager. See the description of the manageprofiles command for more information.

The deployment manager provides a single administrative interface for a logical group of application servers on one or more machines.

You can create profiles with the Profile Management Tool using the typical profile creation process or the advanced profile creation process. The typical profile creation process uses default settings and assigns unique port values. You can optionally set values as allowed. For the advanced profile creation process you can accept the default values, or specify your own values.

1. Start the Profile Management Tool to create a new runtime environment.

   You can use one of the following ways to start the tool.

   - At the end of installation, select the check box to launch the Profile Management Tool.
   - Issue the command directly from a command prompt.

     The command is in the following directory:

     – Linux HP-UX Solaris AIX *app_server_root*/bin/ProfileManagement

     – Windows *app_server_root*\bin\ProfileManagement

     The name of the command varies per platform:

– Linux HP-UX Solaris AIX pmt.sh
– Windows pmt.bat

- Select the Profile Management Tool option from the First steps console.
- Windows Use the **Start** menu to access the Profile Management Tool. For example, click **Start > Programs** or **All Programs > IBM WebSphere > *your_product* > Profile Management Tool**.
- Linux Use the Linux operating system menus used to start programs to start the Profile Management Tool. For example, click ***the_operating_system_menus_to_access_programs* > IBM WebSphere > *your_product* > Profile Management Tool**.

2. Click **Launch Profile Management Tool**, and then click **Create** on the Profiles tab to create a new profile.

   The Profiles tab contains a list of profiles that have been created on your machine. No action can be done on a selected profile unless the profile can be augmented. The Augment button is greyed out unless a profile that you select can be augmented.

   The tool displays the Environment selection panel.
3. Select **Management** and click **Next**.

   The Server type selection panel is displayed.
4. Select **Deployment manager** and click **Next**.

   The Profile creation options panel is displayed.
5. Select either **Typical profile creation** or **Advanced profile creation**, and click **Next**.

   The **Typical profile creation** option creates a profile that uses default configuration settings. With the **Advanced profile creation** option, you can specify your own configuration values for a profile.
6. If you selected **Typical profile creation**, go to the step on administrative security.
7. If you selected **Advanced profile creation**, optionally select to deploy the administrative console, then click **Next**.

   The wizard displays the Profile name and location panel.
8. Specify a name for the profile and the directory path for the profile directory, or accept the default values. Then, click **Next**.

   Profile naming guidelines: Double-byte characters are supported. The profile name can be any unique name with the following restrictions. Do not use any of the following characters when naming your profile:

   - Spaces
   - Special characters that are not supported within the name of a directory on your operating system, such as `*&?`
   - Slashes (/) or (\)

   The default profile

   The first profile that you create on a machine is the default profile. The default profile is the default target for commands that are issued from the `bin` directory in the product installation root. When only one profile exists on a machine, every command works on the single server process in the configuration. You can make another profile the default profile when you create that profile by checking **Make this profile the default** on the Profile name and location panel of the **Advanced profile creation** path. You can also make another profile the default profile using the manageprofiles command after you create the profile.

   Addressing a profile in a multiprofile environment

   When multiple profiles exist on a machine, certain commands require that you specify the profile to which the command applies if the profile is not the default profile. These commands use the -profileName parameter to identify which profile to address. You might find it easier to use the commands that are in the `bin` directory of each profile.

Use these commands to query the command shell to determine the calling profile and to address these commands to the calling profile.

Default profile information

The default profile name is *profileTypeProfileName*:

- *profileType* is a value of `AppSrv`, `Dmgr`, `Custom`, `AdminAgent`, `JobMgr`, or `SecureProxySrv`.
- *ProfileName* is a sequential number that is used to create a unique profile name.

AIX HP-UX Linux Solaris The default profile directory is *app_server_root*/profiles, where *app_server_root* is the installation root.

Windows The default profile directory is *app_server_root*\profiles, where *app_server_root* is the installation root.

9. On the Node, host, and cell names panel, specify a unique node name, the actual host name of the machine, and a unique cell name. Click **Next**.

   The deployment manager node has the following characteristics.

   Some default values in the following table are split on multiple lines for printing purposes.

Field Name	Default Value	Constraints	Description
Node name	*shortHostName* CellManager *NodeNumber* where: • *shortHostName* is the short host name. • *NodeNumber* is a sequential number starting at 01.	Use a unique name for the deployment manager.	The name is used for administration within the deployment manager cell.
Host name	The long form of the domain name server (DNS) name.	The host name must be addressable through your network. Read about Host name considerations.	Use the actual DNS name or IP address of your machine to enable communication with your machine. See additional information about the host name that follows this table.
Cell name	*shortHostName* Cell *CellNumber* where: • *shortHostName* is the short host name. • *CellNumber* is a sequential number starting at 01.	Use a unique name for the deployment manager cell. If you plan to migrate a Version 5 deployment manager cell or a Version 6 deployment manager cell to this Version 7, use the same cell name as the Version 5 or Version 6. A cell name must be unique in any circumstance in which the product is running on the same physical machine or cluster of machines, such as a sysplex. Additionally, a cell name must be unique in any circumstance in which network connectivity between entities is required either between the cells or from a client that must communicate with each of the cells. Cell names must also be unique if their namespaces are federated. Otherwise, you might encounter symptoms such as a javax.naming.NameNotFoundException error, in which case, create uniquely named cells.	All federated nodes become members of the deployment manager cell, which you name in this panel.

Reserved names: Avoid using reserved folder names as field values. The use of reserved folder names can cause unpredictable results. The following terms are reserved folder names:

- cells
- nodes
- servers
- clusters
- applications
- deployments

Directory path considerations

Windows The number of characters in the *profiles_directory_path**profile_name* directory must be less than or equal to 80 characters.

Host name considerations

The host name is the network name for the physical machine on which the node is installed. The host name must resolve to a physical network node on the server. When multiple network cards exist in the server, the host name or IP address must resolve to one of the network cards. Remote nodes use the host name to connect to and communicate with this node. Selecting a host name that other machines can reach within your network is important. Do not use the generic identifier, `localhost`, for this value. Also, do not attempt to install WebSphere Application Server products on a machine with a host name that uses characters from a double-byte character set (DBCS). DBCS characters are not supported when used in the host name.

If you define coexisting nodes on the same computer with unique IP addresses, then define each IP address in a domain name server (DNS) look-up table. Configuration files for stand-alone application servers do not provide domain name resolution for multiple IP addresses on a machine with a single network address.

The value that you specify for the host name is used as the value of the hostName property in configuration documents for the stand-alone application server. Specify the host name value in one of the following formats:

- Fully qualified domain name server (DNS) host name string, such as `xmachine.manhattan.ibm.com`
- The default short DNS host name string, such as `xmachine`
- Numeric IP address, such as `127.1.255.3`

The fully qualified DNS host name has the advantages of being unambiguous and flexible. You have the flexibility of changing the actual IP address for the host system without having to change the application server configuration. This value for the host name is particularly useful if you plan to change the IP address frequently when using Dynamic Host Configuration Protocol (DHCP) to assign IP addresses. A disadvantage of this format is dependency on DNS. If DNS is not available, then connectivity is compromised.

The short host name is also dynamically resolvable. A short name format has the added function of being redefined in the local hosts file so that the system can run the application server, even when disconnected from the network. To run disconnected, define the short name as the loopback address, `127.0.0.1`, in the hosts file to run disconnected. A disadvantage of this format is a dependency on DNS for remote access. If DNS is not available, then connectivity is compromised.

A numeric IP address has the advantage of not requiring name resolution through DNS. A remote node can connect to the node that you name with a numeric IP address without DNS being available. A disadvantage of this format is that the numeric IP address is fixed. You must change the setting of the hostName property in Express configuration documents whenever you change the machine IP address. Therefore, do not use a numeric IP address if you use DHCP, or if you change IP addresses regularly. Another disadvantage of this format is that you cannot use the node if the host is disconnected from the network.

After displaying deployment manager characteristics, the wizard displays the Administrative security panel.

10. Optionally enable administrative security, and click **Next**.

You can enable administrative security now during profile creation, or later from the console. If you enable administrative security now, then enter a user name and password to log onto the administrative console.

After specifying security characteristics, the tool displays the Security certificate panel if you previously selected **Advanced profile creation**.

11. If you selected **Typical profile creation** at the beginning of these steps, then go to the step that displays the Profile summary panel.
12. Create a default personal certificate and a root signing certificate, or import a personal certificate and a root signing certificate from keystore files, and click **Next**.

    You can create both certificates, import both certificates, or create one certificate, and import the other certificate.

    Note: When you import a personal certificate as the default personal certificate, import the root certificate that signed the personal certificate. Otherwise, the Profile Management Tool adds the signer of the personal certificate to the trust.p12 file.

    If you import the default personal certificate or the root signing certificate, specify the path and the password, and select the keystore type and the keystore alias for each certificate that you import.
13. Verify that the certificate information is correct, and click **Next**.

    If you create the certificates, you can use the default values or modify them to create new certificates. The default personal certificate is valid for one year by default and is signed by the root signing certificate. The root signing certificate is a self-signed certificate that is valid for 20 years by default. The default keystore password for the root signing certificate is `WebAS`. You should change the password. The password cannot contain any double-byte character set (DBCS) characters because certain keystore types, including PKCS12, do not support these characters. The keystore types that are supported depend on the providers in the java.security file.

    When you create either or both certificates, or import either or both certificates, the keystore files that are created are key.p12, trust.p12, root-key.p12, default-signers.p12, deleted.p12, and ltpa.jceks. These files all have the same password when you create or import the certificates, which is either the default password, or a password that you specify. The key.p12 file contains the default personal certificate. The trust.p12 file contains the signer certificate from the default root certificate. The root-key.p12 file contains the root signing certificate. The default-signer.p12 file contains signer certificates that are added to any new keystore file that you create after the server is installed and running. By default, the default root certificate signer and a DataPower® signer certificate is in the default-signer.p12 keystore file. The deleted.p12 keystore file is used to hold certificates deleted with the deleteKeyStore task so that they can be recovered if needed. The ltpa.jceks file contains server default Lightweight Third-Party Authentication (LTPA) keys that the servers in your environment use to communicate with each other.

    An imported certificate is added to the key.p12 file or the root-key.p12 file.

    If you import any certificates and the certificates do not contain the information that you want, click **Back** to import another certificate.

    After displaying the Security certificate panels, the tool displays the Ports panel if you previously selected **Advanced profile creation**.
14. Verify that the ports within the deployment manager profile are unique, or intentionally conflicting, and click **Next**.

    If you chose not to deploy the administrative console, then the administrative console ports are disabled on the Ports panel.

    Port conflict resolution

    Ports are recognized as being in use if one of the following conditions exists:

    - The ports are assigned to a profile created from an installation that is performed by the current user.
    - The port is currently in use.

Validation of ports occurs when you access the Port value assignment panel. Conflicts can still occur between the Port value assignment panel and the Profile creation complete panel because ports are not assigned until profile creation completes.

If you suspect a port conflict, then you can investigate the port conflict after the profile is created. Determine the ports that are used during profile creation by examining the following files.

- Linux HP-UX Solaris AIX *profile_root*/properties/portdef.props file
- Windows *profile_root*\properties\portdef.props file

Included in this file are the keys and values that are used in setting the ports. If you discover ports conflicts, then you can reassign ports manually. To reassign ports, run the updatePorts.ant file by using the ws_ant script.

Windows Linux The tool displays the Windows® service definition panel if you are installing on a Windows operating system and the installation ID has the administrative group privilege. The tool displays the Linux service definition panel if you are installing on a supported Linux operating system and the ID that runs the Profile Management Tool is the root user.

15. Choose whether to run the dmgr process as a Windows service on a Windows platform or as a Linux Service on a Linux platform, and click **Next**.

    The Windows service definition panel is displayed for the Windows operating system only if the ID that installs the Windows service has the administrator group privilege. However, you can run the WASService.exe command to create the Windows service as long as the installer ID belongs to the administrator group. Read about automatically restarting server processes for more information.

    Windows The product attempts to start Windows services for dmgr processes that are started by a **startManager** command. For example, if you configure a deployment manager as a Windows service and issue the **startManager** command, then the **wasservice** command attempts to start the defined service.

    If you chose to install a local system service, then you do not have to specify your user ID or password. If you create a specified user type of service, then you must specify the user ID and the password for the user who runs the service. The user must have `Log on as a service` authority for the service to run correctly. If the user does not have `Log on as a service` authority, then the Profile Management tool automatically adds the authority.

    To perform this profile creation task, the user ID must not contain spaces. In addition to belonging to the administrator group, the ID must also have the advanced user right of `Log on as a service`. The Installation wizard grants the user ID the advanced user right if the user ID does not already have the advanced user right and if the user ID belongs to the administrator group.

    You can also create other Windows services after the installation is complete to start other server processes. Read about automatically restarting server processes for more information.

    You can remove the Windows service that is added during profile creation during profile deletion. You can also remove the Windows service with the wasservice command.

    IPv6 considerations

    Profiles created to run as a Windows service fail to start when using IPv6 if the service is configured to run as *Local System*. Create a user-specific environment variable to enable IPv6. Since this environment variable is a user variable instead of a *Local System* variable, only a Windows service that runs as that specific user can access this environment variable. By default, when a new profile is created and configured to run as a Windows service, the service is set to run as *Local System*. When the Windows service for the dmgr process tries to run, the service is unable to access the user environment variable that specifies IPv6, and thus tries to start as IPv4. The server does not start correctly in this case. To resolve the problem, when creating the profile, specify that the Windows service for the dmgr process runs as the same user ID under which the environment variable that specifies IPv6 is defined, instead of as *Local System*.

    Default Windows service information

    Windows The following default values for the Windows service definition panel exist:

- The default is to run as a Windows service.
- The service process is selected to run as a system account.
- The user account is the current user name. User name requirements are the requirements that the Windows operating system imposes for a user ID.
- The startup type is `automatic`. The values for the startup type are those values that the Windows operating system imposes. If you want a startup type other than `automatic`, you can either select another available option from the menu or change the startup type after you create the profile. You can also remove the created service after profile creation, and add it later with the desired startup type. You can choose not to create a service at profile creation time and optionally create the service later with the desired startup type.

Linux The Linux service definition panel is displayed if the current operating system is a supported version of Linux operating systems, and the current user has the appropriate permissions.

The product attempts to start Linux services for application server processes that are started by a startServer command. For example, if you configure an application server as a Linux service and issue the startServer command, then the **wasservice** command attempts to start the defined service.

By default, the product is not selected to run as a Linux service.

To create the service, the user that runs the Profile Management Tool must be the root user. If you run the Profile Management Tool with a non-root user ID, then the Linux service definition panel is not displayed, and no service is created.

When you create a Linux service, you must specify a user name from which the service runs.

To delete a Linux service, the user must be the root user or have appropriate privileges for deleting the service. Otherwise, a removal script is created that the root user can run to delete the service for the user.

The wizard displays the Profile Creation Summary panel.

16. Click **Create** to create the deployment manager, or click **Back** to change the characteristics of the deployment manager.

    The Profile creation progress panel, which shows the configuration commands that are running, is displayed.

    When the profile creation completes, the tool displays the Profile creation complete panel.
17. Optionally, select **Launch the First steps console**. Click **Finish** to exit.

    With the First steps console, you can create additional profiles and start the application server.

Results

You created a deployment manager profile.

Refer to the description of the **manageprofiles** command to learn about creating a profile using a command instead of the Profile Management Tool.

What to do next

Create an application server profile or a custom profile, and add the node into the cell.

Deploy an application to get started.

Read about fast paths for the product to get started deploying applications.

Creating a management profile with an administrative agent server

You can create a management profile for the administrative agent to administer multiple application servers that run customer applications only. The administrative agent provides a single administrative console to administer the application servers.

Before you begin

Before you use the Profile Management Tool, install the product files.

The Profile Management Tool is the graphical user interface for the **manageprofiles** command. See the description of the **manageprofiles** command for more information.

You must provide enough system temporary space to create a profile. For information, read about the file system requirements for profiles.

Note: When you launch the Profile Management Tool, the tool could lock up in the following situation for a non-root user: Log into a machine as root, use the SetPermissions utility to change the user from *x* to *y*. Assume that you are user *x* and log back into the machine. Launch the Profile Management Tool, click **Profile Management Tool**, and click **Create**. The next click after the click on **Create** could lock up the tool.

Note: When you use the Profile Management Tool with the Motif graphical user interface on the Solaris operating system, the default size of the Profile Management Tool might be too small to view all the messages and buttons of the Profile Management Tool. To fix the problem, add the following lines to the *app_server_root*/.Xdefaults file:

```
Eclipse*spacing:0
Eclipse*fontList:-misc-fixed-medium-r-normal-*-10-100-75-75-c-60-iso8859-1
```

After adding the lines, run the following command before launching the Profile Management Tool:

```
xrdb -load user_home/.Xdefaults
```

About this task

After installing the core product files for the product, you must create a profile. This procedure describes creating a management profile with an administrative agent server using the graphical user interface that is provided by the Profile Management Tool. You can also use the **manageprofiles** command to create an administrative agent. See the description of the manageprofiles command for more information.

You can create profiles with the Profile Management Tool using the typical profile creation process or the advanced profile creation process. The typical profile creation process uses default settings and assigns unique port values. You can optionally set values as allowed. For the advanced profile creation process you can accept the default values, or specify your own values.

1. Start the Profile Management Tool to create a new runtime environment.

   You can use one of the following ways to start the tool.

   - At the end of installation, select the check box to launch the Profile Management Tool.
   - Issue the command directly from a command prompt.

     The command is in the following directory:

     – Linux HP-UX Solaris AIX *app_server_root*/bin/ProfileManagement

     – Windows *app_server_root*\bin\ProfileManagement

     The name of the command varies per platform:

     – Linux HP-UX Solaris AIX pmt.sh

     – Windows pmt.bat
   - Select the Profile Management Tool option from the First steps console.
   - Windows Use the **Start** menu to access the Profile Management Tool. For example, click **Start** > **Programs** or **All Programs** > **IBM WebSphere** > ***your_product*** > **Profile Management Tool**.

- Linux Use the Linux operating system menus used to start programs to start the Profile Management Tool. For example, click ***the_operating_system_menus_to_access_programs*** **> IBM WebSphere >** ***your_product*** **> Profile Management Tool**.

2. Click **Launch Profile Management Tool**, and then click **Create** on the Profiles tab to create a new profile.

   The Profiles tab contains a list of profiles that have been created on your machine. No action can be done on a selected profile unless the profile can be augmented. The Augment button is greyed out unless a profile that you select can be augmented.

   The tool displays the Environment selection panel.
3. Select **Management**, and click **Next**.

   The Server type selection panel is displayed.
4. Select **Administrative agent**. Click **Next**.

   The Profile creation options panel is displayed.
5. Select either **Typical profile creation** or **Advanced profile creation**, and click **Next**.

   The **Typical profile creation** option creates a profile that uses default configuration settings. With the **Advanced profile creation** option, you can specify your own configuration values for a profile.
6. If you selected **Typical profile creation**, go to the step on administrative security.
7. If you selected **Advanced profile creation**, optionally select to deploy the administrative console, and then click **Next**.

   If you chose not to deploy the administrative console, then the administrative console ports are disabled on the Ports panel.

   The tool displays the Profile name and location panel.
8. Specify a name for the profile and the directory path for the profile directory, or accept the default values. Then, click **Next**.

   Profile naming guidelines: Double-byte characters are supported. The profile name can be any unique name with the following restrictions. Do not use any of the following characters when naming your profile:

   - Spaces
   - Special characters that are not supported within the name of a directory on your operating system, such as `*&?`
   - Slashes (/) or (\)

   The default profile

   The first profile that you create on a machine is the default profile. The default profile is the default target for commands that are issued from the `bin` directory in the product installation root. When only one profile exists on a machine, every command works on the single server process in the configuration. You can make another profile the default profile when you create that profile by checking **Make this profile the default** on the Profile name and location panel of the **Advanced profile creation** path. You can also make another profile the default profile using the manageprofiles command after you create the profile.

   Addressing a profile in a multiprofile environment

   When multiple profiles exist on a machine, certain commands require that you specify the profile to which the command applies if the profile is not the default profile. These commands use the -profileName parameter to identify which profile to address. You might find it easier to use the commands that are in the `bin` directory of each profile.

   Use these commands to query the command shell to determine the calling profile and to address these commands to the calling profile.

   Default profile information

   The default profile name is *profileTypeProfileName*:

   - *profileType* is a value of `AppSrv`, `Dmgr`, `Custom`, `AdminAgent`, `JobMgr`, or `SecureProxySrv`.

- *ProfileName* is a sequential number that is used to create a unique profile name.

AIX HP-UX Linux Solaris The default profile directory is *app_server_root*/profiles, where *app_server_root* is the installation root.

Windows The default profile directory is *app_server_root*\profiles, where *app_server_root* is the installation root.

9. On the Node, host, and cell names panel, specify a unique node name, the actual host name of the machine, and a unique cell name. Click **Next**.

   The administrative agent node has the following characteristics.

   Some default values in the following table are split on multiple lines for printing purposes.

Field name	Default value	Constraints	Description
Node name	*shortHostName* AANode *NodeNumber* where: • *shortHostName* is the short host name. • *NodeNumber* is a sequential number starting at 01.	Use a unique name for the administrative agent.	The name is used for administration within the administrative agent cell.
Host name	The long form of the domain name server (DNS) name.	The host name must be addressable through your network. Read about Host name considerations.	Use the actual DNS name or IP address of your machine to enable communication with your machine. See additional information about the host name that follows this table.
Cell name	*shortHostName* Cell *CellNumber* where: • *shortHostName* is the short host name. • *CellNumber* is a sequential number starting at 01.	Use a unique name for the cell. If you plan to migrate a Version 5 or Version 6 cell to Version 7, use the same cell name as the Version 5 or Version 7 cell. A cell name must be unique in any circumstance in which the product is running on the same physical machine or cluster of machines, such as a sysplex. Additionally, a cell name must be unique in any circumstance in which network connectivity between entities is required either between the cells or from a client that must communicate with each of the cells. Cell names must also be unique if their namespaces are federated. Otherwise, you might encounter symptoms such as a javax.naming.NameNotFoundException error, in which case, create uniquely named cells.	All federated nodes become members of the cell, which you name in this panel.

Reserved names: Avoid using reserved folder names as field values. The use of reserved folder names can cause unpredictable results. The following terms are reserved folder names:

- cells
- nodes
- servers

- clusters
- applications
- deployments

Directory path length

Windows The number of characters in the *profiles_directory_path**profile_name* directory must be less than or equal to 80 characters.

Host name considerations

The host name is the network name for the physical machine on which the node is installed. The host name must resolve to a physical network node on the server. When multiple network cards exist in the server, the host name or IP address must resolve to one of the network cards. Remote nodes use the host name to connect to and communicate with this node. Selecting a host name that other machines can reach within your network is important. Do not use the generic identifier, `localhost`, for this value. Also, do not attempt to install WebSphere Application Server products on a machine with a host name that uses characters from a double-byte character set (DBCS). DBCS characters are not supported when used in the host name.

If you define coexisting nodes on the same computer with unique IP addresses, then define each IP address in a domain name server (DNS) look-up table. Configuration files for stand-alone application servers do not provide domain name resolution for multiple IP addresses on a machine with a single network address.

The value that you specify for the host name is used as the value of the hostName property in configuration documents for the stand-alone application server. Specify the host name value in one of the following formats:

- Fully qualified domain name server (DNS) host name string, such as `xmachine.manhattan.ibm.com`
- The default short DNS host name string, such as `xmachine`
- Numeric IP address, such as `127.1.255.3`

The fully qualified DNS host name has the advantages of being unambiguous and flexible. You have the flexibility of changing the actual IP address for the host system without having to change the application server configuration. This value for the host name is particularly useful if you plan to change the IP address frequently when using Dynamic Host Configuration Protocol (DHCP) to assign IP addresses. A disadvantage of this format is dependency on DNS. If DNS is not available, then connectivity is compromised.

The short host name is also dynamically resolvable. A short name format has the added function of being redefined in the local hosts file so that the system can run the application server, even when disconnected from the network. To run disconnected, define the short name as the loopback address, `127.0.0.1`, in the hosts file to run disconnected. A disadvantage of this format is a dependency on DNS for remote access. If DNS is not available, then connectivity is compromised.

A numeric IP address has the advantage of not requiring name resolution through DNS. A remote node can connect to the node that you name with a numeric IP address without DNS being available. A disadvantage of this format is that the numeric IP address is fixed. You must change the setting of the hostName property in Express configuration documents whenever you change the machine IP address. Therefore, do not use a numeric IP address if you use DHCP, or if you change IP addresses regularly. Another disadvantage of this format is that you cannot use the node if the host is disconnected from the network.

After displaying characteristics, the tool displays the Administrative security panel.

10. Optionally enable administrative security, and click **Next**.

    You can enable administrative security now during profile creation, or later from the console. If you enable administrative security now, then enter a user name and password to log onto the administrative console.

    After specifying security characteristics, the tool displays the Security certificate panel if you previously selected **Advanced profile creation**.

11. If you selected **Typical profile creation** at the beginning of these steps, then go to the step that displays the Profile summary panel.
12. Create a default personal certificate and a root signing certificate, or import a personal certificate and a root signing certificate from keystore files, and click **Next**.

    You can create both certificates, import both certificates, or create one certificate, and import the other certificate.

    Note: When you import a personal certificate as the default personal certificate, import the root certificate that signed the personal certificate. Otherwise, the Profile Management Tool adds the signer of the personal certificate to the trust.p12 file.

    If you import the default personal certificate or the root signing certificate, specify the path and the password, and select the keystore type and the keystore alias for each certificate that you import.
13. Verify that the certificate information is correct, and click **Next**.

    If you create the certificates, you can use the default values or modify them to create new certificates. The default personal certificate is valid for one year by default and is signed by the root signing certificate. The root signing certificate is a self-signed certificate that is valid for 20 years by default. The default keystore password for the root signing certificate is `WebAS`. You should change the password. The password cannot contain any double-byte character set (DBCS) characters because certain keystore types, including PKCS12, do not support these characters. The keystore types that are supported depend on the providers in the java.security file.

    When you create either or both certificates, or import either or both certificates, the keystore files that are created are key.p12, trust.p12, root-key.p12, default-signers.p12, deleted.p12, and ltpa.jceks. These files all have the same password when you create or import the certificates, which is either the default password, or a password that you specify. The key.p12 file contains the default personal certificate. The trust.p12 file contains the signer certificate from the default root certificate. The root-key.p12 file contains the root signing certificate. The default-signer.p12 file contains signer certificates that are added to any new keystore file that you create after the server is installed and running. By default, the default root certificate signer and a DataPower signer certificate is in the default-signer.p12 keystore file. The deleted.p12 keystore file is used to hold certificates deleted with the deleteKeyStore task so that they can be recovered if needed. The ltpa.jceks file contains server default Lightweight Third-Party Authentication (LTPA) keys that the servers in your environment use to communicate with each other.

    An imported certificate is added to the key.p12 file or the root-key.p12 file.

    If you import any certificates and the certificates do not contain the information that you want, click **Back** to import another certificate.

    After displaying the Security certificate panels, the tool displays the Ports panel if you previously selected **Advanced profile creation**.
14. Verify that the ports within the administrative agent profile are unique, or intentionally conflicting, and click **Next**.

    If you chose not to deploy the administrative console, then the administrative console ports are disabled on the Ports panel.

    Port conflict resolution

    Ports are recognized as being in use if one of the following conditions exists:

    - The ports are assigned to a profile created from an installation that is performed by the current user.
    - The port is currently in use.

    Validation of ports occurs when you access the Port value assignment panel. Conflicts can still occur between the Port value assignment panel and the Profile creation complete panel because ports are not assigned until profile creation completes.

    If you suspect a port conflict, then you can investigate the port conflict after the profile is created. Determine the ports that are used during profile creation by examining the following files.

- Linux HP-UX Solaris AIX *profile_root*/properties/portdef.props file
- Windows *profile_root*\properties\portdef.props file

Included in this file are the keys and values that are used in setting the ports. If you discover ports conflicts, then you can reassign ports manually. To reassign ports, run the updatePorts.ant file by using the ws_ant script.

Windows Linux The tool displays the Windows service definition panel if you are installing on a Windows operating system and the installation ID has the administrative group privilege. The tool displays the Linux service definition panel if you are installing on a supported Linux operating system and the ID that runs the Profile Management Tool is the root user.

15. Choose whether to run the administrative agent process as a Windows service on a Windows operating system or as a Linux service on a Linux operating system, and click **Next**.

    The Windows service definition panel is displayed for the Windows operating system only if the ID that installs the Windows service has the administrator group privilege. However, you can run the WASService.exe command to create the Windows service as long as the installer ID belongs to the administrator group. Read about automatically restarting server processes for more information.

    Windows The product attempts to start Windows services for administrative agent processes that are started by a startServer command. For example, if you configure an administrative agent as a Windows service and issue the startServer command, then the **wasservice** command attempts to start the defined service.

    If you chose to install a local system service, then you do not have to specify your user ID or password. If you create a specified user type of service, then you must specify the user ID and the password for the user who runs the service. The user must have `Log on as a service` authority for the service to run correctly. If the user does not have `Log on as a service` authority, then the Profile Management tool automatically adds the authority.

    To perform this profile creation task, the user ID must not contain spaces. In addition to belonging to the administrator group, the ID must also have the advanced user right of `Log on as a service`. The Installation wizard grants the user ID the advanced user right if the user ID does not already have the advanced user right and if the user ID belongs to the administrator group.

    You can also create other Windows services after the installation is complete to start other server processes. Read about automatically restarting server processes for more information.

    You can remove the Windows service that is added during profile creation during profile deletion. You can also remove the Windows service with the wasservice command.

    IPv6 considerations

    Profiles created to run as a Windows service fail to start when using Internet Protocol version 6 (IPv6) if the service is configured to run as local system. Create a user-specific environment variable to enable IPv6. Since this environment variable is a user variable instead of a local system variable, only a Windows service that runs as that specific user can access this environment variable. By default, when a new profile is created and configured to run as a Windows service, the service is set to run as local system. When the Windows service for the administrative agent process attempts to run, the service is unable to access the user environment variable that specifies IPv6, and thus, attempts to start as IPv4. The server does not start correctly in this case. To resolve the problem, when creating the profile, specify that the Windows service for the administrative agent process runs as the same user ID from which the environment variable that specifies IPv6 is defined, instead of as local system.

    Default Windows service information

    Windows The following default values for the Windows service definition panel exist:

    - The default is to run as a Windows service.
    - The service process is selected to run as a system account.
    - The user account is the current user name. User name requirements are the requirements that the Windows operating system imposes for a user ID.

- The startup type is `automatic`. The values for the startup type are those values that the Windows operating system imposes. If you want a startup type other than `automatic`, you can either select another available option from the menu or change the startup type after you create the profile. You can also remove the created service after profile creation, and add it later with the desired startup type. You can choose not to create a service at profile creation time and optionally create the service later with the desired startup type.

Linux The Linux service definition panel is displayed if the current operating system is a supported version of Linux operating systems, and the current user has the appropriate permissions.

The product attempts to start Linux services for application server processes that are started by a startServer command. For example, if you configure an application server as a Linux service and issue the startServer command, then the **wasservice** command attempts to start the defined service.

By default, the product is not selected to run as a Linux service.

To create the service, the user that runs the Profile Management Tool must be the root user. If you run the Profile Management Tool with a non-root user ID, then the Linux service definition panel is not displayed, and no service is created.

When you create a Linux service, you must specify a user name from which the service runs.

To delete a Linux service, the user must be the root user or have appropriate privileges for deleting the service. Otherwise, a removal script is created that the root user can run to delete the service for the user.

The tool displays the Profile creation summary panel.

16. Click **Create** to create the management profile for the administrative agent, or click **Back** to change the characteristics of the profile.

    The Profile creation progress panel, which shows the configuration commands that are running, is displayed.

    When the profile creation completes, the tool displays the Profile creation complete panel.

17. Optionally, select **Launch the First steps console**. Click **Finish** to exit.

    With the First steps console, you can create additional profiles and start the application server.

Results

You created a management profile for the administrative agent.

Refer to the description of the **manageprofiles** command to learn about creating a profile using a command instead of the Profile Management Tool.

What to do next

Register application servers with the administrative agent using the registerNode command. Then, access the administrative agent console to administer your application servers.

Creating a management profile with a job manager server

You can create a management profile for the job manager to coordinate administrative actions among multiple deployment managers, administer multiple unfederated application servers, asynchronously submit jobs to start servers, and a variety of other tasks.

Before you begin

Before you use the Profile Management Tool, install the product files.

The Profile Management Tool is the graphical user interface for the **manageprofiles** command. See the description of the **manageprofiles** command for more information.

You must provide enough system temporary space to create a profile. For information, read about the file system requirements for profiles.

Note: When you launch the Profile Management Tool, the tool could lock up in the following situation for a non-root user: Log into a machine as root, use the SetPermissions utility to change the user from *x* to *y*. Assume that you are user *x* and log back into the machine. Launch the Profile Management Tool, click **Profile Management Tool**, and click **Create**. The next click after the click on **Create** could lock up the tool.

Note: When you use the Profile Management Tool with the Motif graphical user interface on the Solaris operating system, the default size of the Profile Management Tool might be too small to view all the messages and buttons of the Profile Management Tool. To fix the problem, add the following lines to the *app_server_root*/.Xdefaults file:

```
Eclipse*spacing:0
Eclipse*fontList:-misc-fixed-medium-r-normal-*-10-100-75-75-c-60-iso8859-1
```

After adding the lines, run the following command before launching the Profile Management Tool:

```
xrdb -load user_home/.Xdefaults
```

About this task

After installing the core product files for the product, you must create a profile. This procedure describes creating a management profile with a job manager server using the graphical user interface provided by the Profile Management Tool. You can also use the **manageprofiles** command to create a job manager. See the description of the **manageprofiles** command for more information.

You can create profiles with the Profile Management Tool using the typical profile creation process or the advanced profile creation process. The typical profile creation process uses default settings and assigns unique port values. You can optionally set values as allowed. For the advanced profile creation process you can accept the default values, or specify your own values.

1. Start the Profile Management Tool to create a new runtime environment.

   You can use one of the following ways to start the tool.

   - At the end of installation, select the check box to launch the Profile Management Tool.
   - Issue the command directly from a command prompt.

     The command is in the following directory:

     – Linux HP-UX Solaris AIX *app_server_root*/bin/ProfileManagement

     – Windows *app_server_root*\bin\ProfileManagement

     The name of the command varies per platform:

     – Linux HP-UX Solaris AIX pmt.sh

     – Windows pmt.bat
   - Select the Profile Management Tool option from the First steps console.
   - Windows Use the **Start** menu to access the Profile Management Tool. For example, click **Start > Programs** or **All Programs > IBM WebSphere > *your_product* > Profile Management Tool**.
   - Linux Use the Linux operating system menus used to start programs to start the Profile Management Tool. For example, click ***the_operating_system_menus_to_access_programs* > IBM WebSphere > *your_product* > Profile Management Tool**.
2. Click **Launch Profile Management Tool**, and then click **Create** on the Profiles tab to create a new profile.

The Profiles tab contains a list of profiles that have been created on your machine. No action can be done on a selected profile unless the profile can be augmented. The Augment button is greyed out unless a profile that you select can be augmented.

The tool displays the Environment selection panel.

3. Select **Management**, and click **Next**.

   The Server type selection panel is displayed.

4. Select **Job manager**. Click **Next**.

   The Profile creation options panel is displayed.

5. Select either **Typical profile creation** or **Advanced profile creation**, and click **Next**.

   The **Typical profile creation** option creates a profile that uses default configuration settings. With the **Advanced profile creation** option, you can specify your own configuration values for a profile.

6. If you selected **Typical profile creation**, go to the step on administrative security.

7. If you selected **Advanced profile creation**, optionally select to deploy the administrative console, and then click **Next**.

   If you chose not to deploy the administrative console, then the administrative console ports are disabled on the Ports panel.

   The tool displays the Profile name and location panel.

8. Specify a name for the profile and the directory path for the profile directory, or accept the default values. Then, click **Next**.

   Profile naming guidelines: Double-byte characters are supported. The profile name can be any unique name with the following restrictions. Do not use any of the following characters when naming your profile:

   - Spaces
   - Special characters that are not supported within the name of a directory on your operating system, such as `*&?`
   - Slashes (/) or (\)

   The default profile

   The first profile that you create on a machine is the default profile. The default profile is the default target for commands that are issued from the `bin` directory in the product installation root. When only one profile exists on a machine, every command works on the single server process in the configuration. You can make another profile the default profile when you create that profile by checking **Make this profile the default** on the Profile name and location panel of the **Advanced profile creation** path. You can also make another profile the default profile using the manageprofiles command after you create the profile.

   Addressing a profile in a multiprofile environment

   When multiple profiles exist on a machine, certain commands require that you specify the profile to which the command applies if the profile is not the default profile. These commands use the -profileName parameter to identify which profile to address. You might find it easier to use the commands that are in the `bin` directory of each profile.

   Use these commands to query the command shell to determine the calling profile and to address these commands to the calling profile.

   Default profile information

   The default profile name is *profileTypeProfileName*:

   - *profileType* is a value of `AppSrv`, `Dmgr`, `Custom`, `AdminAgent`, `JobMgr`, or `SecureProxySrv`.
   - *ProfileName* is a sequential number that is used to create a unique profile name.

   AIX HP-UX Linux Solaris The default profile directory is *app_server_root*/profiles, where *app_server_root* is the installation root.

   Windows The default profile directory is *app_server_root*\profiles, where *app_server_root* is the installation root.

9. On the Node, host, and cell names panel, specify a unique node name, the actual host name of the machine, and a unique cell name. Click **Next**.

   The job manager node has the following characteristics.

   Some default values in the following table are split on multiple lines for printing purposes.

Field Name	Default Value	Constraints	Description
Node name	*shortHostName* JobMgr *NodeNumber* where: • *shortHostName* is the short host name. • *NodeNumber* is a sequential number starting at 01.	Use a unique name for the job manager.	The name is used for administration within the job manager cell.
Host name	The long form of the domain name server (DNS) name.	The host name must be addressable through your network. Read about host name considerations.	Use the actual DNS name or IP address of your machine to enable communication with your machine. See additional information about the host name that follows this table.
Cell name	*shortHostName* Cell *CellNumber* where: • *shortHostName* is the short host name. • *CellNumber* is a sequential number starting at 01.	Use a unique name for the cell. If you plan to migrate a Version 5 cell to this Version 7, use the same cell name as the Version 5 cell. A cell name must be unique in any circumstance in which the product is running on the same physical machine or cluster of machines, such as a sysplex. Additionally, a cell name must be unique in any circumstance in which network connectivity between entities is required either between the cells or from a client that must communicate with each of the cells. Cell names must also be unique if their namespaces are federated. Otherwise, you might encounter symptoms such as a javax.naming.NameNotFoundException error, in which case, create uniquely named cells.	All federated nodes become members of the cell, which you name in this panel.

Reserved names: Avoid using reserved folder names as field values. The use of reserved folder names can cause unpredictable results. The following terms are reserved folder names:

- cells
- nodes
- servers
- clusters
- applications
- deployments

Directory path length

Windows The number of characters in the *profiles_directory_path**profile_name* directory must be less than or equal to 80 characters.

Host name considerations

The host name is the network name for the physical machine on which the node is installed. The host name must resolve to a physical network node on the server. When multiple network cards exist in the server, the host name or IP address must resolve to one of the network cards. Remote nodes use the host name to connect to and communicate with this node. Selecting a host name that other machines can reach within your network is important. Do not use the generic identifier, `localhost`, for this value. Also, do not attempt to install WebSphere Application Server products on a machine with a host name that uses characters from a double-byte character set (DBCS). DBCS characters are not supported when used in the host name.

If you define coexisting nodes on the same computer with unique IP addresses, then define each IP address in a domain name server (DNS) look-up table. Configuration files for stand-alone application servers do not provide domain name resolution for multiple IP addresses on a machine with a single network address.

The value that you specify for the host name is used as the value of the hostName property in configuration documents for the stand-alone application server. Specify the host name value in one of the following formats:

- Fully qualified domain name server (DNS) host name string, such as `xmachine.manhattan.ibm.com`
- The default short DNS host name string, such as `xmachine`
- Numeric IP address, such as `127.1.255.3`

The fully qualified DNS host name has the advantages of being unambiguous and flexible. You have the flexibility of changing the actual IP address for the host system without having to change the application server configuration. This value for the host name is particularly useful if you plan to change the IP address frequently when using Dynamic Host Configuration Protocol (DHCP) to assign IP addresses. A disadvantage of this format is dependency on DNS. If DNS is not available, then connectivity is compromised.

The short host name is also dynamically resolvable. A short name format has the added function of being redefined in the local hosts file so that the system can run the application server, even when disconnected from the network. To run disconnected, define the short name as the loopback address, `127.0.0.1`, in the hosts file to run disconnected. A disadvantage of this format is a dependency on DNS for remote access. If DNS is not available, then connectivity is compromised.

A numeric IP address has the advantage of not requiring name resolution through DNS. A remote node can connect to the node that you name with a numeric IP address without DNS being available. A disadvantage of this format is that the numeric IP address is fixed. You must change the setting of the hostName property in Express configuration documents whenever you change the machine IP address. Therefore, do not use a numeric IP address if you use DHCP, or if you change IP addresses regularly. Another disadvantage of this format is that you cannot use the node if the host is disconnected from the network.

After displaying characteristics, the tool displays the Administrative security panel.

10. Optionally enable administrative security, and click **Next**.

    You can enable administrative security now during profile creation, or later from the console. If you enable administrative security now, then enter a user name and password to log onto the administrative console.

    After specifying security characteristics, the tool displays the Security certificate panel if you previously selected **Advanced profile creation**.

11. If you selected **Typical profile creation** at the beginning of these steps, then go to the step that displays the Profile summary panel.

12. Create a default personal certificate and a root signing certificate, or import a personal certificate and a root signing certificate from keystore files, and click **Next**.

    You can create both certificates, import both certificates, or create one certificate, and import the other certificate.

Note: When you import a personal certificate as the default personal certificate, import the root certificate that signed the personal certificate. Otherwise, the Profile Management Tool adds the signer of the personal certificate to the trust.p12 file.

If you import the default personal certificate or the root signing certificate, specify the path and the password, and select the keystore type and the keystore alias for each certificate that you import.

13. Verify that the certificate information is correct, and click **Next**.

    If you create the certificates, you can use the default values or modify them to create new certificates. The default personal certificate is valid for one year by default and is signed by the root signing certificate. The root signing certificate is a self-signed certificate that is valid for 20 years by default. The default keystore password for the root signing certificate is `WebAS`. You should change the password. The password cannot contain any double-byte character set (DBCS) characters because certain keystore types, including PKCS12, do not support these characters. The keystore types that are supported depend on the providers in the java.security file.

    When you create either or both certificates, or import either or both certificates, the keystore files that are created are key.p12, trust.p12, root-key.p12, default-signers.p12, deleted.p12, and ltpa.jceks. These files all have the same password when you create or import the certificates, which is either the default password, or a password that you specify. The key.p12 file contains the default personal certificate. The trust.p12 file contains the signer certificate from the default root certificate. The root-key.p12 file contains the root signing certificate. The default-signer.p12 file contains signer certificates that are added to any new keystore file that you create after the server is installed and running. By default, the default root certificate signer and a DataPower signer certificate is in the default-signer.p12 keystore file. The deleted.p12 keystore file is used to hold certificates deleted with the deleteKeyStore task so that they can be recovered if needed. The ltpa.jceks file contains server default Lightweight Third-Party Authentication (LTPA) keys that the servers in your environment use to communicate with each other.

    An imported certificate is added to the key.p12 file or the root-key.p12 file.

    If you import any certificates and the certificates do not contain the information that you want, click **Back** to import another certificate.

    After displaying the Security certificate panels, the tool displays the Ports panel if you previously selected **Advanced profile creation**.

14. Verify that the ports within the management profile for the job manager are unique, or intentionally conflicting, and click **Next**.

    If you chose not to deploy the administrative console, then the administrative console ports are disabled on the Ports panel.

    Port conflict resolution

    Ports are recognized as being in use if one of the following conditions exists:

    - The ports are assigned to a profile created from an installation that is performed by the current user.
    - The port is currently in use.

    Validation of ports occurs when you access the Port value assignment panel. Conflicts can still occur between the Port value assignment panel and the Profile creation complete panel because ports are not assigned until profile creation completes.

    If you suspect a port conflict, then you can investigate the port conflict after the profile is created. Determine the ports that are used during profile creation by examining the following files.

    - Linux HP-UX Solaris AIX *profile_root*/properties/portdef.props file
    - Windows *profile_root*\properties\portdef.props file

    Included in this file are the keys and values that are used in setting the ports. If you discover ports conflicts, then you can reassign ports manually. To reassign ports, run the updatePorts.ant file by using the ws_ant script.

Windows Linux The tool displays the Windows service definition panel if you are installing on a Windows operating system and the installation ID has the administrative group privilege. The tool displays the Linux service definition panel if you are installing on a supported Linux operating system and the ID that runs the Profile Management Tool is the root user.

15. Choose whether to run the job manager process as a Windows service on a Windows operating system or as a Linux Service on a Linux operating system, and click **Next**.

    The Windows service definition panel is displayed for the Windows operating system only if the ID that installs the Windows service has the administrator group privilege. However, you can run the WASService.exe command to create the Windows service as long as the installer ID belongs to the administrator group. Read about automatically restarting server processes for more information.

    Windows The product attempts to start Windows services for job manager processes that are started by a startServer command. For example, if you configure a job manager as a Windows service and issue the startServer command, then the **wasservice** command attempts to start the defined service.

    If you chose to install a local system service, then you do not have to specify your user ID or password. If you create a specified user type of service, then you must specify the user ID and the password for the user who runs the service. The user must have `Log on as a service` authority for the service to run correctly. If the user does not have `Log on as a service` authority, then the Profile Management tool automatically adds the authority.

    To perform this profile creation task, the user ID must not contain spaces. In addition to belonging to the administrator group, the ID must also have the advanced user right of `Log on as a service`. The Installation wizard grants the user ID the advanced user right if the user ID does not already have the advanced user right and if the user ID belongs to the administrator group.

    You can also create other Windows services after the installation is complete to start other server processes. Read about automatically restarting server processes for more information.

    You can remove the Windows service that is added during profile creation during profile deletion. You can also remove the Windows service with the wasservice command.

    IPv6 considerations

    Profiles created to run as a Windows service fail to start when using Internet Protocol Version 6.0 (IPv6) if the service is configured to run as local system. Create a user-specific environment variable to enable IPv6. Since this environment variable is a user variable instead of a local system variable, only a Windows service that runs as that specific user can access this environment variable. By default, when a new profile is created and configured to run as a Windows service, the service is set to run as local system. When the Windows service for the job manager process tries to run, the service is unable to access the user environment variable that specifies IPv6, and thus tries to start as IPv4. The server does not start correctly in this case. To resolve the problem, when creating the profile, specify that the Windows service for the job manager process runs as the same user ID under which the environment variable that specifies IPv6 is defined, instead of as local system.

    Default Windows service information

    Windows The following default values for the Windows service definition panel exist:

    - The default is to run as a Windows service.
    - The service process is selected to run as a system account.
    - The user account is the current user name. User name requirements are the requirements that the Windows operating system imposes for a user ID.
    - The startup type is `automatic`. The values for the startup type are those values that the Windows operating system imposes. If you want a startup type other than `automatic`, you can either select another available option from the menu or change the startup type after you create the profile. You can also remove the created service after profile creation, and add it later with the desired startup type. You can choose not to create a service at profile creation time and optionally create the service later with the desired startup type.

    Linux The Linux service definition panel is displayed if the current operating system is a supported version of Linux operating systems, and the current user has the appropriate permissions.

The product attempts to start Linux services for application server processes that are started by a startServer command. For example, if you configure an application server as a Linux service and issue the startServer command, then the **wasservice** command attempts to start the defined service.

By default, the product is not selected to run as a Linux service.

To create the service, the user that runs the Profile Management Tool must be the root user. If you run the Profile Management Tool with a non-root user ID, then the Linux service definition panel is not displayed, and no service is created.

When you create a Linux service, you must specify a user name from which the service runs.

To delete a Linux service, the user must be the root user or have appropriate privileges for deleting the service. Otherwise, a removal script is created that the root user can run to delete the service for the user.

The tool displays the Profile Creation Summary panel.

16. Click **Create** to create the management profile for the job manager, or click **Back** to change the characteristics of the profile.

    The Profile creation progress panel, which shows the configuration commands that are running, is displayed.

    When the profile creation completes, the tool displays the Profile creation complete panel.

17. Optionally, select **Launch the First steps console**. Click **Finish** to exit.

    With the First steps console, you can create additional profiles and start the application server.

Results

You created a management profile for the job manager.

Refer to the description of the **manageprofiles** command to learn about creating a profile using a command instead of the Profile Management Tool.

What to do next

Access the job manager console to perform a variety of administrative tasks. You can coordinate management actions among multiple deployment managers, administer multiple unfederated application servers, asynchronously submit jobs to start servers, and so on.

Creating a secure proxy profile

You can create a secure proxy profile to serve as the initial point of entry into your enterprise environment. Typically, a secure proxy server exists in the demilitarized zone (DMZ), accepts requests from clients on theInternet, and forwards the requests to servers in your enterprise environment.

Before you begin

Before you use the Profile Management Tool, install the core product files. You can create two different secure proxy profiles depending on which core product files you install. The core product files could either be for a Network Deployment installation or a DMZ Secure Proxy Server installation. Read about the profiles created for the different installations in About this task.

The Profile Management Tool is the graphical user interface for the **manageprofiles** command. See the description of the **manageprofiles** command for more information.

You must provide enough system temporary space to create a profile. For information, read about the file system requirements for profiles.

Note: When you launch the Profile Management Tool, the tool could lock up in the following situation for a non-root user: Log into a machine as root, use the SetPermissions utility to change the user from *x*

to *y*. Assume that you are user *x* and log back into the machine. Launch the Profile Management Tool, click **Profile Management Tool**, and click **Create**. The next click after the click on **Create** could lock up the tool.

Note: When you use the Profile Management Tool with the Motif graphical user interface on the Solaris operating system, the default size of the Profile Management Tool might be too small to view all the messages and buttons of the Profile Management Tool. To fix the problem, add the following lines to the *app_server_root*/.Xdefaults file:

```
Eclipse*spacing:0
Eclipse*fontList:-misc-fixed-medium-r-normal-*-10-100-75-75-c-60-iso8859-1
```

After adding the lines, run the following command before launching the Profile Management Tool:

```
xrdb -load user_home/.Xdefaults
```

About this task

After installing the core product files for the product, you must create a profile. This procedure describes creating a secure proxy profile using the graphical user interface that is provided by the Profile Management Tool. You can also use the **manageprofiles** command to create a secure proxy profile. See the description of the **manageprofiles** command for more information.

You can create profiles with the Profile Management Tool using the typical profile creation process or the advanced profile creation process. The typical profile creation process uses default settings and assigns unique port values. You can optionally set values as allowed. For the advanced profile creation process you can accept the default values, or specify your own values.

You can create two different profiles for the DMZ Secure Proxy Server using this task. You can create a secure proxy server profile on a Network Deployment installation. However, you can only configure this profile in a Network Deployment installation. To use the secure proxy server of the profile, you must export the profile from the Network Deployment environment and then import it into the DMZ Secure Proxy Server installation. Read about exporting and importing the secure proxy profile in the topic about the ConfigArchiveOperations command group for the AdminTask object. Alternatively, you can create a secure proxy server profile on a DMZ Secure Proxy Server installation. In this situation the secure proxy server does not have a Web container, and so cannot host an administrative console. To administer this secure proxy server, you must employ wsadmin scripting commands.

1. Start the Profile Management Tool to create a new runtime environment.

   You can use one of the following ways to start the tool.

   - At the end of installation, select the check box to launch the Profile Management Tool.
   - Issue the command directly from a command prompt.

     The command is in the following directory:

     – Linux HP-UX Solaris AIX *app_server_root*/bin/ProfileManagement

     – Windows *app_server_root*\bin\ProfileManagement

     The name of the command varies per platform:

     – Linux HP-UX Solaris AIX pmt.sh

     – Windows pmt.bat
   - Select the Profile Management Tool option from the First steps console.
   - Windows Use the **Start** menu to access the Profile Management Tool. For example, click **Start** > **Programs** or **All Programs** > **IBM WebSphere** > ***your_product*** > **Profile Management Tool**.
   - Linux Use the Linux operating system menus used to start programs to start the Profile Management Tool. For example, click ***the_operating_system_menus_to_access_programs*** > **IBM WebSphere** > ***your_product*** > **Profile Management Tool**.

2. Click **Launch Profile Management Tool**, and then click **Create** on the Profiles tab to create a new profile.

   The Profiles tab contains a list of profiles that have been created on your machine. No action can be done on a selected profile unless the profile can be augmented. The Augment button is greyed out unless a profile that you select can be augmented.

   The tool displays the Environment selection panel.
3. Select **Secure proxy (configuration only)** for the Network Deployment image, or **Secure proxy** for the DMZ image, and click **Next**.

   The Profile creation options panel is displayed.
4. Select either **Typical profile creation** or **Advanced profile creation**, and click **Next**.

   The **Typical profile creation** option creates a profile that uses default configuration settings. With the **Advanced profile creation** option, you can specify your own configuration values for a profile.
5. If you selected **Typical profile creation** at the beginning of these steps, then go to the step that displays the administrative security.
6. Specify a name for the profile and the directory path for the profile directory, or accept the default values. Then, click **Next**.

   Profile naming guidelines: Double-byte characters are supported. The profile name can be any unique name with the following restrictions. Do not use any of the following characters when naming your profile:

   - Spaces
   - Special characters that are not supported within the name of a directory on your operating system, such as `*&?`
   - Slashes (/) or (\)

   The default profile

   The first profile that you create on a machine is the default profile. The default profile is the default target for commands that are issued from the `bin` directory in the product installation root. When only one profile exists on a machine, every command works on the single server process in the configuration. You can make another profile the default profile when you create that profile by checking **Make this profile the default** on the Profile name and location panel of the **Advanced profile creation** path. You can also make another profile the default profile using the manageprofiles command after you create the profile.

   Addressing a profile in a multiprofile environment

   When multiple profiles exist on a machine, certain commands require that you specify the profile to which the command applies if the profile is not the default profile. These commands use the -profileName parameter to identify which profile to address. You might find it easier to use the commands that are in the `bin` directory of each profile.

   Use these commands to query the command shell to determine the calling profile and to address these commands to the calling profile.

   Default profile information

   The default profile name is *profileTypeProfileName*:

   - *profileType* is a value of `AppSrv`, `Dmgr`, `Custom`, `AdminAgent`, `JobMgr`, or `SecureProxySrv`.
   - *ProfileName* is a sequential number that is used to create a unique profile name.

   AIX HP-UX Linux Solaris The default profile directory is *app_server_root*/profiles, where *app_server_root* is the installation root.

   Windows The default profile directory is *app_server_root*\profiles, where *app_server_root* is the installation root.
7. On the Node and Host Names panel, specify a unique node name, a server name, and the actual host name of the machine. Click **Next**.

   The secure proxy server node has the following characteristics.

Some default values in the following table are split on multiple lines for printing purposes.

Field name	Default value	Constraints	Description
Node name	*shortHostName* Node where: • *shortHostName* is the short host name. • *NodeNumber* is a sequential number starting at 01.	Use a unique name for the secure proxy server.	The name is used for administration within the deployment manager cell.
Server name	proxy1	Specifies a logical name for the server. Server names must be unique within a node. However, for multiple nodes within a cluster, you might have different servers with the same server name as long as the server and node pair are unique.	The server name is used for administration within the deployment manager cell.
Host name	The long form of the domain name server (DNS) name.	The host name must be addressable through your network. Read about host name considerations.	Use the actual DNS name or IP address of your machine to enable communication with your machine. See additional information about the host name that follows this table.

Reserved names: Avoid using reserved folder names as field values. The use of reserved folder names can cause unpredictable results. The following terms are reserved folder names:

- cells
- nodes
- servers
- clusters
- applications
- deployments

Directory path length

Windows The number of characters in the *profiles_directory_path**profile_name* directory must be less than or equal to 80 characters.

Host name considerations

The host name is the network name for the physical machine on which the node is installed. The host name must resolve to a physical network node on the server. When multiple network cards exist in the server, the host name or IP address must resolve to one of the network cards. Remote nodes use the host name to connect to and communicate with this node. Selecting a host name that other machines can reach within your network is important. Do not use the generic identifier, `localhost`, for this value. Also, do not attempt to install WebSphere Application Server products on a machine with a host name that uses characters from a double-byte character set (DBCS). DBCS characters are not supported when used in the host name.

If you define coexisting nodes on the same computer with unique IP addresses, then define each IP address in a domain name server (DNS) look-up table. Configuration files for stand-alone application servers do not provide domain name resolution for multiple IP addresses on a machine with a single network address.

The value that you specify for the host name is used as the value of the hostName property in configuration documents for the stand-alone application server. Specify the host name value in one of the following formats:

- Fully qualified domain name server (DNS) host name string, such as `xmachine.manhattan.ibm.com`
- The default short DNS host name string, such as `xmachine`
- Numeric IP address, such as `127.1.255.3`

The fully qualified DNS host name has the advantages of being unambiguous and flexible. You have the flexibility of changing the actual IP address for the host system without having to change the application server configuration. This value for the host name is particularly useful if you plan to change the IP address frequently when using Dynamic Host Configuration Protocol (DHCP) to assign IP addresses. A disadvantage of this format is dependency on DNS. If DNS is not available, then connectivity is compromised.

The short host name is also dynamically resolvable. A short name format has the added function of being redefined in the local hosts file so that the system can run the application server, even when disconnected from the network. To run disconnected, define the short name as the loopback address, `127.0.0.1`, in the hosts file to run disconnected. A disadvantage of this format is a dependency on DNS for remote access. If DNS is not available, then connectivity is compromised.

A numeric IP address has the advantage of not requiring name resolution through DNS. A remote node can connect to the node that you name with a numeric IP address without DNS being available. A disadvantage of this format is that the numeric IP address is fixed. You must change the setting of the hostName property in Express configuration documents whenever you change the machine IP address. Therefore, do not use a numeric IP address if you use DHCP, or if you change IP addresses regularly. Another disadvantage of this format is that you cannot use the node if the host is disconnected from the network.

After displaying the node name, server name, and host name for the secure proxy profile, the tool displays the Security Level Selection panel.

8. Accept the defaults or change the proxy security level and the protocols, and click **Next**.

   You can optionally change your security settings after you create the secure proxy server profile. Read about tuning security properties for the secured proxy server.

   After displaying the security level options, the tool displays the Administrative security panel.

9. Optionally enable administrative security, and click **Next**.

   You can enable administrative security now during profile creation, or later from the console. If you enable administrative security now, then enter a user name and password to log onto the administrative console.

   After specifying security characteristics, the tool displays the Security certificate panel if you previously selected **Advanced profile creation**.

10. If you selected **Typical profile creation** at the beginning of these steps, then go to the step that displays the Profile summary panel.

11. Create a default personal certificate and a root signing certificate, or import a personal certificate and a root signing certificate from keystore files, and click **Next**.

    You can create both certificates, import both certificates, or create one certificate, and import the other certificate.

    Note: When you import a personal certificate as the default personal certificate, import the root certificate that signed the personal certificate. Otherwise, the Profile Management Tool adds the signer of the personal certificate to the trust.p12 file.

    If you import the default personal certificate or the root signing certificate, specify the path and the password, and select the keystore type and the keystore alias for each certificate that you import.

12. Verify that the certificate information is correct, and click **Next**.

    If you create the certificates, you can use the default values or modify them to create new certificates. The default personal certificate is valid for one year by default and is signed by the root signing

certificate. The root signing certificate is a self-signed certificate that is valid for 20 years by default. The default keystore password for the root signing certificate is `WebAS`. You should change the password. The password cannot contain any double-byte character set (DBCS) characters because certain keystore types, including PKCS12, do not support these characters. The keystore types that are supported depend on the providers in the java.security file.

When you create either or both certificates, or import either or both certificates, the keystore files that are created are key.p12, trust.p12, root-key.p12, default-signers.p12, deleted.p12, and ltpa.jceks. These files all have the same password when you create or import the certificates, which is either the default password, or a password that you specify. The key.p12 file contains the default personal certificate. The trust.p12 file contains the signer certificate from the default root certificate. The root-key.p12 file contains the root signing certificate. The default-signer.p12 file contains signer certificates that are added to any new keystore file that you create after the server is installed and running. By default, the default root certificate signer and a DataPower signer certificate is in the default-signer.p12 keystore file. The deleted.p12 keystore file is used to hold certificates deleted with the deleteKeyStore task so that they can be recovered if needed. The ltpa.jceks file contains server default Lightweight Third-Party Authentication (LTPA) keys that the servers in your environment use to communicate with each other.

An imported certificate is added to the key.p12 file or the root-key.p12 file.

If you import any certificates and the certificates do not contain the information that you want, click **Back** to import another certificate.

After displaying the Security certificate panels, the tool displays the Ports panel if you previously selected **Advanced profile creation**.

13. Verify that the ports within the secure proxy profile are unique, or intentionally conflicting, and click **Next**.

    Port conflict resolution

    Ports are recognized as being in use if one of the following conditions exists:

    - The ports are assigned to a profile created from an installation that is performed by the current user.
    - The port is currently in use.

    Validation of ports occurs when you access the Port value assignment panel. Conflicts can still occur between the Port value assignment panel and the Profile creation complete panel because ports are not assigned until profile creation completes.

    If you suspect a port conflict, then you can investigate the port conflict after the profile is created. Determine the ports that are used during profile creation by examining the following files.

    - Linux HP-UX Solaris AIX *profile_root*/properties/portdef.props file
    - Windows *profile_root*\properties\portdef.props file

    Included in this file are the keys and values that are used in setting the ports. If you discover ports conflicts, then you can reassign ports manually. To reassign ports, run the updatePorts.ant file by using the ws_ant script.

    Windows Linux The tool displays the Windows service definition panel if you are installing on a Windows operating system and the installation ID has the administrative group privilege. The tool displays the Linux service definition panel if you are installing on a supported Linux operating system and the ID that runs the Profile Management Tool is the root user.

14. Choose whether to run the secure proxy server as a Windows service on a Windows operating system or as a Linux Service on a Linux operating system, and click **Next**.

    The Windows service definition panel is displayed for the Windows operating system only if the ID that installs the Windows service has the administrator group privilege. However, you can run the WASService.exe command to create the Windows service as long as the installer ID belongs to the administrator group. Read about automatically restarting server processes for more information.

Windows The product attempts to start Windows services for secure proxy processes that are started by a startServer command. For example, if you configure a secure proxy server as a Windows service and issue the startServer command, then the **wasservice** command attempts to start the defined service.

If you chose to install a local system service, then you do not have to specify your user ID or password. If you create a specified user type of service, then you must specify the user ID and the password for the user who runs the service. The user must have `Log on as a service` authority for the service to run correctly. If the user does not have `Log on as a service` authority, then the Profile Management tool automatically adds the authority.

To perform this profile creation task, the user ID must not contain spaces. In addition to belonging to the administrator group, the ID must also have the advanced user right of `Log on as a service`. The Installation wizard grants the user ID the advanced user right if the user ID does not already have the advanced user right and if the user ID belongs to the administrator group.

You can also create other Windows services after the installation is complete to start other server processes. Read about automatically restarting server processes for more information.

You can remove the Windows service that is added during profile creation during profile deletion. You can also remove the Windows service with the wasservice command.

IPv6 considerations

Profiles created to run as a Windows service fail to start when using Internet Protocol Version 6.0 (IPv6) if the service is configured to run as local system. Create a user-specific environment variable to enable IPv6. Since this environment variable is a user variable instead of a local system variable, only a Windows service that runs as that specific user can access this environment variable. By default, when a new profile is created and configured to run as a Windows service, the service is set to run as local system. When the Windows service for the secure proxy server process attempts to run, the service is unable to access the user environment variable that specifies IPv6, and thus attempts to start as IPv4. The server does not start correctly in this case. To resolve the problem, when creating the profile, specify that the Windows service for the secure proxy server process runs as the same user ID from which the environment variable that specifies IPv6 is defined, instead of as *Local System*.

Default Windows service information

Windows The following default values for the Windows service definition panel exist:

- The default is to run as a Windows service.
- The service process is selected to run as a system account.
- The user account is the current user name. User name requirements are the requirements that the Windows operating system imposes for a user ID.
- The startup type is `automatic`. The values for the startup type are those values that the Windows operating system imposes. If you want a startup type other than `automatic`, you can either select another available option from the menu or change the startup type after you create the profile. You can also remove the created service after profile creation, and add it later with the desired startup type. You can choose not to create a service at profile creation time and optionally create the service later with the desired startup type.

Linux The Linux service definition panel is displayed if the current operating system is a supported version of Linux operating systems, and the current user has the appropriate permissions.

The product attempts to start Linux services for application server processes that are started by a startServer command. For example, if you configure an application server as a Linux service and issue the startServer command, then the **wasservice** command attempts to start the defined service.

By default, the product is not selected to run as a Linux service.

To create the service, the user that runs the Profile Management Tool must be the root user. If you run the Profile Management Tool with a non-root user ID, then the Linux service definition panel is not displayed, and no service is created.

When you create a Linux service, you must specify a user name from which the service runs.

To delete a Linux service, the user must be the root user or have appropriate privileges for deleting the service. Otherwise, a removal script is created that the root user can run to delete the service for the user.

The tool displays the Profile creation summary panel.

15. Click **Create** to create the secure proxy server profile, or click **Back** to change the characteristics of the profile.

    The Profile creation progress panel, which shows the configuration commands that are running, is displayed.

    When the profile creation completes, the tool displays the Profile creation complete panel.

16. If the secure proxy profile that you are creating is part of the DMZ Secure Proxy Server for IBM WebSphere Application Server installation, optionally select **Launch the First steps console**. Click **Finish** to exit.

    With the First steps console, you can create additional profiles, and start the application server.

    If the secure proxy profile that you are creating is part of the Network Deployment installation, you do not have the option of launching the First steps console.

Results

Depending on your installation, you have either created a secure proxy server profile on a Network Deployment image or a secure proxy profile on a DMZ Secure Proxy Server installatoin..

Refer to the description of the **manageprofiles** command to learn about creating a profile using a command instead of the Profile Management Tool.

What to do next

The secure proxy server can accept requests from clients on the Internet and forward the requests to servers in your enterprise environment.

The secure proxy profile is available both on the Network Deployment and the DMZ images. You cannot start the profile on the Network Deployment image. The profile is used only for configuration on an administrative console. After you configure the profile, you can export it and then import it into the secure proxy profile of the DMZ image. The secure proxy profile is fully operational on the DMZ image.

Creating a cell profile

You can create a cell profile in a single pass with the Profile Management Tool. This cell profile contains a federated application server node and a deployment manager.

Before you begin

Before you use the Profile Management Tool, install the product files.

The Profile Management Tool is the graphical user interface for the **manageprofiles** command. See the description of the **manageprofiles** command for more information.

You must provide enough system temporary space to create a profile. For information, read about the file system requirements for profiles.

Note: When you launch the Profile Management Tool, the tool could lock up in the following situation for a non-root user: Log into a machine as root, use the SetPermissions utility to change the user from *x* to *y*. Assume that you are user *x* and log back into the machine. Launch the Profile Management Tool, click **Profile Management Tool**, and click **Create**. The next click after the click on **Create** could lock up the tool.

Note: When you use the Profile Management Tool with the Motif graphical user interface on the Solaris operating system, the default size of the Profile Management Tool might be too small to view all the messages and buttons of the Profile Management Tool. To fix the problem, add the following lines to the *app_server_root*/.Xdefaults file:

```
Eclipse*spacing:0
Eclipse*fontList:-misc-fixed-medium-r-normal-*-10-100-75-75-c-60-iso8859-1
```

After adding the lines, run the following command before launching the Profile Management Tool:

```
xrdb -load user_home/.Xdefaults
```

About this task

After installing the core product files for the Network Deployment product, you must create a profile. This procedure describes how to create a cell profile with the Profile Management Tool, which is a graphical user interface. You can also use the **manageprofiles** command to create a cell profile. See the description of the manageprofiles for more information.

A cell profile contains a deployment manager profile and a federated application server node profile. You can federate additional Application Server node profiles into this deployment manager profile after initial creation of the cell profile.

You can create profiles with the Profile Management Tool using the typical profile creation process or the advanced profile creation process. The typical profile creation process uses default settings and assigns unique port values. You can optionally set values as allowed. For the advanced profile creation process you can accept the default values, or specify your own values.

1. Start the Profile Management Tool to create a new runtime environment.

   You can use one of the following ways to start the tool.

   - At the end of installation, select the check box to launch the Profile Management Tool.
   - Issue the command directly from a command prompt.

     The command is in the following directory:

     – Linux HP-UX Solaris AIX *app_server_root*/bin/ProfileManagement

     – Windows *app_server_root*\bin\ProfileManagement

     The name of the command varies per platform:

     – Linux HP-UX Solaris AIX pmt.sh

     – Windows pmt.bat
   - Select the Profile Management Tool option from the First steps console.
   - Windows Use the **Start** menu to access the Profile Management Tool. For example, click **Start > Programs** or **All Programs > IBM WebSphere > *your_product* > Profile Management Tool**.
   - Linux Use the Linux operating system menus used to start programs to start the Profile Management Tool. For example, click ***the_operating_system_menus_to_access_programs* > IBM WebSphere > *your_product* > Profile Management Tool**.
2. Click **Launch Profile Management Tool**, and then click **Create** on the Profiles tab to create a new profile.

   The Profiles tab contains a list of profiles that have been created on your machine. No action can be done on a selected profile unless the profile can be augmented. The Augment button is greyed out unless a profile that you select can be augmented.

   The tool displays the Environment selection panel.
3. Select the cell profile, then click **Next**.

   The Profile creation options panel is displayed.

4. Select either **Typical profile creation** or **Advanced profile creation**, and click **Next**.

   The **Typical profile creation** option creates a profile that uses default configuration settings. With the **Advanced profile creation** option, you can specify your own configuration values for a profile.
5. If you selected **Typical profile creation**, go to the step on administrative security.
6. If you selected **Advanced profile creation**, then select the applications that you want to deploy, and click **Next**.

   The tool displays the Profile name and location panel.
7. If you selected **Advanced profile creation**, then specify the deployment manager profile name, the application server profile name and the profile directory on the Profile name and location panel, or accept the defaults. Click **Next**.

   Profile naming guidelines: Double-byte characters are supported. The profile name can be any unique name with the following restrictions. Do not use any of the following characters when naming your profile:

   - Spaces
   - Special characters that are not supported within the name of a directory on your operating system, such as `*&?`
   - Slashes (/) or (\)

   The default profile

   The first profile that you create on a machine is the default profile. The default profile is the default target for commands that are issued from the `bin` directory in the product installation root. When only one profile exists on a machine, every command works on the single server process in the configuration. You can make another profile the default profile when you create that profile by checking **Make this profile the default** on the Profile name and location panel of the **Advanced profile creation** path. You can also make another profile the default profile using the manageprofiles command after you create the profile.

   Addressing a profile in a multiprofile environment

   When multiple profiles exist on a machine, certain commands require that you specify the profile to which the command applies if the profile is not the default profile. These commands use the -profileName parameter to identify which profile to address. You might find it easier to use the commands that are in the `bin` directory of each profile.

   Use these commands to query the command shell to determine the calling profile and to address these commands to the calling profile.

   Default profile information

   The default profile name is *profileTypeProfileName*:

   - *profileType* is a value of `AppSrv`, `Dmgr`, `Custom`, `AdminAgent`, `JobMgr`, or `SecureProxySrv`.
   - *ProfileName* is a sequential number that is used to create a unique profile name.

   AIX HP-UX Linux Solaris The default profile directory is *app_server_root*/profiles, where *app_server_root* is the installation root.

   Windows The default profile directory is *app_server_root*\profiles, where *app_server_root* is the installation root.

   The tool then displays the Node, host, and cell names panel.
8. Specify a unique deployment manager node name, a unique application server node name, the actual host name of the machine, and a unique cell name for the cell, and click **Next**.

   The cell profile has the following characteristics:

   Some default values in the following table are split on multiple lines for printing purposes.

Field Name	Default Value	Constraints	Description
Deployment manager node name	*shortHostName* CellManager *NodeNumber* where: • *shortHostName* is the short host name. • *NodeNumber* is a sequential number starting at 01.	Use a unique name for the deployment manager.	The name is used for administration within the deployment manager cell.
Application server node name	*shortHostName* Node *NodeNumber* where: • *shortHostName* is the short host name. • *NodeNumber* is a sequential number starting at 01.	Use a unique name for the application server.	The name is used for administration within the deployment manager cell.
Host name	The long form of the domain name server (DNS) name.	The host name must be addressable through your network.	Use the actual DNS name or IP address of your machine to enable communication with your machine. See additional information about the host name that follows this table.
Cell name	*shortHostName* Cell *CellNumber* where: • *shortHostName* is the short host name. • *CellNumber* is a sequential number starting at 01.	Use a unique name for the deployment manager cell. If you plan to migrate a Version 5 or Version 6 deployment manager cell to this Version 7 deployment manager, use the same cell name as the Version 5 or Version 6 deployment manager. A cell name must be unique in any circumstance in which the product is running on the same physical machine or cluster of machines, such as a sysplex. Additionally, a cell name must be unique in any circumstance in which network connectivity between entities is required either between the cells or from a client that must communicate with each of the cells. Cell names must also be unique if their namespaces are federated. Otherwise, you might encounter symptoms such as a javax.naming.NameNotFoundException error, in which case, create uniquely named cells.	All federated nodes become members of the deployment manager cell, which you name in this panel.

Reserved names: Avoid using reserved folder names as field values. The use of reserved folder names can cause unpredictable results. The following terms are reserved folder names:

- cells
- nodes
- servers
- clusters

- applications
- deployments

Directory path considerations

Windows The number of characters in the *profiles_directory_path**profile_name* directory must be less than or equal to 80 characters.

Host name considerations

The host name is the network name for the physical machine on which the node is installed. The host name must resolve to a physical network node on the server. When multiple network cards exist in the server, the host name or IP address must resolve to one of the network cards. Remote nodes use the host name to connect to and communicate with this node. Selecting a host name that other machines can reach within your network is important. Do not use the generic identifier, `localhost`, for this value. Also, do not attempt to install WebSphere Application Server products on a machine with a host name that uses characters from a double-byte character set (DBCS). DBCS characters are not supported when used in the host name.

If you define coexisting nodes on the same computer with unique IP addresses, then define each IP address in a domain name server (DNS) look-up table. Configuration files for stand-alone application servers do not provide domain name resolution for multiple IP addresses on a machine with a single network address.

The value that you specify for the host name is used as the value of the hostName property in configuration documents for the stand-alone application server. Specify the host name value in one of the following formats:

- Fully qualified domain name server (DNS) host name string, such as `xmachine.manhattan.ibm.com`
- The default short DNS host name string, such as `xmachine`
- Numeric IP address, such as `127.1.255.3`

The fully qualified DNS host name has the advantages of being unambiguous and flexible. You have the flexibility of changing the actual IP address for the host system without having to change the application server configuration. This value for the host name is particularly useful if you plan to change the IP address frequently when using Dynamic Host Configuration Protocol (DHCP) to assign IP addresses. A disadvantage of this format is dependency on DNS. If DNS is not available, then connectivity is compromised.

The short host name is also dynamically resolvable. A short name format has the added function of being redefined in the local hosts file so that the system can run the application server, even when disconnected from the network. To run disconnected, define the short name as the loopback address, `127.0.0.1`, in the hosts file to run disconnected. A disadvantage of this format is a dependency on DNS for remote access. If DNS is not available, then connectivity is compromised.

A numeric IP address has the advantage of not requiring name resolution through DNS. A remote node can connect to the node that you name with a numeric IP address without DNS being available. A disadvantage of this format is that the numeric IP address is fixed. You must change the setting of the hostName property in Express configuration documents whenever you change the machine IP address. Therefore, do not use a numeric IP address if you use DHCP, or if you change IP addresses regularly. Another disadvantage of this format is that you cannot use the node if the host is disconnected from the network.

After displaying the cell characteristics, the tool displays the Administrative security panel.

9. Optionally enable administrative security, and click **Next**.

   You can enable administrative security now during profile creation, or later from the console. If you enable administrative security now, then enter a user name and password to log onto the administrative console.

   If you installed the Samples, and you chose to deploy them, then the Samples require an account under which to run. Supply the Samples password for the account. You cannot change the user name of the account.

After specifying security characteristics, the tool displays the Security certificate panel if you previously selected **Advanced profile creation**.

10. If you selected **Typical profile creation** at the beginning of these steps, then go to the step that displays the Profile summary panel.
11. Create a default personal certificate and a root signing certificate, or import a personal certificate and a root signing certificate from keystore files, and click **Next**.

    You can create both certificates, import both certificates, or create one certificate, and import the other certificate.

    Note: When you import a personal certificate as the default personal certificate, import the root certificate that signed the personal certificate. Otherwise, the Profile Management Tool adds the signer of the personal certificate to the trust.p12 file.

    If you import the default personal certificate or the root signing certificate, specify the path and the password, and select the keystore type and the keystore alias for each certificate that you import.
12. Verify that the certificate information is correct, and click **Next**.

    If you create the certificates, you can use the default values or modify them to create new certificates. The default personal certificate is valid for one year by default and is signed by the root signing certificate. The root signing certificate is a self-signed certificate that is valid for 20 years by default. The default keystore password for the root signing certificate is `WebAS`. You should change the password. The password cannot contain any double-byte character set (DBCS) characters because certain keystore types, including PKCS12, do not support these characters. The keystore types that are supported depend on the providers in the java.security file.

    When you create either or both certificates, or import either or both certificates, the keystore files that are created are key.p12, trust.p12, root-key.p12, default-signers.p12, deleted.p12, and ltpa.jceks. These files all have the same password when you create or import the certificates, which is either the default password, or a password that you specify. The key.p12 file contains the default personal certificate. The trust.p12 file contains the signer certificate from the default root certificate. The root-key.p12 file contains the root signing certificate. The default-signer.p12 file contains signer certificates that are added to any new keystore file that you create after the server is installed and running. By default, the default root certificate signer and a DataPower signer certificate is in the default-signer.p12 keystore file. The deleted.p12 keystore file is used to hold certificates deleted with the deleteKeyStore task so that they can be recovered if needed. The ltpa.jceks file contains server default Lightweight Third-Party Authentication (LTPA) keys that the servers in your environment use to communicate with each other.

    An imported certificate is added to the key.p12 file or the root-key.p12 file.

    If you import any certificates and the certificates do not contain the information that you want, click **Back** to import another certificate.

    After displaying the Security certificate panels, the tool displays the Ports panel if you previously selected **Advanced profile creation**.
13. Verify that the ports specified for the deployment manager are unique, and click **Next**.

    If you chose not to deploy the administrative console, then the administrative console ports are disabled on the Ports panel.

    Port conflict resolution

    Ports are recognized as being in use if one of the following conditions exists:

    - The ports are assigned to a profile created from an installation that is performed by the current user.
    - The port is currently in use.

    Validation of ports occurs when you access the Port value assignment panel. Conflicts can still occur between the Port value assignment panel and the Profile creation complete panel because ports are not assigned until profile creation completes.

If you suspect a port conflict, then you can investigate the port conflict after the profile is created. Determine the ports that are used during profile creation by examining the following files.

- Linux HP-UX Solaris AIX *profile_root*/properties/portdef.props file
- Windows *profile_root*\properties\portdef.props file

Included in this file are the keys and values that are used in setting the ports. If you discover ports conflicts, then you can reassign ports manually. To reassign ports, run the updatePorts.ant file by using the ws_ant script.

14. Verify that the ports specified for the application server are unique, and click **Next**.

    The same discussion on ports in the previous step applies to this step.

    Windows Linux The tool displays the Windows service definition panel if you are installing on a Windows operating system and the installation ID has the administrative group privilege. The tool displays the Linux service definition panel if you are installing on a supported Linux operating system and the ID that runs the Profile Management Tool is the root user.

15. Choose whether to run the application server as a Windows service on a Windows operating system or as a Linux service on a Linux operating system, then click **Next**.

    - Windows

      The Windows service definition panel is displayed for the Windows operating system only if the ID that installs the Windows service has the administrator group privilege. However, you can run the WASService.exe command to create the Windows service as long as the installer ID belongs to the administrator group. Read about automatically restarting server processes for more information.

      Windows The product attempts to start Windows services for application server processes that are started by a startServer command. For example, if you configure an application server as a Windows service, and issue the startServer command, then the **wasservice** command attempts to start the defined service.

      If you chose to install a local system service, then you do not have to specify your user ID or password. If you create a specified user type of service, then you must specify the user ID and the password for the user who runs the service. The user must have `Log on as a service` authority for the service to run correctly. If the user does not have `Log on as a service` authority, then the Profile Management tool automatically adds the authority.

      To perform this profile creation task, the user ID must not contain spaces. In addition to belonging to the administrator group, the ID must also have the advanced user right of `Log on as a service`. The Installation wizard grants the user ID the advanced user right if the user ID does not already have the advanced user right and if the user ID belongs to the administrator group.

      You can also create other Windows services after the installation is complete to start other server processes. Read about automatically restarting server processes for more information.

      You can remove the Windows service that is added during profile creation during profile deletion. You can also remove the Windows service with the wasservice command.

      IPv6 considerations

      Profiles created to run as a Windows service fail to start when using Internet Protocol Version 6 (IPv6) if the service is configured to run as local system. Create a user-specific environment variable to enable IPv6. Since this environment variable is a user variable instead of a local system variable, only a Windows service that runs as that specific user can access this environment variable. By default, when a new profile is created and configured to run as a Windows service, the service is set to run as local system. When the Windows service for the product tries to run, the service is unable to access the user environment variable that specifies IPv6, and thus, tries to start as Internet Protocol Version 4 (IPv4). The server does not start correctly in this case. To resolve the problem, when creating the profile, specify that the Windows service for the product runs with the same user ID from which the environment variable that specifies IPv6 is defined, instead of as local system.

      Default values for the Windows service

Windows The following default values for the Windows service definition panel exist:

- The default is to run as a Windows service.
- The service process is selected to run as a system account.
- The user account is the current user name. User name requirements are the requirements that the Windows operating system imposes for a user ID.
- The startup type is `automatic`. The values for the startup type are those values that the Windows operating system imposes. If you want a startup type other than `automatic`, you can either select another available option from the menu or change the startup type after you create the profile. You can also remove the created service after profile creation, and add it later with the desired startup type. You can choose not to create a service at profile creation time and optionally create the service later with the desired startup type.

- Linux

  The Linux service definition panel is displayed if the current operating system is a supported version of Linux operating systems, and the current user has the appropriate permissions.

  The product attempts to start Linux services for application server processes that are started by a startServer command. For example, if you configure an application server as a Linux service and issue the startServer command, then the **wasservice** command attempts to start the defined service.

  By default, the product is not selected to run as a Linux service.

  To create the service, the user that runs the Profile Management Tool must be the root user. If you run the Profile Management Tool with a non-root user ID, then the Linux service definition panel is not displayed, and no service is created.

  When you create a Linux service, you must specify a user name from which the service runs.

  To delete a Linux service, the user must be the root user or have appropriate privileges for deleting the service. Otherwise, a removal script is created that the root user can run to delete the service for the user.

If you previously selected **Advanced profile creation**, the next panel displays the Web server definition panel.

16. For advanced profile creation, if you choose to include a Web server definition in the profile now, specify the Web server characteristics on the panels, and click **Next** until you complete the Web server definition panels.

    If you use a Web server to route requests to the product, then you need to include a Web server definition. You can include the definition now, or define the Web server to the product later. If you define the Web server definition during the creation of this profile, then you can install the Web server and its plug-in after you create the profile. However, you must install both to the paths that you specify on the Web server definition panels. If you define the Web server to the product after you create this profile, then you must define the Web server in a separate profile.

    The tool displays the Profile Creation Summary panel.
17. Click **Create** to create the cell profile, or click **Back** to change the characteristics of the cell profile.

    The Profile creation progress panel, which shows the configuration commands that are running, is displayed.

    When the profile creation completes, the tool displays the Profile creation complete panel.
18. Optionally, select **Launch the First steps console**. Click **Finish** to exit.

    With the First steps console, you can create additional profiles and start the application server.

Results

You created a cell profile.

Refer to the description of the **manageprofiles** command to learn about creating a profile using a command instead of the Profile Management Tool.

What to do next

Deploy an application to get started.

Read about fast paths for the product to get started deploying applications.

Creating a custom profile

Create a custom profile so that you can include application servers, clusters, or other Java processes, such as a messaging server, in its empty node. You can use the Profile Management Tool to create a custom profile.

Before you begin

Before you use the Profile Management Tool, install the product files.

The Profile Management Tool is the graphical user interface for the **manageprofiles** command. See the description of the **manageprofiles** command for more information.

You must provide enough system temporary space to create a profile. For information, read about the file system requirements for profiles.

Note: When you launch the Profile Management Tool, the tool could lock up in the following situation for a non-root user: Log into a machine as root, use the SetPermissions utility to change the user from *x* to *y*. Assume that you are user *x* and log back into the machine. Launch the Profile Management Tool, click **Profile Management Tool**, and click **Create**. The next click after the click on **Create** could lock up the tool.

Note: When you use the Profile Management Tool with the Motif graphical user interface on the Solaris operating system, the default size of the Profile Management Tool might be too small to view all the messages and buttons of the Profile Management Tool. To fix the problem, add the following lines to the *app_server_root*/.Xdefaults file:

```
Eclipse*spacing:0
Eclipse*fontList:-misc-fixed-medium-r-normal-*-10-100-75-75-c-60-iso8859-1
```

After adding the lines, run the following command before launching the Profile Management Tool:

```
xrdb -load user_home/.Xdefaults
```

About this task

After installing the core product files for the Network Deployment product, you must create a profile. This topic describes creating a custom profile using the Profile Management Tool. A custom profile is an empty node that you can customize to include application servers, clusters, or other Java processes, such as a messaging server.

You can also use the **manageprofiles** command to create a custom profile. See the description of the **manageprofiles** for more information.

By default, the Profile Management Tool federates a custom node when you create a custom profile. Federating the node makes the node operational. You must have access to a running deployment manager to federate the node. Otherwise, a connection error displays. You can federate the node later if you do not have access to a running deployment manager, or for any other reason.

If the custom profile is on a machine that does not have a deployment manager, then the deployment manager must be accessible over the network to support the federation of the node.

You can create profiles with the Profile Management Tool using the typical profile creation process or the advanced profile creation process. The typical profile creation process uses default settings and assigns unique port values. You can optionally set values as allowed. For the advanced profile creation process you can accept the default values, or specify your own values.

1. Install the product to create the core product files.
2. Start the Profile Management Tool to create a new runtime environment.

   You can use one of the following ways to start the tool.

   - At the end of installation, select the check box to launch the Profile Management Tool.
   - Issue the command directly from a command prompt.

     The command is in the following directory:

     – Linux HP-UX Solaris AIX *app_server_root*/bin/ProfileManagement

     – Windows *app_server_root*\bin\ProfileManagement

     The name of the command varies per platform:

     – Linux HP-UX Solaris AIX pmt.sh

     – Windows pmt.bat
   - Select the Profile Management Tool option from the First steps console.
   - Windows Use the **Start** menu to access the Profile Management Tool. For example, click **Start > Programs** or **All Programs > IBM WebSphere > *your_product* > Profile Management Tool**.
   - Linux Use the Linux operating system menus used to start programs to start the Profile Management Tool. For example, click ***the_operating_system_menus_to_access_programs* > IBM WebSphere > *your_product* > Profile Management Tool**.
3. Click **Launch Profile Management Tool**, and then click **Create** on the Profiles tab to create a new profile.

   The Profiles tab contains a list of profiles that have been created on your machine. No action can be done on a selected profile unless the profile can be augmented. The Augment button is greyed out unless a profile that you select can be augmented.

   The tool displays the Environment selection panel.
4. Select the custom profile, and click **Next**.

   The Profile creation options panel is displayed.
5. Select either **Typical profile creation** or **Advanced profile creation**, and click **Next**.

   The **Typical profile creation** option creates a profile that uses default configuration settings. With the **Advanced profile creation** option, you can specify your own configuration values for a profile.
6. If you selected **Typical profile creation**, then go to the step on federating the node.
7. If you selected **Advanced profile creation**, then specify the custom profile name and the profile directory on the Profile name and location panel, or accept the defaults, and click **Next**.

   Profile naming guidelines: Double-byte characters are supported. The profile name can be any unique name with the following restrictions. Do not use any of the following characters when naming your profile:

   - Spaces
   - Special characters that are not supported within the name of a directory on your operating system, such as `*&?`
   - Slashes (/) or (\)

   The default profile

The first profile that you create on a machine is the default profile. The default profile is the default target for commands that are issued from the `bin` directory in the product installation root. When only one profile exists on a machine, every command works on the single server process in the configuration. You can make another profile the default profile when you create that profile by checking **Make this profile the default** on the Profile name and location panel of the **Advanced profile creation** path. You can also make another profile the default profile using the manageprofiles command after you create the profile.

Addressing a profile in a multiprofile environment

When multiple profiles exist on a machine, certain commands require that you specify the profile to which the command applies if the profile is not the default profile. These commands use the -profileName parameter to identify which profile to address. You might find it easier to use the commands that are in the `bin` directory of each profile.

Use these commands to query the command shell to determine the calling profile and to address these commands to the calling profile.

Default profile information

The default profile name is *profileTypeProfileName*:

- *profileType* is a value of `AppSrv`, `Dmgr`, `Custom`, `AdminAgent`, `JobMgr`, or `SecureProxySrv`.
- *ProfileName* is a sequential number that is used to create a unique profile name.

AIX HP-UX Linux Solaris The default profile directory is *app_server_root*/profiles, where *app_server_root* is the installation root.

Windows The default profile directory is *app_server_root*\profiles, where *app_server_root* is the installation root.

The tool then displays the Node and host names panel.

8. Specify the node and host characteristics for the custom profile, and click **Next**.

Migration considerations

If you plan to migrate an installation of Network Deployment Version 5 to Version 6, then use the same cell name for the Version 6 deployment manager that you used for the Version 5 cell. A cell name must be unique in any circumstance in which the product is running on the same physical machine or cluster of machines, such as a sysplex. Additionally, a cell name must be unique in any circumstance in which network connectivity between entities is required either between the cells or from a client that must communicate with each of the cells. Cell names must also be unique if their namespaces are federated. Otherwise, you might encounter symptoms such as a javax.naming.NameNotFoundException error, in which case, create uniquely named cells.

After migrating the cell, the Version 5 managed nodes are now managed by the Version 6 deployment manager in compatibility mode. You can migrate individual Version 5 managed nodes in the cell to Version 6. To do so, you must create a Version 6 profile with the same node name as the Version 5 managed node.

Reserved names: Avoid using reserved folder names as field values. The use of reserved folder names can cause unpredictable results. The following terms are reserved folder names:

- cells
- nodes
- servers
- clusters
- applications
- deployments

Some default values in the following table are split on multiple lines for printing purposes.

The custom profile has the following characteristics:

Field Name	Default Value	Constraints	Description
Node name	*shortHostName* Node *NodeNumber* where: • *shortHostName* is the short host name. • *NodeNumber* is a sequential number starting at 01.	Avoid using the reserved terms. Use a unique name within the deployment manager cell. If you plan to migrate a Version 5 managed node, then use the same node name for this Version 6 custom profile.	The name is used for administration within the deployment manager cell to which the custom profile is added. Use a unique name within the deployment manager cell. After migrating a Version 5 deployment manager cell to a Version 6 deployment manager, you can migrate the Version 5 custom profiles that are running in compatibility mode in the Version 6 deployment manager.
Host name	The long form of the domain name server (DNS) name.	The host name must be addressable through your network.	Use the actual DNS name or IP address of your machine to enable communication with your machine. See additional information about the host name that follows this table.

Directory path considerations

Windows The number of characters in the *profiles_directory_path\profile_name* directory must be less than or equal to 80 characters.

Host name considerations

The host name is the network name for the physical machine on which the node is installed. The host name must resolve to a physical network node on the server. When multiple network cards exist in the server, the host name or IP address must resolve to one of the network cards. Remote nodes use the host name to connect to and communicate with this node. Selecting a host name that other machines can reach within your network is important. Do not use the generic identifier, `localhost`, for this value. Also, do not attempt to install WebSphere Application Server products on a machine with a host name that uses characters from a double-byte character set (DBCS). DBCS characters are not supported when used in the host name.

If you define coexisting nodes on the same computer with unique IP addresses, then define each IP address in a domain name server (DNS) look-up table. Configuration files for stand-alone application servers do not provide domain name resolution for multiple IP addresses on a machine with a single network address.

The value that you specify for the host name is used as the value of the hostName property in configuration documents for the stand-alone application server. Specify the host name value in one of the following formats:

- Fully qualified domain name server (DNS) host name string, such as `xmachine.manhattan.ibm.com`
- The default short DNS host name string, such as `xmachine`
- Numeric IP address, such as `127.1.255.3`

The fully qualified DNS host name has the advantages of being unambiguous and flexible. You have the flexibility of changing the actual IP address for the host system without having to change the application server configuration. This value for the host name is particularly useful if you plan to change the IP address frequently when using Dynamic Host Configuration Protocol (DHCP) to assign IP addresses. A disadvantage of this format is dependency on DNS. If DNS is not available, then connectivity is compromised.

The short host name is also dynamically resolvable. A short name format has the added function of being redefined in the local hosts file so that the system can run the application server, even when disconnected from the network. To run disconnected, define the short name as the loopback address, `127.0.0.1`, in the hosts file to run disconnected. A disadvantage of this format is a dependency on DNS for remote access. If DNS is not available, then connectivity is compromised.

A numeric IP address has the advantage of not requiring name resolution through DNS. A remote node can connect to the node that you name with a numeric IP address without DNS being available. A disadvantage of this format is that the numeric IP address is fixed. You must change the setting of the hostName property in Express configuration documents whenever you change the machine IP address. Therefore, do not use a numeric IP address if you use DHCP, or if you change IP addresses regularly. Another disadvantage of this format is that you cannot use the node if the host is disconnected from the network.

After specifying custom profile characteristics, the tool displays the Federation panel.

9. If administrative security is enabled for the deployment manager, specify the host name and SOAP port of the deployment manager, and the user name and password for the deployment manager. Click **Next**.

   After federation, the process in the custom profile is the node agent process. The node agent process is the agent of the deployment manager for the custom node. The node agent responds to commands from the deployment manager to perform tasks that include the following actions:

   - Creating application server processes, clusters, and cluster members
   - Starting and stopping application server processes
   - Synchronizing configurations between the current edition on the deployment manager and the copy that exists on the node
   - Deleting application server processes

   See the system administration section in the information center for more information about node agents and their tasks.

   Should you federate the node?

   The recommendation is that you federate the custom node at this time. The deployment manager must be running and accessible when you click **Next** on the Federation panel to federate the custom node. If the custom profile is on a machine that does not have a deployment manager, then the deployment manager must be running and accessible over the network to allow the federation of the node. If the deployment manager is not running or not accessible before you click **Next**, but you can start it and make it accessible at this time, then do so. Otherwise, select the **Federate the node later** check box.

   If you are unsure whether the deployment manager is running or accessible, then do not federate now. Federate the node when you can verify the availability of the deployment manager.

   A possibility exists that the deployment manager is reconfigured to use the non-default remote method invocation (RMI) as the preferred Java Management Extensions (JMX) connector. Click **System Administration > Deployment manager > Administrative services** in the administrative console of the deployment manager to verify the preferred connector type.

   If RMI is the preferred JMX connector, then you must use the addNode command to federate the custom profile later. Use the addNode command so that you can specify the JMX connector type and the RMI port.

   If the deployment manager uses the default SOAP JMX connector type, specify the host name and SOAP port and federate the node now to create a functional node that you can customize.

   Federating when the deployment manager is not available

   If you federate a custom node when the deployment manager is not running or is not accessible, then an error message is displayed. If the deployment manager becomes unavailable during the profile creation process, then the installation indicator in the logs is `INSTCONFFAIL`, to indicate a complete failure. The resulting custom profile is unusable. You must delete the profile. Read about deleting a profile for more information.

If you chose to federate now, and you previously selected **Advanced profile creation**, then the Security certificate panel displays next. Go to the step on creating and importing certificates.

Otherwise, the Profile Creation Summary panel displays for the typical profile creation option. Go to the step on creating the custom profile.

10. Create a default personal certificate and a root signing certificate, or import a personal certificate and a root signing certificate from keystore files, and click **Next**.

    You can create both certificates, import both certificates, or create one certificate, and import the other certificate.

    Note: When you import a personal certificate as the default personal certificate, import the root certificate that signed the personal certificate. Otherwise, the Profile Management Tool adds the signer of the personal certificate to the trust.p12 file.

    If you import the default personal certificate or the root signing certificate, specify the path and the password, and select the keystore type and the keystore alias for each certificate that you import.

11. Verify that the certificate information is correct, and click **Next**.

    If you create the certificates, you can use the default values or modify them to create new certificates. The default personal certificate is valid for one year by default and is signed by the root signing certificate. The root signing certificate is a self-signed certificate that is valid for 20 years by default. The default keystore password for the root signing certificate is `WebAS`. You should change the password. The password cannot contain any double-byte character set (DBCS) characters because certain keystore types, including PKCS12, do not support these characters. The keystore types that are supported depend on the providers in the java.security file.

    When you create either or both certificates, or import either or both certificates, the keystore files that are created are key.p12, trust.p12, root-key.p12, default-signers.p12, deleted.p12, and ltpa.jceks. These files all have the same password when you create or import the certificates, which is either the default password, or a password that you specify. The key.p12 file contains the default personal certificate. The trust.p12 file contains the signer certificate from the default root certificate. The root-key.p12 file contains the root signing certificate. The default-signer.p12 file contains signer certificates that are added to any new keystore file that you create after the server is installed and running. By default, the default root certificate signer and a DataPower signer certificate is in the default-signer.p12 keystore file. The deleted.p12 keystore file is used to hold certificates deleted with the deleteKeyStore task so that they can be recovered if needed. The ltpa.jceks file contains server default Lightweight Third-Party Authentication (LTPA) keys that the servers in your environment use to communicate with each other.

    An imported certificate is added to the key.p12 file or the root-key.p12 file.

    If you import any certificates and the certificates do not contain the information that you want, click **Back** to import another certificate.

    After displaying the Security certificate panels, the tool displays the Ports panel if you previously selected **Advanced profile creation**.

12. Verify that the ports within the custom profile are unique, or intentionally conflicting, and click **Next**.

    Port conflict resolution

    If you suspect a port conflict, then you can investigate the port conflict after the profile is created. Determine the ports that are used during profile creation by examining the following files.

    - Linux HP-UX Solaris AIX *profile_root*/properties/portdef.props file
    - Windows *profile_root*\properties\portdef.props file

    Included in this file are the keys and values that are used in setting the ports. If you discover ports conflicts, then you can reassign ports manually. To reassign ports, run the updatePorts.ant file by using the ws_ant script.

    The Profile Creation Summary panel is displayed.

13. Click **Create** to create the custom profile, or click **Back** to change the characteristics of the custom profile.

If you previously chose to federate the custom node on the Federation panel, the deployment manager had to be running and accessible. The deployment manager must be running and accessible when you click **Create**. If you think the deployment manager might no longer be running or might have become inaccessible, then start the deployment manager and make it accessible, or make it accessible if it is already running.

The Profile creation progress panel, which shows the configuration commands that are running, is displayed.

When the profile creation completes, the tool displays the Profile creation complete panel.

14. Optionally, select **Launch the First steps console**. Click **Finish** to exit.

    With the First steps console, you can create additional profiles and start the application server.

Results

You created a custom profile. The node within the profile is empty until you federate the node and use the deployment manager to customize the node.

The directory structure shows the new profile folder within the profiles directory. The profile folder has the same name as the profile that you create.

Refer to the description of the **manageprofiles** command to learn about creating a profile using a command instead of the Profile Management Tool.

The Profile Management Tool creates a log during profile creation. The logs are in the *install_dir*/logs/manageprofiles directory. The files are named in this pattern: `manageprofiles_create_`*`profile_name`*`.log`.

What to do next

Federate the node into the deployment manager cell if you did not already do so when you created the node. Then, use the deployment manager to create an application server on the node.

Deploy an application to get started.

Read about fast paths for the product to get started deploying applications.

Creating an application server profile

Create an application server profile so that you can make applications available to the Internet or to an intranet, typically using Java technology. You can create an application server profile using the Profile Management Tool.

Before you begin

Before you use the Profile Management Tool, install the product files.

The Profile Management Tool is the graphical user interface for the **manageprofiles** command. See the description of the **manageprofiles** command for more information.

You must provide enough system temporary space to create a profile. For information, read about the file system requirements for profiles.

Note: When you launch the Profile Management Tool, the tool could lock up in the following situation for a non-root user: Log into a machine as root, use the SetPermissions utility to change the user from *x*

to *y*. Assume that you are user *x* and log back into the machine. Launch the Profile Management Tool, click **Profile Management Tool**, and click **Create**. The next click after the click on **Create** could lock up the tool.

Note: When you use the Profile Management Tool with the Motif graphical user interface on the Solaris operating system, the default size of the Profile Management Tool might be too small to view all the messages and buttons of the Profile Management Tool. To fix the problem, add the following lines to the *app_server_root*/.Xdefaults file:

```
Eclipse*spacing:0
Eclipse*fontList:-misc-fixed-medium-r-normal-*-10-100-75-75-c-60-iso8859-1
```

After adding the lines, run the following command before launching the Profile Management Tool:

```
xrdb -load user_home/.Xdefaults
```

About this task

After installing the core product files for the Network Deployment product, you must create a profile. This procedure describes creating an application server profile using the graphical user interface provided by the Profile Management Tool. You can also use the **manageprofiles** command to create an application server profile. See the description of the manageprofiles command for more information.

An application server profile has a default server, which is server1, the default application that includes the Snoop servlet and the Hitcount servlet, and application Samples. You can federate the application server or use it as a stand-alone application server.

You can create profiles with the Profile Management Tool using the typical profile creation process or the advanced profile creation process. The typical profile creation process uses default settings and assigns unique port values. You can optionally set values as allowed. For the advanced profile creation process you can accept the default values, or specify your own values.

1. Start the Profile Management Tool to create a new runtime environment.

   You can use one of the following ways to start the tool.

   - At the end of installation, select the check box to launch the Profile Management Tool.
   - Issue the command directly from a command prompt.

     The command is in the following directory:

     – Linux HP-UX Solaris AIX *app_server_root*/bin/ProfileManagement

     – Windows *app_server_root*\bin\ProfileManagement

     The name of the command varies per platform:

     – Linux HP-UX Solaris AIX pmt.sh

     – Windows pmt.bat
   - Select the Profile Management Tool option from the First steps console.
   - Windows Use the **Start** menu to access the Profile Management Tool. For example, click **Start > Programs** or **All Programs > IBM WebSphere > *your_product* > Profile Management Tool**.
   - Linux Use the Linux operating system menus used to start programs to start the Profile Management Tool. For example, click ***the_operating_system_menus_to_access_programs* > IBM WebSphere > *your_product* > Profile Management Tool**.
2. Click **Launch Profile Management Tool**, and then click **Create** on the Profiles tab to create a new profile.

   The Profiles tab contains a list of profiles that have been created on your machine. No action can be done on a selected profile unless the profile can be augmented. The Augment button is greyed out unless a profile that you select can be augmented.

The tool displays the Environment selection panel.

3. Select **Application server** and click **Next**.

   The Profile creation options panel is displayed.

4. Select either **Typical profile creation** or **Advanced profile creation**, and click **Next**.

   The **Typical profile creation** option creates a profile that uses default configuration settings. With the **Advanced profile creation** option, you can specify your own configuration values for a profile.

5. If you selected **Typical profile creation**, then go to the step on administrative security.

6. If you selected **Advanced profile creation**, then select the applications that you want to deploy; and click **Next**.

   The tool displays the Profile name and location panel.

7. Specify a name for the profile and the directory path for the profile directory, or accept the default values. Then, click **Next**.

   Profile naming guidelines: Double-byte characters are supported. The profile name can be any unique name with the following restrictions. Do not use any of the following characters when naming your profile:

   - Spaces
   - Special characters that are not supported within the name of a directory on your operating system, such as `*&?`
   - Slashes (/) or (\)

   The default profile

   The first profile that you create on a machine is the default profile. The default profile is the default target for commands that are issued from the `bin` directory in the product installation root. When only one profile exists on a machine, every command works on the single server process in the configuration. You can make another profile the default profile when you create that profile by checking **Make this profile the default** on the Profile name and location panel of the **Advanced profile creation** path. You can also make another profile the default profile using the manageprofiles command after you create the profile.

   Addressing a profile in a multiprofile environment

   When multiple profiles exist on a machine, certain commands require that you specify the profile to which the command applies if the profile is not the default profile. These commands use the -profileName parameter to identify which profile to address. You might find it easier to use the commands that are in the `bin` directory of each profile.

   Use these commands to query the command shell to determine the calling profile and to address these commands to the calling profile.

   Default profile information

   The default profile name is *profileTypeProfileName*:

   - *profileType* is a value of `AppSrv`, `Dmgr`, `Custom`, `AdminAgent`, `JobMgr`, or `SecureProxySrv`.
   - *ProfileName* is a sequential number that is used to create a unique profile name.

   AIX HP-UX Linux Solaris The default profile directory is *app_server_root*/profiles, where *app_server_root* is the installation root.

   Windows The default profile directory is *app_server_root*\profiles, where *app_server_root* is the installation root.

8. On the Node and host names panel, specify the characteristics for the application server, and click **Next**.

   Use unique names for each application server that you create.

   Reserved names: Avoid using reserved folder names as field values. The use of reserved folder names can cause unpredictable results. The following terms are reserved folder names:

   - cells

- nodes
- servers
- clusters
- applications
- deployments

Some default values in the following table are split on multiple lines for printing purposes.

Field Name	Default Value	Constraints	Description
Node name	*shortHostName* Node *NodeNumber* where: • *shortHostName* is the short host name. • *NodeNumber* is a sequential number starting at 01.	Avoid using the reserved terms.	Select any name you want. To help organize your installation, use a unique name if you plan to create more than one application server on the machine.
Server name	server1	Use a unique name for the application server.	The name is a logical name for the application server.
Host name	The long form of the domain name server (DNS) name.	Addressable through your network.	Use the DNS name or IP address of your machine to enable communication with your machine. See additional information about the host name following this table.

Node name considerations: If you plan to migrate an installation of Version 5 Network Deployment to Version 6 and migrate one of the managed nodes in the cell, use the same node name for the Version 6 application server that you used for the Version 5 managed node.

Windows **Directory path considerations:** The installation directory path must be less than or equal to 60 characters.

Host name considerations:

The host name is the network name for the physical machine on which the node is installed. The host name must resolve to a physical network node on the server. When multiple network cards exist in the server, the host name or IP address must resolve to one of the network cards. Remote nodes use the host name to connect to and communicate with this node. Selecting a host name that other machines can reach within your network is important. Do not use the generic identifier, `localhost`, for this value. Also, do not attempt to install WebSphere Application Server products on a machine with a host name that uses characters from a double-byte character set (DBCS). DBCS characters are not supported when used in the host name.

If you define coexisting nodes on the same computer with unique IP addresses, then define each IP address in a domain name server (DNS) look-up table. Configuration files for stand-alone application servers do not provide domain name resolution for multiple IP addresses on a machine with a single network address.

The value that you specify for the host name is used as the value of the hostName property in configuration documents for the stand-alone application server. Specify the host name value in one of the following formats:

- Fully qualified domain name server (DNS) host name string, such as `xmachine.manhattan.ibm.com`
- The default short DNS host name string, such as `xmachine`
- Numeric IP address, such as `127.1.255.3`

The fully qualified DNS host name has the advantages of being unambiguous and flexible. You have the flexibility of changing the actual IP address for the host system without having to change the application server configuration. This value for the host name is particularly useful if you plan to change the IP address frequently when using Dynamic Host Configuration Protocol (DHCP) to assign IP addresses. A disadvantage of this format is dependency on DNS. If DNS is not available, then connectivity is compromised.

The short host name is also dynamically resolvable. A short name format has the added function of being redefined in the local hosts file so that the system can run the application server, even when disconnected from the network. To run disconnected, define the short name as the loopback address, `127.0.0.1`, in the hosts file to run disconnected. A disadvantage of this format is a dependency on DNS for remote access. If DNS is not available, then connectivity is compromised.

A numeric IP address has the advantage of not requiring name resolution through DNS. A remote node can connect to the node that you name with a numeric IP address without DNS being available. A disadvantage of this format is that the numeric IP address is fixed. You must change the setting of the hostName property in Express configuration documents whenever you change the machine IP address. Therefore, do not use a numeric IP address if you use DHCP, or if you change IP addresses regularly. Another disadvantage of this format is that you cannot use the node if the host is disconnected from the network.

After specifying application server characteristics, the tool displays the Administrative security panel.

9. Optionally enable administrative security, and click **Next**.

   You can enable administrative security now during profile creation, or later from the console. If you enable administrative security now, then enter a user name and password to log onto the administrative console.

   If you installed the Samples, and you chose to deploy them, then the Samples require an account under which to run. Supply the Samples password for the account. You cannot change the user name of the account.

   After specifying security characteristics, the tool displays the Security certificate panel if you previously selected **Advanced profile creation**.

10. If you selected **Typical profile creation** at the beginning of these steps, go to the step that displays the Profile summary panel.

11. Create a default personal certificate and a root signing certificate, or import a personal certificate and a root signing certificate from keystore files, and click **Next**.

    You can create both certificates, import both certificates, or create one certificate, and import the other certificate.

    Note: When you import a personal certificate as the default personal certificate, import the root certificate that signed the personal certificate. Otherwise, the Profile Management Tool adds the signer of the personal certificate to the trust.p12 file.

    If you import the default personal certificate or the root signing certificate, specify the path and the password, and select the keystore type and the keystore alias for each certificate that you import.

12. Verify that the certificate information is correct, and click **Next**.

    If you create the certificates, you can use the default values or modify them to create new certificates. The default personal certificate is valid for one year by default and is signed by the root signing certificate. The root signing certificate is a self-signed certificate that is valid for 20 years by default. The default keystore password for the root signing certificate is `WebAS`. You should change the password. The password cannot contain any double-byte character set (DBCS) characters because certain keystore types, including PKCS12, do not support these characters. The keystore types that are supported depend on the providers in the java.security file.

    When you create either or both certificates, or import either or both certificates, the keystore files that are created are key.p12, trust.p12, root-key.p12, default-signers.p12, deleted.p12, and ltpa.jceks. These files all have the same password when you create or import the certificates, which is either the default password, or a password that you specify. The key.p12 file contains the default personal

certificate. The trust.p12 file contains the signer certificate from the default root certificate. The root-key.p12 file contains the root signing certificate. The default-signer.p12 file contains signer certificates that are added to any new keystore file that you create after the server is installed and running. By default, the default root certificate signer and a DataPower signer certificate is in the default-signer.p12 keystore file. The deleted.p12 keystore file is used to hold certificates deleted with the deleteKeyStore task so that they can be recovered if needed. The ltpa.jceks file contains server default Lightweight Third-Party Authentication (LTPA) keys that the servers in your environment use to communicate with each other.

An imported certificate is added to the key.p12 file or the root-key.p12 file.

If you import any certificates and the certificates do not contain the information that you want, click **Back** to import another certificate.

After displaying the Security certificate panels, the tool displays the Ports panel if you previously selected **Advanced profile creation**.

13. Verify that the ports specified for the stand-alone application server are unique, and click **Next**.

    If you chose not to deploy the administrative console, then the administrative console ports are disabled on the Ports panel.

    Port conflict resolution

    Ports are recognized as being in use if one of the following conditions exists:

    - The ports are assigned to a profile created from an installation that is performed by the current user.
    - The port is currently in use.

    Validation of ports occurs when you access the Port value assignment panel. Conflicts can still occur between the Port value assignment panel and the Profile creation complete panel because ports are not assigned until profile creation completes.

    If you suspect a port conflict, then you can investigate the port conflict after the profile is created. Determine the ports that are used during profile creation by examining the following files.

    - Linux HP-UX Solaris AIX *profile_root*/properties/portdef.props file
    - Windows *profile_root*\properties\portdef.props file

    Included in this file are the keys and values that are used in setting the ports. If you discover ports conflicts, then you can reassign ports manually. To reassign ports, run the updatePorts.ant file by using the ws_ant script.

    Windows Linux The tool displays the Windows service definition panel if you are installing on a Windows operating system and the installation ID has the administrative group privilege. The tool displays the Linux service definition panel if you are installing on a supported Linux operating system and the ID that runs the Profile Management Tool is the root user.

14. Choose whether to run the application server as a Windows service on a Windows operating system or as a Linux service on a Linux operating system, then click **Next**.

    - Windows

      The Windows service definition panel is displayed for the Windows operating system only if the ID that installs the Windows service has the administrator group privilege. However, you can run the WASService.exe command to create the Windows service as long as the installer ID belongs to the administrator group. Read about automatically restarting server processes for more information.

      Windows The product attempts to start Windows services for application server processes that are started by a startServer command. For example, if you configure an application server as a Windows service, and issue the startServer command, then the **wasservice** command attempts to start the defined service.

      If you chose to install a local system service, then you do not have to specify your user ID or password. If you create a specified user type of service, then you must specify the user ID and the password for the user who runs the service. The user must have `Log on as a service` authority for

the service to run correctly. If the user does not have `Log on as a service` authority, then the Profile Management tool automatically adds the authority.

To perform this profile creation task, the user ID must not contain spaces. In addition to belonging to the administrator group, the ID must also have the advanced user right of `Log on as a service`. The Installation wizard grants the user ID the advanced user right if the user ID does not already have the advanced user right and if the user ID belongs to the administrator group.

You can also create other Windows services after the installation is complete to start other server processes. Read about automatically restarting server processes for more information.

You can remove the Windows service that is added during profile creation during profile deletion. You can also remove the Windows service with the wasservice command.

IPv6 considerations

Profiles created to run as a Windows service fail to start when using Internet Protocol Version 6 (IPv6) if the service is configured to run as local system. Create a user-specific environment variable to enable IPv6. Since this environment variable is a user variable instead of a local system variable, only a Windows service that runs as that specific user can access this environment variable. By default, when a new profile is created and configured to run as a Windows service, the service is set to run as local system. When the Windows service for the product tries to run, the service is unable to access the user environment variable that specifies IPv6, and thus, tries to start as Internet Protocol Version 4 (IPv4). The server does not start correctly in this case. To resolve the problem, when creating the profile, specify that the Windows service for the product runs with the same user ID from which the environment variable that specifies IPv6 is defined, instead of as local system.

Default values for the Windows service

Windows The following default values for the Windows service definition panel exist:

- The default is to run as a Windows service.
- The service process is selected to run as a system account.
- The user account is the current user name. User name requirements are the requirements that the Windows operating system imposes for a user ID.
- The startup type is `automatic`. The values for the startup type are those values that the Windows operating system imposes. If you want a startup type other than `automatic`, you can either select another available option from the menu or change the startup type after you create the profile. You can also remove the created service after profile creation, and add it later with the desired startup type. You can choose not to create a service at profile creation time and optionally create the service later with the desired startup type.

- Linux

  The Linux service definition panel is displayed if the current operating system is a supported version of Linux operating systems, and the current user has the appropriate permissions.

  The product attempts to start Linux services for application server processes that are started by a startServer command. For example, if you configure an application server as a Linux service and issue the startServer command, then the **wasservice** command attempts to start the defined service.

  By default, the product is not selected to run as a Linux service.

  To create the service, the user that runs the Profile Management Tool must be the root user. If you run the Profile Management Tool with a non-root user ID, then the Linux service definition panel is not displayed, and no service is created.

  When you create a Linux service, you must specify a user name from which the service runs.

  To delete a Linux service, the user must be the root user or have appropriate privileges for deleting the service. Otherwise, a removal script is created that the root user can run to delete the service for the user.

If you previously selected **Advanced profile creation**, the next panel displays the Web server definition panel.

15. For advanced profile creation, if you choose to include a Web server definition in the profile now, specify the Web server characteristics on the panels, and click **Next** until you complete the Web server definition panels.

    If you use a Web server to route requests to the product, then you need to include a Web server definition. You can include the definition now, or define the Web server to the product later. If you define the Web server definition during the creation of this profile, then you can install the Web server and its plug-in after you create the profile. However, you must install both to the paths that you specify on the Web server definition panels. If you define the Web server to the product after you create this profile, then you must define the Web server in a separate profile.

    The tool displays the Profile Creation Summary panel.
16. Click **Create** to create the application server, or click **Back** to change the characteristics of the application server.

    The Profile creation progress panel, which shows the configuration commands that are running, is displayed.

    When the profile creation completes, the tool displays the Profile creation complete panel.
17. Optionally, select **Launch the First steps console**. Click **Finish** to exit.

    With the First steps console, you can create additional profiles and start the application server.

Results

You created an application server profile. The node within the profile has an application server named server1.

Refer to the description of the **manageprofiles** command to learn about creating a profile using a command instead of the Profile Management Tool.

What to do next

Deploy an application to get started.

Read about fast paths for the product to get started deploying applications.

When you create the application server profile, a default server1 process is created. You can federate the server1 node into the deployment manager cell with the **addNode** command or from the administrative console of the deployment manager. The server1 process must be running to begin the federation from the deployment manager.

If you include all of the applications from the application server, then the act of federation installs the applications on the deployment manager where they can be redeployed.

Creating profiles for non-root users

The non-root user can receive permissions for files and directories so that the non-root user can create a profile.

Before you begin

This task assumes a basic familiarity with the **manageprofiles** command, the Profile Management Tool, and system commands.

This task uses the following terms:

- *Root users* refers to:
  - Linux HP-UX Solaris AIX Root users

  - Windows Administrators
- *Non-root users* refers to:
  - Linux HP-UX Solaris AIX Non-root users
  - Windows Non-administrators
- *Installer* refers to a root user or a non-root user.

Note: An ease-of-use limitation exists for non-root users who create profiles. Mechanisms within the Profile Management Tool that suggest unique names and port values are disabled for non-root users. The non-root user must change the default field values in the Profile Management Tool for the profile name, node name, cell name, and port assignments. Consider assigning non-root users a range of values for each of the fields. You can assign responsibility to the non-root users for adhering to their assigned value ranges and for maintaining the integrity of their own definitions.

About this task

Non-root users might typically need these tasks completed so that they can start their own application servers in development environments. For instance, an application developer might test an application on a application server in a profile assigned to that application developer.

- Create a profile as an installer and assign ownership to a non-root user.

  This topic describes how the installer creates a profile and assigns ownership of the profile directory to a non-root user so that the non-root user can start the application server for a specific profile.
- Grant write permission of files and directories to a non-root user for profile creation.

  This topic describes how an installer authorizes a group to certain files and directories so that non-root users in the group can create profiles.
- Install maintenance as an installer and change the ownership of profile related files.

  This topic describes how to install product maintenance and change the ownership of new profile files to the non-root user that owns the profile. The installer changes ownership of the files so that the non-root user can then successfully start the application server.

Results

Depending on the tasks that the installer followed, the installer has completed the following actions:

- Created a profile for a non-root user and assigned ownership of the profile directory to the non-root user
- Granted permission to the appropriate directories so that non-root users can create profiles
- After installing maintenance, changed ownership of new profile files in a directory that is owned by a non-root user, so that the non-root user can successfully start the application server

What to do next

Depending on the tasks that the installer completes, a non-root user can create a profile, start WebSphere Application Server, or do both.

Creating a profile as an installer and assigning ownership to a non-root user

An installer can create a profile and assign ownership of the profile directory to a non-root user so that the non-root user can start the product for a specific profile. Use this example to accomplish the tasks through commands.

Before you begin

This task assumes a basic familiarity with the **manageprofiles** command and system commands.

This task uses the following terms:

- *Root users* refers to:
  - Linux HP-UX Solaris AIX Root users
  - Windows Administrators
- *Non-root users* refers to:
  - Linux HP-UX Solaris AIX Non-root users
  - Windows Non-administrators
- *Installer* refers to a root user or a non-root user.

Before you can create a profile, you must install the product.

About this task

Have the installer perform the following steps to create a profile and assign ownership for the profile directory and the logs directory. The ownership is assigned to a non-root user ID that is different from the installer ID. The non-root user needs access to these directories to start the product.

This example creates a default profile.

The commands are split on multiple lines for printing purposes.

1. Create the profile by issuing the following code from a command prompt:

   Linux HP-UX Solaris AIX

   ```
   ./manageprofiles.sh -create -profileName profile01 -profilePath
   app_server_root/profiles/profile01 -templatePath
   app_server_root/profileTemplates/default
   ```

   Windows

   ```
   manageprofiles.bat -create -profileName profile01 -profilePath
   app_server_root\profiles\profile01 -templatePath
   app_server_root\profileTemplates\default
   ```

2. Change ownership of the profile01 profile directory to the user1 non-root user.

   Linux HP-UX Solaris AIX For example, issue the following command:

   ```
   chown -R user1 app_server_root/profiles/profile01
   ```

   Windows Follow instructions in the Windows documentation to grant user1 access to the following directory:

   ```
   app_server_root\profiles\profile01
   ```

3. Change the ownership of the logs directory for the profile01 profile to the user1 non-root user to prevent displaying log messages to the console.

   Linux HP-UX Solaris AIX Issue the following command:

   ```
   chown -R user1 app_server_root/logs/manageprofiles/profile01
   ```

   Windows Follow instructions in the Windows documentation to grant user1 access to the following directory:

   ```
   app_server_root\logs\manageprofiles\profile01
   ```

Results

The installer has created a default profile and changed ownership of the profile directory and log directory to a non-root user.

What to do next

As the installer, you can continue to create profiles and assign ownership to non-root users as needed.

A non-root user ID can manage multiple profiles. Have the same non-root user ID manage an entire profile, whether it is the deployment manager profile, a profile that contains the application servers and the node agent, or a custom profile. A different user ID can be used for each profile in a cell, whether global security or administrative security is enabled or disabled. The user IDs can be a mix of root and non-root user IDs. For example, the root user might manage the deployment manager profile, while a non-root user might manage a profile that contains application servers and the node agent, or vice versa. However, typically, a root user or a non-root user manages all profiles in a cell.

The non-root user can use the same tasks to manage a profile that the root user uses.

Granting write permission of files and directories to a non-root user for profile creation

The installer can grant write permission of the appropriate files and directories to a non-root user. The non-root user can then create the profile. The installer can create a group for users who are authorized to create profiles, or the installer can give individual users the authority to create profiles. The following example task shows how to create a group that is authorized to create profiles.

Before you begin

This task assumes a basic familiarity with system commands.

This task uses the following terms:

- *Root users* refers to:
  - Linux HP-UX Solaris AIX Root users
  - Windows Administrators
- *Non-root users* refers to:
  - Linux HP-UX Solaris AIX Non-root users
  - Windows Non-administrators
- *Installer* refers to a root user or a non-root user.

About this task

The steps that you follow to grant write permission of files and directories to a non-root user for profile creation depend on whether a profile was previously created.

If at least one profile was created prior to implementing the following steps, then certain directories and files were created. Because these directories and files were created, skip the steps that create these directories and files. If no profile was previously created, then you must complete the steps to create the required directories and files. In most cases, a profile has been created previously.

The installer can perform the following steps to create the profilers group and give the group appropriate permissions to create a profile.

1. Log on as the installer to the system where the product is installed.

2. Create the profilers group that you can use to create profiles.
3. Create a user named user1 to create profiles.
4. Add the installer and user1 to the profilers group.
5. Linux HP-UX Solaris AIX Log off and log back on again as the installer to use the new group.
6. Create the following directories as the installer, if no profile was previously created:
   - Linux HP-UX Solaris AIX Create the *app_server_root*/logs/manageprofiles directory:

     ```
     mkdir app_server_root/logs/manageprofiles
     ```

     Windows Create the *app_server_root*\logs\manageprofiles directory by following instructions in the Windows documentation. For this example procedure the directory is:

     ```
     app_server_root\logs\manageprofiles
     ```

   - Linux HP-UX Solaris AIX Create the *app_server_root*/properties/fsdb directory:

     ```
     mkdir app_server_root/properties/fsdb
     ```

     Windows Create the *app_server_root*\properties\fsdb directory by following instructions in the Windows documentation. For this example procedure the directory is:

     ```
     app_server_root\properties\fsdb
     ```

7. As the installer, create the profileRegistry.xml file and add the appropriate information, if no profile was previously created.

   Follow directions for your operating system to create the profileRegistry.xml file. For this example, the file paths are: Linux HP-UX Solaris AIX

   ```
   app_server_root/properties/profileRegistry.xml
   ```

   Windows

   ```
   app_server_root\properties\profileRegistry.xml
   ```

   Follow instructions for your operating system to add the following information to the profileRegistry.xml file. The file must be encoded as UTF-8.

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <profiles/>
   ```

8. As the installer, use operating system tools to change directory and file permissions.

   Linux HP-UX Solaris AIX The following example assumes that the installation root directory is /opt/IBM/WebSphere/AppServer:

   ```
   chgrp  profilers /opt/IBM/WebSphere/AppServer/logs/manageprofiles
   chmod  g+wr  /opt/IBM/WebSphere/AppServer/logs/manageprofiles
   chgrp  profilers /opt/IBM/WebSphere/AppServer/properties
   chmod  g+wr  /opt/IBM/WebSphere/AppServer/properties
   chgrp  profilers /opt/IBM/WebSphere/AppServer/properties/fsdb
   chmod  g+wr  /opt/IBM/WebSphere/AppServer/properties/fsdb
   chgrp  profilers /opt/IBM/WebSphere/AppServer/properties/profileRegistry.xml
   chmod  g+wr  /opt/IBM/WebSphere/AppServer/properties/profileRegistry.xml
   chgrp -R profilers /opt/IBM/WebSphere/AppServer/profileTemplates
   ```

   HP-UX If you create a cell profile, additionally issue the following commands:

   ```
   chmod  -R g+wr /opt/IBM/WebSphere/AppServer/profileTemplates/cell/default/documents
   chmod  -R g+wr /opt/IBM/WebSphere/AppServer/profileTemplates/cell/dmgr/documents
   ```

   HP-UX If you create an application server profile, a deployment manager profile, or a custom profile, then additionally issue the following command:

   ```
   chmod  -R g+wr /opt/IBM/WebSphere/AppServer/profileTemplates/profile_template_name/documents
   ```

   profile_template_name is `default`, `dmgr`, or `managed`, respectively.

HP-UX The ownership of files is preserved when the files are copied to the profile directory during profile creation. You granted write permission to the profile directory so that files copied to the profile directory can be modified as part of the profile creation process. Files that are already in the profileTemplate directory structure prior to the start of profile creation are not modified during profile creation. Linux

```
chgrp profilers /opt/IBM/WebSphere/AppServer/properties/Profiles.menu
chmod  g+wr /opt/IBM/WebSphere/AppServer/properties/Profiles.menu
```

Windows The following example assumes that the installation root directory is C:\Program Files\IBM\WebSphere\AppServer. Follow instructions in the Windows documentation to give the profilers group read and write permission to the following directories and their files:

```
C:\Program Files\IBM\WebSphere\AppServer\logs\manageprofiles
C:\Program Files\IBM\WebSphere\AppServer\properties
C:\Program Files\IBM\WebSphere\AppServer\properties\fsdb
C:\Program Files\IBM\WebSphere\AppServer\properties\profileRegistry.xml
```

You might have to change the permissions on additional files if the non-root user encounters permission errors. For example, if you authorize a non-root user to delete a profile, then the user might have to delete the following file:

Linux HP-UX Solaris AIX *app_server_root*/properties/profileRegistry.xml_LOCK

Windows *app_server_root*\properties\profileRegistry.xml_LOCK

- Give write access to the non-root user for the file to authorize the user to delete the file. If the non-root user still cannot delete the profile, then the installer can delete the profile.

Results

The installer created the profilers group and gave the group proper permissions to certain directories and files to create profiles.

These directories and files are the only ones in the installation root of the product to which a non-root user needs to write to create profiles.

What to do next

The non-root user that belongs to the profilers group can create profiles in a directory that the non-root user owns and to which the non-root user has write permission. However, the non-root user cannot create profiles in the installation root directory of the product.

A non-root user ID can manage multiple profiles. The same non-root user ID can manage an entire profile, whether it is the deployment manager profile, a profile that contains the application servers and the node agent, or a custom profile. A different user ID can be used for each profile in a cell, whether global security or administrative security is enabled or disabled. The user IDs can be a mix of root and non-root user IDs. For example, the root user might manage the deployment manager profile, while a non-root user might manage a profile that contains application servers and the node agent, or vice versa. However, typically, a root user or a non-root user can manage all profiles in a cell.

The non-root user can use the same tasks to manage a profile that the root user uses.

Installing maintenance packages and changing the ownership of profile-related files

When an installer installs a maintenance package that contains service for a profile that a non-root user owns, the installer owns any new files that the maintenance package creates. The installer can change the ownership of the new files so that a non-root user can successfully start the product.

Before you begin

This task assumes a basic familiarity with the Update Installer wizard and system commands.

This task uses the following terms:

- *Root users* refers to:
  - Linux HP-UX Solaris AIX Root users
  - Windows Administrators
- *Non-root users* refers to:
  - Linux HP-UX Solaris AIX Non-root users
  - Windows Non-administrators
- *Installer* refers to a root user or a non-root user.

Before you can update a profile, you must install the product, and create a profile.

About this task

This example assumes that the installer completes the following actions:

- Applies service that creates new files in a profiles directory that the wsdemo non-root user owns
- Changes ownership of new profile files from the installer to the wsdemo non-root user.

If the installer does not change ownership, then when the non-root user starts the product, the application server encounters an error and issues a message that is similar to the following example:

```
ADMR0104E:
The system is unable to read document
cells/express1Cell/nodes/express1/node-metadata.properties:
java.io.IOException: No such file or directory
```

1. Run the update installer wizard to install maintenance packages for the product.

   The installer owns the new files that the Update Installer wizard creates in the *profile_root* directory. The original owner of existing files continues to own those files that the Update Installer wizard only modifies.
2. Reassign ownership of the entire profile directory to the wsdemo non-root user.

   The *profile_root* variable in the following examples is the profile directory that the non-root user owns.

   Linux HP-UX Solaris AIX Issue the **chown** command.

   ```
   chown -R wsdemo profile_root
   ```

   Windows Follow instructions in the Windows documentation to reassign ownership of the *profile_root* profile directory to the wsdemo non-root user.

Results

The installer installed a maintenance package that creates new files in a non-root user profile directory and changes ownership of the new files to the non-root owner.

What to do next

The non-root user can start the product without receiving the ADMR0104E error message.

Deleting a profile

You can delete a profile using the manageprofiles command. If the command fails, you can delete the profile using operating system commands.

Before you begin

If a node within a profile is federated to a deployment manager, before you delete the profile, stop the node and remove the node from the deployment manager. Otherwise, an orphan node is left in the deployment manager.

If you delete a profile that has augmenting templates registered to it in the profile registry, then unaugment actions are attempted prior to the deletion.

You cannot delete a profile using the Profile Management Tool.

About this task

The following example attempts to delete a profile using the manageprofiles command, and then using operating system commands.

1. Issue the manageprofiles command to delete a profile.

   Substitute your profile name for the *profile_name* value in the following commands.

   Linux HP-UX Solaris AIX

   ```
   ./manageprofiles.sh -delete
                       -profileName profile_name
   ```

   Windows

   ```
   manageprofiles.bat -delete
                      -profileName profile_name
   ```

   If the command is successful, you have completed the task and can skip the remaining steps. If the command is partially successful or unsuccessful, proceed to the next step to delete the profile manually. If you receive the `INSTCONFFAILED: Cannot delete profile.` message, the command was unsuccessful. If the deletion is partially successful, you could receive message information similar to the following wording:

   ```
   INSTCONFPARTIALSUCCESS: The profiles no longer exist, but errors occurred.
   For more information, consult
   app_server_root/logs/manageprofiles/deleteAll.log.
   ```

   or

   ```
   The current user does not have sufficient permissions to detect or
   remove services. If a service does exist, then an administrative or root user has
   to remove it. If a service does not exist, then no further action is
   required.
   ```

2. Issue operating system commands to delete the profile directory.
3. Issue the following command to remove references in the registry to deleted profiles:

   Linux HP-UX Solaris AIX

   ```
   ./manageprofiles.sh -validateAndUpdateRegistry
   ```

   Windows

   ```
   manageprofiles.bat -validateAndUpdateRegistry
   ```

   Editing of the registry is not recommended.

Results

You have now deleted a profile.

What to do next

You can delete other profiles using this procedure, or create other profiles using the manageprofiles command or the Profile Management Tool.

Chapter 3. Setting up the administrative architecture

You can monitor and control incorporated nodes and the resources on those nodes by using these tasks with the administrative console or other administrative tools.

About this task

After you install and set up the Network Deployment product, you mainly need to monitor and control incorporated nodes and the resources on those nodes by using the administrative console or other administrative tools. Use the following tasks to perform these activities.

- Use the settings page for an administrative service to configure administrative services.
- Configure cells.
- Configure deployment managers.
- Manage nodes.
- Manage node agents.
- Manage node groups.
- Configure remote file services.
- Administer job managers.
- Change the host name.
- Administer multiple application servers through an administrative agent.

Configuring cells

This topic describes how to change the cell protocol information, define custom properties for the cell, and add additional nodes.

Before you begin

Before you can configure cells, you must install the WebSphere Application Server Network Deployment product.

About this task

When you create a deployment manager profile, a cell is created. A cell provides a way to group one or more nodes of your Network Deployment product. You define the nodes that make up a cell, according to the specific criteria that make sense in your organizational environment. You probably do not need to configure the cell again.

Administrative configuration data is stored in XML files. A cell retains master configuration files for each server in every node in the cell. Each node and server also have their own local configuration files. Changes to a local node or to a server configuration file are temporary, if the server belongs to the cell. While in effect, local changes override cell configurations. Changes to the master server and master node configuration files made at the cell level replace any temporary changes made at the node when the cell configuration documents are synchronized to the nodes. Synchronization occurs at designated events, such as when a server starts.

To view information about and to manage a cell, use the settings page for a cell.

1. Access the settings page for a cell. Click **System Administration > Cell** from the navigation tree of the administrative console.
2. If the protocol that the cell uses to retrieve information from a network is not appropriate for your system, select the appropriate protocol. By default, a cell uses Transmission Control Protocol (TCP). If

you want the cell to use User Datagram Protocol, select **UDP** from the list for **Cell discovery protocol** on the settings page for the cell. It is unlikely that you need to change the cell protocol configuration from TCP.

3. Click **Custom Properties** and define any name-value pairs that your deployment manager needs.
   a. Click **New**.
   b. Specify a name and value for the custom property.

      The IBM_CLUSTER_RIPPLESTART_NOTIFICATION_TIMEOUT custom property:

      Specify this custom property with an integer value in milliseconds to indicate the amount of time the ripplestart function waits for processes to shut down before restarting them. If you attempt a ripplestart and the processes have not shutdown before the start operation begins, one or more of the processes will not restart.

Property	IBM_CLUSTER_RIPPLESTART_NOTIFICATION_TIMEOUT
Data type	integer
Default	300000 milliseconds (5 minutes)

The com.ibm.websphere.management.launcher.options custom property:

Specify this custom property with a value of `displayServerInFront` to display the name of the cell, node, and server in front of the output for the ps -ef command. Use of this property is intended to help you identify the process ID of a server. The property has no impact on the server process.

Property	com.ibm.websphere.management.launcher.options
Data type	String
Default	None

4. When you install the WebSphere Application Server Network Deployment product, a node may have been added to the cell. You can add additional nodes on the Node page. Click **Nodes** to access the Node page, which you use to manage nodes.

   Both Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6) are now supported by WebSphere Application Server, but there are restrictions that apply to using both IPv4 and IPv6 in the same cell. Note that when you add a node to a cell, the format in which you specify the host name is based on the version of IP the node will be using. For further information, read the topic on IP version considerations for cells.

Results

Depending on which steps you performed, you changed the cell protocol information, defined custom properties for the cell, and added additional nodes.

What to do next

You can continue to administer your Network Deployment product by doing such tasks as managing nodes, node agents, and node groups.

IP version considerations for cells

There are compatibility issues to consider when configuring the IP version for cells.

WebSphere Application Server has support for Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6). IPv6 support was introduced with i5/OS® release V5R4, and although WebSphere Application Server V6.0 only supports IPv4, when running on V5R4 and later releases of i5/OS, WebSphere Application Server V6.1 and later support IPv4 and IPv6.When defining a node, you must specify the host name as a string or as a 32-bit numerical address.

WebSphere Application Server support for Internet Protocol Version 4 and Internet Protocol Version 6

Internet Protocol Version 4 is no longer viable for many businesses. Because it is based on 32-bit architecture, there is a growing shortage of Internet Protocol Version 4 (IPv4) addresses. Internet Protocol Version 6 (IPv6) is based on 128-bit architecture, which allows a far greater number of addresses to be available for use over the Internet.

In response, WebSphere Application Server now includes support for IPv6, in addition to continued support for IPv4. This means that nodes running WebSphere Application Server Version 6 and later can use IPv6. However, note that nodes running WebSphere Application Server Version 5.x cannot use IPv6.

WebSphere Application Server supports a *dual mode* environment in which you can have older legacy applications running on IPv4 and IPv6-enabled applications running on IPv6. Note, however, that there are restrictions on using IPv4 and IPv6 in the same cell. This article documents those restrictions as well as outlines the ways in which you can set up your cells, depending on the version of IP that you will be using.

Note: IPv6 is not supported on native transports. If you need this function, you must configure a channel chain. Channel chains may not be configured automatically for a server migrated from WebSphere Application Server V5 to WebSphere Application Server V6. Channel chains are automatically configured when you create a V6 server.

When defining a node, you must specify the host name as a string or as a 32-bit numerical address.

Dual mode cell

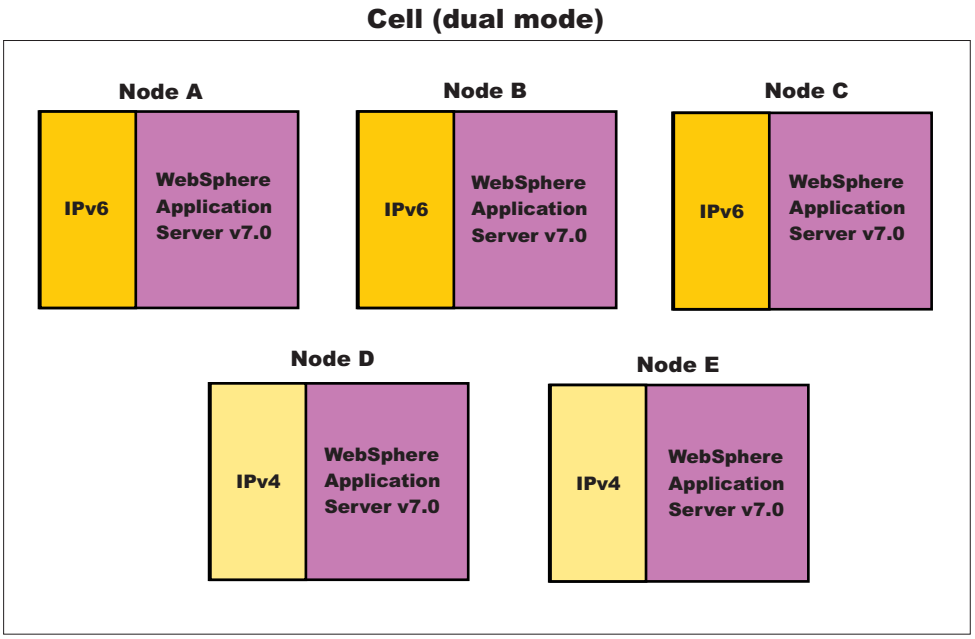
In a dual mode cell, mixed IPv4 and IPv6 communications are supported. By default, a cell is set to dual mode when it is created. Note, however, that only nodes running WebSphere Application Server Version 6 and later are valid in a dual mode cell.

The deployment manager can manage both IPv4 and IPv6 nodes. Application servers can listen to both IPv4 and IPv6 communications.

IPv4 and IPv6 nodes cannot communicate with each other, so the purpose of the dual mode cell is to enable this communication, thereby allowing you to use your existing applications, running over IPv4, with newer applications that have been enabled for IPv6.

The following illustration shows a dual mode cell:

Dual mode cell

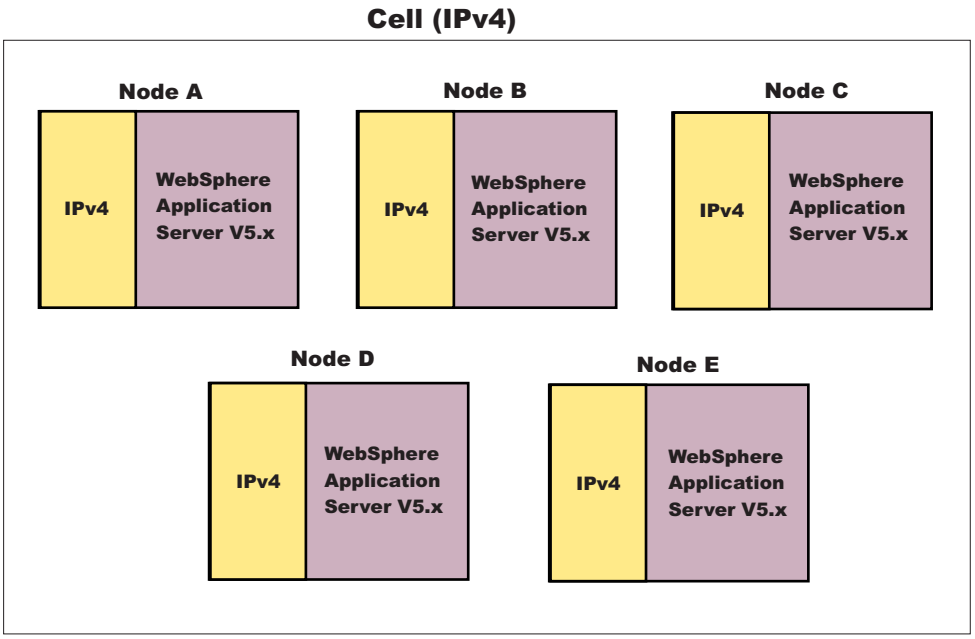


IPv4-only cell

In an IPv4-only cell, all nodes must:

- Use IPv4
- Run WebSphere Application Server Version 5.x
- Have host names defined as strings or 32-bit numerical addresses.

IPv4-only cell



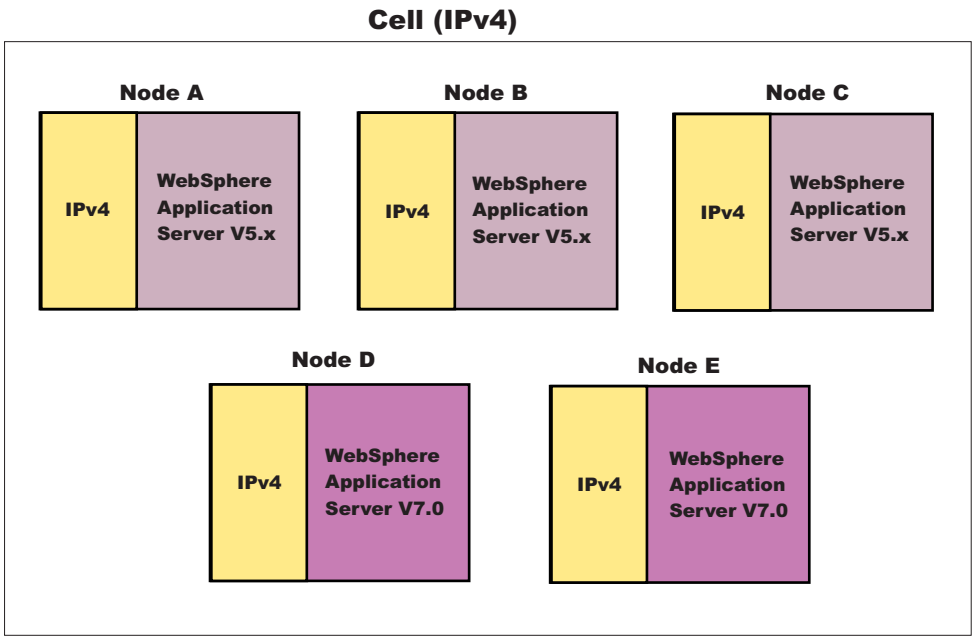
It is important to note that, by default, a cell is set to dual mode. However, in order to run in an IPv4-only environment, you will need to explicitly set the cell to IPv4. See the topic on Java virtual machine (JVM) settings for more information.

Note: If you want to run a combination of WebSphere Application Server Version 5.x and WebSphere Application Server Version 6.0 or later nodes over IPv4, see the section on setting up a *mixed node cell*, below.

Mixed node cell

A mixed node cell consists of some nodes running WebSphere Application Server Version 5.x and other nodes running WebSphere Application Server Version 6 or later. In a mixed node cell, all nodes must use IPv4. When defining a node that will be used in a mixed node cell, you must specify the host name as a string or as a 32-bit numerical address, regardless of whether the node is running WebSphere Application Server Version 5.x or WebSphere Application Server Version 6 and later, 128-bit numerical addresses cannot be specified.

Mixed node cell



In a mixed node cell, even though the WebSphere Application Server Version 6 and later nodes will be configured to use IPv4, the operating system running on them can still support both IPv4 and IPv6. This is true as long as the WebSphere Application Server Version 6 and later nodes are configured with string-based host names or 32-bit numerical addresses.

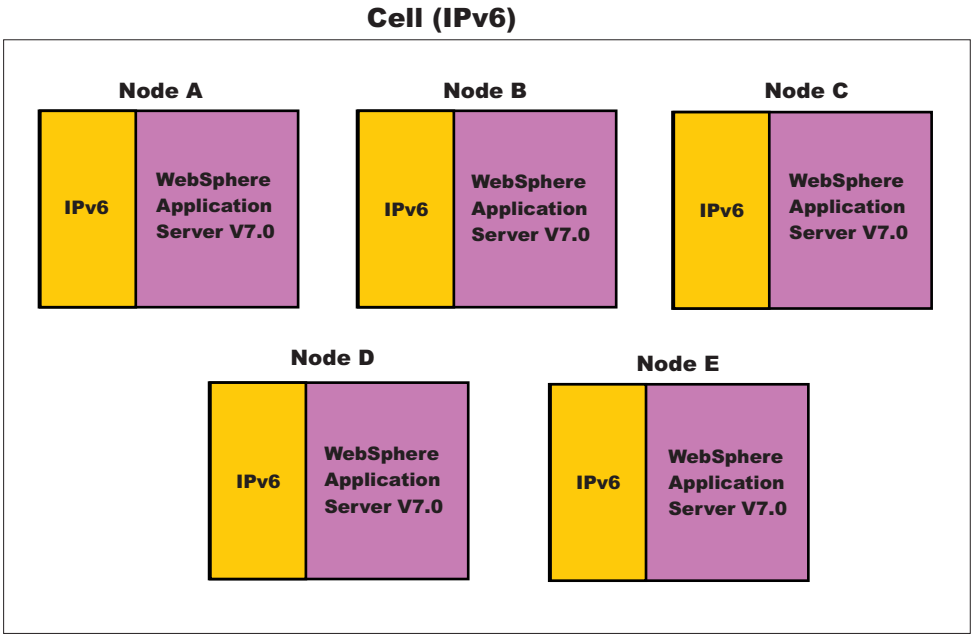
Note also, that you can only add Version 5.x nodes into a mixed node cell through migration. You first need to migrate from a Version 5.x Deployment Manager to a Version 6.0 or later Deployment Manager, and then either keep the Version 5.x nodes or migrate them to Version 6.0 or later nodes.

IPv6-only cell

In an IPv6-only cell, all nodes must:

- Use IPv6
- Run WebSphere Application Server Version 6 or later
- Have host names defined as strings or 128-bit numerical addresses.

IPv6-only cell



Specifying host names

During profile creation for WebSphere Application Server, you are asked to provide the host name or IP address of the machine on which the profile is being created in the *Host Name or IP address* field. The host name or IP address that you specify is used to advertise this profile to all other WebSphere Application Server profiles in the cell configurations. All nodes in the cell will use the host names or IP addresses that are defined in this way to reach each other. In general, it is best to always use a host name to identify a WebSphere Application Server profile. By using a host name, you will not have to be concerned about which IP address is being used (32-bit versus 128-bit), whether it runs on IPv4 or IPv6, and so on. As long as the DNS service is properly configured, the nodes should all be able to work together.

However, if you prefer, you can control which IP stack or address is used. To do this, enter the specific IP address (32-bit for IPv4 or 128-bit for IPv6) into the *Host Name or IP Address* field. This profile will then be identified with this IP address and other WebSphere Application Server nodes will use this IP address to communicate with this node.

When specifying IPv6 addresses, it is good practice to surround them with protective square brackets. For example, `[fe80::202:57ff:fec4:2334]`. The reason for this is that in system internal processing, IP addresses are often combined with port numbers in the form of `<IP address>:<port number>` , and the colons in IPv6 addresses could be confusing in such circumstances. **However, note that you cannot use IPv6 addresses that are surrounded by square brackets within the administrative console or the Profile Management Tool.**

Note that the use of IPv6 (Internet Protocol Version 6) and WS-AT (Web Services Atomic Transactions) are not supported on HTTP transports; they are only supported on HTTP transport channel chains.

Note that in scripting, the square brackets might have special meaning, depending on the language binding used (for example, Jacl). You can work around this problem by using a special escape character in front of the opening and closing brackets. Using the Jacl binding, for example, the same IPv6 address cited earlier can be entered as `\[fe80::202:57ff:fec4:2334\]`

Note: While you cannot use square brackets with IPv6 addresses within the administrative console, you must use square brackets to specify an IPv6 address as part of the administrative console's URL in a browser. This allows the browser to distinguish the IPv6 address from the port value.

Multicast configuration

WebSphere Application Server uses multicast broadcasting at the node level to allow a node agent to discover the managed processes in the node. IPv4 and IPv6 addresses are not compatible. Therefore, to allow a WebSphere Application Server node to run after initial installation, both IPv4 and IPv6 multicast addresses are initially defined in the node agent configuration, and when a node agent starts, both addresses are tried in sequence. Delete either the NODE_MULTICAST_DISCOVERY_ADDRESS address or the NODE_IPV6_MULTICAST_DISCOVERY_ADDRESS address after installation. By that time, you should know whether the node is running IPv4 or IPv6, so limiting multicast discovery to the known protocol, the node agent runs more efficiently.

Deleting the Internet Protocol Version 4 or the Internet Protocol Version 6 multicast port

This topic describes how to delete the Internet Protocol Version 4 (IPv4) or the Internet Protocol Version 6 (IPv6) for multicast ports so that the node agent runs more efficiently.

Before you begin

You must install the WebSphere Application Server Network Deployment product before you can delete a multicast port.

About this task

To allow node installation to run out-of-the box, both IPv4 and IPv6 are initially defined in the node agent configuration. To make the node agent run more efficiently, delete the multicast port that the node is not using.

To delete one of the multicast ports using the administrative console, perform these steps:

1. Click **System Administration > Node agents**.
2. Select the node agent.
3. On the next panel, under Additional Properties, select **Ports**.

   The next panel shows a list of existing ports.
4. Delete a multicast port.

   Select either NODE_MULTICAST_DISCOVERY_ADDRESS to delete IPv4 or NODE_IPV6_MULTICAST_DISCOVERY_ADDRESS to delete IPv6.
5. Click **Delete**.

Results

You deleted either IPv4 or IPv6.

What to do next

You can continue to administer your Network Deployment product by doing such tasks as managing nodes, node agents, and node groups.

Cell settings

Use this page to set the discovery protocol and address end point for an existing cell. A cell is a configuration concept, a way for an administrator to logically associate nodes according to whatever criteria make sense in the administrator's organizational environment.

To view this administrative console page, click **System Administration > Cell**.

Name

Specifies the name of the existing cell.

A cell name must be unique in any circumstance in which the product is running on the same physical machine or cluster of machines, such as a sysplex. Additionally, a cell name must be unique in any circumstance in which network connectivity between entities is required either between the cells or from a client that must communicate with each of the cells. Cell names must also be unique if their namespaces are federated. Otherwise, you might encounter symptoms such as a javax.naming.NameNotFoundException error, in which case, create uniquely named cells.

Cell Discovery Protocol

Specifies the protocol that the nodes use to contact and discover the deployment manager in the cell.

Select one of these protocol options:
UDP User Datagram Protocol (UDP)
TCP Transmission Control Protocol (TCP)

Default TCP

Configuring deployment managers

Configure deployment managers for a single, central point of administrative control for all elements in a WebSphere Application Server distributed cell.

Before you begin

If you plan to change the system clock, stop all the application servers, the node agent servers, the deployment manager server, the administrative agent server, and the job manager server first. After you stop the servers, change the system clock, and then restart the servers. If you change the system clock on one system, you must ensure the clocks on all systems that communicate with each other and have WebSphere Application Server installed are synchronized. Otherwise, you might experience errors, such as security tokens no longer being valid.

About this task

Deployment managers are administrative agents that provide a centralized management view for all nodes in a cell, as well as management of clusters and workload balancing of application servers across one or several nodes in some editions. Each cell contains one deployment manager.

A deployment manager hosts the administrative console.

When you create a deployment manager profile, a deployment manager is created. You can run the deployment manager with its default settings. However, you can change the deployment manager configuration settings, such as the ports that the process uses, custom services, logging and tracing settings, and so on. To view information about managing a deployment manager, use the settings page for a deployment manager.

1. Click **System Administration > Deployment manager** from the navigation tree of the administrative console to access the settings page for a deployment manager.

2. Configure the deployment manager by clicking a property, such as **Custom services**, and specifying settings.
3. Optionally register or unregister the deployment manager with the job manager.

   A job manager allows you to submit administrative jobs asynchronously for deployment managers and for application servers registered to administrative agents. Click **System Administration > Deployment manager**. Under Additional Properties, click **Job Managers > Register/unregister with job manager**.

Results

You configured a deployment manager with options that you selected.

What to do next

You can continue to administer your product by doing such tasks as configuring cells and managing nodes, node agents, and node groups.

Deployment manager settings

Use this page to stop the deployment manager, and to link to other pages that you can use to define additional properties for the deployment manager. A deployment manager provides a single, central point of administrative control for all of the elements in the WebSphere Application Server distributed cell.

To view this administrative console page, click **System administration > Deployment manager**.

Name

Specifies a logical name for the deployment manager. The name must be unique within the cell.

Data type String

Start components as needed

Select this property if you want the server components started as they are needed for applications that run on this server.

When this property is not selected, all of the server components are started during the startup process. Therefore, selecting this property usually results in improved startup time because fewer components are started during the startup process.

Note: If you are running other WebSphere products on top of the this product, make sure that those other products support this functionality before you select this property.

Process ID

Specifies a string that identifies the process.

Data type String
Default None

Cell name

Specifies the name of the cell for the deployment manager. The default is the name of the host computer on which the deployment manager is installed with `Cell##` appended, where *##* is a two-digit number.

Data type String
Default *host_name*Cell01

Node name
Specifies the name of the node for the deployment manager. The default is the name of the host computer on which the deployment manager is installed with `CellManager##` appended, where *##* is a two-digit number.

Data type	String
Default	*host_name*CellManager01

State
Indicates the state of the deployment manager. The state is *Started* when the deployment manager is running and *Stopped* when the deployment manager is not running.

Data type	String
Default	Started

Node

A *node* is a logical grouping of managed servers.

A node usually corresponds to a logical or physical computer system with a distinct IP host address. Nodes cannot span multiple computers.

By default, node names are based on the host name of the computer, for example MyHostNode01.

Nodes can be managed or unmanaged. An unmanaged node does not have a node agent or administrative agent to manage its servers, whereas a managed node does. Both application servers and supported Web servers can be on unmanaged or managed nodes.

A stand-alone application server is an unmanaged node. The application server node becomes a managed node when it is either federated into a cell or registered with an administrative agent.

When you create a managed node by federating the application server node into a deployment manager cell, a node agent is automatically created. The node agent process manages the application server configurations and servers on the node.

When you create a managed node by registering an application server node with an administrative agent, the application server must be an unfederated application server node. The administrative agent is a single interface that monitors and controls one or more application server nodes so that you can use the application servers only to run your applications. Using a single interface reduces the overhead of running administrative services in every application server.

A managed node in a cell can have WebSphere Application Servers, Java Message Service (JMS) servers (on Version 5 nodes only), Web servers, or generic servers. A managed node that is not in a cell, but is instead registered to an administrative agent, can have application servers, web servers, and generic servers on the node.

An unmanaged node can exist in a cell as long as the unmanaged node only has a supported Web server defined on it. Unsupported Web servers can be on unmanaged nodes only and cannot be in a cell.

You can use the command line only to create a managed node that is registered to an administrative agent.

You can create a managed node in a cell in one of the following ways:

- Administrative console

- Command line
- Administrative script
- Java program

Each of these methods for adding a node to a Network Deployment cell includes the option of specifying a target node group for the managed node to join. If you do not specify a node group, or you do not have the option of specifying a node group, the default node group of DefaultNodeGroup is the target node group.

Whether you specify an explicit node group for a cell or accept the default, the node group membership rules must be satisfied. If the node that you are adding does not satisfy the node group membership rules for the target node group, the add node operation fails with an error message.

Each managed node that is joined to a cell must be a member of a node group. However, a managed node that is registered to an administrative agent cannot be a member of a node group.

The concepts of managed and unmanaged nodes are not applied to the registration of nodes to the job manager.

Administrative functions for Web server nodes

WebSphere Application Server supports basic administrative functions for all supported Web servers. For example, the generation of a plug-in configuration can be performed for all Web servers. However, propagation of a plug-in configuration to remote Web servers is supported only for IBM HTTP Servers that are defined on an unmanaged node. If the Web server is defined on a managed node, propagation of the plug-in configuration is done for all the Web servers by using node synchronization. The Web server plug-in configuration file is created according to the Web server definition and is based on the list of applications that are deployed on the Web server. You can also map all supported Web servers as potential targets for the modules during application deployment.

WebSphere Application Server supports some additional administrative console tasks for IBM HTTP Servers on managed and unmanaged nodes. For instance, you can start IBM HTTP Servers, stop them, terminate them, display their log files, and edit their configuration files.

Managing nodes

This topic describes how to add a node, select the discovery protocol for a node, define a custom property for a node, stop servers on a node, and remove a node.

Before you begin

A node is a grouping of managed or unmanaged servers. You can add both managed and unmanaged nodes to the WebSphere Application Server topology. If you add a new node for an existing WebSphere Application Server to the Network Deployment cell, you add a managed node. If you create a new node in the topology for managing Web servers or servers other than WebSphere Application Servers, you add an unmanaged node.

To view information about nodes and managed nodes, use the Nodes page. To access the Nodes page, click **System Administration > Nodes** in the administrative console navigation tree.

About this task

You can manage nodes on an application server through the wsadmin scripting tool, through the Java application programming interfaces (APIs), or through the administrative console. Perform the following tasks to manage nodes on an application server through the administrative console.

- **Add a node.**

1. Go to the Nodes page and click **Add Node**. Choose whether you want to add a managed or unmanaged node, and click **Next**.
2. For a managed node, verify that an application server is running on the remote host for the node that you are adding. On the Add Node page, specify a host name, connector type, and port for the application server at the node you are adding.
3. For a managed node, perform one of the following sets of actions listed in the table:

If the deployment manager is on	And the node that you add to the cell is on	Complete the appropriate set of actions:
The distributed platform or the i5/OS platform	The distributed platform or the i5/OS platform	Optionally specify a node group and a core group. Click **OK**.
The distributed platform or the i5/OS platform	A z/OS® system	Specify a node group that contains nodes from the same sysplex as the node you are now adding. If no such node group exists, create a node group and then specify that node group. Optionally specify a core group. Click **OK**.
A z/OS system	The distributed platform or the i5/OS platform	Specify a node group that contains distributed nodes. If no such node group exists, create a node group and then specify that node group. Optionally specify a core group. Click **OK**.

   For the node group option to display, a group other than the default node group must first be created. Likewise, for the core group option to display, a group other than the default core group must first be created.
4. For managed nodes, another administrative console panel is displayed if the node to federate is on a Windows operating system. Specify on the panel whether you want to register the node agent to run as a Windows service. If security is enabled, you can optionally enter the local operating system user name and password under which you will run the service. If you do not specify a user name and password, the service runs under the local system identity. When you run remove the node, the node agent is de-registered as a Window service.
5. For an unmanaged node, on the **Nodes > New** page, specify a node name, a host name, and a platform for the new node. Click **OK**.

The node is added to the WebSphere Application Server environment and the name of the node is displayed in the collection on the Nodes page.

Both Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6) are now supported by WebSphere Application Server, but restrictions do apply when using both IPv4 and IPv6 in the same cell. When you add a node to a cell, the format in which you specify the name is based on the version of IP that the node is using. For details, see IP version considerations for cells. On completing this step, you will have added one or more nodes.

Note: When nodes are added while LDAP security is enabled, the following exception is generated in the deployment manager System.out log under certain circumstances. If this happens, restart the deployment manager to resolve the problem.

```
0000004d ORBRas E com.ibm.ws.security.orbssl.WSSSLClientSocketFactoryImpl
createSSLSocket ProcessDiscovery : 0 JSSL0080E: javax.net.ssl.SSLHandshakeException -
The client and server could not negotiate the desired level of security.
Reason?com.ibm.jsse2.util.h: No trusted certificate found
```

- **Select the discovery protocol.**

  If the discovery protocol that a node uses is not appropriate for the node, select the appropriate protocol. On the Nodes page, click the node to access the Settings for the node. Select a value for

Discovery protocol. User Datagram Protocol (UDP) is faster than Transmission Control Protocol (TCP). However, TCP is more reliable than UDP because UDP does not guarantee the delivery of datagrams to the destination. The default of TCP is the recommended value.

For a node agent or deployment manager, use **TCP** or **UDP**.

A managed process uses multicast as its discovery protocol. The discovery protocol is fixed for a managed process. The main benefit of using multicast on managed processes is efficiency for the node agent. Suppose you have forty servers in a node. A node agent that uses multicast sends one broadcast to all forty servers. If a node agent did not use multicast, it would send discovery queries to all managed processes one at a time, totaling forty sends. Additional benefits of using multicast are that you do not have to configure the discovery port for each server or prevent port conflicts because all servers in one node listen to one port instead of to one port for each server.

On the Windows platform, multicast requires a router. If you run WebSphere Application Server on the Windows platform, but the machine the Application Server is on is not connected to the network, the multicast address is not shared with the application servers.

- **Define a custom property for a node.**
  1. On the Nodes page, click the node for which you want to define a custom property.
  2. On the Settings for the node, click **Custom Properties**.
  3. On the Property collection page, click **New**.
  4. On the Settings page for a property instance, specify a name-value pair and a description for the property, and click **OK**.
- Synchronize the node configuration.

  If you add a managed node or change a managed node configuration, synchronize the node configuration. On the Node Agents page, ensure that the node agent for the node is running. Then, on the Nodes page, select the check box beside the node whose configuration files you want to synchronize and click **Synchronize** or **Full Resynchronize**.

  Clicking either option sends a request to the node agent for that node to perform a configuration synchronization immediately, instead of waiting for the periodic synchronization to occur. This action is important if automatic configuration synchronization is disabled, or if the synchronization interval is set to a long time, and a configuration change is made to the cell repository that needs to replicate to that node. Settings for automatic synchronization are on the File Synchronization Service page.

  Synchronize requests that a node synchronization operation be performed using the normal synchronization optimization algorithm. This operation is fast, but might not fix problems from manual file edits that occur on the node. It is still possible for the node and cell configuration to be out of synchronization after this operation is performed.

  Full Resynchronize clears all synchronization optimization settings and performs configuration synchronization anew, so there is no mismatch between node and cell configuration after this operation is performed. This operation can take longer than the **Synchronize** operation.

  Unmanaged nodes cannot be synchronized.
- **Stop servers on a node.**

  On the Nodes page, Select the check box beside the managed node whose servers that you want to stop running, and click **Stop**.
- **Remove a node.**

  On the Nodes page, Select the check box beside the node that you want to delete and click **Remove Node**. If you cannot remove the node by clicking **Remove Node**, remove the node from the configuration by clicking **Force Delete**.
- **View node capabilities.**

  Review the node capabilities, such as the product version through the administrative console. You can also query them through the Application Server application programming interface (API) or the wsadmin tool. For information on the wsadmin tool, see the *Using the administrative clients* PDF.

The product versions for WebSphere Application Server are as follows: The base edition of WebSphere Application Server is listed in the version column as `Base`. The express edition of WebSphere Application Server is listed in the version column as `Express`. The Network Deployment product is listed in the version column as `ND`.

Node collection

Use this page to manage nodes in the WebSphere Application Server environment. Nodes group managed servers. The table lists the managed and unmanaged nodes in this cell. The first node is the deployment manager. Add new nodes to the cell and to the list by clicking **Add Node**.

To view this administrative console page, click **System administration > Nodes**.

Name

Specifies a name for a node that is unique within the cell.

A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer.

Version

Specifies the product name and version number of the node.

The product version is the version of a WebSphere Application Server for managed nodes.

For unmanaged nodes on which you can define Web servers, the version displays as not applicable

The base edition of WebSphere Application Server is listed in the version column as `Base`. The express edition of WebSphere Application Server is listed in the version column as `Express`. The Network Deployment product is listed in the version column as `ND`.

The product in the version column indicates the product that you used to create the profile, not the type of profile that you installed. For example, if you use the Network Deployment product to install a profile type of application server, the version column indicates `ND`.

Discovery protocol

Specifies the protocol that servers use to discover the presence of other servers on this node.

The possible protocol options follow:
UDP User Datagram Protocol (UDP)
TCP Transmission Control Protocol (TCP)

Status

Indicates that the node is either synchronized, not synchronized, unknown, or not applicable.

	**Synchronized**	The configuration files on this node are synchronized with the deployment manager.
	**Not synchronized**	The configuration files on this node are not synchronized with the deployment manager and are out-of-date. Perform a synchronize operation to get the latest configuration changes on the node.
	**Unknown**	The state of the configuration file cannot be determined because the node agent cannot be reached for this node.
	Not applicable	The status column is not applicable for this node because the node is an unmanaged node.

Node settings

Use this page to view or change the configuration or topology settings for either a managed node instance or an unmanaged node instance.

A managed node is a node with an Application Server and a node agent that belongs to a cell. An unmanaged node is a node defined in the cell topology that does not have a node agent running to manage the process. Unmanaged nodes are typically used to manage Web servers.

To view this administrative console page, click **System administration > Nodes >** *node_name*.

Name:

Specifies a logical name for the node. The name must be unique within the cell.

A node name usually is identical to the host name for the computer. However, you choose the node name. You can make the node name some name other than the host name.

Data type	String

Host name:

Specifies the host name of the unmanaged node that is added to the configuration.

Date type	String
Default	None

Discovery Protocol:

Specifies the protocol that the node follows to retrieve information from a network. The Discovery protocol setting is only valid for managed nodes.

Select from one of these protocol options:
UDP User Datagram Protocol (UDP)
TCP Transmission Control Protocol (TCP)

Data type	String
Default	TCP
Range	Valid values are UDP or, TCP.

UDP is faster than TCP, but TCP is more reliable than UDP because UDP does not guarantee delivery of datagrams to the destination. Between these two protocols, the default of TCP is recommended.

File permissions: Specifies the most lenient file permissions for the application files that WebSphere Application Server extracts into the application destination location. A deployer can override the permissions by configuring the permissions at the application level. However, if the file permissions specified at the application level are more lenient than the ones specified at the node, the ones specified at the node are used. The File permissions setting is only valid for managed nodes.

Data type	String
Default	`755` , or `rwx-rx-rx`, for files that end in `.dll`, `.so`, `.a` and `.sl` if no value is set

Platform type:

Specifies the operating system on which the unmanaged node runs.

Valid options are:

Windows
AIX®
HP-UX
Solaris
Linux
OS/400®
z/OS

Add a Windows based node as a Windows service

Use this page to run the node agent as a Windows service.

To view this administrative console page, click **System Administration > Nodes > Add node > managed node > Add managed node** .

Run the node agent as a Windows service:

Specifies whether to run a node agent as a Windows service.

Default false (cleared)

User name:

Specifies the ID for running the service process for the node agent. The user name and password fields are only available if security is enabled. If you do not specify the user name, the node agent runs under the authority of the local system. User name requirements are the requirements that the Windows operating system imposes for a user ID.

Password:

Specifies the password for the user name that you supply. Password requirements are the requirements that the Windows operating system imposes for a password.

Confirm password:

Specifies the same password that you typed for Password so that you can verify the correct password.

Add managed nodes

A managed node is a node with an application server and a node agent that belongs to a deployment manager cell. Use this page to add an application server node to a deployment manager cell.

To view this deployment manager administrative console page, click **System Administration > Nodes > Add node > Next** .

Node connection

Specifies connection information for WebSphere Application Server.

- **Host**

  Specifies the host name or IP address of the node to add to the cell. A WebSphere Application Server instance must be running on this machine.

Data type String

Default None

- **JMX connector type**

  Specifies the Java Management Extensions (JMX) connectors that communicate with the WebSphere Application Server when you invoke a scripting process.

  Select from one of these JMX connector types:

Simple Object Access Protocol (SOAP)

Use when the Application Server connects to a SOAP server.
Remote Method Invocation (RMI)

Use when the Application Server connects to an RMI server.

- **JMX connector port**

  Specifies the port number of the JMX connector on the instance to add to the cell. The default SOAP connector port is 8880.

Date type Integer
Default 8880

- **Application server user name**

  Specifies the administration user name that connects to the remote Application Server whose node is being added to the cell. The Application Server user name and password are used to connect to the Application Server and start the add node process at the Application Server. The Application Server user name and password settings always display. You must specify values for them if security is enabled at the Application Server. Otherwise, leave them blank.
- **Application server password**

  Specifies the password for the Application Server user name that you supply.
- **Deployment manager user name**

  Specifies the deployment manager administration user name that the Application Server uses when connecting to the deployment manager to add its node to the cell. The deployment manager user name and password settings display only if security is enabled at the deployment manager. The deployment manager user name and password are required if their settings display.
- **Deployment manager password**

  Specifies the password for the deployment manager user name that you supply.

Options

Select from the following settings to further specify characteristics when adding a managed node to a cell.

- **Include Applications**

  Copies the applications installed on the remote instance into a cell. If the applications to copy have the same name as the applications that currently exist in the cell, the Application Server does not copy the applications.
- **Include buses**

  Specifies whether to move the bus configuration at the node to the deployment manager.
- **Starting port**

  Specifies the port numbers for the node agent process.

Use default Specifies whether to use the default node agent port numbers.

Specify	Allows you to specify the starting port number in the Port number field. WebSphere Application Server administration assigns the port numbers in order from the starting port number. For example, if you specify 9950, the administration program configures the node agent ports as 9950, 9951, 9952, and so on.

- **Core Group**

  Specifies the group to which you can add a cluster or node agent. By default, clusters or node agents are added to the DefaultCoreGroup group.

  Select from one of the core groups if a list is displayed. The list displays if a core group in addition to the default core group exists.
- **Node group**

  Specifies the group to which you can add the node. By default, nodes are added to the DefaultNodeGroup group.

  Select from one of the node groups if a list is displayed. The list displays if a node group in addition to the default node group exists.

Node installation properties

Use this page to view read-only installation properties for this node. These properties provide information about the capabilities of the node that are collected during product installation time, such as the operating system name, architecture and version, or WebSphere Application Server product levels that are installed on the node.

To view this administrative console page, click **System administration > Nodes > *node name* > Node installation properties**.

Information about a node, such as operating system platform and product features, is maintained in the configuration repository in the form of properties. As product features are installed on a node, new property settings are added.

WebSphere Application Server system management uses the managed object metadata properties as follows:

- To display the node version in the administrative console
- To ensure that new configuration types or attributes are not created or set on older release nodes
- To ensure that new resource types are not created on old release nodes
- To ensure that new applications are not installed on old release nodes because the old run time cannot support the new applications

For detailed information about the following properties, see the Application Server application programming interface (API).

com.ibm.websphere.baseProductShortName

The product short name for the WebSphere Application Server that is installed.

com.ibm.websphere.baseProductVersion

The version of WebSphere Application Server that is installed.

com.ibm.websphere.nodeOperatingSystem

The operating system platform on which the node runs.

Node group

A *node group* is a collection of managed nodes. Managed nodes are WebSphere Application Server nodes. A node group defines a boundary for server cluster formation.

- “Node groups”
- “Sysplex node groups”
- “Example: Using node groups” on page 96

Node groups

Nodes that you organize into a node group need to be similar in terms of installed software, available resources, and configuration to enable servers on those nodes to host the same applications as part of a server cluster. The deployment manager does no validation to guarantee that nodes in a given node group have anything in common.

Node groups are optional and are established at the discretion of the WebSphere Application Server administrator. However, a node must be a member of a node group. Initially, all Application Server nodes are members of the default DefaultNodeGroup node group.

A node can be a member of more than one node group.

Nodes on distributed platforms and the i5/OS platform cannot be members of a node group that contains a node on a z/OS platform. However, nodes on distributed platforms and nodes on the i5/OS platform can be members of the same node group.

To delete a node group, the node group must be empty. The default node group cannot be deleted.

Sysplex node groups

A sysplex node group is a node group unique to the z/OS operating system. The sysplex node group includes a sysplex name and a z/OS operating system location service configuration. A sysplex is a collection of z/OS systems that cooperate by using certain hardware and software products to process workloads.

You cannot explicitly create a sysplex node group. The z/OS operating system creates sysplex node groups in the following ways:

- When you configure a deployment manager server on the z/OS operating system, the default node group is a sysplex node group. The deployment manager is automatically a member of the sysplex node group. Application Server for z/OS nodes that you add to the network deployment cell are automatically members of this node group.
- You can add an Application Server for z/OS node to a network deployment cell whose deployment manager is on a distributed platform node. In this case, you must add the first Application Server for z/OS node for the network deployment cell to an empty node group. The system automatically configures the node group into a sysplex node group by using the sysplex name and the z/OS location service configuration that belongs to the Application Server for z/OS node.

You cannot remove a node from a sysplex node group. However, if a node is the only member of a sysplex node group, you can add that node to an empty node group. The empty node group is converted into a sysplex node group and the former sysplex node group of the node is converted into a regular node group.

You cannot delete a node group that is a sysplex node group.

Example: Using node groups

By organizing nodes that satisfy your application requirements into a node group, you establish an administrative policy that governs which nodes can be used together to form a cluster. The people who define the cell configuration and the people who create server clusters can operate with greater independence from one another, if they are different people.

Example 1

Assume the following information:

- A cell is comprised of nodes one to eight.
- Each node is a managed node, which means that each node is configured with an Application Server.
- Nodes six, seven, and eight are additionally configured as WebSphere Business Integration Server Foundation nodes.
- All nodes are either z/OS system nodes from the same sysplex, or some combination of distributed platform nodes and i5/OS platform nodes.
- By default, all the nodes are in the default DefaultNodeGroup node group.

Applications that exploit WebSphere Business Integration Server Foundation functions can run successfully only on nodes six, seven, and eight. Therefore, clusters that host these applications can be formed only on nodes six, seven, and eight. To define a clustering policy that guides users of your WebSphere cell into building clusters that can span only predetermined nodes, create an additional node group called WBINodeGroup, for example. Add to the node group nodes six, seven, and eight. If you create a cluster on a node from the WBINodeGroup node group, the system allows only nodes from the WBINodeGroup node group to be members of the cluster.

Example 2

Assume the following information:

- A cell is comprised of nodes one to six.
- Each node is a managed node, which means that each node is configured with an Application Server.
- Nodes one to four are some combination of distributed platform nodes and i5/OS platform nodes.
- Nodes five and six are nodes on the z/OS operating system and are in the PLEX1 sysplex.
- The deployment manager is on a distributed platform node.
- Nodes one to four are members of the DefaultNodeGroup node group by default.
- You created empty PLEX1NodeGroup node group to group the z/OS operating system nodes on the PLEX1 sysplex.
- You joined the nodes on the z/OS operating system to the PLEX1NodeGroup node group when you added them to the cell. Nodes on the z/OS operating system cannot be in the same node group with the distributed platform nodes.

Applications that exploit z/OS functions in the PLEX1 sysplex can run successfully on nodes five and six only. Therefore, clusters that host these applications can be formed only on nodes five and six. The required separation of distributed platform nodes and i5/OS platform nodes from z/OS system nodes establishes a natural clustering policy that guides users of your Application Server cell into building clusters that can span only predetermined nodes. If you create a cluster on a node from the PLEX1NodeGroup node group, the system allows only nodes from the PLEX1NodeGroup node group to be members of the cluster.

Example: Using node groups

Use node groups to define groups of nodes that are capable of hosting members of the same cluster. An application that is deployed to a cluster must be capable of running on any of the cluster members. The node that hosts each of the cluster members must be configured with software and settings that are necessary to support the application.

By organizing nodes that satisfy your application requirements into a node group, you establish an administrative policy that governs which nodes can be used together to form a cluster. The people who define the cell configuration and the people who create server clusters can operate with greater independence from one another, if they are different people.

Example 1

Assume the following information:

- A cell is comprised of nodes one to eight.
- Each node is a managed node, which means that each node is configured with an Application Server.
- Nodes six, seven, and eight are additionally configured as WebSphere Business Integration Server Foundation nodes.
- All nodes are either z/OS system nodes from the same sysplex, or some combination of distributed platform nodes and i5/OS platform nodes.
- By default, all the nodes are in the default DefaultNodeGroup node group.

Applications that exploit WebSphere Business Integration Server Foundation functions can run successfully only on nodes six, seven, and eight. Therefore, clusters that host these applications can be formed only on nodes six, seven, and eight. To define a clustering policy that guides users of your WebSphere cell into building clusters that can span only predetermined nodes, create an additional node group called WBINodeGroup, for example. Add to the node group nodes six, seven, and eight. If you create a cluster on a node from the WBINodeGroup node group, the system allows only nodes from the WBINodeGroup node group to be members of the cluster.

Example 2

Assume the following information:

- A cell is comprised of nodes one to six.
- Each node is a managed node, which means that each node is configured with an Application Server.
- Nodes one to four are some combination of distributed platform nodes and i5/OS platform nodes.
- Nodes five and six are nodes on the z/OS operating system and are in the PLEX1 sysplex.
- The deployment manager is on a distributed platform node.
- Nodes one to four are members of the DefaultNodeGroup node group by default.
- You created empty PLEX1NodeGroup node group to group the z/OS operating system nodes on the PLEX1 sysplex.
- You joined the nodes on the z/OS operating system to the PLEX1NodeGroup node group when you added them to the cell. Nodes on the z/OS operating system cannot be in the same node group with the distributed platform nodes.

Applications that exploit z/OS functions in the PLEX1 sysplex can run successfully on nodes five and six only. Therefore, clusters that host these applications can be formed only on nodes five and six. The required separation of distributed platform nodes and i5/OS platform nodes from z/OS system nodes establishes a natural clustering policy that guides users of your Application Server cell into building clusters that can span only predetermined nodes. If you create a cluster on a node from the PLEX1NodeGroup node group, the system allows only nodes from the PLEX1NodeGroup node group to be members of the cluster.

Managing node groups

This task discusses how to create and manage node groups.

Before you begin

Read about Nodes groups if you are unfamiliar with them.

About this task

Your WebSphere Application Server environment has a default node group. However, if you need additional node groups to manage your Application Server environment, you can create and configure additional node groups. You can delete a node group as long as it is not a default node group.

- View and configure node groups.
  1. Click **System Administration > Node groups** in the console navigation tree.
  2. To view additional information about a particular node group or to further configure a node group, click on the node group name under **Name**.
- Create a node group.
  1. Click **System Administration > Node groups** in the console navigation tree.
  2. Click **New**.
  3. Specify the node group name and description.

  The node group is added to the WebSphere Application Server environment . The name of the node group appears in the name column of the Node group page.

  You can now add nodes to the node group. See “Managing nodes” on page 87 and “Managing node group members” on page 99 for information on how to add the nodes.
- Delete a node group if the node group is not the default node group.
  1. If the node group contains members, delete the members:
     a. Click **System Administration > Node groups** in the console navigation tree.
     b. Under **Name**, click the node group whose members you want to delete.
     c. Click **Node group members**.
     d. Select all the node group members.
     e. Click **Remove**.
  2. Click **System Administration > Node groups**.
  3. Select an empty node group.
  4. Click delete.

Node group collection

Use this page to manage node groups. A node group is a collection of WebSphere Application Server nodes. A node group defines a boundary for server cluster formation.

Nodes that are organized into a node group should be enough alike in terms of installed software, available resources, and configuration to enable servers on those nodes to host the same applications as part of a server cluster. The deployment manager does no validation to guarantee that nodes in a given node group have anything in common.

Node groups are optional and are established at the discretion of the WebSphere administrator. However, a node must be a member of a node group. Initially, all Application Server nodes are members of the default node group. The default node group is DefaultNodeGroup.

A node can be a member of more than one node group.

Nodes on distributed platforms and the i5/OS platform cannot be members of a node group that contains a node on a z/OS platform. However, nodes on distributed platforms and nodes on the i5/OS platform can be members of the same node group.

To delete a node group, the node group must be empty. The default node group cannot be deleted.

To view this administrative console page, click **System Administration > Node groups**.

Name
Specifies a name for a node group that is unique within the cell.

Members
Specifies the number of members or nodes in the node group.

Description
Specifies a description that you define for the node group.

Node group settings
Use this page to view or change the configuration or topology settings for a node group instance.

To view this administrative console page, click **System Administration > Node groups >** *node group name*.

Name:

Specifies a logical name for the node group. The name must be unique within the cell. The name can start with a number.

Data type	String
Maximum length	64 characters

Members:

Specifies the number of nodes within the node group.

Data type	Integer

Description:

Specifies the description that you define for the node group. The description has no specific maximum length.

Managing node group members

Use this topic to manage the nodes in your node groups by viewing, adding or deleting the nodes in a node group.

Before you begin

Read about Nodes groups if you are unfamiliar with them.

About this task

Make the nodes that you organize into a node group enough alike in terms of installed software, available resources, and configuration to enable servers on those nodes to host the same applications as part of a server cluster.

Node group membership must adhere to the following rules:

- A node in a node group must be a managed node.
- A managed node must be a member of at least one node group. Initially, all WebSphere Application Server nodes are members of the default node group named DefaultNodeGroup.
- Nodes on distributed platforms and nodes on i5/OS platforms can be members of the same node group.
- Nodes on distributed platforms and i5/OS platforms cannot be members of a node group that contains a node on a z/OS platform.
- View node groups members.
  1. Click **System Administration > Node groups >** ***node group name*** **> Nodes > Node group members** in the console navigation tree.
  2. To view additional information about a particular node group member for this node group, click on the node group member name under **Name**.
- Add a node to a node group.
  1. Click **System Administration > Node groups >** ***node group name*** **> Nodes > Node group members** in the console navigation tree.
  2. Click **Add**.
  3. Select the node from a list. The node group member name is the node name.

  The node group member is added to the node group specified on the breadcrumb trail. The name of the node group member appears in the name column of the Node group member page. You can add additional nodes of similar characteristics to the node group by repeating the steps for adding a node to a node group.

  If the node you add does not satisfy the node group membership rules for the target node group, the add node operation fails with an error message.
- Remove a node from a node group.
  1. Click **System Administration > Node groups >** ***node group name*** **> Nodes > Node group members** in the console navigation tree.
  2. Select the box next to each node group member that you want to remove from the node group.
  3. Click **Remove**.

  Each node group member that you selected is removed from the node group specified on the breadcrumb trail.

Node group member collection

Use this page to manage node groups members. A node group member is a WebSphere Application Server node.

Click **Add** to add node members to the node group. Click **Remove** to remove node members from the node group.

To view this administrative console page, click **System Administration > Node groups >** *node group name* **> Node group members**.

Name

Specifies the name of a node group member.

Node group member settings

Use this page to view or change the configuration or topology settings for a node group member.

To view this administrative console page, click **System Administration > Node groups >** *node group name* **> Node group members >** *node group member name*.

Name:

Specifies a logical name for the node group member. A node group member is a node. The name must be unique within the cell.

A node group member name usually is identical to the host name for the computer.

Data type	String
Maximum length	64 characters

The name must contain alphanumeric or national language characters and can start with a number.

Managing node agents

Node agents are administrative agents that represent a node to your system and manage the servers on that node. Node agents monitor application servers on a host system and route administrative requests to servers.

Before you begin

Before you can manage a node agent, you must install the Network Deployment product.

If you plan to change the system clock, stop all the application servers, the node agent servers, the deployment manager server, the administrative agent server, and the job manager server first. After you stop the servers, change the system clock, and then restart the servers. If you change the system clock on one system, you must ensure the clocks on all systems that communicate with each other and have WebSphere Application Server installed are synchronized. Otherwise, you might experience errors, such as security tokens no longer being valid.

About this task

A node agent is a server that is created automatically when a node is added to a cell. A node agent runs on every host computer system that participates in the Network Deployment product. You can view information about a node agent, stop and start the processing of a node agent, stop and restart application servers on the node that is managed by the node agent, and so on.

A node agent is purely an administrative agent and is not involved in application serving functions. A node agent also hosts other important administrative functions, such as file transfer services, configuration synchronization, and performance monitoring.

You can manage nodes through the wsadmin scripting tool, through the Java application programming interfaces (APIs), or through the administrative console. Perform the following tasks to manage nodes on an application server through the administrative console.

- View information about a node agent. Click **System Administration > Node agents** in the console navigation tree. To view additional information about a particular node agent or to further configure a node agent, click the node agent name under **Name**.

  IP verions: Both Internet Protocol Version 4 (IPv4) and Internet Protocol Version 6 (IPv6) are now supported by WebSphere Application Server, but there are restrictions that apply to using both IPv4 and

IPv6 in the same cell. Note that when a node is added to a cell, the format in which the name is specified is based on the version of IP the node will be using.

- Stop and then restart the processing of a node agent. On the Node Agents page, select the check box beside the node agent that you want to restart; then click **Restart**. It is important to keep a node agent running because a node agent must be running for application servers on the node managed by the node agent to run.
- Stop and then restart all of the application servers on the node that is managed by the node agent. On the Node Agents page, select the check box beside the node agent that manages the node with servers that you want to restart, and click **Restart all servers on node**.

  Clicking **Restart all servers on node** also stops and then restarts the node agent. Servers that were stopped when you clicked **Restart all Servers on Node** remain stopped.

  Tip: The node agent for the node must be processing to restart application servers on the node.
- Stop the processing of a node agent. On the Node Agents page, select the check box beside the node agent that you want to stop processing; click **Stop**.

Results

Depending on the steps that you completed, you have viewed information about a node agent, stopped and started the processing of a node agent, and stopped and restarted application servers on the node that is managed by the node agent.

What to do next

You can administer other aspects of the Network Deployment environment, such as the deployment manager, nodes, and cells.

Node agent collection

Use this page to view information about node agents. Node agents are administrative agents that monitor application servers on a host system and route administrative requests to servers. A node agent is the running server that represents a node in a Network Deployment environment.

To view this administrative console page, click **System Administration > Node agents** .

You must initially start a node agent outside the administrative console. For information on how to initially start a node agent, see the addNode command and the startNode command.

Name

Specifies a logical name for the node agent server.

Node

Specifies a name for the node. The node name is unique within the cell.

A node name usually is identical to the host name for the computer. That is, a node usually corresponds to a physical computer system with a distinct IP host address.

However, the node name is a purely logical name for a group of servers. You can name the node anything you please. The node name does not have to be the host name.

Version

Specifies the product version of the node.

The product version is the version of a WebSphere Application Server node agent and Application Servers that run on the node.

Host Name
Specifies the IP address, the full domain name system (DNS) host name with a domain name suffix, or the short DNS host name for the node agent.

Status
Indicates whether the node agent server is started or stopped.

Note that

	**Started**	The node agent is running.
	**Stopped**	The node agent is not running.

Node agent server settings
Use this page to view information about and to configure a node agent. A node agent coordinates administrative requests and event notifications among servers on a machine. A node agent is the running server that represents a node in a Network Deployment environment.

To view this administrative console page, click **System Administration > Node Agents >** *node_agent_name*.

A node agent must be started on each node for the deployment manager node to collect and control servers that are configured on that node. If you use configuration synchronization support, a node agent coordinates with the deployment manager server to synchronize the configuration data of the node with the master copy that the deployment manager manages.

You must initially start a node agent outside the administrative console. For information on how to initially start a node agent, see the addNode command and the startNode command.

The Runtime tab displays only when a node agent runs.

Name:

Specifies a logical name for the node agent server.

Data type String

Node:

Specifies the name of the node for the node agent server.

Data type String

Start components as needed: Select this property if you want the server components started as they are needed for applications that run on this server.

When this property is not selected, all of the server components are started during the startup process. Therefore, selecting this property usually results in improved startup time because fewer components are started during the startup process.

Note: If you are running other WebSphere products on top of the this product, make sure that those other products support this functionality before you select this property.

Process ID:

Specifies a string identifying the process.

Data type String

Cell Name:

Specifies the name of the cell for the node agent server.

Data type String
Default *host_name*Network

Node Name:

Specifies the name of the node for the node agent server.

Data type String

State:

Indicates whether the node agent server is started or stopped.

Data type String
Default Started

Administrative agent

An administrative agent provides a single interface to administer multiple unfederated application server nodes in environments such as development, unit test or that portion of a server farm that resides on a single machine.

The administrative agent and application servers must be on the same machine, but you can connect to the machine from a browser or the wsadmin tool on another machine.

Multiple customers administer application servers in their development, test, and production environments by federating the application server nodes into a cell and administering the application servers from the deployment manager. However, if you have development and unit test environments, then you might prefer to run application servers whose nodes have not been federated. These application servers have some administrative disadvantages. The application servers lack a common administrative interface. Remote administration is limited to installing applications and changing application server configurations. As an alternative, you can register these application servers with an administrative agent to administer application servers from a single interface and to more fully administer application servers remotely.

You can register an application server node with the administrative agent or federate the node with a deployment manager, but not both.

An administrative agent can monitor and control multiple application servers on one or more nodes. Use the application servers only to run your applications. By using a single interface to administer your application servers, you reduce the overhead of running administrative services in every application server.

You can use the administrative agent to remotely install applications on application servers, change application server configurations, stop and restart application servers, and create additional application servers.

Administering nodes using the administrative agent

You can configure an administrative agent, view or change unfederated application server nodes registered to the administrative agent, and view or change job manager configurations for a registered node. An administrative agent provides a single interface to administer application servers in, development, unit test, or server farm environments, for example.

Before you begin

If you plan to change the system clock, stop all the application servers, the node agent servers, the deployment manager server, the administrative agent server, and the job manager server first. After you stop the servers, change the system clock, and then restart the servers. If you change the system clock on one system, you must ensure the clocks on all systems that communicate with each other and have WebSphere Application Server installed are synchronized. Otherwise, you might experience errors, such as security tokens no longer being valid.

Before you use the administrative agent, install the core product files, create an administrative agent profile, and start the administrative agent. Use the registerNode command to register at least one application server node with the administrative agent.

About this task

Note: The administrative agent provides a single interface to administer multiple unfederated application server nodes in, for example, development, unit test, or server farm environments. By using a single interface to administer your application servers, you reduce the overhead of running administrative services in every application server.

You can use the administrative console of the administrative agent to configure the administrative agent, view and change properties for nodes registered to the administrative agent, register and unregister application server nodes with job managers, and view and change job manager configurations for a registered node. A job manager allows you to asynchronously submit and administer jobs for a node registered to the administrative agent when the node is also registered to the job manager. Read the section on planning the installation for topologies that include administrative agents and job managers.

- View and change properties for the administrative agent.
  1. Click **System Administration > Administrative agent** from the navigation of the administrative agent administrative console.
     - Optionally click **Register with Job manager** to register an application server node with a job manager.
     - Optionally click **Unregister from a Job Manager** to unregister an application server node from a job manager.
     - Optionally view the administrative agent properties on the Configuration tab and the Runtime tab.
     - Optionally select **Start components as needed** on the Configuration tab. Click **Apply**, and then click **OK**.

       Selecting the setting allows administrative agent components to start dynamically as needed for applications.
- View and change properties for a node registered to the administrative agent.
  1. Click **System Administration > Administrative agent > Nodes**.

     You can view the nodes registered to the administrative agent.
  2. Click **System Administration > Administrative agent > Nodes >*node_name*** .
     - Optionally select Poll jobs from job manager to have the administrative agent retrieve jobs from the job manager for this node.
     - To change other properties for the node, click the links under Additional Properties.

- View and change job manager configurations for a registered node.
  1. Click **System Administration > Administrative agent > Nodes > *node_name* > Job managers** to view the job managers to which the node is registered.
  2. Click **System Administration > Administrative agent > Nodes > *node_name* > Job managers > *job_manager_UUID*** to change job manager-related properties for the registered node.
     – Optionally change the polling interval by entering an integer value.

       The administrative agent uses the polling interval to check for jobs from the job manager for this registered node.
     – Optionally change the Web address of the job manager that the administrative agent polls for this registered node.

Results

Depending on the tasks that you completed, you might have configured the administrative agent, registered and unregistered application server nodes with job managers, viewed or changed properties for a node registered to the administrative agent, or viewed and changed the job manager configuration for a registered node.

What to do next

You can continue to administer registered nodes from the administrative agent. You can further configure the administrative agent using the links on the configuration tab of the administrative agent panel. You can register more nodes with the administrative agent using the registerNode command. You can deregister nodes from the administrative agent using the deregisterNode command. You can register and unregister nodes with a job manager.

Administrative agent settings

This panel allows you to configure the administrative agent and view its properties.

To view this administrative console page, click **System Administration > Administrative agent**.

Name

Specifies the administrative agent server name. The name is read-only.

Node

Specifies a name for the administrative agent node. The node name is unique within the cell. The node name is read-only.

By default, a node name is the hostname appended with Node01. For example, a node on a computer with the host name of MyComputer is named MyComputerNode01 by default.

However, the node name is a purely logical name for a group of servers. The node name does not have to contain the host name.

Start components as needed

Select this property if you want the server components started as they are needed for applications that run on this server.

When this property is not selected, all of the server components are started during the startup process. Therefore, selecting this property usually results in improved startup time because fewer components are started during the startup process.

Note: If you are running other WebSphere products on top of the this product, make sure that those other products support this functionality before you select this property.

Process ID
Specifies the read-only process ID of the administrative agent.

Cell name
Specifies the read-only cell name of the administrative agent.

Node name
Specifies the read-only node name of the administrative agent.

State
Specifies the read-only state of the administrative agent, such as started or stopped.

Nodes collection for the administrative agent
This panel allows you to view the application server nodes that are registered to the administrative agent. The administrative agent provides a single interface to the registered nodes.

To view this administrative console page, click **System Administration > Administrative agent > Nodes**.

Application server nodes already registered to an administrative agent can also be registered to job managers. Job managers allow you to asynchronously submit and administer jobs for large numbers of unfederated application servers and deployment managers over a geographically dispersed area.

Register with Job manager
Allows you to register a node with the job manager

Unregister from a Job Manager
Allows you to unregister a node from the job manager

Name
Specifies a name for an application server node that is registered to the administrative agent. The name is read-only.

Registered nodes settings
This panel allows you to view properties for a node registered to the administrative agent. The properties are name, unique ID, and poll jobs from job manager.

To view this administrative console page, click **System Administration > Administrative agent > Nodes > *node_name***.

Name: Specifies the name of an application server registered to the administrative agent. The name is read-only.

Poll jobs from job manager: Select the option so that the administrative agent retrieves jobs from the job manager for this node. The jobs start when the administrative agent retrieves them. The polling interval is defined on the Job manager panel of the administrative console for the administrative agent and controls how often the administrative agent checks for new jobs. The default is to retrieve the jobs.

Register or unregister with job manager
This panel allows you to either register a node to a job manager or unregister a node from a job manager. The node can be a deployment manager or a node registered to an administrative agent.

You can use the administrative agent administrative console to register or unregister a node with the job manager. The node that you register with the job manager is already registered to the administrative agent. For example, click **System administration > Administrative agent > Nodes**. Select **Register with Job Manager** to register a node to a job manager or **Unregister from a Job Manager** to unregister a node from a job manager.

You can use the deployment manager administrative console to register or unregister a deployment manager with the job manager. Click **System administration > Deployment manager**. Under Additional properties, click **Job Managers > Register/unregister with job manager**.

Managed node name
A required setting that specifies the name of the managed node. For a deployment manager, the managed node name is the name of the deployment manager node, usually *host_name*CellManager01.

Alias
An optional setting that specifies the alias of the managed node to enroll. Specify an alias if the managed node name is in use by another node.

Host name
An optional setting that specifies the host name to use to identify the job manager. The default value is `localhost`.

Port
An optional setting that specifies the job manager administrative console port number. If security is enabled, use the secure port number. If security is disabled, use the unsecure port number. The default secure port number is 9943, and the default unsecure port number is 9960. If no port number is specified, 9943 is used.

User name
Specifies the user name to log into the job manager when security is enabled.

Password
Specifies the password to log into the job manager. This setting is required when the user name setting is required.

Confirm password
Specifies the password a second time. This setting is required when a user name and a password are required.

Job managers collection
This panel allows you to view the job mangers to which this node is registered. The job managers allow you to asynchronously submit and administer jobs, such as manage applications, for this node.

To view this administrative console page, click **System Administration > Administrative agent > Nodes > *node_name* > Job managers**.

UUID
Specifies the Universal Unique Identifier (UUID), which uniquely identifies the job manager to which the administrative agent connects.

URL
Specifies the Web address of the job manger to which the node is registered.

Job manager settings
This panel allows you to view the Universal Unique Identifier (UUID) of the job manager, specify the polling interval to check for jobs on the job manager, and specify the Web address of the job manager.

To view this administrative console page, click **System Administration > Administrative agent > Nodes > *node_name* > Job managers >*job_manager_UUID*** .

Job Manager UUID: Specifies the Universal Unique Identifier (UUID), which uniquely identifies the job manager to which the administrative agent connects. The UUID is read-only.

Polling interval:

Specifies in seconds the time that elapses during a polling interval.

The administrative agent uses the polling interval to determine how often to poll a job manager for new jobs for the registered node. The default value is 30 seconds.

URL:

Specifies the Web address that the administrative agent uses to connect to the job manager.

The formats are `http://`*`host`*`:`*`port`*`/otis/OMADMServlet` or `https://`*`host`*`:`*`port`*`/otis/OMADMServlet`, where *host* is the host name of the job manager, and *port* is the port of the job manager.

Job manager

In a flexible management environment, a job manager allows you to submit administrative jobs asynchronously for application servers registered to administrative agents and for deployment managers. You can submit these jobs to a large number of servers over a geographically dispersed area.

You can make both application server nodes that are registered to administrative agents and deployment managers known to the job manager through a registration process. After you register application server nodes and deployment managers with the job manager, you can queue administrative jobs directed at the application server nodes or deployment managers through the job manager.

To register application server nodes and deployment managers with the job manager, use the wsadmin registerWithJobManager command. The command is in the ManagedNodeAgent command group.

Use the job manager to asynchronously administer job submissions. You can complete the following tasks:

- Set the job submission to take effect at a specified time.
- Set the job submission to expire at a specified time.
- Specify that the job submission occur at a specified time interval.
- Notify the administrator through e-mail that the job has completed.

Nodes in terms of the job manager are the application server nodes and deployment managers registered to the job manager. Groups of nodes are those groups that you create so that you can make job submission easier. You can submit a job for a group of nodes instead of entering multiple node names for a job.

Many of the management tasks that you can perform with the job manager are tasks that you can already perform with the product, such as application management, server management, and node management. However, with the job manager, you can aggregate tasks and perform those tasks across multiple application servers or deployment managers.

The following hypothetical company environments are examples of situations where a job manager is useful:

Branch office environment

A business has a thousand stores geographically dispersed across the continent. Each store contains either a few application servers, or a small Network Deployment cell consisting of two or three machines. Each store is managed locally for daily operations. However, each store is also connected to the data center at the company headquarters, potentially thousands of miles away. Some connections to the headquarters are at modem speeds. The headquarters uses the job manager to periodically submit administrative jobs for the stores.

Environment consisting of hundreds of application servers

An administrator sets up hundreds of low-cost machines running identical clones of an application server. Each application server node, which is registered with an administrative agent, is registered with the job manager. The administrator uses the job manager to aggregate administration commands across all the application servers, for example, to create a new server, or to install or update an application.

Environment consisting of dozens of deployment manager cells

An administrator sets up hundreds of application servers, which are divided into thirty different groups. Each group is configured within a cell. The cells are geographically distributed over five regions, consisting of three to seven cells per region. Each cell is used to support one to fifteen member institutions, with a total of 230 institutions supported. Each cell contains approximately thirty applications, each running on a cluster of two for failover purposes, resulting in a total of 1800 application servers. The administrator uses the job manager to aggregate administration commands across all the cells, for example, to start and stop servers, or to install or update an application.

Administering nodes using the job manager

In a flexible management environment, you can asynchronously submit and administer jobs for large numbers of unfederated application servers and deployment managers over a geographically dispersed area. At the remote machines, you can use jobs to manage applications, modify the product configuration, or do general purpose tasks such as run a script.

Before you begin

If you plan to change the system clock, stop all the application servers, the node agent servers, the deployment manager server, the administrative agent server, and the job manager server first. After you stop the servers, change the system clock, and then restart the servers. If you change the system clock on one system, you must ensure the clocks on all systems that communicate with each other and have WebSphere Application Server installed are synchronized. Otherwise, you might experience errors, such as security tokens no longer being valid.

Before you use the job manager, install the core product files, create a job manager management profile, and start the job manager. To administer unfederated nodes using the job manager, register the nodes with the administrative agent before you register the nodes with the job manager. To register unfederated nodes and deployment managers with the job manager, use the wsadmin registerWithJobManager command. The command is in the ManagedNodeAgent command group.

About this task

Note: In a flexible management environment, the job manager allows you to asynchronously submit and administer jobs for large numbers of unfederated application servers and deployment managers over a geographically dispersed area. Many of the management tasks that you can perform with the job manager are tasks that you can already perform with the product, such as application management, server management, and node management. However, with the job manager, you can aggregate the tasks and perform the tasks across multiple application servers or deployment managers.

You can submit jobs for groups of nodes that you define or for individual nodes. After you submit a job, you can check the job status, check the status of nodes, and check the status of node resources. You can view node resources for nodes and groups of nodes that you administer. You can configure the job manager and view its properties. Read the section on planning the installation for topologies that include administrative agents and job managers.

- Submit a job.

You can submit jobs to remote nodes to manage applications, modify the product configuration on remote machines, or do general purpose tasks such as run a script. You can specify when the jobs start, whether they are recurring, and when they are no longer available for submission.

- Check the status of a job.

  You can check the status of jobs, the status of jobs at their nodes, and the job history of nodes. You can suspend, resume, or delete jobs on the Job status collection panel.
- Administer nodes of the job manager.

  You can view nodes with their version numbers based on the results of the Find option and view node resources for nodes that you select. You can also view the properties and property values for a particular node.
- Administer node resources of the job manager.

  You can view server, application, node, and cluster resources that are associated with nodes and groups of nodes registered to the job manager. You can also view the status of specific resources at each node and view properties for a particular node resource as a name-value pair.
- Administer groups of nodes for the job manager.

  You can create, modify, delete, and view groups of nodes. Groups of nodes make job submission simpler because you can submit a job for a group of nodes instead of entering multiple node names for a job submission.
- Configure job managers.

  You can specify settings such as the default job expiration, the job manager Web address, and the mail provider Java Naming and Directory Interface (JNDI) name for the job manager. You can view job manager properties such as the process ID and the state of the job manager.

Results

Depending on the tasks that you completed, you might have submitted jobs, checked the status of jobs, viewed nodes and node resources, or administered groups of nodes.

What to do next

You can continue to administer jobs as described in the procedure. You can register nodes using the wsadmin registerWithJobManager command, or unregister nodes using the wsadmin unregisterWithJobManager command. Both commands are in the ManagedNodeAgent command group. You can stop and restart the job manager.

Submitting a job

In a flexible management environment, you can submit jobs to remote nodes to manage applications, modify the product configuration on remote machines, or do a general purpose task such as run a script. You can specify when the jobs start, whether they are recurring, and when they expire.

Before you begin

Before you can submit a job, you must have registered at least one node with the job manager. A node can be an application server node that was first registered with an administrative agent or a deployment manger node.

Your ID must be authorized for the administrator role or the operator role to submit jobs.

You can simplify administration of a large number of nodes by submitting jobs against groups of nodes. Each group of nodes represents a group of nodes. Before you can submit a job for a group of nodes, you must have created the group of nodes.

About this task

You can use the administrative console of the job manager to submit jobs to do tasks such as manage applications, modify the product configuration on remote workstations, or do general purpose tasks such as run a script. To complete the job submission, choose the type of job, choose the nodes on which you want the job to run, specify the job parameters that are specific to the job type, schedule the job, review the summary, and submit the job.

1. Click **Jobs > Submit** from the navigation tree of the job manager administrative console.
2. Choose the job type.
   a. Select the job type from the list.

      The list of job types varies based on the nodes that you have registered with the job manager. The values displayed in the list are retrieved from the getJobTypes and getJobTypeMetadata commands of the AdminTask object. You can have job types that manage applications, modify the product configuration on remote machines, or do general purpose tasks such as run a script.
   b. Optionally specify a description of the job.

      The description is a string that can be up to 256 characters. The default description is the job type. You can change or add to the default description. The description is useful when using the Find option to view existing jobs.
   c. Click **Next**.
3. Choose the job targets.

   You are determining the nodes on which you want the job to run.
   a. Select a group of nodes from the list, or select **Node name**.

      Only groups of nodes that are valid for the job type that you selected are displayed in the list of groups of nodes.
   b. If you selected **Node name**, then enter a node name, and click **Add**, or compile a list of nodes by using the **Find** option.

      Node name that you enter

      If you enter a node name, it must be a node that has been registered to the job manager. The node name is validated when you click **Next**.

      List of node names

      1) Click **Find**.

         The Find nodes panel is displayed.
      2) If you want to run the Find operation on specific keywords, specify a valid operator and a text string.

         The list of keywords is dynamic. Valid operators are `=` (equal to), `!=` (not equal to), `is null`, and `is not null`. The text string can be complete or partial and can contain an asterisk (*) to include variable or unknown characters.
      3) Click **Find**.

         The results are displayed in the Available nodes list and are selected.
      4) Move nodes between the Available nodes list and the Chosen nodes list.
         - To move specific nodes from the Available nodes list to the Chosen nodes list, select nodes in the Available nodes list and click **>**.
         - To move specific nodes from the Chosen nodes list to the Available nodes list, select nodes in the Chosen nodes list and click **<**.
      5) After you have a list of the desired nodes in the Chosen nodes list, click **OK**.

         The nodes display on the Choose job targets panel.
   c. Click **Next**.

4. Specify the job parameters.

   The list of job parameters is dynamic and based on the job type. For example, if the job type is to install an application, you would specify the application name, the location of the application to install, and optionally the name of the server where the system installs the application. The following table describes the types of parameters.

Parameter Type	Description
String	You can enter text for the appropriate parameters. The text is not validated until the job is submitted.
Node resource	You can select a node resource. The Find option is available for you to search for the resource, depending on the job type that you selected in the first step.

   a. Optionally click **Find** if it is available.

      The Find node resources panel is displayed.

   b. If you want to run the Find operation on specific keywords, specify a valid operator and a text string.

      The list of keywords is dynamic. Valid operators are `=` (equal to), `!=` (not equal to), `is null`, and `is not null`. The text string can be complete or partial and can contain an asterisk (*) to include variable or unknown characters.

   c. Click **Find**.

      The results are displayed in the Available resources common to all selected endpoints list.

   d. Click **OK** to save the results and return to the panel on specifying job parameters.

   e. Click **Next**.

5. Schedule the job.

   The times and dates that you specify are relative to the job manager.

   a. Optionally specify one or more e-mail addresses where notifications are sent when the job is done.

      If you specify multiple e-mail addresses, separate them by commas. The e-mail addresses are saved in your console preferences. Each e-mail address is validated for format errors.

   b. Select when the job is available for submission.

      You can submit the job to be available now, or specify a time and date that the job is retrieved from the job manager.

   c. Select the job expiration.

      The job expiration is the time at which the job will no longer be available for nodes to run. You can use the default expiration, specify a time and date for the job expiration, or specify an amount of time in which the job expires. The default expiration is defined on the Job manager configuration panel.

   d. Optionally specify a recurring interval for the job, a start date and time for the interval, and an end date and time for the interval.

   e. Click **Next**.

6. Review the summary, and submit the job.

   a. If you want to make changes to the options, click **Previous** until you reach the panel that you want to change. Make the correction on that panel, and then proceed through the panels until you reach the Summary and submit panel.

   b. When you are satisfied with the options, click **Finish** to submit the job.

      The Job status collection panel is displayed where only the status for the job that you submitted is displayed.

Results

Depending on the type of job that you selected, you have submitted a job to manage applications, modify the product configuration on remote machines, or do a general purpose task such as run a script.

What to do next

You can check the job status or submit other jobs.

Find node resources

This panel allows you to find node resources for resource job parameters when you submit a job through the job submission wizard. This panel is used for job types such as start server, stop server, and start cluster.

To access this panel, first click **Jobs > Submit**. The job type that you select in step 1 of the Job submission wizard affects whether you can access the Find node resources panel in step 3. For example, if you select the job type, start server, in step 1, then in step 3 you can select **Find** to access the Find node resources panel.

Find:

You can use the Find option to determine the node resources to display. The Find results are displayed in **Available resources common to all selected nodes**, after you click **Find**. Click **Reset** to clear the operators in column 2 and text in column 3.

Column 1	Column 2	Column 3
Except for the maximum results parameter, this column lists parameters that define node resources. The node name, job type, and unique identifier parameters are always available.	• All parameters except type and maximum results: Valid operators for the find operation are `=` (equal to), `!=` (not equal to), `is null`, and `is not null`. • Type: They type is preselected to the type of resource option that you need for job submission. You cannot change the type. The list contains one or more values of `Server`, `Application`, or `Cluster`.	• All parameters except type and maximum results: Specifies the string or partial string of a parameter. A partial string is designated using an asterisk (*). For example, setting the node resource parameter to `Server*` finds all server type resources that start with `Server`. You can search for an exact match for multiple items by including comma-separated items. For example, you can search on two resource names by entering `server1, server2`. When you search for more than one item, you cannot use the asterisk. • Node names are included based on your step 2 choices. • Maximum results: Specifies the number of records that the find operation displays. Enter a value between one and the maximum number of records that can be retrieved as defined in the job manager configuration.

Example: If the resources are server1, server2, and server3, you can enter `server` or `server*` for column 3 and `=` for column 2 of the resource name parameter.

Available resources common to all selected nodes: Specifies the results of the find operation. If no choices are listed or you do not see the resource you want, you can search again. If your job is targeted to multiple nodes, only those resources that are present on all nodes will be listed. If at least one resource is listed, you can select one resource and click `OK` to return it to the parameter on the previous console panel.

Checking job status

In a flexible management environment, you can check the overall status of jobs, the status of specific job nodes, and the job history of nodes. You can suspend, resume, or delete jobs on the Job Status collection panel.

Before you begin

Before you can check the job status, you must have registered at least one node with the job manager and submitted a job for the node.

To suspend a started job, resume a suspended job, or delete selected jobs, your ID must be authorized for the operator role.

About this task

After you finish submitting a job through the Job Submission wizard, the Job status collection panel is displayed. When this panel is displayed, the panel contains only the job that you submitted along with its status information. You can check the status of the job, the status of job at its nodes, and the job history of the nodes.

If you access the Job status collection panel by selecting **Jobs > Status** in the job manager administrative console navigation, you can use the Find option to limit the number of jobs that are displayed based on the criteria you specify. The first time you access the Job status collection, no jobs are listed. You must enter parameters for the Find option to obtain a list of jobs based on the parameter information that you provide. The next time you select **Jobs > Status**, a list of jobs is displayed based on the parameters you last specified on the Find option for this job manager administrative console panel. You can then optionally modify the Find option criteria to display a different set of jobs. After at least one job displays, you can check the status of the displayed jobs, the status of specific job nodes, and the job history for nodes of a particular job.

- Optionally use the Find option to display a set of jobs.

  If no jobs are displayed, you must use the Find option to display jobs based on the parameter information that you enter.

  1. Click **Jobs > Status** in the job manager administrative console navigation.
  2. For the parameters on which you want to do a Find operation, specify an operator and a text string.
  3. Click **Find**.

  The list of jobs along with their status information are in the collection table.
- Check the status of a job at its nodes.
  1. Select **Jobs > Status** in the console navigation to access the Job status collection panel if you did not get to the panel as a result of a job submission.
  2. Select either a job from the Job ID column or a number on the graph in the Status Summary column for a particular job. The graph is divided in up to four sections, indicating success, partial success, failure, or other, in that order, of the nodes in the job.
  3. Optionally use the Find option to display the status of specific job nodes based on the parameter information that you enter.
     a. If you want to run the Find operation on specific parameters, specify an operator and a text string as appropriate.

b. Click **Find**.

A list of nodes for the job, along with the status for each node, are displayed on the Job status detail panel.

- Check the job history of nodes.
  1. Select **Jobs > Status** in the job manager administrative console navigation to access the Job status collection panel if you did not get to the panel as a result of a job submission.
  2. Select either a job from the Job ID column or a number on the graph in the Status Summary column for a particular job. The graph is divided in up to four sections, indicating success, partial success, failure, or other, in that order, of the nodes in the job.
  3. On the Job status detail panel, click a link in the Status column for one of the nodes names.
  4. Optionally use the Find option to display job history based on the parameter information that you enter.
     a. If you want to run the Find operation on specific parameters, specify an operator and a text string as appropriate.
     b. Click **Find**.

  The status of the job for the node is displayed on the Job status history panel.
- Suspend a job.
  1. Select **Jobs > Status** in the job manager administrative console navigation to access the Job Status Collection if you did not get to the panel as a result of a job submission.
  2. Select the check box next to a job with an active or pending state.
  3. Click **Suspend**.
- Resume a job.
  1. Select **Jobs > Status** in the job manager administrative console navigation to access the Job status collection panel if you did not get to the panel as a result of a job submission.
  2. Select the check box next to a job whose state is Suspended.
  3. Click **Resume**.
- Delete a job.
  1. Select **Jobs > Status** in the job manager administrative console navigation to access the Job status collection panel if you did not get to the panel as a result of a job submission.
  2. Select the check box next to the job that you want to delete.
  3. Click **Delete**.

Results

You might have run a Find operation to display job status based on criteria that you specify, checked the status of jobs at their nodes, checked the jobs history of nodes, suspended a job, resumed a job, or deleted a job.

What to do next

You can continue to check job status and do other job management tasks such as submit other jobs, create node groups for job submission, view node resources, or view nodes.

Job status collection

This panel displays information about submitted jobs, including the job ID, description, state, activation time, expiration time, and status summary. Jobs are submitted to administer nodes that have been registered with the job manager.

Deployment manager nodes and unfederated application server nodes that have been registered with an administrative agent are eligible to be registered with the job manager. The job manager can administer the entire deployment manager cell, not just the deployment manager.

This console panel is navigated to in two ways.

- When the job submission wizard completes, this console panel is displayed only with the job that you just submitted.
- Click **Jobs > Status** to view the status of any your submitted jobs.

To suspend a started job, resume a suspended job, or delete selected jobs, your ID must be authorized for the operator role.

Use one of the following buttons to suspend, resume, or delete a job.

Button	Description
Suspend	Specifies that an administrative agent or deployment manager can no longer retrieve the job.
Resume	Specifies that an administrative agent or a deployment manager can retrieve the job.
Delete	Specifies that a job and all its history are removed and no longer available. When you click **Delete**, you are given an opportunity to confirm or cancel the delete operation.

Find:

When you click **Jobs > Status**, you can use the Find option to limit the submitted jobs to display. The Find results are displayed in the table that follows the Find and Preference options after you click **Find**. Clicking **Reset** clears the operators in column 2 and text in column 3.

Column 1	Column 2	Column 3
Except for the maximum results parameter, this column lists parameters that define a job.	• Job ID, state, node name, group of nodes, and description parameters: Valid operators for the find operation are `=` (equal to), `!=` (not equal to), `is null`, and `is not null`. • Activation time and the expiration time parameters: Valid operators are >=, and <=.	• All parameters except maximum results: Specifies the string or partial string of a parameter. A partial string is designated using an asterisk (*). For example, setting the description to `inventory*` finds all jobs with a description that starts with `inventory`. You cannot use the asterisk in the job ID field or the fields related to time. You can search on an exact match for multiple items by including the items to match separated by commas. For example, you can search on two job IDs by entering `119489625729609463`, `119489651801509472`. When you search on more than one item, you cannot use the asterisk. • Maximum results: Specifies the number of records that the find operation displays. Enter a value between one and the maximum number of records that can be retrieved as defined in the job manager configuration.

Example: If the nodes are EastCoast1, EastCoast2, WestCoast1, and WestCoast2, you can enter `East` or `East*` for column 3 and `=` for column 2 of the node parameter. Job status is displayed for the jobs that include the EastCoast1 and EastCoast2 nodes.

Job ID: Specifies the job number of the job that you submitted.

Description: Specifies the description that you entered when you submitted the job.

State:

Specifies whether the state of the job is Active, Pending, Expired, or Suspended.

State	Description
Active	The job is activated and not expired. The job does not have to be running to be active.
Pending	The job has been submitted, but has not been activated.
Expired	The expiration time that you specified during job submission was reached before the job was started. A job does not start on a node after the expiration time is reached. If the expiration time is reached while the job is running on any of its nodes, the job continues on those nodes.
Suspended	The job has been suspended. If the suspension occurs while the job is running on a node, the job continues on that node. If the suspended job has not started when the suspension occurs, the job will not start on any of its nodes unless someone resumes the job. You can suspend a job from this console panel by clicking **Suspend**. You can resume a job from this console panel by clicking **Resume**.

Activation time:

Specifies the activation time that you entered when you submitted the job or the actual time when the job was submitted if you did not specify an activation time.

The activation time is the time the job is available to run on the targeted nodes or groups of nodes.

Expiration time:

Specifies the time the job is to expire. If the job has not started by the expiration time, the job will not start and the state will change to `Expired`. If the expiration time occurs while the job is running on a node, the job continues on that node.

When you submit the job, you can set the expiration date and time, choose the default expiration time option, or specify the amount of time until the job expires.

Status summary:

Graphically provides an overview of how the job is running at its nodes. The graph is divided in up to four sections, indicating success, partial success, failure, or other, in that order, of the nodes in the job.

Status	Description
Success	Indicates the number of nodes on which the job completed successfully.

Status	Description
Partial success	Indicates the number of nodes on which the job partially completed. Partial success can occur, for example, when a node represents multiple servers, and only some of the servers on the node complete successfully.
Failed	Indicates the number of nodes on which the job failed to complete.
Other	Indicates the number of nodes on which the job has some other status than success, partial success, or failure. A status of other can include nodes on which the job is currently running, or nodes on which the job has not started.

Job status detail settings:

This panel displays the job status such as success, partial success, or failed, at each node of the job. Job information, including the job ID, description, activation time, and expiration time, is also displayed.

To view the status of all the nodes for a particular job, click **Jobs** > **Status** > *job ID*.

To view only the successful nodes, only the failed nodes or only nodes with a status of other for a particular job, click **Jobs** > **Status** > *number for the successful, failed or other nodes in the status summary*.

Find:

When you click **Jobs** > **Status** > *job ID*, you can use the Find option to limit the nodes to display. The Find results are displayed in the table that follows the Find option after you click **Find**. Clicking **Reset** clears the operators in column 2 and text in column 3.

Column 1	Column 2	Column 3
This column lists the node name parameter, the status parameter and the maximum results parameter.	• Job ID parameter: Valid operators for the find operation are `=` (equal to), `!=` (not equal to), `is null`, and `is not null`. • Status parameter: Valid operators for the find operation are `Succeeded`, `Failed`, `Rejected`, `In progress`, `Distributed`, and `Not attempted`. `In progress` is the same as `ASYNC_IN_PROGRESS` in the status column. `Not attempted` is the same as `NOT_ATTEMPTED` in the status column.	• For the node name parameter, specifies the string or partial string of a parameter. A partial string is designated using an asterisk (*). For example, setting the node name to `AppSrv*` finds all jobs with a node name that starts with `AppSrv`. You can search on an exact match for multiple items by including the items to match separated by commas. For example, you can search on two node names by entering `AppSrv01,` `AppSrv02`. • Maximum results: Specifies the number of records that the find operation displays. Enter a value between one and the maximum number of records that can be retrieved as defined in the job manager configuration.

Example: If the nodes are EastCoast1, EastCoast2, WestCoast1, and WestCoast2, you can enter `East` or `East*` for column 3 and `=` for column 2 of the node parameter. Job status is displayed for the jobs that include the EastCoast1 and EastCoast2 nodes.

Job ID: Specifies the job number of the job that you submitted.

Description: Specifies the description that you entered when you submitted the job.

Activation time:

The activation time is the time that the job is available to start, but not necessarily the time that the job actually starts.

You entered the activation time when you submitted the job.

Expiration time:

Specifies the expiration time that you entered when you submitted the job.

If the job has not started by the expiration time, the job will not start.

Node name: Specifies the name of each node that is included in the job.

Status:

Specifies the status of the job at its nodes.

Status	Description
Succeeded	The job completed successfully on the node.
Partially succeeded	The job partially completed on the node. Partial success can occur, for example, when a node represents multiple servers, and only some of the servers on the node complete successfully.
Failed	The running of the job on the node failed.
Not attempted	The agent for the node has not received the job.
Distributed	The agent for the node has received the job, but the job has not completed.
In progress	The agent for the node has received the job and is running the job concurrent with other jobs for other nodes that belong to the agent.
Rejected	The agent rejected the job because the agent does not support the job type. The rejection can happen if a new node is added to the job group after the job is submitted.

Job status history:

This panel displays the job status of the node at various stages of the job for the node.

To view this administrative console panel, click **Jobs > Status >** *job ID* **>** *node name*.

The **Previous records** button and **Next records** button allow you to scroll backwards and forwards through the list of records if you are not viewing all records in the job history for the node. More records can be generated as the job progresses. These records are retrieved and included in the job history records as you click the **Previous records** button and the **Next records** button.

Find:

When you click **Jobs > Status >** *job ID* **>** *node name*, you can use the Find option to limit the submitted jobs to display. The Find results are displayed in the table that follows the Find option after you click **Find**. Clicking **Reset** clears the operators in column 2 and text in column 3.

Column 1	Column 2	Column 3
Except for the maximum results parameter, this column lists parameters that define a job.	• Job ID and node name parameters: Valid operators for the find operation are = (equal to), != (not equal to), `is null`, and `is not null`. • Time stamp: Valid operators are >=, and <=.	• For the node name parameter, specifies the string or partial string of a parameter. A partial string is designated using an asterisk (*). For example, setting the node name to `AppSrv*` finds all jobs with a node name that starts with `AppSrv`. You cannot use the asterisk in the job ID field or the time stamp field. You can search on an exact match for multiple items by including the items to match separated by commas. For example, you can search on two job IDs by entering `119489625729609463, 119489651801509472`. • Maximum results: Specifies the number of records that the find operation displays. Enter a value between one and the maximum number of records that can be retrieved as defined in the job manager configuration.

Time stamp: Specifies the date and time at which the status is logged for the job running at the node.

Status:

Specifies the status of the job running at the node for various stages of the job. The following table lists valid statuses with a description of each status.

Status	Description
Succeeded	The job completed successfully on the node.
Partially succeeded	The job partially completed on the node. Partial success can occur, for example, when a node represents multiple servers, and only some of the servers on the node complete successfully.
Failed	The running of the job on the node failed.
Not attempted	The agent for the node has not received the job.
Distributed	The agent for the node has received the job, but the job has not completed.
In progress	The agent for the node has received the job and is running the job concurrent with other jobs for other nodes that belong to the agent.
Rejected	The agent rejected the job because the agent does not support the job type. The rejection can happen if a new node is added to the job group after the job is submitted.

Message: Specifies error messages associated with the job when the status is `Failed`.

Administering nodes of the job manager

In a flexible management environment, you can view nodes with their version numbers based on the results of the Find option, and view node resources for nodes that you select. You can also view the properties and property values for a particular node.

Before you begin

Before you can view information about nodes, you must have registered at least one node with the job manager.

To display resources for selected nodes, your ID must be authorized for the monitor role.

About this task

The first time you access the Nodes collection panel, no nodes are listed. You must enter parameters for the Find option to obtain a list of nodes based on the parameter information that you provide. The next time you select **Jobs > Nodes**, a list of nodes are displayed based on the parameters you last specified on the Find option for this job manager administrative console panel. You can then optionally modify the Find option criteria to display a different set of nodes. After at least one node is displayed, you can view information about the nodes. You can display all nodes by setting the string for the node name on the Find option to `*` .

- Optionally use the Find option to display a set of nodes.

  If no nodes are displayed, you must use the Find option to display nodes based on the parameter information that you enter.

  1. Click **Jobs > Nodes** in the job manager administrative console navigation.
  2. If you want to run the Find operation on specific parameters, specify a valid operator and a text string. You can use the asterisk (*) character on the node name, job type, and unique identifier parameters to represent unspecified characters in conjunction with the characters that you specify. The node Find option has some advanced Find options that you can view by selecting the plus (+) character. You can enter partial search strings for the advanced find options as well.
  3. Click **Find**.

  The list of nodes along with their version number is displayed in the collection table.
- Optionally display resources for selected nodes.
  1. Click **Jobs > Nodes** in the job manager administrative console navigation.
  2. Select the check box next to nodes for which you want to display resources.
  3. Click **Display resources**.
  4. Choose the type of resources to display.

  The resources are displayed for the nodes that you selected.
- Optionally display properties and property values for a particular node by clicking **Jobs > Nodes > *managed_node_name*** in the job manager administrative console navigation.

Results

Depending on the tasks that you completed, you might have viewed nodes with their version numbers based on the results of the Find option, viewed node resources for nodes that you selected, or viewed the properties and property values for a particular node.

What to do next

You can continue to view nodes and do other job management tasks such as submit jobs, create node groups for job submission, and view node resources.

Nodes collection for Find results

This panel displays nodes with their version numbers based on the results of the Find option. You can additionally display node resources for nodes that you select.

To view this administrative console page, click **Jobs > Nodes**.

Find:

When you click **Jobs > Nodes**, you can use the Find option to determine the nodes to display. After you click **Find**, the results are displayed in the table. The table follows the Find and Preferences options. Clicking **Reset** clears the operators in column 2 and text in column 3.

Column 1	Column 2	Column 3
Except for the maximum results parameter, this column lists parameters that define nodes. The node name, job type, and unique identifier parameters are always available. The rest of the parameters are for the node properties, dynamic, and built at run time. The dynamic parameters are displayed when you click the Advanced find options link.	• All parameters except maximum results: Valid operators for the find operation are = (equal to), != (not equal to), `is null`, and `is not null`.	• All parameters except maximum results and type: Specifies the string or partial string of a parameter. A partial string is designated using an asterisk (*). For example, setting the node name parameter to `Node*` finds all jobs with a node name that starts with `Node`. You can search on an exact match for multiple items by including the items to match separated by commas. For example, you can search on two node names by entering `Node1,Node2`. When you search on more than one item, you cannot use the asterisk. • Maximum results: Specifies the number of records that the find operation displays. Enter a value between one and the maximum number of records that can be retrieved as defined in the job manager configuration.

Example: If the nodes are AppSvr01, AppSvr02, AppSvr03, and Test01, you can enter `App` or `App*` for column 3 and = for column 2 of the node parameter. The node names displayed in the table are AppSvr01, AppSvr02, and AppSvr03.

Display resources: To display node resources of selected nodes, click **Display resources**. Your ID must be authorized for the monitor role. A dropdown list displays all available resources. Possible values are `All`, `Applications`, `Servers`, and `Clusters`. The choices in the list depend on the nodes registered with the job manager. For example, if you do not have a deployment manager registered to the job manager, clusters are not in the list.

Node name: Specifies the node names that are found as a result of the find option.

Version:

Specifies the product name and version number of the node.

The product version is the version of a WebSphere Application Server for nodes.

The base edition of WebSphere Application Server is listed in the version column as `Base`. The Network Deployment product is listed in the version column as `ND`.

The product in the version column indicates the product that you used to create the profile, not the type of profile that you installed. For example, if you use the Network Deployment product to install a profile type of application server, the version column indicates `ND`.

Node properties:

This panel allows you to view the properties and property values for a particular node.

To view this administrative console page, click **Jobs > Nodes >** *managed_node_name*.

Node name: Specifies the node name that you selected from the nodes collection.

Name: Specifies the name of each property for the node. The list of names varies from one runtime to another and is read-only.

Value: Specifies a value of the specified name. The value is read-only.

Administering node resources of the job manager

In a flexible management environment, you can view server, application, and cluster resources associated with nodes and node groups registered to the job manager. You can also view the status of specific resources at each node, and view properties for a particular node resource as a name-value pair.

Before you begin

Before you can view information about node resources, you must have registered at least one node with the job manager.

About this task

The type of resources you can view depends on your topology. For example, you cannot view clusters if a deployment manager that you registered to the job manager does not have a defined cluster.

The first time you access the Node resource collection panel, no node resources are listed. You must enter parameters for the Find option to obtain a list of node resources based on the parameter information that you provide. The next time you select **Jobs > Node resources**, a list of node resources are displayed based on the parameters you last specified on the Find option for this console panel. You can then optionally modify the Find option criteria to display a different set of node resources. After at least one node resource displays, you can view information about the node resources.

- Optionally use the Find option to display a set of node resources.

  If no node resources are displayed, you must use the Find option to display node resources based on the parameter information that you enter.

  1. Click **Jobs > Node resources** in the console navigation.
  2. Choose the resource type in the **Type** list.
  3. If you want to do a Find operation on specific parameters, specify a valid operator and a text string.
  4. Click **Find**.

  The list of node resources along with the number of node resources for a given node resource in the list and the node for the resource are displayed.

- Optionally display the status of specific node resources for each node.
  1. Click **Jobs > Node resources > *managed_resource*** in the job manager administrative console navigation

  The resource ID, node name, and resource status are displayed.
- Optionally display the properties of a node resource by clicking **Jobs > Node resources > *managed_resource* > *managed_resource_ID*** in the job manager administrative console navigation.

Results

Depending on the tasks that you completed, you might have viewed server, application, and cluster resources associated with nodes and node groups registered to the job manager. You might have also viewed the status of specific resources at each node, and viewed properties for a particular node resource as a name-value pair.

What to do next

You can continue to view node resources and do other job management tasks such as submit jobs, create node groups for job submission, and view nodes.

Node resources collection

This panel displays server, application, node, and cluster resources associated with nodes and node groups registered to the job manager.

To view this administrative console panel, click **Jobs > Node resources**.

Find:

You can use the Find option to determine the resources to display. The Find results are displayed in the table that follows the Find and Preference options after you click **Find**. Clicking **Reset** clears the operators in column 2 and text in column 3.

Column 1	Column 2	Column 3
Except for the maximum results parameter, this column lists parameters that define the resources of nodes. The node name and type parameters are always displayed. The rest of the resource parameters in the list vary from node to node. **Node name** Specifies the name of the node on which the resource resides. **Type** Specifies the type of resource. Options are server, application, and cluster. **Status** Specifies the status of the resource. Valid values for the status vary because different resources have different status. Examples of status for a server are started and stopped. **Context** Specifies topology information of the resource, such as cell/application. **Resource name** Specifies the name of a resource type. For example, a server resource could have a resource name of server1. **Maximum results** The maximum number of results that display after the find option completes.	• Type: Valid values for the type of resource are server, application, and cluster. • Status, node name, context, and resource parameters: Valid operators for the find operation are `=` (equal to), `!=` (not equal to), `is null`, and `is not null`. When you search on more than one item, you cannot use the asterisk.	• All parameters except type and maximum results: Specifies the string or partial string of a parameter. • Maximum results: Specifies the number of records that the find operation displays. Enter a value between one and the maximum number of records that can be retrieved as defined in the job manager configuration.

Example: If the resource names are server1, server2, application01, and application03, you can select an operator of `!=` in column 2 for the resource name parameter, and a resource name of `server` or `*erver` in column 3, The results of the find operation display the application01 and application03 resource information.

Resources: Specifies the ID of the resource.

Number: Specifies the number of resources with this name registered with the job manager.

Node name: Specifies the name of the node for the resource. When the resource is defined on multiple nodes, the node name is displayed as `multiple`, and the number is greater than one. When you click on the resource name, you can see the details of the nodes where this resource resides.

Node resources for nodes collection:

This panel displays the status of specific resources at each node, and includes the resource ID, the name of the node, and the resource status.

To view this administrative console panel, click **Jobs > Node resources >** *resource_ID*.

Resource ID: Specifies a unique identifier for the resource in the form of context and values. For example, servers display as

server/*server_name* for nodes registered with an administrative agent

node/*node_name*/server/*server_name* for nodes federated in a registered deployment manager

Node name: Specifies the node on which the resource is located.

Status: Specifies the status of the resource. Valid values for the status varies because different resources have different status. Examples of status for a server are started and stopped.

Node resource properties:

This panel displays a read-only view of the properties for a particular node resource as a name-value pair. Updates that you make to the node resource properties must be done at the administrative agent.

To view this administrative console panel, click **Jobs > Node resources >** *resource_ID* **>** *resource_ID*.

Name:

Specifies the name of the property. The property name is unique.

Value:

Specifies the value paired with the specified name.

Administering groups of nodes for the job manager

In a flexible management environment, you can create, modify, delete, and view groups of nodes. Groups of nodes make job submission simpler because you can submit a job for a group of nodes instead of a entering multiple node names for a job submission.

Before you begin

Before you can add a node to a group of nodes, you must have registered at least one node with the job manager.

About this task

Groups of nodes are particularly useful if you submit multiple jobs to the same set of nodes.

The first time you access the Groups of nodes collection panel, no groups of nodes are listed. You must create at least one group. You must then enter parameters for the Find option to obtain a list of groups of nodes based on the parameter information that you provide. The next time you select **Jobs > Groups of nodes**, a list of groups of nodes are displayed based on the parameters you last specified on the Find option for this job manager administrative console panel. You can then optionally modify the Find option criteria to display a different set of groups of nodes. After at least one group of nodes is displayed, you can administer the groups of nodes by doing such tasks as adding and removing members for node groups, or deleting node groups.

- Create a group of nodes.
  1. Click **Jobs > Groups of nodes** in the job manager administrative console navigation.
  2. Click **New**.
  3. Enter the name of the group of nodes.
  4. Optionally enter a description.
  5. Optionally add members to or remove members from the group of nodes.

Members are nodes. You can add members to the group or delete members from the group now or later. You can use the Add option, the Find option, or both options to add the members. Follow the steps on adding to or removing members from a group of nodes.

6. Click **Apply** to save the changes, and then click **OK** to return to the collection page.

- Optionally use the Find option to display groups of nodes.

  If no groups of nodes are displayed, you must use the Find option to display groups of nodes based on the parameter information that you enter.

  1. Click **Jobs > Groups of nodes** in the job manager administrative console navigation.
  2. Specify a valid operator and a text string.
  3. Click **Find**.

- Optionally add or remove the members in a group of node.

  You can add and remove members from the group of nodes. Members are nodes.

  1. Click **Jobs > Groups of nodes** in the job manager administrative console navigation.
  2. Click one of the names of a group of nodes.
  3. To add a node, use the Add option, the Find option, or both.
     a. To use the Add option, type the name of the node in the **Member** list box, and then click **Add**.
     b. Continue to type the name of a node, and then click **Add** until you have added all the members.
     c. To use the Find option, click **Find**.
        1) Enter criteria for the Find option by adding text for one or more options. For example, specify the node name as `test*` or `test*a`.
        2) After getting the list of nodes in the **Chosen nodes** list, click **OK** to return the list to the Groups of nodes panel.

        You can change the find criteria, and select the Find option multiple times to create the list you want.
  4. To remove a node, select the node, and click **Remove**.
  5. Click **Apply**, and then click **OK**.

- Optionally delete one or more groups of nodes.
  1. Click **Jobs > Groups of nodes** in the job manager administrative console navigation.
  2. Select one or more groups of nodes.
  3. Click **Delete**.

Results

Depending on the tasks that you completed, you might have created a group of nodes, used the Find option to display groups of nodes, added or deleted members in the group of node, or deleted groups of nodes.

What to do next

You can continue to administer groups of nodes and do other job management tasks such as view nodes, submit jobs, and view node resources.

Groups of nodes collection

This panel allows you to create and view groups of nodes. Groups of nodes make job submission easier because you can submit a job for a group of nodes instead of a separate job for each node.

To view this administrative console page, click **Jobs > Groups of nodes**.

To create or delete a group, you must be authorized for the configurator role.

Find:

When you click **Jobs > Groups of nodes**, you can use the Find option to limit the groups of nodes to display. The Find results are displayed in the table that follows the Find and preferences options after you click **Find**. Clicking **Reset** clears the operators in column 2 and text in column 3.

Column 1	Column 2	Column 3
This column lists the group name parameter, the job type parameter, the description parameter, and the maximum results parameter.	• Group name parameter, job type parameter, and description parameter: Valid operators for the find operation are = (equal to), != (not equal to), `is null`, and `is not null`.	• Group name parameter, job type parameter, and description parameter: Specifies the string or partial string of a parameter. A partial string is designated using an asterisk (*). For example, setting the group name parameter to `Region*` finds all jobs with a group name that starts with `Region`. You can search on an exact match for multiple items by including the items to match separated by commas. For example, you can search on two group names by entering `Region1, Region2`. When you search on more than one item, you cannot use the asterisk. • Maximum results: Specifies the number of records that the find operation displays. Enter a value between one and the maximum number of records that can be retrieved as defined in the job manager configuration.

Example: If you have already defined groups Region1, Region2, Center1, and Center2, you can enter `Region` or `Region*` for column 3 and = for column 2 of the group name parameter. When you click **Find**, only groups Region1 and Region2 will be displayed in the collection.

Group name: Specifies the name of the group.

Members: Specifies the number of members in the group. A member is a node.

Description: Specifies a description of the group.

Groups of nodes settings:

This panel allows you to set the name of a group of nodes, description, and group members. Groups of nodes make job submission easier because you can submit a job for a group of nodes instead of a separate job for each node.

To view this administrative console panel, click **Jobs > Groups of nodes >** *group_name*.

Group name: Specifies the name of the group. When you create a new group of nodes, the name is verified to be unique.

Data type	String
Maximum length	64 characters
Default	None

Description: Specifies an optional string that describes the group.

Data type	String
Maximum length	256 characters
Default	None

Member list: Specifies nodes to add to the group. You can either type the node name or use the **Find** option to add nodes to the list. The node must already exist. There are two types of nodes for the job manager. A node that is registered to an administrative agent and to the job manager is a node. A deployment manager that is registered to the job manager is a node.

Find nodes:

This panel allows you to build a list of nodes that you can use to choose the nodes on which you want the job to run. You can also find nodes to add to a group of nodes.

This console panel is navigated to in two ways.

- On step 2 of the Job submission wizard you can select **Find**.
- Click **Jobs > Groups of nodes > *group_name* > Find**.

Find:

When you click **Jobs > Nodes**, you can use the Find option to determine the nodes to display. The Find results are displayed in Available nodes after you click **Find**. Click **Reset** to clear the operators in column 2 and text in column 3.

Column 1	Column 2	Column 3
Except for the maximum results parameter, this column lists parameters that define nodes. The node name, job type, and unique identifier parameters are always available. The remaining parameters are for the node properties, dynamic, and built at runtime. The dynamic parameters are displayed when you click **Advanced find options**.	• All parameters except maximum results: Valid operators for the find operation are `=` (equal to), `!=` (not equal to), `is null`, and `is not null`.	• All parameters except maximum results: Specifies the string or partial string of a parameter. A partial string is designated using an asterisk (*). For example, setting the node name parameter to `Node*` finds all jobs with a node name that starts with `Node`. You can search for an exact match for multiple items by including comma-separated items. For example, you can search on two node names by entering `Node1`, `Node2` When you search for more than one item, you cannot use the asterisk. • Maximum results: Specifies the number of records that the find operation displays. Enter a value between one and the maximum number of records that can be retrieved as defined in the job manager configuration.

Example: If the nodes are AppSvr01, AppSvr02, AppSvr03, and Test01, you can enter `App` or `App*` for column 3 and = for column 2 of the node parameter. The node names displayed in the table are AppSvr01, AppSvr02, and AppSvr03.

Available nodes: Specifies the initial results of the Find option. The nodes are not included in the list you are building until you use the right arrow to move nodes into the chosen nodes list. You can select multiple items in the list, and move them simultaneously.

Chosen nodes: Specifies the list of nodes to be included in the group of nodes that you are creating. The left arrow is used to move nodes into the available nodes list. You can select multiple items in the list, and move them simultaneously.

Configuring job managers

In a flexible management environment, you can specify settings such as the default job expiration, the job manager Web address, and the mail provider Java Naming and Directory Interface (JNDI) name for the job manager. You can view job manager properties such as the process ID and the state of the job manager.

Before you begin

Before you can configure the job manager, you must have started the job manager.

If you plan to change the system clock, stop all the application servers, the node agent servers, the deployment manager server, the administrative agent server, and the job manager server first. After you stop the servers, change the system clock, and then restart the servers. If you change the system clock on one system, you must ensure the clocks on all systems that communicate with each other and have WebSphere Application Server installed are synchronized. Otherwise, you might experience errors, such as security tokens no longer being valid.

About this task

When you create a job manager profile, a job manager is created. You can run the job manager with its default settings. However, you can change the job manager configuration settings, such as the expiration time of jobs, the maximum number of database results to display, and so on.

1. Click **System administration > Job manager** from the navigation tree of the job manager administrative console to access the settings page for a job manager.
2. Configure the job manager by clicking a property and specifying settings.
   a. Optionally change the default job expiration by specifying an integer value.
   b. Optionally change the maximum number of rows that a query can return to the job manager by specifying an integer value on the **Maximum database results** setting.
   c. Optionally change the Web address of the job manager that the administrative agent uses to retrieve jobs by specifying a Web address for the **Job manager URL** setting.

      The Web address is used only when the job manager is configured as a proxy server.
   d. Optionally specify an e-mail address on the E-mail sender's address setting.

      The e-mail address that you specify is the e-mail address of the sender of the notification message that the job manager provides when jobs have completed.
   e. Optionally select the **Start components as needed** setting to start components dynamically as needed for applications.
   f. Optionally stop the job manager by clicking **Stop**.

      Selecting this option stops the job manager and its console.
3. If you changed any of the settings, click **Apply**, and then **OK**.

Results

You configured the job manager with options that you selected.

What to do next

You can do other job management tasks such as administer node groups, view nodes, submit jobs, and view node resources.

Job manager settings

This panel allows you to configure the job manager server and view its properties. You can specify the default job expiration, the job manager Web address, and the mail provider Java Naming and Directory Interface (JNDI) name.

To view this administrative console page, click **System administration > Job manager**.

Name: Specifies the job manager server name. The name is read-only.

Default job expiration:

Specifies the default job expiration time in days.

Data type	Integer
Default	60

Maximum database results: The maximum database results property applies to find operations for jobs, nodes, and node resources. The number represents the maximum number of records that can be retrieved during a job manager find operation. This number can be further reduced by the maximum results property on a find operation. Assume, for example, that you specify the maximum results to display for finding nodes at 50, but the maximum database results property is set to 10000. If you have 20000 jobs, the find operation finds 50 nodes.

Data type	Integer
Default	10000

Job manager URL:

Specifies the Web address of the job manager that the administrative agent uses to fetch jobs.

The Web address that you specify is used only when the job manager is configured as a proxy server. The Web address overrides the default Web address. If you modify the Web address, you must reregister the nodes with the job manager. The change affects only the nodes previously registered.

Data type	String
Default	http://*host*:*port*/otis/OMADMServlet The *host* and *port* are those of the job manager unless you use a Web server. In that case, change the *host* and *port* to that of the Web server.

Mail provider JNDI name:

Specifies an optional JNDI mail provider to send an e-mail notification that a job has completed.

Data type	String

Default None

E-mail sender's address: Specifies the e-mail address of the sender of the notification message that the job manager provides when jobs have completed. This setting is required if you specify a JNDI mail provider on the Mail provider JNDI name setting.

Start components as needed: Select this property if you want the server components started as they are needed for applications that run on this server.

When this property is not selected, all of the server components are started during the startup process. Therefore, selecting this property usually results in improved startup time because fewer components are started during the startup process.

Note: If you are running other WebSphere products on top of the this product, make sure that those other products support this functionality before you select this property.

Process ID: Specifies the read-only process ID of the job manager.

Cell name: Specifies the read-only cell name of the job manager.

Node name: Specifies the read-only node name of the job manager.

State: Specifies the read-only state of the job manager, such as started or stopped.

Administration service settings

Use this page to view and change the configuration for an administration service.

To view this administrative console page, click **Servers > Application Servers >** *server_name* **> Administration > Administration Services**

Remote connector

Specifies the remote JMX Connector type. The remote JMX connector is the connector that is used between server processes that reside on different physical machines, for example, between the deployment manager and the node agent. Available options of SOAPConnector, RMIConnector, and JSR160RMI Connector are defined using the JMX Connectors page.

Data type String
Default SOAPConnector

Local connector

Specifies the local JMX Connector type. The local JMX connector is the connector used between server processes that reside on the same physical machine, for example, between the node agent and its application servers. Available options of SOAPConnector, RMIConnector, JSR160RMI Connector, and IPC Connector are defined using the JMX Connectors page.

Data type String
Default IPC Connector

Administration services custom properties

This topic discusses the administration services custom properties that you can set on the administrative console.

To view the administration services custom properties administrative console page that goes with this topic, click: **Servers > Application Server >** *server_name* **> Administration > Administration Services > Custom Property**.

Specify a property and its value as a name-value pair on the Administration services custom properties page.

com.ibm.websphere.mbeans.disableRouting

When a custom managed bean (MBean) is registered directly with the MBean server that runs in a WebSphere Application Server process, the MBean object name is enhanced by default to include the cell, node, and process names as key properties. To turn off the default behavior, set the following custom property on the application server.

With this enhancement, in a Network Deployment environment, the MBean that is registered on an application server is addressable through a client that is connected to the deployment manager.

If this custom property is set, an administrative client needs to connect directly to the application server on which the MBean is registered to invoke methods. The MBean cannot participate in all the distributed functions of the administrative system.

One or more MBean object names tagged with `<on>...</on>`. You can specify the object name of your MBean or a pattern that matches the names of several MBeans.

Example:

If you register a custom MBean with the `WebSphere:type=custom,name=custommbean1` object name and another custom MBean with the `WebSphere:type=custom,name=custommbean2` object name, each of the following values is valid:

- `<on>WebSphere:type=custom,name=custommbean1</on>`

  The value disables the MBean object name modification for this MBean.
- `<on>WebSphere:type=custom,*</on>`

  The value disables the MBean object name modification for both MBeans.
- `<on>WebSphere:type=custom,name=custommbean1</on><on>WebSphere:type=custom,name=custommbean2</on>`

  The value disables the object name modification for both MBeans.

Administrative audits

This topic discusses aspects of administrative audits, such as log files that contain the audit information, the administrative actions that are audited, and the types of audit messages that are logged.

Administrative audits use the same logging facility as the rest of the product. The audits are available in both the `activity.log` file and the `SystemOut.log` of the server that performs the action. You do not need to enable trace to produce the audits. However, through the Repository service console page, you can control whether configuration change auditing is done. This type of audit is done by default. Operational command auditing is always enabled. Information about which user performed the change is available only when security is enabled.

You can do administrative audits with or without the security audit facility. The security audit facility can record unauthorized access in audit log files. You can sign and encrypt the file-based audit logs to ensure data integrity. You can protect the audit files using directory and file permissions.

The following administrative actions are audited:

- All configuration changes, in terms of the configuration documents that are created, modified, or deleted.
- Certain operational changes like starting and stopping nodes, clusters, servers, and applications. These managed bean (MBean) operations provide administrative auditing:

Table 10.

MBean type	MBean operations
CellSync	syncNode
Cluster	start, stop, stopImmediate, rippleStart
NodeAgent	launchProcess, stopNode, restart
Server	stop, stopImmediate
AppManagement	startApplication, stopApplication

Configuration change audits have ADMRxxxxI message IDs, where xxxx is the message number. Operational audits have ADMN10xxI message IDs, where 10xx is the message number.

Here are some audit examples from a Network Deployment environment.

The following audit example is from the deployment manager `SystemOut.log` file:

```
[7/23/03 17:04:49:089 CDT] 39c26dad FileRepositor A ADMR0015I: Document
cells/ellingtonNetwork/security.xml was modified by user u1.
   [7/23/03 17:04:49:269 CDT] 3ea0edb5 FileRepositor A ADMR0016I: Document
cells/ellingtonNetwork/nodes/ellington/app.policy was created by user u1.
   ...
   [7/23/03 17:13:54:081 CDT] 39a572a1 AdminHelper   A ADMN1008I: Attempt
made to start the SamplesGallery application. (User ID = u1)
   ...
```

The following audit example is from the node agent `SystemOut.log` file:

```
[7/23/03 17:38:43:461 CDT]  23d1326 AdminHelper   A ADMN1000I: Attempt
made to launch server1 on node ellington. (User ID = u1)
```

The following audit example is from the application server`SystemOut.log` file:

```
[7/23/03 17:39:59:360 CDT] 24865373 AdminHelper   A ADMN1020I: Attempt
made to stop the server1 server. (User ID = u1)
```

The message text is split for printing purposes.

Remote file services

Configuration documents describe the available application servers, their configurations, and their contents. Two file services manage configuration documents: the file transfer service and the file synchronization service.

The following information describes what the file services do:

File transfer service

The file transfer service enables the moving of files between the deployment manager and the nodes as well as between the wsadmin scripting process and either the deployment manager or the application server. It uses the HTTP protocol to transfer files. When you enable security in the WebSphere Application Server product, the file transfer service uses certificate-based mutual authentication. You can use the default key files in a test environment. Ensure that you change the default key file to secure your system.

The ports used for file transfer are the HTTP_Transport port, the HTTPS transport port, the administrative console port, and the administrative console secure port. For more information, see the topic on port number settings in WebSphere Application Server versions.

File synchronization service

The file synchronization service ensures that a file set on each node matches that on the deployment manager node. This service promotes consistent configuration data across a cell. You can adjust several configuration settings to control file synchronization on individual nodes and throughout a system.

This service runs in the deployment manager and node agents, and ensures that configuration changes made to the cell repository are propagated to the appropriate node repositories. The cell repository is the master repository, and configuration changes made to node repositories are not propagated up to the cell. 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.

The default behavior, which is enabled, is for each node agent to periodically run a synchronization operation. You can configure the interval between operations or disable the periodic behavior. You can also configure the synchronization service to synchronize a node repository before starting a server on the node.

Configuring remote file services

Configuration data for the WebSphere Application Server product resides in files. Two services help you reconfigure and otherwise manage these files: the file transfer service and file synchronization service.

About this task

By default, the file transfer service is always configured and enabled at a node agent, so you do not need to take additional steps to configure this service. However, you might need to configure the file synchronization service.

1. Go to the File Synchronization Service page. Click **System Administration > Node agents** in the console navigation tree. Then, click the node agent for which you want to configure a synchronization server and click **File synchronization service**.
2. On the File Synchronization Service page, customize the service that helps make configuration data consistent across a cell by moving updated configuration files from the deployment manager to the node. Change the values for properties on the File Synchronization Service page. The file synchronization service is always started, but you can control how it runs by changing the values.
3. Optionally add a custom property for file synchronization.
   a. Click **Custom properties** on the File Synchronization Service page.
   b. Click **New**.
   c. Specify a name and value for the custom property.

      The com.ibm.websphere.management.sync.allowfailure custom property:

      When synchronization fails five times consecutively, automatic synchronization is disabled. To keep automatic synchronization enabled, specify this custom property with a value of `true`.

Property	com.ibm.websphere.management.sync.allowfailure
Data type	Boolean
Default	False

File transfer service settings

Use this page to configure the service that transfers files from the deployment manager to individual remote nodes.

To view this administrative console page, click **System Administration** > **Node agents** > *node_agent_name* > **File transfer service**.

Enable service at server startup

Specifies whether the server attempts to start the specified service. Some services are always enabled and disregard this property if set. This setting is enabled by default.

Data type Boolean
Default true

Retries count

Specifies the number of times you want the file transfer service to retry sending or receiving a file after a communication failure occurs.

Data type Integer
Default 3

If the retries count setting is blank, the file transfer service sets the default to 3. If the retries count setting is `0`, the file transfer service does not retry. The default is the recommended value.

Retry wait time

Specifies the number of seconds that the file transfer service waits before it retries a failed file transfer.

Data type Integer
Default 10

If the retry wait time setting is blank, the code sets the default to 10. If the retry wait time setting is `0`, the file transfer service does not wait between retries. The default is the recommended value.

File synchronization service settings

Use this page to specify that a file set on one node matches that on the central deployment manager node and to ensure consistent configuration data across a cell.

You can synchronize files on individual nodes or throughout your system.

Note: If your installation includes mixed release cells, a large numbers of nodes, and runs a large number of applications, you might want to use the **Generic JVM Arguments** field, on the Java Virtual Machine Settings page of the administrative console, to enable the hot restart sync feature of the synchronization service. This feature indicates to the synchronization service that the installation is running in an environment where configuration updates are not made when the deployment manager is not active. Therefore, the service does not have to perform a complete repository comparison when the deployment manager or node agent servers restart.

To view this administrative console page, click **System Administration** > **Node agents** > *node_agent_name* > **File synchronization service**.

Enable service at server startup

Specifies whether the server attempts to start the file synchronization service. This setting does not cause a file synchronization operation to start. This setting is enabled by default.

Data type Boolean

Default true

Synchronization Interval

Specifies the number of minutes that elapse between synchronizations. Increase the time interval to synchronize files less often. Decrease the time interval to synchronize files more often.

Data type Integer
Units Minutes
Default 1
The minimum value that the application server uses is 1. If you specify a value of 0, the application server ignores the value and uses the default of 1.

Automatic Synchronization

Specifies whether to synchronize files automatically after a designated interval. When this setting is enabled, the node agent automatically contacts the deployment manager every synchronization interval to attempt to synchronize the node's configuration repository with the master repository owned by the deployment manager.

If the Automatic synchronization setting is enabled, the node agent attempts file synchronization when it establishes contact with the deployment manager. The node agent waits the synchronization interval before it attempts the next synchronization.

Remove the check mark from the check box if you want to control when files are sent to the node.

Data type Boolean
Default true

Startup Synchronization

Specifies whether the node agent attempts to synchronize the node configuration with the latest configurations in the master repository prior to starting an application server.

The default is to not synchronize files prior to starting an application server. Enabling the setting ensures that the node agent has the latest configuration but increases the amount of time it takes to start the application server.

Note that this setting has no effect on the startServer command. The startServer command launches a server directly and does not use the node agent.

Data type Boolean
Default false

Exclusions

Specifies files or patterns that should not be part of the synchronization of configuration data. Files in this list are not copied from the master configuration repository to the node, and are not deleted from the repository at the node.

The default is to have no files specified.

To specify a file, use a complete name or a name with a leading or trailing asterisk (*) for a wildcard. For example:

cells/*cell name*/nodes/*node name*/*file name*	Excludes this specific file

*/*file name*	Excludes files named *file name* in any context
dirname/*	Excludes the subtree under dirname

Press **Enter** at the end of each entry. Each file name appears on a separate line.

Since these strings represent logical document locations and not actual file paths, only forward slashes are needed no matter the platform.

Changes to the exclusion list are picked up when the node agent is restarted.

Data type String
Units File names or patterns

Changing the node host names

After creating a profile or adding a node, the host name of the server or its ports might be incorrect. You can follow the examples to change the server host name using command line tools and the wsadmin scripting tool, and the host name of the server ports using the administrative console and command line tools.

Before you begin

Create a profile or add a node to a cell. Verify that the host name of the server and the server ports are correct.

About this task

If the host name of a server or its ports is incorrect, then you might experience problems such as errors when you attempt to stop a server. One example task shows how to correct the server host name through command line tools and the wsadmin scripting tool. The other example task shows how to correct the host name of the server ports using the administrative console and command line tools.

- Correct the host name for an application server node, a node agent, or a deployment manager node using the wsadmin scripting tool and command line tools.
  1. Launch the wsadmin tool.

     Enter the following command:

     ```
     wsadmin -lang jython
     ```
  2. List the contents of the server configuration file.

     Enter the following line of code:

     ```
     AdminConfig.list('ServerIndex')
     ```
  3. In the output, find the ServerIndex object for the application server node, the node agent, or the deployment manager, similar to the following examples:

     Application server and node agent:

     ```
     cells/isthmusCell16/nodes/isthmusNode06|serverindex.xml#ServerIndex_1
     ```

     Deployment manager:

     ```
     cells/isthmusCell16/nodes/isthmusCellManager06|serverindex.xml#ServerIndex_1
     ```
  4. Modify the host name for the application server node, the node agent, or the deployment manager, similar to the following examples:

     Application server and node agent:

     Enter the following line of code:

```
AdminConfig.modify('(cells/isthmusCell16/nodes/isthmusNode06|serverindex.xml
#ServerIndex_1)',  "[[hostName new_host_name]]")
```

Deployment manager:

Enter the following line of code:

```
AdminConfig.modify('(cells/isthmusCell16/nodes/isthmusCellManager06|
serverindex.xml#ServerIndex_1)',  "[[hostName new_host_name]]")
```

The commands are split on multiple lines for printing purposes.

5. Verify that the host names are correct, similar to the following examples:

   Application server and node agent:

   Enter the following line of code:

   ```
   AdminConfig.show('(cells/isthmusCell07/nodes/isthmusCellManager07|
   serverindex.xml#ServerIndex_1)', 'hostName')
   ```

   The response is:

   ```
   '[hostName isthmus]'
   ```

   Deployment manager:

   Enter the following line of code:

   ```
   AdminConfig.show('(cells/isthmusCell07/nodes/isthmusNode04|
   serverindex.xml#ServerIndex_1)', 'hostName')
   ```

   The response is:

   ```
   '[hostName isthmus]'
   ```

   The commands are split on multiple lines for printing purposes.

6. Save the configuration.

   Enter the following line of code:

   ```
   AdminConfig.save()
   ```

7. Type `exit` to end the wsadmin session.
8. If you changed the host names for the application server and node agent, update the node with the changes.
   a. Stop the node agent.

      Enter the following command:

      ```
      stopNode -profileName AppSrv01
      ```

   b. Stop the application server.

      Enter the following command:

      ```
      stopServer server1 -profileName AppSrv01
      ```

   c. Synchronize the nodes.

      Enter the following command:

      ```
      syncNode deployment_manager_host deployment_manager_port
      ```

   d. Restart the node agent.

      Enter the following command:

      ```
      startNode -profileName AppSrv01
      ```

   e. Restart the application server.

      Enter the following command:

      ```
      startServer server1 -profileName AppSrv01
      ```

9. If you changed the host name for the deployment manager, restart the deployment manager to apply the changes.
   a. Stop the deployment manager.

      Enter the following command:

```
stopManager -profileName DMgr01
```

b. Start the deployment manager.

Enter the following command:

```
startManager -profileName DMgr01
```

- Correct the host names for the ports that an application server, node agent, or deployment manager opens.

  If you have to correct the host names of the server ports, then you can make the correction using command line tools and either the wsadmin scripting tool or the administrative console. You might have to correct the host names of multiple ports for a particular server. This example shows you how to correct the host names using the administrative console and command line tools.

  1. For the application server, select **Servers > Application servers > *application server* > Ports**. For the node agent, select **System administration > Node agents >*node agent* > Ports**. For the deployment manager, select **System administration > Deployment manager > Ports**.
  2. Select a port whose host name needs changing.
  3. Change the host name in the **Host** field; Click **OK**.
  4. Continue selecting ports and changing host names until you correct each of the host names for the server ports.
  5. Save the changes to the master configuration.
  6. If you changed the host names for the application server and node agent, update the node with the changes.
     a. Stop the node agent.
        – Select **System administration > Node agents**.
        – Select the node agent that you want to stop.
        – Click **Stop**.
     b. Stop the application server.
        – Select **Servers > Application servers**.
        – Select the server that you want to stop.
        – Click **Stop**.
     c. Synchronize the nodes.

        Enter the following command:

        ```
        syncNode deployment_manager_host deployment_manager_port
        ```

     d. Restart the node agent.
        – Select **System administration > Node agents**.
        – Select the node agent that you want to restart.
        – Click **Restart**.
     e. Restart the application server.
        – Select **Servers > Application servers**.
        – Select the server that you want to restart.
        – Click **Start**.
  7. If you changed the host name for the deployment manager, restart the deployment manager to apply the changes.
     a. Stop the deployment manager.
        – Select **System administration > Deployment manager**.
        – Click **Stop**.
     b. Start the deployment manager.

        Enter the following command:

        ```
        startManager -profileName DMgr01
        ```

Results

You have changed the host name of the server, the host names of the server ports, or both.

What to do next

You can continue to administer the product by doing such tasks as managing nodes, node agents, and node groups.

Administrative topology: Resources for learning

Use the following links to find relevant supplemental information about WebSphere Application Server administrative topologies and distributed administration. The information resides on IBM and non-IBM Internet sites, whose sponsors control the technical accuracy of the information.

These links are provided for convenience. Often, the information is not specific to the IBM WebSphere Application Server product, but is useful all or in part for understanding the product. When possible, links are provided to technical papers and IBM Redbooks® that supplement the broad coverage of the release documentation with in-depth examinations of particular product areas.

View links to additional information:

Administration

- IBM WebSphere Application Server Redbooks

  This site contains a listing of all WebSphere Application Server Redbooks.
- IBM WebSphere developerWorks®

  This site is the home of technical information for developers working with WebSphere products. You can download WebSphere software, take a fast path to developerWorks zones, such as VisualAge® Java or WebSphere Application Server, learn about WebSphere products through a newcomers page, tutorials, technology previews, training, and Redbooks, get answers to questions about WebSphere products, and join the WebSphere community, where you can keep up with the latest developments and technical papers.
- WebSphere Application Server Support page

  Take advantage of the Web-based Support and Service resources from IBM to quickly find answers to your technical questions. You can easily access this extensive Web-based support through the IBM Software Support portal at URL `http://www.ibm.com/software/support/` and search by product category, or by product name. For example, if you are experiencing problems specific to WebSphere Application Server, click **WebSphere Application Server** in the product list. The WebSphere Application Server Support page appears.

Extension MBean Providers collection

Use this page to view and change the configuration for JMX extension MBean providers.

You can configure JMX extension MBean providers to be used to extend the existing WebSphere managed resources in the core administrative system. Each MBean provider is a library containing an implementation of a JMX MBean and its MBean XML Descriptor file.

To view this administrative console page, click **Servers > Application Servers >** *server_name* **> Administration > Administration Services > Extension MBean Providers**

Name

The name used to identify the Extension MBean provider library.

Description
An arbitrary descriptive text for the Extension MBean Provider configuration.

Classpath
The path to the Java archive (JAR) file that contains the Extension MBean provider library. This class path is automatically added to the Application Server class path.

Extension MBean Provider settings
Use this page to view and change the configuration for a JMX extension MBean provider.

You can configure a library containing an implementation of a JMX MBean, and its MBean XML Descriptor file, to be used to extend the existing WebSphere managed resources in the core administrative system

To view this administrative console page, click **Servers > Application Servers >** *server_name* **> Administration > Administration Services > Extension MBean Providers >** *provider_library_name*

Name
The name used to identify the Extension MBean provider library.

Data type String

Classpath
The path to the Java archive (JAR) file that contains the Extension MBean provider library. This class path is automatically added to the Application Server class path. The class loader needs this information to load and parse the Extension MBean XML Descriptor file.

Data type String

Description
An arbitrary descriptive text for the Extension MBean Provider configuration. Use this field for any text that helps identify or differentiate the provider configuration.

Data type String

Extension MBean collection

You can configure Java Management Extension (JMX) MBeans to extend the existing WebSphere Application Server managed resources in the administrative console. Use this page to register JMX MBeans. Any MBeans that are listed have already been registered.

To view this administrative console page, click **Servers > Application Servers >** *server name* **> Administration > Administration Services > Extension MBean Providers >** *provider library name***> extensionMBeans**

DescriptorURI
Specifies the location, relative to the provider class path, where the MBean XML descriptor file is located.

Type Specifies the type to use for registering this MBean. The type must match the type that is declared in the MBean descriptor file.

Extension MBean settings
Use this page to view and configure Java Management Extension (JMX) MBeans.

To view this administrative console page, click **Servers** > **Application Servers** > *server name* > **Administration** > **Administration Services** > **Extension MBean Providers** > *provider library name* > **ExtensionMBeans** > *descriptorURI*

descriptorURI

Specifies the location, relative to the provider class path, where the MBean XML descriptor file is located.

Data type String

type

Specifies the type to use for registering this MBean. The type must match the type that is declared in the MBean descriptor file.

Data type String

Java Management Extensions connector properties

You can specify or set a property in the administrative console, the wsadmin tool, Application Server commands, the scripts that run from a command-line interface, or a custom Java administrative client program that you write. You can also set SOAP connector properties in the soap.client.props file and IPC connector properties in the ipc.client.props file.

A Java Management Extensions (JMX) connector can be a Remote Method Invocation (RMI) connector, a Simple Object Access Protocol (SOAP) connector, a JMX Remote application programming interface (JSR 160) Remote Method Invocation (JSR160RMI) connector, or an Inter-Process Communications (IPC) connector.

Note: You should eventually convert all of your RMI connectors to JSR160RMI connectors because support for the RMI connector is deprecated.

For specific information on how to code the JMX connector properties for the wsadmin tool, the Application Server commands, or scripts, see the particular tool or command. Read the application programming interfaces documentation to learn how to code the JMX connector properties for a custom Java administrative client program.

The JMX connectors that servers create use JMX connector properties that are accessible in the administrative console. The wsadmin tool and the Java administrative client use JMX connector properties in the soap.client.props, ipc.client.props, and sas.client.prop files.

For the administrative console, this topic specifies the coding of the particular setting or property. Coding of properties in the `soap.client.props` file and the `ipc.client.props` file that are specific to JMX connectors is specified. These SOAP properties begin with com.ibm.SOAP and the IPC properties begin with com.ibm.IPC. Other properties in the `soap.client.props` file and the `ipc.client.props` file that contain information that can be set elsewhere in the application server are not documented here. The coding for the com.ibm.ssl.contextProvider property, which can be set only in the `soap.client.props` file and the `ipc.client.props` file, is specified.

Each profile has property files at the following locations:

- For the SOAP connector:
  - Linux HP-UX Solaris AIX *installation root*/profiles/*profile name*/properties/soap.client.props
  - Windows *installation root*\profiles*profile name*\properties\soap.client.props
- For the IPC connector:

– Linux HP-UX Solaris AIX *installation root*/profiles/*profile name*/properties/ipc.client.props

– Windows *installation root*\profiles*profile name*\properties\ipc.client.props

These property files allow you to set different properties, including security and timeout properties. These properties are the default for all the administrative connections that use either the SOAP JMX connector or the IPC JMX connector between processes that run in a particular profile. For instance, the wsadmin program running under a particular profile uses the property values from these files for the SOAP connector behavior and the IPC connector behavior unless the properties are overridden by some other programmatic means.

To view the JMX connector custom properties administrative console panel that goes with this article, click **Servers > Application servers >***server name* **> Server Infrastructure > Administration > Administration Services > Additional properties > JMX Connectors>***connector type* **> Additional Properties > Custom properties**.

SOAP connector properties

This section discusses JMX connector properties that pertain to SOAP connectors.

SOAP request timeout

The value that you choose depends on a number of factors, such as the size and the number of the applications that are installed on the server, the speed of your machine, and the usage of your machine.

The program default value for the request timeout is `600 seconds`. However, other components that connect to the SOAP client can override the default. Components that use the `soap.client.props` file have a default value of `180 seconds`.

Set the property by using one of the following options:

- Scripts that run from a command-line interface.
- The `soap.client.props` file.

Property	com.ibm.SOAP.requestTimeout
Data type	Integer
Range in seconds	0 to n If the property is zero (0), the request never times out.
Default	180

- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	requestTimeout
Data type	Integer
Range in seconds	0 to n If the property is zero (0), the request never times out.
Default	600

- A Java administrative client. The property is AdminClient.CONNECTOR_SOAP_REQUEST_TIMEOUT.

Configuration URL

Specify the configuration Universal Resource Locator (URL) property if you want a program to read SOAP properties from this file. You can set the property by using one of the following options:

- Scripts run from a command-line interface. Scripts can pass the Configuration URL property to the application server on the com.ibm.SOAP.ConfigURL system property.
- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	ConfigURL
Data type	String
Valid Value	http://*Path*/soap.client.props
Default	None

- A Java administrative client. Use the AdminClient.CONNECTOR_SOAP_CONFIG property.

Security context provider

This property indicates the Secure Sockets Layer (SSL) implementation to use between the application server and the SOAP client.

Set the property by using the `soap.client.props` file.

Property	com.ibm.ssl.contextProvider
Data type	String
Valid Values	`IBMJSSE2`
Default	IBMJSSE2

Secure Sockets Layer (SSL) security

Use this property to enable SSL security between the application server and the SOAP client. Set the property by using one of the following options:

- Scripts that run from a command-line interface.
- The `soap.client.props` file.

Property	com.ibm.SOAP.securityEnabled
Data type	Boolean
Default	False

- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	securityEnabled
Data type	Boolean
Default	False

- A Java administrative client. Use the AdminClient.CONNECTOR_SECURITY_ENABLED property.

SSL alias

This property specifies the alias to use for an SSL configuration for client connections. The value of the alias is what you want it to be.

Set the property in the `soap.client.props` file.

Property	com.ibm.ssl.alias
Data type	String
Default	DefaultSSLSettings

IPC connector properties

This section discusses JMX connector properties that pertain to IPC connectors.

IPC request timeout

The value that you choose depends on a number of factors, such as the size and the number of the applications that are installed on the server, the speed of your machine, and the usage of your machine.

The program default value for the request timeout is `600 seconds`. However, other components that connect to the IPC client can override the default. Components that use the `ipc.client.props` file have a default value of `180 seconds`.

Set the property by using one of the following options:

- Scripts that run from a command-line interface.
- The `ipc.client.props` file.

Property	com.ibm.IPC.requestTimeout
Data type	Integer
Range in seconds	0 to n If the property is zero (0), the request never times out.
Default	180

- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	requestTimeout
Data type	Integer
Range in seconds	0 to n If the property is zero (0), the request never times out.
Default	600

- A Java administrative client. The property is AdminClient.CONNECTOR_IPC_REQUEST_TIMEOUT.

Configuration URL

Specify the configuration URL property if you want a program to read IPC properties from this file. You can set the property by using one of the following options:

- Scripts run from a command-line interface. Scripts can pass the Configuration URL property to the Application Server on the com.ibm.IPC.ConfigURL system property.
- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	ConfigURL
Data type	String
Valid Value	http://*Path*/ipc.client.props
Default	None

- A Java administrative client. Use the AdminClient.CONNECTOR_IPC_CONFIG property.

Security context provider

This property indicates the SSL implementation to use between the application server and the IPC client.

Set the property by using the `ipc.client.props` file.

Property	com.ibm.ssl.contextProvider
Data type	String
Valid Values	`IBMJSSE2`
Default	IBMJSSE2

Secure Sockets Layer (SSL) security

Use this property to enable SSL security between Application Server and the IPC client. Set the property by using one of the following options:

- Scripts that run from a command-line interface.
- The `ipc.client.props` file.

Property	com.ibm.IPC.securityEnabled
Data type	Boolean
Default	False

- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	securityEnabled
Data type	Boolean
Default	False

- A Java administrative client. Use the AdminClient.CONNECTOR_SECURITY_ENABLED property.

SSL alias

This property specifies the alias to use for an SSL configuration for client connections. The value of the alias is what you want it to be.

Set the property in the `ipc.client.props` file.

Property	com.ibm.ssl.alias
Data type	String
Default	DefaultSSLSettings

SOAP, RMI, JSR160RMI, and IPC connector properties

This section discusses JMX connector properties that pertain to SOAP connectors, RMI connectors, JSR160RMI connectors, and IPC connectors.

Connector type

A connector type of SOAP, RMI, JSR160RMI, or IPC depends on whether the application server connects to a SOAP server, an RMI server, a JSR160RMI server, or an IPC server. You can set the property by using one of the following options:

- The wsadmin tool.
- Scripts that run from a command-line interface.
- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	Type

Data type	String		

Valid values	`SOAPConnector`
	`RMIConnector`
	`JSR160RMIConnector`
	`IPCConnector`		

Default	SOAPConnector	JSR160RMI	IPC

- A Java administrative client. Use the AdminClient.CONNECTOR_TYPE property. Specify the connector type by using the AdminClient.CONNECTOR_TYPE_RMI, the AdminClient.CONNECTOR_TYPE_SOAP, the AdminClient.CONNECTOR_TYPE_JSR160RMI, or the AdminClient.CONNECTOR_TYPE_IPC constants.

Host

The host name or the IP address of the server to which the application server connects. The server can be a SOAP server, an RMI server, a JSR160RMI server, or an IPC server. You can set the property by using one of the following options:

- The wsadmin tool.
- Scripts that run from a command-line interface.
- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	host
Data type	String
Valid values	Host name or IP address
Default	None

- A Java administrative client. Use the AdminClient.CONNECTOR_HOST property.

Port

The port number of the server to which the application server connects. The server can be a SOAP server, an RMI server, a JSR160RMI server, or an IPC server. You can set the property by using one of the following options:

- The wsadmin tool.
- Scripts run from a command-line interface.
- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	port
Data type	Integer
Valid value	Port number
Default	None

- A Java administrative client. Use the AdminClient.CONNECTOR_PORT property.

User name

The user name that the application server uses to access the SOAP server, an RMI server, a JSR160RMI server, or an IPC server. You can set the property by using one of the following options:

- The wsadmin tool.
- Scripts run from a command-line interface.

- The `soap.client.props` file for the SOAP server, an RMI server, a JSR160RMI server.

Property	com.ibm.SOAP.loginUserid
Data type	String
Valid value	The value must match the global SSL settings for SOAP, RMI, or JSR160RMI.
Default	None

- The `ipc.client.props` file for the IPC server.

Property	com.ibm.IPC.loginUserid
Data type	String
Valid value	The value must match the global SSL settings for IPC.
Default	None

- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	username
Data type	String
Valid value	The value must match the global SSL settings for SOAP, RMI, JSR160RMI, or IPC.
Default	None

- A Java administrative client. Use the AdminClient.USERNAME property.

Password

The password that the application server uses to access the SOAP server, the RMI server, the JSR160RMI server, or the IPC server. You can set the property by using one of the following options:

- The wsadmin tool.
- Scripts run from a command-line interface.
- The `soap.client.props` file for the SOAP server, the RMI server, or the JSR160RMI server.

Property	com.ibm.SOAP.loginPassword
Data type	String
Valid values	The value must match the global SSL settings for SOAP, RMI, or JSR160RMI.
Default	None

- The `ipc.client.props` file for the IPC server.

Property	com.ibm.IPC.loginPassword
Data type	String
Valid values	The value must match the global SSL settings for IPC.
Default	None

- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	password
Data type	String
Valid values	The value must match the global SSL settings for SOAP, RMI, JSR160RMI, or IPC.
Default	None

- A Java administrative client. Use the AdminClient.PASSWORD property.

Disabling a connector

You can enable or disable any of the JMX connectors from the administrative console.

- The wsadmin tool.
- The administrative console. Select the box next to the connector to enable the connector. Clear the box next to the connector to disable the connector.

Property	enabled
Data type	Boolean
Value	truelfalse

RMI connector properties

This section discusses JMX connector properties that pertain to RMI connectors.

Disabling the JSR 160 RMI connector

Support for JMX Remote application programming interface (JSR 160) is enabled by default so that you automatically receive specification-compliant JMX function. To disable the function for a particular server, set the property by using one of the following options:

- The wsadmin tool.
- The administrative console. Specify the property and the value as a name-value pair on the JMX connector custom properties panel of the administrative console.

Property	disableJDKJMXConnector
Data type	string
Value	true

SOAP connector and Inter-Process Communications connector properties files

Use the soap.client.props file to set properties for the SOAP connector and the ipc.client.props file to set properties for the Inter-Process Communications (IPC) connector. Most of the properties in the ipc.client.props file have corresponding properties in the soap.client.props file.

The SOAP connector properties file for a particular profile is at the following location:

- Windows *profile_root*\properties\soap.client.props
- Linux HP-UX Solaris AIX *profile_root*/properties/soap.client.props

The IPC connector properties file for a particular profile is at the following location:

- Windows *profile_root*\properties\ipc.client.props
- Linux HP-UX Solaris AIX *profile_root*/properties/ipc.client.props

The following table provides basic information on the various properties. Read the properties files to obtain more detailed information.

Some properties are split on multiple lines for printing purposes.

SOAP connector properties	IPC connector properties	Description
com.ibm.SOAP. securityEnabled	com.ibm.IPC. securityEnabled	Specifies enablement of security for the connector. Set the property to true to enable security.
com.ibm.SOAP. authenticationTarget	com.ibm.IPC. authenticationTarget	Specifies the type of authentication for the connector if security is enabled. You can specify <codeph>BasicAuth</codeph> for basic authentication. If no value is specified, basic authentication is used.
com.ibm.SOAP.loginUserid	com.ibm.IPC.loginUserid	Specifies the user ID for the connector if security is enabled, and you do not enter a user ID through a command prompt or standard in.
com.ibm.SOAP.loginPassword	com.ibm.IPC.loginPassword	Specifies the password for the connector if security is enabled, and you do not enter a password through a command prompt or standard in.
com.ibm.SOAP.loginSource	com.ibm.IPC.loginSource	Specifies automatic prompting for the user ID and password when you specify prompt. Prerequisites for using this property are discussed in the properties file for the particular connector.
com.ibm.SOAP. requestTimeout	com.ibm.IPC. requestTimeout	Specifies how long in seconds the connector waits for a server response. The property for the SOAP connector and the property for the IPC connector are each initially set to 180 in their respective properties files.
com.ibm.ssl.alias	com.ibm.ssl.alias	This property specifies the alias to use for a Secure Sockets Layer (SSL) configuration for client connections. The value of the alias is what you want it to be.
	timeToExpiration	Specifies the time in seconds that connections can be idle in the connection pool. Beyond this time the connections are purged. The initial setting for the property is 360.

Java Management Extensions connectors

Use this page to view and change the configuration for Java Management Extensions (JMX) connectors, which make connections between server processes.

To view this administrative console page, click one of the following paths:
- **Servers > Application Servers >** *server_name* **> Administration > Administration Services > JMX Connectors**
- **Servers > JMS Servers >** *server_name* **> Administration > Administration Services > JMX Connectors**

Java Management Extensions (JMX) connectors communicate with WebSphere Application Server when you invoke a scripting process. There is no default for the type and parameters of a connector. The `wsadmin.properties` file specifies the Simple Object Access Protocol (SOAP) connector and an appropriate port number. You can also use the Remote Method Invocation (RMI) connector, the JMX Remote application programming interface (JSR 160) Remote Method Invocation (JSR160RMI) connector, or the Inter-Process Communications (IPC) connector.

Use one of the following methods to select the connector type and attributes:
- Specify properties in a properties file.
- Indicate options on the command line.

Type

Specifies the type of the JMX connector.

Data type Enumeration

Default SOAPConnector

Range

SOAPConnector
For JMX connections using Simple Object Access Protocol (SOAP).

RMIConnector
For JMX connections using Remote Method Invocation (RMI).

JSR160RMIConnector
For JMX connections using JMX Remote application programming interface (JSR 160) Remote Method Invocation (JSR160RMI).

IPCConnector
For JMX connections using Inter-Process Communications (IPC).

Enabled

Specifies whether a JMX connector is enabled. If `Yes` is specified, the connector is enabled. All JMX connectors are enabled by default.

Data type Boolean

JMX connector settings

Use this page to view the configuration for a Java Management Extensions (JMX) connector, which makes connections between server processes.

To view this administrative console page, click one of the following paths:
- **Servers > Application Servers >** *server_name* **> Administration > Administration Services > JMX Connectors >** *connector_type*

Type

Specifies the type of the JMX connector.

Data type Enumeration

Default SOAPConnector

Range

SOAPConnector
For JMX connections using Simple Object Access Protocol (SOAP).

RMIConnector
For JMX connections using Remote Method Invocation (RMI).

JSR160RMIConnector
For JMX connections using JMX Remote application programming interface (JSR 160) Remote Method Invocation (JSR160RMI).

IPCConnector
For JMX connections using Inter-Process Communications (IPC).

Repository service settings

Use this page to view and change the configuration for an administrative service repository.

To view this administrative console page, click **Servers > Application Servers >** *server_name* **Administration > Administration Services > Repository Service**.

Audit Enabled

Specifies whether to audit repository updates in the log file. The default is to audit repository updates.

Data type Boolean
Default true

Chapter 4. Working with server configuration files

This topic show how to manage application server configuration files.

About this task

Application server configuration files define the available application servers, their configurations, and their contents.

A configuration repository stores configuration data.

By default, configuration repositories reside in the *config* subdirectory of the profile root directory.

A cell-level repository stores configuration data for the entire cell and is managed by a file repository service that runs in the deployment manager. The deployment manager and each node have their own repositories. A node-level repository stores configuration data that is needed by processes on that node and is accessed by the node agent and application servers on that node.

When you change a WebSphere Application Server configuration by creating an application server, installing an application, changing a variable definition or the like, and then save the changes, the cell-level repository is updated. The file synchronization service distributes the changes to the appropriate nodes.

You should periodically save changes to your administrative configuration. You can change the default locations of configuration files, as needed.

- Edit configuration files.

  The master repository is comprised of .xml configuration files

  You can edit configuration files using
  - The administrative console. See the Using the administrative console topic in the *Using the administrative clients* PDF.
  - Scripting. See the Getting started with scripting topic in the *Using the administrative clients* PDF.
  - The wsadmin commands. See the Using command line tools topic in the *Using the administrative clients* PDF.
  - Programing. See the Using administrative programs (JMX) topic in the *Using the administrative clients* PDF.
  - By editing a configuration file directly.
- Save changes made to configuration files. Using the console, you can save changes as follows:
  1. In the navigation select **System Administration > Save changes to master repository**.
  2. Put a check mark in the **Synchronize changes with Nodes** check box.
  3. Click **Save**.
- Handle temporary configuration files resulting from a session timing out.
- Change the location of temporary configuration files.
- Change the location of backed-up configuration files.
- Change the location of temporary workspace files.
- Back up and restore configurations.

Configuration documents

WebSphere Application Server stores configuration data in several documents in a cascading hierarchy of directories. Most configuration documents have XML content.

The configuration documents describe the available application servers, their configurations, and their contents.

Hierarchy of directories of documents

The cascading hierarchy of directories and the documents' structure support multinode replication to synchronize the activities of all servers in a cell. In a Network Deployment environment, changes made to configuration documents in the cell repository, are automatically replicated to the same configuration documents that are stored on nodes throughout the cell.

At the top of the hierarchy is the **cells** directory. It holds a subdirectory for each cell. The names of the cell subdirectories match the names of the cells. For example, a cell named *cell1* has its configuration documents in the subdirectory *cell1*. The name of the cell must be different from the cluster name pair.

On the Network Deployment node, the subdirectories under the cell contain the entire set of documents for every node and server throughout the cell. On other nodes, the set of documents is limited to what applies to that specific node. If a configuration document only applies to *node1*, then that document exists in the configuration on *node1* and in the Network Deployment configuration, but not on any other node in the cell.

Each cell subdirectory has the following files and subdirectories:

- The `cell.xml` file, which provides configuration data for the cell
- Files such as `security.xml`, `virtualhosts.xml`, `resources.xml`, and `variables.xml`, which provide configuration data that applies across every node in the cell
- The **clusters** subdirectory, which holds a subdirectory for each cluster defined in the cell. The names of the subdirectories under clusters match the names of the clusters.

  Each cluster subdirectory holds a cluster.xml file, which provides configuration data specifically for that cluster.
- The **nodes** subdirectory, which holds a subdirectory for each node in the cell. The names of the nodes subdirectories match the names of the nodes.

  Each node subdirectory holds files such as `variables.xml` and `resources.xml`, which provide configuration data that applies across the node. Note that these files have the same name as those in the containing cell's directory. The configurations specified in these node documents override the configurations specified in cell documents having the same name. For example, if a particular variable is in both cell- and node-level `variables.xml` files, all servers on the node use the variable definition in the node document and ignore the definition in the cell document.

  Each node subdirectory holds a subdirectory for each server defined on the node. The names of the subdirectories match the names of the servers. Each server subdirectory holds a `server.xml` file, which provides configuration data specific to that server. Server subdirectories might hold files such as `security.xml`, `resources.xml` and `variables.xml`, which provide configuration data that applies only to the server. The configurations specified in these server documents override the configurations specified in containing cell and node documents having the same name.
- The **applications** subdirectory, which holds a subdirectory for each application deployed in the cell. The names of the applications subdirectories match the names of the deployed applications.

  Each deployed application subdirectory holds a `deployment.xml` file that contains configuration data on the application deployment. Each subdirectory also holds a **META-INF** subdirectory that holds a Java 2 Platform, Enterprise Edition (J2EE) application deployment descriptor file as well as IBM deployment extensions files and bindings files. Deployed application subdirectories also hold subdirectories for all .war and entity bean .jar files in the application. Binary files such as .jar files are also part of the configuration structure.

An example file structure is as follows:

```
cells
  cell1
     cell.xml resources.xml virtualhosts.xml variables.xml security.xml
     nodes
        nodeX
           node.xml variables.xml resources.xml serverindex.xml
           serverA
              server.xml variables.xml
           nodeAgent
              server.xml variables.xml
        nodeY
           node.xml variables.xml resources.xml serverindex.xml
     applications
        sampleApp1
           deployment.xml
           META-INF
              application.xml ibm-application-ext.xml ibm-application-bnd.xml
        sampleApp2
           deployment.xml
           META-INF
              application.xml ibm-application-ext.xml ibm-application-bnd.xml
cells
  cell1
     cell.xml resources.xml virtualhosts.xml variables.xml security.xml
     nodes
        nodeX
           node.xml variables.xml resources.xml serverindex.xml
           serverA
              server.xml variables.xml
     applications
        sampleApp1
           deployment.xml
           META-INF
              application.xml ibm-application-ext.xml ibm-application-bnd.xml
        sampleApp2
           deployment.xml
           META-INF
              application.xml ibm-application-ext.xml ibm-application-bnd.xml
```

Changing configuration documents

You can use one of the administrative tools (console, wsadmin, Java APIs) to modify configuration documents or edit them directly. It is preferable to use the administrative console because it validates changes made to configurations. ″“Configuration document descriptions” on page 158″ states whether you can edit a document using the administrative tools or must edit it directly.

Transformation of configuration files

The WebSphere Application Server master configuration repository stores configuration files for all the nodes in the cell. When you upgrade the deployment manager from one release of WebSphere Application Server to another, the configuration files that are stored in the master repository for the nodes on the old release are converted into the format of the new release.

With this conversion, the deployment manager can process the configuration files uniformly. However, nodes on an old release cannot readily use configuration files that are in the format of the new release. WebSphere Application Server addresses the problem when it synchronizes the configuration files from the master repository to a node on an old release. The configuration files are first transformed into the old release format before they ship to the node. WebSphere Application Server performs the following transformations on configuration documents:

- Changes the XML name space from the format of the new release to the format of the old release

- Strips out attributes of cell-level documents that are applicable to the new release only
- Strips out new resource definitions that are not understood by old release nodes

Configuration document descriptions

Most configuration documents have XML content. The table describes the documents and states whether you can edit them using an administrative tool or must edit them directly.

If possible, edit a configuration document using the administrative console because it validates any changes that you make to configurations. You can also use one of the other administrative tools (wsadmin or Java APIs) to modify configuration documents. Using the administrative console or wsadmin scripting to update configurations is less error prone and likely quicker and easier than other methods.

However, you cannot edit some files using the administrative tools. Configuration files that you must edit manually have an X in the **Manual editing required** column in the table below.

Document descriptions

(The paths in the Locations column are split on multiple lines for publishing purposes.)

Configuration file	Locations	Purpose	Manual editing required
admin-authz.xml	config/cells/ *cell_name*/	Define a role for administrative operation authorization.	
app.policy	config/cells/ *cell_name*/ nodes/*node_name*/	Define security permissions for application code.	X
cell.xml	config/cells/ *cell_name*/	Identify a cell.	
cluster.xml	config/cells/ *cell_name*/ clusters/ *cluster_name*/	Identify a cluster and its members and weights. This file is only available with the Network Deployment product.	
deployment.xml	config/cells/ *cell_name*/ applications/ *application_name*/	Configure application deployment settings such as target servers and application-specific server configuration.	
filter.policy	config/cells/ *cell_name*/	Specify security permissions to be filtered out of other policy files.	X
integral-jms-authorizations.xml	config/cells/ *cell_name*/	Provide security configuration data for the integrated messaging system.	X
library.policy	config/cells/ *cell_name*/ nodes/*node_name*/	Define security permissions for shared library code.	X
multibroker.xml	config/cells/ *cell_name*/	Configure a data replication message broker.	
namestore.xml	config/cells/ *cell_name*/	Provide persistent name binding data.	X

naming-authz.xml	config/cells/*cell_name*/	Define roles for a naming operation authorization.	X
node.xml	config/cells/*cell_name*/nodes/*node_name*/	Identify a node.	
pmirm.xml	config/cells/*cell_name*/	Configure PMI request metrics.	X
resources.xml	config/cells/*cell_name*/ config/cells/*cell_name*/nodes/*node_name*/ config/cells/*cell_name*/nodes/*node_name*/servers/*server_name*/	Define operating environment resources, including JDBC, JMS, JavaMail, URL, JCA resource providers and factories.	
security.xml	config/cells/*cell_name*/	Configure security, including all user ID and password data.	
server.xml	config/cells/*cell_name*/nodes/*node_name*/servers/*server_name*/	Identify a server and its components.	
serverindex.xml	config/cells/*cell_name*/nodes/*node_name*/	Specify communication ports used on a specific node.	
spi.policy	config/cells/*cell_name*/nodes/*node_name*/	Define security permissions for service provider libraries such as resource providers.	X
variables.xml	config/cells/*cell_name*/ config/cells/*cell_name*/nodes/*node_name*/ config/cells/*cell_name*/nodes/*node_name*/servers/*server_name*/	Configure variables used to parameterize any part of the configuration settings.	
virtualhosts.xml	config/cells/*cell_name*/	Configure a virtual host and its MIME types.	

Object names: What the name string cannot contain

When you create a new object using the administrative console or a wsadmin command, you often must specify a string for a name attribute.

Most characters are allowed in the name string. However, the name string cannot contain the following characters. The name string also cannot contain leading and trailing spaces.

/	forward slash
****	backslash
*****	asterisk
,	comma
:	colon
;	semi-colon
=	equal sign
+	plus sign
?	question mark
|	vertical bar
<	left angle bracket
>	right angle bracket
&	ampersand (and sign)
%	percent sign
'	single quote mark
"	double quote mark
]]>	No specific name exists for this character combination.
.	period (not valid if first character; valid if a later character)
#	Hash mark
$	Dollar sign
~	Tilde

Handling temporary configuration files resulting from session timeout

If the console is not used for 15 minutes or more, the session times out. The same thing happens if you close the browser window without saving the configuration file. Changes to the file are saved to a temporary file when the session times out, after 15 minutes. This topic discusses what happens depending on whether you load the saved file.

Before you begin

A configuration file must have been saved from a previous administrative console session for the user ID that you are currently using to access the administrative console.

About this task

When a session times out, the configuration file in use is saved under the `userid/timeout` directory under the ServletContext's temp area. This value is the value of the javax.servlet.context.tempdir attribute of the ServletContext context. By default, it is: *profile_root*`/temp/hostname/Administration/admin/admin.war`

You can change the temp area by specifying it as a value for the tempDir init-param of the action servlet in the deployment descriptor (web.xml) of the administrative application.

The configuration file is also saved automatically when the same user ID logs into the non-secured console again, effectively starting a different session. This process is equivalent to forcing the existing user ID out of session, similar to a session timing out.

The next time you log on to the administrative console, you are prompted to load the saved configuration file. Do one of the following actions:

- Load the saved file.
  1. If a file with the same name exists in the *profile_root*`/config` directory, that file is moved to the `userid/backup` directory in the temp area.

2. The saved file is moved to the *profile_root*/config directory.
3. The file is then loaded.

- Do not load the saved file.

  The saved file is deleted from the userid/timeout directory in the temp area.

Results

You loaded the saved configuration file if you chose to do so.

What to do next

Once you have logged into the administrative console, do whatever administration of WebSphere Application Server that you need to do.

Changing the location of temporary configuration files

You can change the default directory where temporary configuration files are stored.

About this task

The configuration repository uses copies of configuration files and temporary files while processing repository requests. It also uses a backup directory while managing the configuration. You can change the default locations of these files from the configuration directory to a directory of your choice by using the administrative console.

The default location for the configuration temporary directory is *profile_root*/config/temp. Use the administrative console to change the location of the temporary repository file location for all types of server processes. For example, to change the setting for Application Server, do the following steps:

1. Click **Servers > Application servers** in the navigation tree of the administrative console. Then, click ***server name*** **> Administration > Administration services > Repository service > Custom properties**.
2. On the Properties page, click **New**.
3. On the settings page for a property, define a property for the temporary file location. The key for this property is was.repository.temp. The value is the full path name to the desired location.
4. Click **OK**.

Changing the location of backed-up configuration files

You can change the default directory where backup files are stored.

About this task

During administrative processes like adding a node to a cell or updating a file, configuration files are temporarily backed up to a backup location.

The default location for the backup configuration directory is *profile_root*/config/backup. Use the administrative console to change the location of the repository backup directory for all types of server processes. For example, to change the setting for Application Server, do the following steps:

1. Click **Servers > Application servers** in the navigation tree of the administrative console. Then, click ***server name*** **> Administration > Administration services > Repository service > Custom properties**.
2. On the Properties page, click **New**.
3. On the settings page for a property, define a property for the backup file location. The key for this property is was.repository.backup. The value is the full path name to the desired location.

4. Click **OK**.

Changing the location of the wstemp temporary workspace directory

Configuration changes are stored in the wstemp temporary workspace directory until the changes are merged with the master configuration repository. This topic discusses how to change the location of the wstemp temporary workspace directory.

Before you begin

You must first install WebSphere Application Server before you change the location of the wstemp directory, which is a temporary workspace directory.

About this task

Whenever a user logs into the administrative console, or uses wsadmin scripting to make a configuration change, the changes are stored in the workspace. When a user uses the ConfigService configuration service interface of the Java application programming interfaces (APIs), the user specifies a session object that is associated with the workspace in order to store the changes. Only when the user performs a save operation under the administrative console, wsadmin scripting, or the Java APIs are the changes propagated and merged with the master configuration repository. For each administrative console user or each invocation of wsadmin scripting, the application server creates a separate workspace directory to store the intermediate changes until the changes are merged with the master configuration repository. Users of the Java APIs use different session objects to decide where the workspace directory resides. Both the administrative console and wsadmin scripting generate user IDs randomly. The user IDs are different from the user IDs that you use to log into the administrative console or wsadmin scripting. The Java APIs can either randomly generate the user ID or specify the user ID as an option when creating the session object.

You might want to change the location of the wstemp directory if you want to keep it in a separate place from the product installation.

The product determines the location of the workspace in the following order by using the first Java Virtual Machine (JVM) property in the list that is set. If no JVM property is set, the product uses the default workspace location.

JVM System Property	Location	Comments

websphere.workspace.root	The wstemp directory location is the value of the websphere.workspace.root JVM system property plus
	• Linux HP-UX Solaris AIX /wstemp
	• Windows \wstemp
	For example, the websphere.workspace.root JVM system property and its value could be
	• Linux HP-UX Solaris AIX
	`-Dwebsphere.workspace.root =/temp`
	• Windows
	`-Dwebsphere.workspace.root =c:\temp`
	The property and its value are split on multiple lines for printing purposes.
		Set the JVM system property for the deployment manager to change the wstemp directory location. Use the full path rather than a relative path for this property.

If the websphere.workspace.root property is not set, the value of the user.install.root property is used.	The default wstemp location is the value of the user.install.root JVM system property plus • Linux HP-UX Solaris AIX /wstemp • Windows \wstemp	Do not change the user.install.root property as the profile creation process sets this property by pointing to the *profile_root* directory. In this case, the wstemp location is: • Linux HP-UX Solaris AIX *profile_root*/wstemp • Windows *profile_root*\wstemp

- Change the workspace location for a particular JVM property by setting the -D option on the **java** command.

  This method of changing the workspace location is only needed when you run a standalone administrative program in local mode.

  For example, use the following option:

  `-Dwebsphere.workspace.root=`*`the location of the new workspace directory`*
- Change the JVM custom property through the administrative console by setting the JVM property as a name-value pair on the Custom properties page.

  For example,

  1. Click **Deployment Manager > Java and Process Management > Process Definition > Java Virtual Machine > Custom Properties**.
  2. Click **New**.
  3. Specify `websphere.workspace.root` as the name.
  4. Specify the full path of the new workspace directory as the value. The wstemp directory is created under that path.
  5. Stop the server.

     This step is optional if you want to keep your existing workspace files.

6. Copy files from the old location of the workspace directory to the new location of the workspace directory.

   This step is optional if you want to keep your existing workspace files.
7. Start the server.

   This step is optional if you want to keep your existing workspace files.

Results

You have used either the administrative console or the -D option on the **java** command to change the location of the wstemp temporary workspace directory.

Backing up and restoring administrative configuration files

This topic discusses how to back up and restore administrative configuration files.

About this task

WebSphere Application Server represents its administrative configurations as XML files. You should back up configuration files on a regular basis.

Restore the configuration only if the configuration files that you backed up are at the same level of the release, including fixes, as the release to which you are restoring.

1. Synchronize administrative configuration files.
   a. Click **System Administration > Nodes** in the console navigation tree to access the Nodes page.
   b. Click **Full Resynchronize**. The resynchronize operation resolves conflicts among configuration files and can take several minutes to run.
2. Run the backupConfig command to back up configuration files. See the backupConfig command topic in the *Using the administrative clients* PDF for information.
3. Run the restoreConfig command to restore configuration files. See the restoreConfig command topic in the *Using the administrative clients* PDF for information. Specify backup files that do not contain invalid or inconsistent configurations.

Server configuration files: Resources for learning

Use the following links to find relevant supplemental information about administering WebSphere Application Server configuration files. The information resides on IBM and non-IBM Internet sites, whose sponsors control the technical accuracy of the information.

These links are provided for convenience. Often, the information is not specific to the IBM WebSphere Application Server product, but is useful all or in part for understanding the product. When possible, links are provided to technical papers and IBM Redbooks that supplement the broad coverage of the release documentation with in-depth examinations of particular product areas.

View links to additional information:

Administration

- IBM WebSphere Application Server Redbooks

  This site contains a listing of all WebSphere Application Server Redbooks.
- IBM WebSphere developerWorks

  This site is the home of technical information for developers working with WebSphere products. You can download WebSphere software, take a fast path to developerWorks zones, such as VisualAge Java or WebSphere Application Server, learn about WebSphere products through a newcomers page, tutorials,

technology previews, training, and Redbooks, get answers to questions about WebSphere products, and join the WebSphere community, where you can keep up with the latest developments and technical papers.

- WebSphere Application Server Support page

  Take advantage of the Web-based Support and Service resources from IBM to quickly find answers to your technical questions. You can easily access this extensive Web-based support through the IBM Software Support portal at URL `http://www.ibm.com/software/support/` and search by product category, or by product name. For example, if you are experiencing problems specific to WebSphere Application Server, click **WebSphere Application Server** in the product list. The WebSphere Application Server Support page appears.

Chapter 5. Administering application servers

An application server configuration provides settings that control how an application server provides services for running applications and their components.

About this task

After you install the product, you might have to perform one or more of the following tasks. Unless the task you want to perform is dependent on the existence of an application server, you can perform these tasks in any order.

- Create an application server.
- Create clusters for workload balancing.
- Configure the server startup process such that only server components that are initially needed are started.

  When the server is configured such that only the components that are initially needed are started during the startup process, the remaining components are dynamically started as they are needed.

  Note: If you are running other WebSphere products on top of this product, make sure that those other products support this functionality before you select this property.
- Configure transport chains to handle client requests.
- Develop custom services.
- Define processes for the application server.
- Configure the Java virtual machine.

Results

Any new application servers you create are displayed in the list of servers on the administrative console Application servers page.

What to do next

- Manage your application servers. Any newly created application servers are configured with many default settings that do not display when you run the Create New Application Server wizard. You might need to change some of these settings to better fit the needs of your environment.
- Deploy an application or component on the application server.
- View the status of the applications running on the application server.

Virtual hosts

A virtual host is a configuration entity that enables a single host machine to resemble multiple host machines. It maintains a list of Multipurpose Internet Mail Extensions (MIME) types that it processes. You can associate a virtual host to one or more Web modules, but you can associate each Web module with one and only one virtual host. Resources associated with one virtual host cannot share data with resources associated with another virtual host, even if the virtual hosts share the same physical machine.

Each virtual host has a logical name and a list of one or more DNS aliases by which it is known. A DNS alias is the TCP/IP hostname and port number that is used to request the servlet, for example `yourHostName:80`. When no port number is specified, 80 is assumed.

The virtual host configuration uses wildcard entries with the ports for its virtual host entries.

- The default alias is *:80, using an external port that is not secure.
- Aliases of the form *:9080 use the internal port that is not secure.
- Aliases of the form *:9443 use the secure internal port.

- Aliases of the form *:443 use the secure external port.

A client request for a servlet, JavaServer Pages file, or related resource contains a DNS alias and a Uniform Resource Indicator (URI) that is unique to that resource. When a client request for a servlet, JavaServer Pages file, or related resource is received, the DNS alias is compared to the list of all known virtual host groups to locate the correct virtual host, and the URI is compared to the list of all known URI groups to locate the correct URI group. If the virtual host group and URI group are found, the request is sent to the corresponding server group for processing and a response is returned to browser. If a matching virtual host group or URI group is not found, an error is returned to the browser.

A virtual host is not associated with a particular node (machine). It is a configuration, rather than a live object, which is why you can create it, but cannot start or stop it. A default virtual host, named default_host, is automatically configured the first time you start an application server. Unless you specifically want to isolate resources from one another on the same node (physical machine), you probably do not need any virtual hosts in addition to the default host.

The DNS aliases for the default virtual host are configured as `*:80` and `*:9080`, where port 80 is the HTTP server port and port 9080 is the port for the default server's HTTP transport. The default virtual host includes common aliases, such as the machine's IP address, short host name, and fully qualified host name. One of these aliases comprises the first part of the path for accessing a resource such as a servlet. For example, the alias `localhost:80` is used in the request `http://localhost:80/myServlet`.

Adding a localhost to the virtual hosts adds the host name and IP address of the localhost machine to the alias table. This allows a remote user to access the administrative console.

You can use the administrative console to add or change DNS aliases if you want to use ports other than the default ports. If you do make a change to a DNS alias, you must regenerate the Web server plug-in configuration. You can use the administrative console to initiate the plug-in regeneration.

Note: You might want to add additional aliases or change the default aliases if:

- The HTTP server instance is running on a port other than 80. Add the correct port number to each of the aliases. For example, change `yourhost` to `yourhost:8000`.
- You want to make HTTPS requests, which use Secure Sockets Layer (SSL). To make HTTPS requests you must add port 443 to each of the aliases. Port 443 is the default port for SSL requests.
- Your Web server instance is listening for SSL requests on a port other than 443. In this situation, you must add that port number to each of the aliases.
- You want to use a port other then default port (9080) for the application server.
- You want to use other aliases that are not listed.

When you request a resource, the product tries to map the request to an alias of a defined virtual host. The `http://host:port/` portion of the virtual host is not case sensitive, but the URL that follows is case sensitive. The match for the URL must be alphanumerically exact. Different port numbers are treated as different aliases.

For example, the request `http://www.myhost.com/myservlet` maps successfully to `http://WWW.MYHOST.COM/myservlet` but not to `http://WWW.MYHOST.COM/MYSERVLET` or `Www.Myhost.Com/Myservlet`. In the latter two cases, these mappings fail because of case sensitivity. The request `http://www.myhost.com/myservlet` does not map successfully to `http://myhost/myservlet` or to `http://myhost:9876/myservlet`. These mappings fail because they are not alphanumerically correct.

You can use wildcard entries for aliases by port and specify that all valid host name and address combinations on a particular port map to a particular virtual host.

If you request a resource using an alias that cannot be mapped to an alias of a defined virtual host, you receive a 404 error in the browser that you used to issue the request. A message states that the virtual host could not be found.

Two sets of associations occur for virtual hosts. Application deployment associates an application with a virtual host. Virtual host definitions associate the network address of the machine and the HTTP transport or Web server port assignment of the application server with the virtual host. Looking at the flow from the Web client request for the snoop servlet, for example, the following actions occur:

1. The Web client asks for the snoop servlet: at Web address `http://www.some_host.some_company.com:9080/snoop`
2. The some_host machine has the 9080 port assigned to the standalone application server, server1.
3. server1 looks at the virtual host assignments to determine the virtual host that is assigned to the alias `some_host.some_company.com:9080`.
4. The application server finds that no explicit alias for that DNS string exists. However, a wild card assignment for host name * at port 9080 does exist. This is a match. The virtual host that defines the match is default_host.
5. The application server looks at the applications deployed on the default_host and finds the snoop servlet.
6. The application server serves the application to the Web client and the requester is able to use the snoop servlet.

You can have any number of aliases for a virtual host. You can even have overlapping aliases, such as:

Virtual host	Alias	Port
default_host	*	9080
	localhost	9080
	my_machine	9080
	my_machine.my_company.com	9080
	localhost	80

The Application Server looks for a match using the explicit address specified on the Web client address. However, it might resolve the match to any other alias that matches the pattern before matching the explicit address. Simply defining an alias first in the list of aliases does not guarantee the search order whenever the product is looking for a matching alias.

A problem can occur if you use the same alias for two different virtual hosts. For example, assume that you installed the default application and the snoop servlet on the default_host. You also have another virtual host called the admin_host. However, you have not installed the default application or the snoop servlet on the admin_host.

Assume that you define overlapping aliases for both virtual hosts because you accidentally defined port 9080 for the admin_host instead of port 9060:

Virtual host	Alias	Port
default_host	*	9080
	localhost	9080
admin_host	*	9060
	my_machine.com	9080

Assume that a Web client request comes in for `http://my_machine.com:9080/snoop`.

If the application server matches the request against `*:9080`, the application is served from the default_host. If the application server matches the request to `my.machine.com:9080`, the application cannot be found. A 404 error occurs in the browser that issues the request. A message states that the virtual host could not be found.

This problem is the result of not finding the requested application in the first virtual host that has a matching alias. The correct way to code aliases is for the alias name on an incoming request to match only one virtual host in all of your virtual host definitions. If the URL can match more than one virtual host, you can see the problem just described.

Configuring virtual hosts

Virtual hosts let you manage a single application server on a single machine as if the application server were multiple application servers each on their own host machine. You can separate and control which resources are available for client requests by combining multiple host machines into a single virtual host, or by assigning host machines to different virtual hosts.

Before you begin

If your external HTTP server configuration uses the default port, 9080, you do not have to perform these steps.

About this task

Virtual hosts isolate and independently manage multiple sets of resources on the same physical machine. Resources associated with one virtual host cannot share data with resources associated with another virtual host. This is true even though the virtual hosts share the same application server on the same physical machine.

For example, suppose that:

- An Internet service provider (ISP) has two customers with Internet sites hosted on the same machine. The ISP keeps the two sites isolated from one another, despite their sharing a machine, by using virtual hosts. The ISP associates the resources of the first company with VirtualHost1 and the resources of the second company with VirtualHost2. Both virtual hosts map to the same application server.
- Both company sites offer the same servlet. Each site has its own instance of the servlet, and is unaware of the same servlet on the other site. If the company whose site is organized on VirtualHost2 is past due in paying its account with the ISP, the ISP can refuse all servlet requests that are routed to VirtualHost2. Even though the same servlet is available on VirtualHost1, the requests directed at VirtualHost2 do not go to the other virtual host.

Because the servlet is associated with a virtual host instead of the actual DNS address, The servlet on virtual host VirtualHost1 does not share its context with the servlet that has the same name on virtual host VirtualHost2. Requests for the servlet on VirtualHost1 can continue as usual, even though VirtualHost2 is refusing to fill requests for the servlet with the same name.

If any of the following conditions exist, you must update the HTTP port numbers associated with the default virtual host. or define a new virtual host and associate it with the ports your HTTP server configuration uses:

- Your external HTTP server configuration uses a port other than the default port of 9080, you must define the port that you are using.
- You are using the default HTTP port 9080, but the port is no longer defined. You must define port 9080.
- You have created multiple application servers as either stand-alone servers or cluster members, and these servers use the same virtual host. Because each server must be listening on a different port, you must define a virtual host alias for the HTTP port of each server.

If you define new virtual host aliases, identify the port values that the aliases use on the Host alias settings page in the administrative console.

Perform the following steps to create a new virtual host or change the configuration of an existing virtual host.

1. In the administrative console, click **Environment > Virtual hosts**.
2. Optional: Create a new virtual host. If you create a new virtual host, a default set of 90 MIME entries are automatically created for that virtual host.
   a. In the administrative console, click **New**.
   b. Enter the name of the new virtual host and click **OK**. The new virtual host appears in the list of virtual hosts you can configure.
3. Select the virtual host whose configuration you want to change.
4. Under Additional Properties, click **Host aliases**.
5. Create new host aliases or update existing host aliases to associate each of your HTTP port numbers with this virtual host.

   There must be a virtual host alias corresponding to each port your HTTP server configuration uses. There is one HTTP port associated with each Web container, and it is usually assigned to the virtual host named `default_host`. You can change the default assignment to any valid virtual host.

   The host aliases associated with the `default_host` virtual host are set to * when you install the product. The * (an asterisk) indicates that the alias name does not have to be specified or that any name can be specified.

   When the URL for the application is entered into a Web browser, the port number is included. For example, if 9082 is the port number, the specified URL might look like the following:

   ```
   http://localhost:9082/wlm/SimpleServlet
   ```

   To create a new host alias:

   a. Click **New**.
   b. Specify a host alias name in the Host Name field and one of your HTTP ports in the Port field.

      You can specify * (an asterisk) for the alias name if you do not want to require the specification of the alias name or if you want to allow any name to be specified.
   c. Click **OK** and **Save** to save your configuration change.

   To update an existing host alias:

   a. Select an existing host alias name.
   b. Change the value specified in the Port field to one of your HTTP ports.
   c. Click **OK** and **Save** to save your configuration change.
6. Optional: Define a MIME object type and its file name extension if you require a MIME type other than the pre-defined types.
   a. For each needed MIME entry on the MIME type collection page, click **New**.
   b. On the MIME type settings page, specify a MIME type and extension.
   c. Click **OK** and **Save** to save your configuration change.
7. Regenerate the Web server plug-in configuration.
   a. **Servers > Server Types > Web servers**, then select the appropriate Web server.
   b. Click **Generate pug-in**, then click **Propagate plug-in**.
8. Restart the application server.

Virtual host collection

Use this page to create and manage configurations that each let a single host machine resemble multiple host machines. Such configurations are known as *virtual hosts.*

To view this administrative console page, click **Environment > Virtual hosts**.

Each virtual host has a logical name (which you define on this panel) and is known by its list of one or more domain name system (DNS) aliases. A DNS alias is the TCP/IP host name and port number used to request the servlet, for example yourHostName:80. (Port 80 is the default.)

You define one or more alias associations by clicking an existing virtual host or by adding a new virtual host.

When a servlet request is made, the server name and port number entered into the browser are compared to a list of all known aliases in an effort to locate the correct virtual host to serve the servlet. No match returns an error to the browser.

An application server profile provides a default virtual host with some common aliases, such as the internet protocol (IP) address, the DNS short host name, and the DNS fully qualified host name. The alias comprises the first part of the path for accessing a resource such as a servlet.

For example, the alias is localhost:80 in the request http://localhost:80/myServlet.

A virtual host is not associated with a particular profile or node (machine), but is associated with a particular server instead. It is a configuration, rather than a ″live object.″ You can create a virtual host, but you cannot start or stop it.

For many users, creating virtual hosts is unnecessary because the default_host that is provided is sufficient.

Adding the host name and IP address of the localhost machine to the alias table lets a remote user access the administrative console.

Resources associated with one virtual host cannot share data with resources associated with another virtual host, even if the virtual hosts share the same physical machine.

Name

Specifies a logical name for configuring Web applications to a particular host name. The default virtual host is suitable for most simple configurations.

Virtual hosts enable you to isolate, and independently manage, multiple sets of resources on the same physical machine. Determine whether you need a virtual host alias for each port associated with an HTTP transport channel or an HTTP transport. There must be a virtual host alias corresponding to each port used by an HTTP transport channel or an HTTP transport. There is one HTTP transport channel or HTTP transport associated with each Web container, and there is one Web container in each application server.

When you create a virtual host, a default set of 90 MIME entries is created for the virtual host.

You must create a virtual host for each HTTP port in the following cases:

- You use the internal HTTP transport with a port other than the default value of 9080, or for some reason the virtual host does not contain the usual entry for port 9080.
- You create multiple application servers, such as stand-alone servers, managed servers, or cluster members, that are using the same virtual host. Because each server must be listening on a different HTTP port, you need a virtual host alias for the HTTP port of each server.

Virtual host settings

Use this page to configure a virtual host instance.

To view this administrative console page, click **Environment > WebSphere variables***virtual_host_name*.

Name:

Specifies a logical name for configuring Web applications to a particular host name. The default virtual host is suitable for most simple configurations.

Data type	String
Default	default_host

Host alias collection

Use this page to manage host name aliases defined for a virtual host. An alias is the DNS host name and port number that a client uses to form the URL request for a Web application resource.

To view this administrative console page, click **Environment > Virtual host***virtual_host_name* **> Host aliases**.

Host name:

Specifies the IP address, DNS host name with domain name suffix, or just the DNS host name, used by a client to request a Web application resource (such as a servlet, JavaServer Pages (JSP) file, or HTML page). For example, the host alias name is `myhost` in a DNS name of `myhost:8080`.

The product provides a default virtual host (named default_host). The virtual host configuration uses the wildcard character * (asterisk) along with the port number for its virtual host entries. Unless you specifically want to isolate resources from one another on the same node (physical machine), you probably do not need any virtual hosts in addition to the default host.

Port:

Specifies the port for which the Web server has been configured to accept client requests. For example, the port assignment is *8080* in a DNS name of `myhost:8080`. A URL refers to this DNS as: http://*myhost:8080*/servlet/snoop.

Host alias settings:

Use this page to view and configure a host alias.

To view this administrative console page, click **Environment > Virtual hosts >** *virtual_host_name* **> Host aliases >** *host_alias_name*.

Host name:

Specifies the IP address, domain name system (DNS) host name with domain name suffix, or the DNS host name that clients use to request a Web application resource, such as a servlet, JSP file, or HTML page.

For example, when the DNS name is `myhost`, the host alias is `myhost:8080`, where *8080* is the port. A URL request can refer to the snoop servlet on the host alias as: http://*myhost:8080*/servlet/snoop.

When there is no port number specified for a host alias, the default port is 80. For existing virtual hosts, the default host name and port reflect the values specified at product installation or configuration. For new virtual hosts, the default can be `*` to allow any value or no specification.

Data type	String

Default *

You can also use the IP address or the long or short DNS name.

Port:

Specifies the port where the Web server accepts client requests. Specify a port value in conjunction with the host name.

Specifies the port where the virtual host accepts Web client requests. The port number that you specify must be a unique in conjunction with the host name to avoid conflicts with other virtual hosts. The port number default is port 80, which is the default Web server port. You can assign another port number if you want to use the internal HTTP transport capability of the application server, or to use another port that you have designated as the Web server port. For example, you can create a new virtual host and assign port 9085 to that virtual host if you want to serve application resources over the internal HTTP transport of the application server that uses port 9085.

Data type Integer
Default 80

MIME type collection

Use this page to view and configure multi-purpose internet mail extensions (MIME) object types and their file name extensions.

The list shows a collection of MIME type extension mappings defined for the virtual host. Virtual host MIME entries apply when you do not specify MIME entries at the Web module level.

To view a list of current virtual host Mime types in the administrative console, click **Environment > Virtual hosts***virtual_host_name* **> Mime types**.

MIME type:

Specifies a MIME type, which can be application, audio, image, text, video, www, or x-world. An example value for MIME type is `text/html`.

Extensions:

Specifies file extensions of files that map the MIME type. Do not specify the period before the extension. Example extensions for a `text/html` MIME type are `htm` and `html`.

MIME type settings:

Use this page to configure a multi-purpose internet mail extensions (MIME) object type.

To view this administrative console page, click **Environment > Virtual hosts***virtual_host_name* **> Mine types >** *mime_type*.

MIME type:

Specifies a MIME type, which can be application, audio, image, text, video, www, or x-world. An example value for MIME type is `text/html`.

An example value for MIME type is `text/html`. A default value appears only if you are viewing the configuration for an existing instance.

Data type String

Extensions:

Specifies file extensions of files that map the MIME type. Do not specify the period before the extension. Example extensions for a `text/html` MIME type are `htm` and `html`.

File extensions for a `text/html` MIME type are `.htm` and `.html`. A default value appears only if you are viewing the configuration for an existing MIME type.

Data type String

Creating, editing, and deleting WebSphere variables

You can use WebSphere variables to provide settings for any of the string data type attributes that are contained in the product configuration files.

Before you begin

Because applications cannot directly access WebSphere variables, if you define a WebSphere variable inside of an application, an error message, such as ″Unknown variable,″ is returned. If you must reference a WebSphere variable from within an application, include the following method in the application to expand the string that uses the WebSphere variable:

```
private String expandVariable(String s) throws
javax.management.JMException {
com.ibm.websphere.management.AdminService as =
com.ibm.websphere.management.AdminServiceFactory.getAdminService
();

String server = as.getProcessName();

java.util.Set result = as.queryNames(new javax.management.ObjectName("*:*,type=AdminOperations,process="
+ server), null);

return (String)as.invoke((javax.management.ObjectName)
result.iterator().next(),"expandVariable",new Object[]
{"${"+s+"}"}, new String[] {"java.lang.String"});
```

About this task

WebSphere variables are usually used to specify file paths. The ″Variable settings″ topic supplies further details about specifying variables and highlights further details about product components that use them.

WebSphere variables are also used to configure:

- Product path names, such as JAVA_HOME, and APP_INSTALL_ROOT.
- Configure certain cell-wide or cluster-wide customization values.

The variable scoping mechanism for WebSphere variables enables you to define a variable at the node, cluster, or cell level, as well as at the server level. This mechanism enables you to specify a setting for all of the servers in a node, cluster, or cell, instead of individually specifying the setting for each server.

To define a new variable, change the value of an existing variable, or delete an existing variable complete the following steps, as appropriate.

1. Click **Environment > WebSphere variables** in the administrative console
2. Select the scope of the variable from the list of available scopes.

If you create a new variable, it will be created at the selected scope. If you define the same variable at multiple levels, the more granular definition overrides the higher level setting. For example, if you specify the same variable on a cell level and at a node level, the node level setting overrides the cell level setting.

Scoping variables is particularly important if you are testing data source objects. Variable scoping can cause a data source to fail the test connection, but to succeed at run time, or to pass the test connection, but fail at run time.

3. Create a new variable.
   a. Click **New**.
   b. Specify a name, a value, and, optionally, a description for the variable.

      You can create WebSphere variables that support substitution. For example, if you enter ${<*variable name*>} in the **Name** field, the value of <*variable name*> becomes the name of your new WebSphere variable. For example if you enter ${JAVA_HOME} as the name of your variable, the name of the WebSphere variable that is created is the Java home directory.
   c. Click **OK**.
   d. Click **Environment > WebSphere variables** in the administrative console navigation, and verify that the variable is displayed in the list of variables for the selected scope.

      The administrative console does not pick up typing errors. The variable is ignored if it is referred to incorrectly.
4. Modify the setting for an existing variable.
   a. Click on the name of the variable that you want to change.
   b. Modify the content of the Values field.

      The Values field for some of the variables that are already defined when you install the product are read-only because changing the values that are specified for those variables might cause product processing errors.
   c. Click **OK**.
5. Delete an existing variable.
   a. Select the variable that you want to delete.
   b. Click **Delete**.
   c. Click **OK**.
   d. Verify that this variable was removed from the list of variables for the selected scope.
6. Save your configuration.
7. Stop the affected servers and start those servers again to put the variable configuration change into effect.

   If the change you made affects a node, you must stop and restart all of the servers on that node. Similarly if the change you made affects a cell, you must stop and restart all of the servers in that cell.

WebSphere variables collection

Use this page to view and change the defined product variables with their values. You can also use this page to create a new variable, or delete an existing variable. These variables are name and value pairs that are used to provide the settings for the string data type configuration attributes that are contained in one of the XML formatted configuration files that reside in the product repository.

To view this administrative console page, click **Environment > WebSphere variables**.

To display a list of all of the variables that are defined for a specific scope, select that scope.

To view additional information about a specific variable, or to change the setting for that variable, click the variable name. Some of the pre-defined variables, that is, variables that already exist when you install the product, are set at values that are required for the product to function properly. The Value fields for these variables are read-only and cannot be edited.

To define a new variable, select the appropriate scope from the list of available options and then click **New**. The selected scope indicates the level at which the variable setting is visible.

To delete an existing variable, select the appropriate variable, and then click **Delete**. Do not delete any of the pre-defined variables. Before deleting a variable that you defined, make sure that none of your applications require the configuration attribute setting that the variable provides.

Name
Specifies the symbolic name for a WebSphere Application Server variable. For example, a variable name might represent a physical path or URL root used by WebSphere Application Server.

Value
Specifies the value that the symbolic name represents. For example, the value might be an absolute path value for a file or URL root.

Scope
Specifies the level at which a WebSphere variable is visible on the administrative console panel. The scope is specified when a new variable is defined.

A resource can be visible in the administrative console collection table at the node or server scope.

On a multiple-server product, a resource also can be visible at the cell or cluster scope. The cluster scope is only available if a cluster is defined for the cell.

WebSphere variables settings
Use this page to define the name and value of a WebSphere variable. A WebSphere variable is a name and value pair that is used to provide the setting for one of the string data type attributes contained in one of the XML formatted configuration files that reside in the product repository.

To view this administrative console page, click **Environment > WebSphere variables > *WebSphere_variable_name***.

Name:

Specifies the symbolic name for a product variable. After the variable is defined, this symbolic name can be specified in the **Value** field of any other product configuration field that accepts a string value. Whenever the application server encounters a configuration field that contains one or more symbolic names, it replaces the symbolic names with their defined values. For example, you might define a variable name that represent a commonly used file path or URL.

WebSphere Application Server variables are used for:

- Configuring WebSphere Application Server path names, such as *JAVA_HOME*, and APP_INSTALL_ROOT.
- Configuring certain cell-wide customization values.

For example, WAS_SERVER_NAME is the pre-defined symbolic name of the variable that represents the name of the default application server that is provided with the product..

Value:

Specifies the value that the symbolic name represents.

For example, server1 is the value of a pre-defined variable WAS_SERVER_NAME.

Data type String

Description:

Documents the purpose of a variable.

Data type String

Introduction: Variables

Variables come in many varieties. They are used to control settings and properties relating to the server environment. The three main types of variables that you should understand are environment variables, WebSphere variables, and custom properties.

Environment variables. Environment variables, also called *native environment variables*, are not specific to WebSphere Application Server and are defined by other elements, such as UNIX®, Language Environment® (LE), or third-party vendors, among others. Some of the UNIX-specific native variables are LIBPATH and STEPLIB. These variables tend to be operating system-specific.

Environment variables can also be specified as an application server environment entry. To specify an environment variable as an environment entry, in the administrative console, click **Servers > Server Types > WebSphere application servers***server_name*. Then, under Server Infrastructure, click **Java process management > Process definition > Environment entries**.

WebSphere variables

WebSphere variables are name and value pairs that are used to provide settings for any of the string data type attributes contained in one of the XML formatted configuration files that reside in the product repository. After a variable is defined, the value specified for the variable replaces the variable name whenever the variable name is encountered during configuration processing.

WebSphere variables can be used to configure:

- WebSphere Application Server path names, such as JAVA_HOME, and APP_INSTALL_ROOT
- Certain cell-wide customization values

To create or modify a WebSphere variable, in the administrative console click **Environment > WebSphere variables**.

A variable can apply to a cell, a cluster, a node, or a server.

How the variable is set determines its scope. If the variable is set:

- At the server level, it applies to the entire server.
- At the node level, it applies to all servers in the node, unless you set the same variable at the server level. In that case, for that server, the setting that is specified at the server level overrides the setting that is specified at the node level.
- At the cell level, it applies to all nodes in that cell, unless you set the same variable at the node or server level.
  - If you set the same variable at the server level, for that server, the setting that is specified at the server level overrides the setting that is specified at the cell level.
  - If you set the same variable at the node level, for all servers in that node, the setting that is specified at the node level overrides the setting that is specified at the cell level.

Custom properties

Custom properties are property settings meant for a specific functional component. Any configuration element can have a custom property. Common configuration elements are cell, node, server, Web container, and transaction service. A limited number of supported custom properties are available and these properties can be set in the administrative console using the custom properties link that is associated with the functional component.

For example, to set Web container custom properties, click **Servers > Server Types > WebSphere application servers >** *server_name*, and then, in the Container settings section, click **Web container > Custom properties**

Custom properties set from the Web container custom properties page apply to all transports that are associated with that Web container; custom properties set from one of the Web container transport chain or HTTP transport custom properties pages apply only to that specific HTTP transport chain or HTTP transport. If the same property is set on both the Web container page and either a transport chain or HTTP transport page, the settings on the transport chain or HTTP transport page override the settings that are defined for the Web container for that specific transport.

Note: You can only specify custom properties for an HTTP transport that is being used by an application server that is running on a Version 5.1.x node in a mixed cell environment.

WebSphere Variables

WebSphere variables are name and value pairs that are used to provide settings for any of the string data type attributes that are used to configure the product. After a variable is defined, the symbolic name that is specified for that variable can be specified in the **Value** field of any other configuration field for the product that accepts a string value.

When a variable is defined, it is given a scope. The scope is the range of locations within the product network where the variable is applicable.

- A variable with a cell-wide scope is available across the entire deployment manager cell.
- A variable with a cluster-wide scope is available across the entire cluster in the cell.
- A variable with a node-level scope is available only on the node and the servers on that node. If a node-level variable has the same name as a cell-wide variable, the node-level variable value takes precedence.
- A server variable is available only on the one server process. A server variable takes precedence over a variable with the same name that is defined at a higher level.

The value of a configuration attribute can contain references to one or more variables. The syntax for such an attribute is the name of the variable, enclosed in either a pair of curly braces { } or a pair of parenthesis ( ). In either case, the variable is proceeded by the dollar sign.

A string configuration attribute value can consist of:

- String literals, including the null value and an empty string
- Variable references that each includes one or more levels of indirection
- Nested variable references.
- Any combination of non-null and non-empty string literals, variable references, and nested variable references.

The following table illustrates all of the possible combinations.

Windows For Microsoft® Windows operating systems, the specified file paths are prefixed wth /Program Files.

Table 11.

Configuration attribute consists of:	Configuration attribute value	Variable name	Second variable value	Third variable value	Fourth variable value	Expanded configuration attribute value
String literal	/IBM/ WebSphere/ AppServer	N/A	N/A	N/A	N/A	/IBM/ WebSphere/ AppServer
Variable reference	$(WAS_ INSTALL_ ROOT)	WAS_ INSTALL_ ROOT	/IBM/ WebSphere/ AppServer	N/A	N/A	/IBM/ WebSphere/ AppServer
Variable reference with a string literal	$(USER_ INSTALL_ ROOT)/temp	USER_ INSTALL_ ROOT	N/A	N/A	/IBM/ WebSphere/ AppServer/ profiles/ AppSrv01	/IBM/ WebSphere/ AppServer/ profiles/ AppSrv01/temp
Indirect variable reference with a string literal	$(WAS_ INSTALL_ ROOT)/lib	WAS_ INSTALL_ ROOT	$(MY_INSTALL_ ROOT)	MY_INSTALL_ ROOT	N/A	N/A
Nested variable references with string literal (Example 1)	$(${INSTALL_ TYPE}_ INSTALL_ ROOT)/lib	INSTALL_ TYPE	USER	USER_INSTALL_ ROOT	/IBM/ WebSphere/ AppServer/ profiles/ AppSrv01	/IBM/ WebSphere/ AppServer/ profiles/ AppSrv01/lib
Nested variable references with string literal (Example 2)	$(${INSTALL_ TYPE}_ INSTALL_ ROOT)/lib	INSTALL_ TYPE	WAS	WAS_INSTALL_ ROOT	/IBM/ WebSphere/ AppServer/ AppServer	/IBM/ WebSphere/ AppServer/ AppServer/lib

During the configuration process, whenever a variable is encountered as the value for a configuration attribute, a variable expansion is performed on that variable. A variable expansion is the process of recursively replacing variable references with variable values until only a string literal remains as the value for the configuration attribute. If the expansion process encounters a variable that is not properly defined, the expansion of that variable stops and a VariableExpansionException exception is issued. The product configuration process continues. However, processing errors might occur because the value for this configuration attribute is not properly established.

Note: The variable expansion syntax that is provided in Versions 5.1.x, 6.0.x, and 6.1.x, of the product, includes a variant that consists of a dollar sign, and a single letter variable name without any surrounding braces or parenthesis. This syntax is not supported in Version 7.0 or higher. All WebSphere variables references must be surrounded by matching parenthesis or braces, even if it is a single letter. That syntax required escaping of dollar signs to avoid ambiguity. For backward compatibility, the escaping of the literal dollar sign is still supported, and the literal dollar sign is interpreted as indicated in the following table.

Table 12.

Input value	Value after expansion
$	$
$$	$
$$$	$$
$$$$	$$
$$$$$	$$$

Configuring the IBM Toolbox for Java

The IBM Toolbox for DB2® is a library of Java classes that are optimized for accessing i5/OS data and resources. You can use the IBM Toolbox for Java JDBC driver to access local or remote DB2 UDB for iSeries® databases from server-side and client Java applications that run on any platform that supports Java.

Before you begin

Determine which version of the IBM Toolbox for Java you want to use on your system.

About this task

The IBM Toolbox for Java is available in these versions:

IBM Toolbox for Java licensed program

The licensed program is available with every i5/OS release. You can install the licensed program on your i5/OS system, and then either copy the IBM Toolbox for Java JAR file (*jt400.jar*) to your system or update your system *classpath* to locate the server installation. Product documentation for IBM Toolbox for Java is available from the i5/OS information center: http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp Locate the documentation by traversing the following path in the left-hand navigation window of the iSeries information center:

Programming > **Java** > **IBM Toolbox for Java**.

JTOpen

JTOpen is the open source version of IBM Toolbox for Java, and is more frequently updated than the licensed program version. You can download JTOpen from http://www.ibm.com/servers/eserver/iseries/toolbox/downloads.htm. You can also download the *JTOpen Programming Guide*. The guide includes instructions for installing JTOpen and information about the JDBC driver.

The IBM Toolbox for Java JDBC driver is included with both versions of the IBM Toolbox for Java. This JDBC driver supports JDBC 3.0. For more information about IBM Toolbox for Java and JTOpen, see the product Web site at http://www.ibm.com/servers/eserver/iseries/toolbox/index.html.

Note: If you are using the product on platforms other than iSeries, use the **JTOpen** version of the Toolbox JDBC driver.

1. Download the *jt400.jar* file from the **JTOpen** URL at http://www.ibm.com/servers/eserver/iseries/toolbox/downloads.htm.

   Place it in a directory on your workstation such as /JDBC_Drivers/Toolbox.
2. Open the administrative console.
3. Select **Environment > WebSphere variables**.
4. In the list of available scopes, select the appropriate node.
5. Locate the WebSphere variable OS400_TOOLBOX_JDBC_DRIVER_PATH in the list of variables that are defined for that scope.

   Depending on how many variables are defined for the selected node, you might have to navigate through multiple pages of variables to find the OS400_TOOLBOX_JDBC_DRIVER_PATH variable. In this situation, clicking the arrow at the bottom of the page takes you to the next page of variables for the selected node.
6. Click **OS400_TOOLBOX_JDBC_DRIVER_PATH** in the name column.
7. Set the value to the full directory path to the `jt400.jar` file downloaded in step one. Do not include jt400.jar in this value.

   For example, if the fully qualified path to the `jt400.jar` file is:

   ```
   JDBC_Drivers/Toolbox/jt400.jar
   ```

   Specify JDBC_Drivers/Toolbox as the value for the OS400_TOOLBOX_JDBC_DRIVER_PATH variable.

8. Click **Apply** and then click **Save** to save your changes.

Repository service custom properties

Use this page to add custom properties for the repository service.

You can specify repository service custom properties in the administrative console:

1. In the administrative console navigation, click **System Administration > Node agents**.
2. Select a node agent from the list.
3. Under Additional Properties, click **File synchronization service**.
4. Under Additional Properties, click **Custom properties**.
5. Click **New**.
6. Enter the name of the custom property in the Name field, and the value in the Value field. You can leave the Description field blank.

Managing shared libraries

Shared libraries are files used by multiple applications. Each shared library consists of a symbolic name, a Java class path, and a native path for loading Java Native Interface (JNI) libraries. You can use shared libraries to reduce the number of duplicate library files on your system.

Before you begin

Your applications use the same library files. The applications already are deployed on a server or you currently are deploying the applications.

About this task

Suppose that you have four applications that use the same library file, my_sample.jar. Instead of having four copies of my_sample.jar on your system after the four applications are deployed, you can define a shared library for my_sample.jar and have the four deployed applications use that one my_sample.jar library file.

Isolated shared libraries provide another way to reduce the number of library files. Isolated shared libraries each have their own class loader, enabling a single instance of the classes to be shared across the applications. Each application can specify which isolated shared libraries that it wants to reference. Different applications can reference different versions of the isolated shared library, resulting in a set of applications sharing an isolated shared library. With isolated shared libraries, some applications can share a single copy of Library A, Version 1 while other applications share a single copy of Library A, Version 2, for a total of two instances in memory.

Using the administrative console, you can define shared libraries for the library files that multiple applications use and then associate the libraries with specific applications or modules or with an application server. Guidelines for associating shared libraries are as follows:

- Associate a shared library file with an application or module to load the classes represented by the shared library in a local class loader, which can be an application-wide or module-wide class loader.
- Associate an isolated shared library file with an application or module to load the classes represented by the shared library in a separate class loader created for that shared library.
- Associate a shared library file with a server to load the classes represented by the shared library in a server-wide class loader. This class loader is the parent of the application class loader, and the WebSphere Application Server extensions class loader is its parent. Associating a shared library file with a server associates the file with all applications on the server.
- Do not associate an isolated shared library file with a server if you want a separate class loader for a shared library. If you associate the shared library with a server, the product ignores the isolation setting

and still adds files in the shared library to the application server class loader. That is, associating an isolated shared library file with a server associates the file with all applications on the server. The product does not use an isolated shared library when you associate the shared library with a server. Associate an isolated shared library with an application or module.

Instead of using the administrative console to associate a shared library with an application, you can use an installed optional package. You associate a shared library to an application by declaring the dependent library .jar file in the MANIFEST.MF file of the application. Refer to the Java 2 Platform, Enterprise Edition (J2EE) 1.4 specification, section 8.2 for an example.

- Use the administrative console to define a shared library.
  1. Create a shared library.

     On a single-server product, you can define a shared library at the cell, node, or server level.

     On a multiple-server product, you can define a shared library at the cell, node, server, or cluster level.

     Defining a library at one of the these levels does not automatically place the library into a class loader. You must associate the library with an application, module, or server before the product loads the classes represented by the shared library into a local or server-wide class loader.
  2. Associate each shared library with an application, module, or server.
     - Associate a shared library with an application or module that uses the shared library file.

       If you enabled the **Use an isolated class loader for this shared library** setting when creating the shared library, associate the isolated shared library with an application or module to use a separate class loader for the shared library.
     - Associate a shared library with an application server so every application on the server can use the shared library file.
- Use an installed optional package to declare a shared library for an application.
- Remove a shared library.
  1. Click **Environment** → **Shared libraries** in the console navigation tree to access the Shared libraries page.
  2. Select the library to be removed.
  3. Click **Delete**.

  The list of shared libraries is refreshed. The library file no longer displays in the list.

Creating shared libraries

Shared libraries are files used by multiple applications. Create a shared library to reduce the number of duplicate library files on your system.

Before you begin

Determine the full path name or directory of each library file for which you want a shared library.

About this task

To make a library file available to multiple applications deployed on a server, create one or more shared libraries for library files that your applications need. When you create the shared libraries, you can use variables within the library file class paths.

You can create one shared library that points to multiple files or directories. This enables you to maintain a single shared library for files that your applications need.

Or you can create a shared library for each library file that your applications need. This approach is recommended only when you have few library files and few applications that use the files. After you create a shared library, you associate it with each application that uses the library files. If you have multiple

shared libraries and multiple applications that use the library files, you must complete many steps to create and associate those shared libraries. It is simpler to use one shared library for related files.

Use the Shared libraries page to create and configure shared libraries.

1. Go to the Shared libraries page.

   Click **Environment** → **Shared libraries** in the console navigation tree.
2. Select a shared library scope.

   Change the scope of the collection table to see what shared libraries are in a particular cell, node or server.

   a. Select a cell, node, or server.

      On a multiple-server product, you also can select a cluster. To see the cluster scope, you first must create a cluster on the Server clusters page (**Servers** → **Clusters** → **WebSphere application server clusters**).

   b. Click **Apply**.

   After creating a shared library, you can see whether a shared library can be used on a specific node. Select a scope to see what shared libraries are available to applications installed on or mapped to that scope.
3. Click **New**.
4. Configure the shared library.

   a. On the shared library settings page, specify the name, class path, and any other variables for the library file that are needed.

      If the shared library specifies a native library path, refer to “Configuring native libraries in shared libraries.”

      To have only one instance of a version of a class shared among applications or modules, make the shared library an isolated shared library. Select **Use an isolated class loader for this shared library**. Using an isolated shared library can reduce the memory footprint when a large number of applications share the library.

   b. Click **Apply**.

What to do next

Using the administrative console, associate your shared libraries with specific applications or modules or with the class loader of an application server. Associating a shared library file with a server class loader associates the file with all applications on the server.

If you enabled the **Use an isolated class loader for this shared library** setting when creating your shared library, associate the shared library with applications or Web modules. If you associate the shared library with a server, the product ignores this setting and still adds files in the shared library to the application server class loader. The product does not use an isolated shared library when you associate the shared library with a server.

Alternatively, you can use an installed optional package to associate your shared libraries with an application.

Configuring native libraries in shared libraries

Native libraries are platform-specific library files, including .dll, .so, or *SRVPGM objects, that can be configured within shared libraries. Native libraries are visible to an application class loader whenever the shared library is associated with an application. Similarly, native libraries are visible to an application server class loader whenever the shared library is associated with an application server.

Before you begin

When designing a shared library, consider the following conditions regarding Java native library support:

- The Java virtual machine (JVM) allows only one class loader to load a particular native library.
- There is no application programming interface (API) to unload a native library from a class loader.

  Native libraries are unloaded by the JVM when the class loader that found the library is collected from the heap during garbage collection.
- Application server class loaders, unlike the native JVM class loader, only load native shared libraries that use the default operating system extension for the current platform. For example, on AIX, native shared libraries must end in .a when loaded by application server class loaders. The JVM class loader loads files ending in .a or .so.
- Application server class loaders persist for the duration of the application server.
- Application class loaders persist until an application is stopped or dynamically reloaded.

  If a shared library that is configured with a native library path is associated with an application, whenever the application is restarted or dynamically reloaded the application might fail with an UnsatisfiedLinkError indicating that the library is already loaded. The error occurs because, when the application restarts, it invokes the shared library class to reload the native library. The native library, however, is still loaded in memory because the application class loader which previously loaded the native library has not yet been garbage collected.
- Only the JVM class loader can load a dependent native library.

  For example, if *NativeLib1* is dependent on *NativeLib2*, then *NativeLib2* must be visible to the JVM class loader. The path containing *NativeLib2* must be specified on Java library path defined by the LIBPATH environment variable.

  If a native library configured in a shared library is dependent on other native libraries, the dependent libraries must be configured on the LIBPATH of the JVM hosting the application server in order for that library to load successfully.

About this task

When configuring a shared library on a shared library settings page, if you specify a value for **Native library path**, the native libraries on this path are not located by the WebSphere Application Server application or shared library class loaders unless the class which loads the native library was itself loaded by the same class loader.

Because a native library cannot be loaded more than once by a class loader, it is preferable for native libraries to be loaded within shared libraries associated with the class loader of an application server, because these class loaders persist for the lifetime of the server.

1. Implement a static method in the class that loads the native library.

   In the class that loads the native library, call `System.loadLibrary(`*`native_library`*`)` in a static block. For example:

   ```
   static {System.loadLibrary("native_library");
   ```

   native_library loads during the static initialization of the class, which occurs exactly once when the class loads.
2. On the shared library settings page, set values for **Classpath** and **Native library path** that enable the shared library to load the native library.

   If you want to associate your shared library with an application or module, also select **Use an isolated class loader for this shared library**. If you do not enable this setting, associate the shared library with an application server.
3. Associate the shared library.
   - If you did not enable **Use an isolated class loader for this shared library**, associate the shared library with an application server.

Associating a shared library with the class loader of an application server, rather than with an application, ensures that the shared library is loaded exactly once by the application server class loader, even though applications on the server are restarted or dynamically reloaded. Because the native library is loaded within a static block, the native library is never loaded more than once.

- If you enabled **Use an isolated class loader for this shared library**, associate the shared library with an application or module.

  Associating an isolated shared library file with an application or module loads the classes represented by the shared library in a separate class loader created for that shared library. Do not associate an isolated shared library file with a server if you want a separate class loader for a shared library. If you associate the shared library with a server, the product ignores the isolation setting and still adds files in the shared library to the application server class loader. That is, associating an isolated shared library file with a server associates the file with all applications on the server.

  The class loader created for an isolated shared library does not reload and, like a server class loader, exists for the lifetime of a server. For shared native libraries, you can use an isolated shared library to avoid errors resulting from reloading of native libraries.

What to do next

To verify that an application can use a shared library, test the application or examine the class loader in the Class loader viewer. Click **Troubleshooting** → **Class loader viewer** → ***module_name*** → **Table View**. The classpath of the application module class loader lists the classes used by the shared library.

Shared library collection

Use this page to define a list of shared library files that deployed applications can use.

To view this administrative console page, click **Environment** → **Shared libraries**.

Change the scope to see what shared libraries are in a particular node or server. By default, a shared library is accessible to applications deployed (or installed) on the same node as the shared library file. To change the scope, select the cell, a node, or a server under **Scope**.

On a multiple-server product, you also can select a cluster. To see the cluster scope, you first must create a cluster on the Server clusters page (**Servers** → **Clusters** → **WebSphere application server clusters**). The cluster scope limits the scope of a shared library to cluster members of a particular cluster.

Select a scope before you click **New** and create a shared library. After you create a shared library and map an application to the selected scope, you can associate the shared library with the application or its modules.

- To associate a shared library with an application or module, use the Shared library references page for the application. Click **Applications** → **Application Types** → **WebSphere enterprise applications** → ***application_name*** → **Shared library references**.
- To associate a shared library with a server class loader, use the settings page for the library reference for the server class loader. Click **Servers** → **Server Types** → **WebSphere application servers** → ***server_name*** → **Java and Process Management** → **Class loader** → ***class_loader_ID*** → **Shared library references** → ***shared_library_name***.

Name

Specifies a name for the shared library.

Description

Describes the shared library file.

Shared library settings

Use this page to make a library file available to deployed applications.

To view this administrative console page, click **Environment** → **Shared libraries** → ***shared_library_name***.

Scope:

Specifies the level of the location of the shared library configuration file.

On single-server installations, the shared library has its configuration file in a location that pertains to the cell, node, or server level.

On multiple-server installations, the shared library has its configuration file in a location that pertains to the cell, node, server, or cluster level.

Data type String

Name:

Specifies a name for the shared library.

Data type String

Description:

Describes the shared library.

Data type String

Classpath:

Specifies a list of paths that the product searches for classes and resources of the shared library.

If a path in the list is a file, the product searches the contents of that Java archive (JAR) or compressed (zip) file. If a path in the list is a directory, then the product searches the contents of JAR and zip files in that directory. For performance reasons, the product searches the directory itself only if the directory contains subdirectories or files other than JAR or zip files.

Press Enter to separate class path entries. Entries must not contain path separator characters such as a semicolon (**;**) or colon (**:**). Class paths can contain variable names that can be substituted using a variable map.

Data type String
Units Class path

Native library path:

Specifies the class path for locating platform-specific library files for shared library support; for example, .dll, .so, or *SRVPGM objects.

If you specify a value for **Native library path**, the native libraries are not located by application or shared library class loaders unless the following conditions exist:

- A class loads the native libraries.

- The application invokes a method in this class which loads the libraries.

  For example, in the class that loads the native library, call `System.loadLibrary(native_library)` in a static block:

  ```
  static {System.loadLibrary("native_library");
  ```

- The **Classpath** specified on this page contains the class that loads the libraries.

Native libraries cannot be loaded more than once by a class loader. Thus, it is preferable for native libraries to use an isolated shared library or to be loaded within shared libraries associated with the class loader of an application server. See the **Use an isolated class loader for this shared library** setting.

Data type	String
Units	Class path

Use an isolated class loader for this shared library:

Specifies whether the shared library has a single isolated shared library shared across its associated applications or Web modules.

Note: An isolated shared library enables one instance of the library classes to be shared only among associated applications and Web modules. An isolated shared library enables multiple applications or Web modules to share a common set of classes across a subset of the applications. Further, an isolated shared library supports versioning and loads the minimum number of library copies. The class loader created for an isolated shared library does not reload and, like a server class loader, exists for the lifetime of a server. For shared native libraries, you can use an isolated shared library to avoid errors resulting from reloading of native libraries.

The default, `false`, is not to isolate the shared library so that each application loads its own instances of the shared library classes.

Using an isolated shared library can reduce the memory footprint when a large number of applications share the library. If you select this option, associate the shared library with applications or Web modules.

Note: If you associate the shared library with a server, the product ignores this setting and still adds files in the shared library to the application server class loader. The product does not use an isolated shared library when you associate the shared library with a server. To use an isolated shared library, you must associate the shared library with applications or Web modules.

Selecting this option affects the class loader order of the associated application or Web module. If the class loader order for a class loader associated with an isolated shared library is **Classes loaded with the parent class loader first** (Parent first), the class loader checks whether a class can be loaded in the following order:
1. Checks whether the associated library class loaders can load the class.
2. Checks whether its parent class loader can load the class.
3. Checks whether it (application or WAR module class loader) can load the class.

If the order is **Classes loaded with the local class loader first (Parent last)**, the class loader checks in the following order:
1. Checks whether it (application or WAR module class loader) can load the class.
2. Checks whether the associated library class loaders can load the class.
3. Checks whether its parent class loader can load the class.

This setting maps to the `isolatedClassLoader` Boolean attribute of the Library object.

Boolean	false

Associating shared libraries with applications or modules

You can associate a shared library with an application or module. Classes represented by the shared library are then loaded in the application's class loader, making the classes available to the application.

Before you begin

This topic assumes that you have defined a shared library. The shared library represents a library file used by multiple deployed applications.

You can define a shared library at the cell, node, server, or cluster level.

On a multiple-server product, you also can define a shared library at the cluster level. To see the cluster scope, you first must create a cluster on the Server clusters page (**Servers** → **Clusters** → **WebSphere application server clusters**).

This topic also assumes that you want to use the administrative console, and not an installed optional package, to associate a shared library with an application.

About this task

To associate a shared library with an application or module, create and configure a library reference using the administrative console. A library reference specifies the name of the shared library file.

If you associate a shared library with an application, do not associate the same shared library with a server class loader.

1. If you have not done so already, map your application to a target server that is within the scope of the shared library.

   For example, if the shared library scope is the *my_cluster* cluster, map your application to the target *my_cluster* cluster.
2. Click **Applications** → **Application Types** → **WebSphere enterprise applications** → ***application_name*** → **Shared library references** in the console navigation tree to access the Shared library references page.
3. On the Shared library references page, select an application or module to which you want to associate a shared library.
4. Click **Reference shared libraries**.
5. On the Shared library mapping page, select one or more shared libraries that the application or modules use in the **Available** list, click **>>** to add them to the **Selected** list, and click **OK**.
6. Repeat steps 2 through 4 until you define a library reference instance for each shared library that your application or module requires.
7. On the Shared library references page, click **OK**.
8. Save the changes to the configuration.

Results

When you run the application, classes represented by the shared library are loaded in the application class loader.

The classes are now available to the application or module.

What to do next

To verify an association between an application and a shared library, examine the application class loader in the Class loader viewer. Click **Troubleshooting** → **Class loader viewer** → ***module_name*** → **Table View**.

The classpath of the application module class loader lists the classes used by the shared library.

Shared library reference and mapping settings

Use the Shared library references and Shared library mapping pages to associate defined shared libraries with an application or Web module. A shared library is an external Java archive (JAR) file that is used by one or more applications. Using shared libraries enables multiple applications deployed on a server to use a single library, rather than use multiple copies of the same library. After you associate shared libraries with an application or module, the application or module class loader loads classes represented by the shared libraries and makes those classes available to the application or module.

To view the Shared library references console page, click **Applications** → **Application Types** → **WebSphere enterprise applications** → ***application_name*** → **Shared library references**. To view the Shared library mapping page, click **Reference shared libraries** on the Shared library references page. These pages are the same as the Map shared libraries and Map shared libraries to an entire application or module pages in the application installation and update wizards.

On the Shared library references page, the first element listed is the application. The other elements are modules in the application.

To associate shared libraries with your application or module:

1. Select an application or module.
2. Click **Reference shared libraries**.
3. On the Shared library mapping page, select one or more shared libraries that the application or modules uses in the **Available** list, click **>>** to add them to the **Selected** list, and click **OK**.

A defined shared library for a file that your application or module uses must exist to associate your application or module to the library.

If no shared libraries are defined and the application is installed already, on the Shared library mapping page, click **New** and define a shared library.

You can otherwise define a shared library as follows:

1. Click **Environment** → **Shared libraries**.
2. Specify whether the shared library is visible at the cell, node or server level.
3. Click **New**.
4. On the settings page for the new shared library, specify a name and one or more class paths. If the libraries are platform-specific files such as .dll, .so, or *SRVPGM objects, also specify a native library path. Then, click **Apply**.
5. Save the administrative configuration.

Application:

Specifies the name of the application that you are installing or that you selected on the Enterprise applications page.

Module:

Specifies the name of the module associated with the shared libraries.

URI:

Specifies the location of the module relative to the root of the application EAR file.

Shared libraries:

Specifies the name of the shared library files associated with the application or module.

Associating shared libraries with servers

You can associate shared libraries with the class loader of a server. Classes represented by the shared library are then loaded in a server-wide class loader, making the classes available to all applications deployed on the server.

Before you begin

This topic assumes that you have defined a shared library. The shared library represents a library file used by multiple deployed applications.

About this task

To associate a shared library with the class loader of a server, create and configure a library reference using the administrative console. A library reference specifies the name of the shared library file.

If you associate a shared library with a server class loader, do not associate the same shared library with an application.

1. Configure class loaders for applications deployed on the server.
   a. Click **Servers** → **Server Types** → **WebSphere application servers** → ***server_name*** to access the application server setting page.
   b. Set values for the application **Class loader policy** and **Class loading mode** of the server.

      For information on these settings, see Application server settings in the *Administering applications and their environment* PDF.
2. Create a library reference for each shared library file that your application needs.
   a. In the administrative console, click **Servers** → **Server Types** → **WebSphere application servers** → ***server_name*** → **Java and Process Management** → **Class loader** → ***class_loader_ID*** .
   b. Click **Shared library references** to access the Library reference page.
   c. Click **Add**.
   d. On the library reference settings page, name the library reference. The name identifies the shared library file that your application uses.
   e. Click **Apply**. The name of the library reference is shown in the list on the Library reference page.

   Repeat the previous steps until you define a library reference for each shared library that your application needs.

What to do next

To verify that an application can use a shared library, test the application or examine the class loader in the Class loader viewer. Click **Troubleshooting** → **Class loader viewer** → ***module_name*** → **Table View**. The classpath of the application module class loader lists the classes used by the shared library.

Installed optional packages

Installed optional packages enable applications to use the classes in Java archive (.jar) files without having to include them explicitly in a class path. An installed optional package is a .jar file containing specialized tags in its manifest file that enable the application server to identify it. An installed optional package declares one or more shared library .jar files in the manifest file of an application. When the application is installed on a server or cluster, the classes represented by the shared libraries are loaded in the class loader of the application, making the classes available to the application.

When a Java Platform, Enterprise Edition (Java EE) application is installed on a server or cluster, dependency information is specified in its manifest file. The product reads the dependency information of

the application (.ear file) to automatically associate the application with an installed optional package .jar file. The product adds the .jar files in associated optional packages to the application class path. Classes in the installed optional packages are then available to application classes.

Installed optional packages used by the product are described in section 8.2 of the Java 2 Platform, Enterprise Edition (J2EE) specification, Version 1.4 at http://java.sun.com/j2ee/j2ee-1_4-fr-spec.pdf.

The product supports using the manifest file (manifest.mf) in shared library .jar files and application .ear files. The product does not support the Java 2 Platform Standard Edition (J2SE) Installed Optional Package semantics used in the J2SE specification (http://java.sun.com/j2se/1.3/docs/guide/extensions/spec.html), which primarily serve the applet environment. The product ignores applet-specific tags within manifest files.

Sample manifest.mf file

A sample manifest file follows for an application app1.ear that refers to a single shared library file util.jar:

```
app1.ear:
    META-INF/application.xml
    ejb1.jar:
         META-INF/MANIFEST.MF:
             Extension-List: util
             util-Extension-Name: com/example/util
             util-Specification-Version: 1.4
         META-INF/ejb-jar.xml

util.jar:
    META-INF/MANIFEST.MF:
        Extension-Name: com/example/util
        Specification-Title: example.com's util package
        Specification-Version: 1.4
        Specification-Vendor: example.com
        Implementation-Version: build96
```

The syntax of a manifest entry depends on whether the entry applies to a member with a defining role (the shared library) or a member with a referencing role (a Java EE application or a module within a Java EE application).

Manifest entry tagging

Main tags used for manifest entries include the following:

Extension-List
A required tag with variable syntax. Within the context of the referencing role (application's manifest), this is a space delimited list that identifies and constructs unique Extension-Name, Extension-Specification tags for each element in the list. Within the context of the defining role (shared library), this tag is not valid.

Extension-Name
A required tag that provides a name and links the defining and referencing members. The syntax of the element within the referencing role is to prefix the element with the `<ListElement>` string. For each element in the Extension-List, there is a corresponding `<ListElement>`-Extension-Name tag. The defining string literal value for this tag (in the above sample `com/example/util1`) is used to match (in an equality test) the corresponding tags between the defining and referencing roles.

Specification-Version
A required tag that identifies the specification version and links the defining and referencing members.

Implementation-Version
An optional tag that identifies the implementation version and links the defining and referencing members.

Further information on these tags is in the .jar file specification at http://java.sun.com/j2se/1.4.2/docs/guide/jar/jar.html#Manifest%20Specification.

Using installed optional packages

You can associate one or more shared libraries with an application using an installed optional package that declares the shared libraries in the application's manifest file. Classes represented by the shared libraries are then loaded in the application's class loader, making the classes available to the application.

Before you begin

Read about installed optional packages in "Installed optional packages" on page 191 and in section 8.2 of the Java 2 Platform, Enterprise Edition (J2EE) specification, Version 1.4 at http://java.sun.com/j2ee/j2ee-1_4-fr-spec.pdf.

WebSphere Application Server does not support the Java 2 Platform Standard Edition (J2SE) Installed Optional Package semantics used in the J2SE specification (http://java.sun.com/j2se/1.3/docs/guide/extensions/spec.html), which primarily serve the applet environment. WebSphere Application Server ignores applet-specific tags within manifest files.

About this task

Installed optional packages expand the existing shared library capabilities of an application server. Prior to Version 6.0, an administrator was required to associate a shared library to an application or server. Installed optional packages enable an administrator to declare a dependency in an application's manifest file to a shared library, with installed optional package elements listed in the manifest file, and automatically associate the application to the shared library. During application installation, the shared library `.jar` file is added to the class path of the application class loader.

If you use an installed optional package to associate a shared library with an application, do not associate the same shared library with an application class loader or a server class loader using the administrative console.

1. Assemble the library file, including the manifest information that identifies it as an extension. Two sample manifest files follow. The first sample manifest file has application `app1.ear` refer to a single shared library file `util.jar`:

```
app1.ear:
    META-INF/application.xml
    ejb1.jar:
         META-INF/MANIFEST.MF:
             Extension-List: util
             util-Extension-Name: com/example/util
             util-Specification-Version: 1.4
         META-INF/ejb-jar.xml

util.jar:
    META-INF/MANIFEST.MF:
        Extension-Name: com/example/util
        Specification-Title: example.com's util package
        Specification-Version: 1.4
        Specification-Vendor: example.com
        Implementation-Version: build96
```

   The second sample manifest file has application `app1.ear` refer to multiple shared library `.jar` files:

```
app1.ear:
    META-INF/application.xml
    ejb1.jar:
         META-INF/MANIFEST.MF:
             Extension-List: util1 util2 util3
             Util1-Extension-Name: com/example/util1
             Util1-Specification-Version: 1.4
             Util2-Extension-Name: com/example/util2
             Util2-Specification-Version: 1.4
             Util3-Extension-Name: com/example/util3
             Util3-Specification-Version: 1.4
         META-INF/ejb-jar.xml

util1.jar:
    META-INF/MANIFEST.MF:
        Extension-Name: com/example/util1
        Specification-Title: example.com's util package
        Specification-Version: 1.4
        Specification-Vendor: example.com
        Implementation-Version: build96

util2.jar:
    META-INF/MANIFEST.MF:
        Extension-Name: com/example/util2
        Specification-Title: example.com's util package
        Specification-Version: 1.4
        Specification-Vendor: example.com
        Implementation-Version: build96

util3.jar:
    META-INF/MANIFEST.MF:
        Extension-Name: com/example/util3
        Specification-Title: example.com's util package
        Specification-Version: 1.4
        Specification-Vendor: example.com
        Implementation-Version: build96
```

2. Create a shared library that represents the library file assembled in step 1. This installs the library file as a shared library.
3. Copy the shared library `.jar` file to the cluster members.
4. Assemble the application, declaring in the application manifest file dependencies to the library files named the manifest created for step 1.

   See the *Developing and deploying applications* PDF for more information.
5. Install the application on the server or cluster.

   See the *Developing and deploying applications* PDF for more information.

Results

During application installation, the shared library `.jar` files are added to the class path of the application class loader.

Library reference collection

Use this page to view and manage library references that define how to use global libraries. For example, you can use this page to associate shared library files with a deployed application.

To view this administrative console page, click **Servers** → **Server Types** → **WebSphere application servers** → ***server_name*** → **Java and Process Management** → **Class loader** → ***class_loader_ID*** → **Shared library references** .

If no shared libraries are defined in your environment, such as at the node or server scope, after you click **Add** a message is displayed stating that you must define a shared library before you can create a library

reference. A shared library is a container-wide library file that deployed applications can use. To define a shared library, click **Environment** → **Shared libraries** and specify the scope of the container. Then, click **New** and specify a name and one or more paths for the shared library. After you define a shared library, return to this page, click **Add**, and create a library reference.

Library name
Specifies a name for the library reference.

Library reference settings
Use this page to define library references, which specify how to use global libraries.

To view this administrative console page, click **Servers** → **Server Types** → **WebSphere application servers** → ***server_name*** → **Java and Process Management** → **Class loader** → ***class_loader_ID*** → **Shared library references** → ***library_reference_name***.

A shared library is a container-wide library file that deployed applications can use. To define a shared library, click **Environment** → **Shared libraries** and specify the scope of the container. Then, click **New** and specify a name and one or more paths for the shared library.

Library name:

Specifies the name of the shared library to use for the library reference.

Data type String

Creating application servers

During the installation process, the product creates a default application server, named server1. Most installations require several application servers to handle the application serving needs of their production environment. You can use the command-line tool or the administrative console to create additional application servers.

Before you begin

Determine if you want to use the application server that you are creating as part of a cluster. If this application server is going to be part of a cluster, you must use the Create a new cluster wizard instead of the Create a new application server wizard to create this application server. The topic *Adding members to a cluster* describes how to use the Create a new cluster wizard.

About this task

To create a new application server that is not part of a cluster, you can either use the createApplicationServer, createWebServer, or createGenericServer wsadmin command, or you can use the administrative console.

If you are migrating from a previous version of the product, you can upgrade a portion of the nodes in a cell, while leaving others at the previous product level. This means that, for a period of time, you might be managing servers that are running at two different release levels in the same cell. However, when you create a new server definition, you must use a server configuration template, and that template must be created from a server instance that matches the version of the node for which you are creating the server.

There are no restrictions on what you can do with the servers running on the more current release level.

Complete the following steps if you want to use the administrative console to create a new application server that is not part of a cluster.

1. In the administrative console, click **Servers > Server Types > WebSphere application servers > New**.

   The Create a new application server wizard starts.
2. Select a node for the application server.
3. Enter a name for the application server. The name must be unique within the node.
4. Click **Next**.
5. Select a server template for the new server.

   You can use a default application server template for your new server, or you can use the template that is optimized for development uses. The new application server inherits all of the configuration settings of the template server.
6. Click **Next**.

   By default, this option is enabled. If you select this option, then you might need to update the alias list for the virtual host that you plan to use with this server to contain these new port values. If you deselect this option, then ensure that the default port values do not conflict with other servers on the same physical machine.
7. Select **Generate unique HTTP ports** if you want the wizard to generate unique ports for the application server.
8. Click **Next**. Review the settings for the new server.
9. If you want to change any of the settings, click **Previous** until you return to a page where you can change that setting.
10. Click **Finish** when you do not want to make any additional changes.
11. Click **Review**, select **Synchronize changes with nodes**, and then click **Save** to save your changes.

Results

The new application server is in the list of servers on the administrative console Application servers page.

What to do next

This newly created application server is configured with default settings that are not displayed when you run the Create New Application Server wizard.

You can:

- In the administrative console, click **Servers > Server Types > WebSphere application servers** , and then click the name of this application server to view all of the configuration settings for this application server. You can then use this page to change some of the configuration settings for this server.

  For example, if you do not need to have all of the sever components start during the server startup process, you might want to select **Start components as needed**, which is not automatically selected when a new server is created. When this property is selected, server components are dynamically started as they are needed. When this property is not selected, all of the server components are started during the startup process. Therefore, selecting this property usually results in improved startup time because fewer components are started during the startup process.

  Note: If you are running other WebSphere products on top of this product, make sure that those other products support this functionality before you select this property.
- Set the client.encoding.override Java virtual machine (JVM) argument to `UTF-8` if you need to use multiple language encoding support in the administrative console.

Creating server templates

A server template is used to define the configuration settings for a new application server. When you create a new application server, you either select the default server template or a template you previously

created, that is based on another, already existing application server. The default template is used if you do not specify a different template when you create the server.

About this task

To create a server template.

1. In the administrative console, click **Servers > Server Types > WebSphere application servers >** , and then click **Templates**.

   You can also use the **createServerTemplate** command for the AdminTask object to create a server template.
2. On the Server templates page, click **New**.
3. From the list of servers, select the server that you want to use to create the new template, and then click **OK**.
4. Enter the name of the new template and, optionally, a description of that template that distinguishes it from your other templates.
5. click **OK**.

Results

Your new template is on the list of server templates that you can use to create a new application server or cluster member.

What to do next

You can perform one of the following actions to display a list of all of the server templates that are available on your system:

1. In the administrative console, click **Servers > Server Types > WebSphere application servers >** , and then click **Templates**.
2. Issue the **listServerTemplates** wsadmin command.

Deleting server templates

The steps below describe how to delete a server template that you no longer need.

1. In the administrative console, click **Servers > Server Types > WebSphere application servers**, and then click **Templates**.

   You can also use the **deleteServerTemplate** wsadmin command to delete server templates.
2. Select the template you want to delete, and click **Delete**.

Results

The template you chose is removed from the list and cannot be used to create a new application server.

What to do next

You can perform one of the following actions to display a list of the server templates that are still available on your system:

1. In the administrative console, click **Servers > Server Types > WebSphere application servers**, and then click **Templates**.
2. Issue the **listServerTemplates** wsadmin command.

Managing application servers

You can use either the administrative console or command-line tools to manage your application servers.

Before you begin

If you plan to change the system clock, stop all the application servers, the node agent servers, the deployment manager server, the administrative agent server, and the job manager server first. After you stop the servers, change the system clock, and then restart the servers. If you change the system clock on one system, you must ensure the clocks on all systems that communicate with each other and have WebSphere Application Server installed are synchronized. Otherwise, you might experience errors, such as security tokens no longer being valid.

About this task

Note: If you are migrating from a previous version of the product, you can upgrade a portion of the nodes in a cell, while leaving others at the previous release level. This means that, for a period of time, you might be managing servers that are running at different release levels in the same cell. In this mixed environment, some restrictions exist for what you can do with servers that are running at a previous release level. No restrictions exist for what you can do with the servers that are running on the newest release level.

You can perform the following steps to view and manage an application server from the administrative console.

1. In the administrative console click **Servers > Server Types > WebSphere application servers**.

   The Application servers page lists the application servers in your environment and the status of each of these servers. You can use this page to complete the following actions:

   - Create additional servers.
   - Monitor running servers.
   - Control the status of a server.
   - Create a server template
   - Delete a server. When you select a server for deletion, you must click **Delete** and **OK** before the server is deleted.

     Note: If the server you are deleting has applications or modules mapped to it and is not part of a cluster, remap the modules to another server, or create a new server and remap the modules to the new server, before you delete this server. After a server to which modules are mapped is deleted, you cannot remap these modules to another server. Therefore, if you do not remap the modules to another server before you delete this server, you must uninstall all of the modules that were mapped to this server, and then reinstall them on a different server.

     If the server you are deleting is part of a cluster, any application that is installed on this server is automatically installed on all of the other servers in the cluster. Therefore, deleting one cluster member does not affect the other cluster members, and the application remains installed in the cluster. Similarly when a new member is added to an existing cluster, any applications that are installed on the servers in that cluster are automatically installed on the new cluster member.

2. Click the name of a listed server to view or change the configuration settings for that server.

   You can use this administrative console page to:

   - Change the configuration settings for the selected server.

     For example, if you do not need to have all of the sever components start during the server startup process, you might want to select **Start components as needed**, which is not automatically selected when a new server is created. When this property is selected, server components are dynamically started as they are needed. When this property is not selected, all of the server components are started during the startup process. Therefore, selecting this property usually results in improved startup time because fewer components are started during the startup process.

- View the status of applications running on the selected server. To view the status of applications running on this server, under Applications, click **Installed Applications**.

3. Click **Review**, select **Synchronize changes with Nodes**.
4. Click **Save** to save any configuration changes that you made.
5. If you made any configuration or custom property changes, start the application server, or stop and restart the application server if it is already running.

Results

When you click **Servers > Server Types > WebSphere application servers**, you can view the state of each server.

When you click **Servers > Server Types > WebSphere application servers >** *server_name*, you can view any configuration changes you made.

What to do next

You can deploy applications or components to your application servers.

Server collection

Use this topic to learn how to navigate within the administrative console to the pages where you can view information about the application servers, generic servers, Java message service (JMS) servers, and Web servers that are defined for your system.

You can use these respective administrative console pages to view the status of the listed servers. The status indicates whether a server is running, stopped, or encountering problems. You can also use these pages to perform the following actions for the listed servers:

- Select one or more of the listed servers, and then click **Start** to start those servers.
- Select one or more of the listed servers, and then click one of the following options to stop those servers:

  STOP When you click this option, the normal server quiesce process is followed. This process allows in-flight requests to complete before the entire server process shuts down.

  Immediate Stop
  This option is only available for application servers.

  When you click this button, the selected sever stops but the normal server quiesce process is not followed. This shutdown mode is faster than the normal server stop processing, but some application clients might receive exceptions if an in-flight request does not complete before the server process shuts down.

  Terminate
  This option is not available for Version 5 JMS servers.

  You should only click **Terminate** if the server does not respond when you click **Stop**, or, **Immediate Stop** or when you issue the Stop or ImmediateStop commands. Some application clients can receive exceptions. Therefore, you should always attempt an immediate stop before clicking **Terminate**.
- Click **New** to create a new server.
- Click **Templates** to create a new server template.
- Select one or more of the listed servers, and then click **Delete** those servers. This option is not available for Version 5 JMS servers.

To view the Application servers page, in the administrative console page, click **Servers > Server Types > WebSphere application servers**. This page lists all of the application servers in the cell.

To view the Generic servers page, in the administrative console, click **Servers > Server Types > Generic servers**. This page lists all of the generic servers in the cell.

To view the Web servers page, in the administrative console, click **Servers > Server Types > Web servers**. This page lists all of the Web servers in your administrative domain. In addition to the previously mentioned actions, you can use this page to generate and propagate a Web server plug-in configuration file.

To view the Version 5 Java message service (JMS) servers page, in the administrative console page, click **Servers > Server Types > Version 5 JMS servers.** This page lists all of the JMS servers in the cell. Each JMS server provides the functions of the JMS provider for a node in your administrative domain. There can be, at most, one JMS server on each node in the administration domain, and any application server within the domain can access JMS resources served by any JMS server on any node in the domain.

Note: JMS servers apply only to WebSphere Application Server Version 5.x nodes. You cannot create a JMS server on a node that is not running WebSphere Application Server Version 5.x, but existing Version 5.x JMS servers continue to be displayed, and you can modify their properties. However, you cannot use this page to delete a Version 5.x JMS server.

Name
Specifies a logical name for the server. For WebSphere Application Server, server names must be unique within a node.

Node
Specifies the node on which the server resides.

Host Name
Specifies the IP address, the full domain name system (DNS) host name with a domain name suffix, or the short DNS host name for the server.

Version
Specifies the version of the product on which the server runs.

Cluster Name
Specifies the name of the cluster to which the application server belongs. This field only displays when the **Include cluster members in the collection** console page preference is selected on the Applications server page.

If you have created clusters, and if the **Include cluster members in the collection** console page preference is selected, application servers that are cluster members are included in the list of application servers that displays on the Application servers page. These cluster members can be managed in the same manner as any of the other application servers in the list.

If the **Include cluster members in the collection** console page preference is not selected, application servers that are cluster members are not listed in the list of application servers that can be managed from this page.

Status
Specifies whether the server is started, stopped, partially stopped, or unavailable. If the status is unavailable, the node agent is not running in that node and you must restart the node agent before you can start the server.

	**Started**	The server is running.

	**Partially stopped**	The server is in the process of changing from a started state to a stopped state.
	**Stopped**	The server is not running.
	**Unknown**	The server status cannot be determined.

Application server settings

Use this page to configure an application server or a cluster member template. An application server is a server that provides services required to run enterprise applications. A cluster member template is the set of application server configuration settings that are assigned to new members of a cluster.

To view this administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name*.

On the **Configuration** tab, you can change field settings. You can also click **Installed applications** to view the status of applications that are running on this server. On the **Runtime** tab, you can view read only information. The **Runtime** tab is available only when the server is running.

Name

Specifies a logical name for the server. Server names must be unique within a node. However, for multiple nodes within a cluster, you might have different servers with the same server name as long as the server and node pair are unique. You cannot change the value that appears in this field.

For example, a server named *server1* in a node named *node1* in the same cluster with a server named *server1* in a node named *node2* is allowed. However, you cannot have two servers named *server1* in the same node. The product uses the server name for administrative actions, such as referencing the server in scripting.

Default server1

Run in development mode

Enabling this option might reduce application server start-up time because it changes some of the JVM settings, such as disabling bytecode verification, and reducing just-in-time (JIT) compiler compilation costs. Do not enable this setting on production servers. This setting is only available on an application server that is running in a Version 6.0 or later cell.

Specifies that you want to use the **-Xverify** and **-Xquickstart** JVM properties as startup values. Before selecting this option, add the **-Xverify** and **-Xquickstart** properties as generic arguments to the JVM configuration.

If you select this option, then you must save the configuration, and restart the server before this configuration change takes effect.

The default setting for this option is `false`, which indicates that the server does not start in development mode. Setting this option to `true` specifies that the server starts in development mode with settings that decrease server start-up time.

Data type Boolean
Default false

Parallel start

Select this field to start the server on multiple threads. This might shorten the startup time.

Specifies that you want the server components, services, and applications to start in parallel rather than sequentially.

The default setting for this option is `true`, which indicates that when the server starts, the server components, services, and applications start on multiple threads. Setting this option to `false` specifies that when the server starts, the server components, services, and applications start on a single thread, which might lengthen start-up time.

The order in which the applications start depends on the weights that you assign to them. Applications that have the same weight start 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. The more important an application is, the lower the startup order value should be. For example, you might specify a startup order value of `1` for your most important application, and a value of `2` for the next most important application. You might then specify a startup order of `3` for the next four applications because you want all four of those applications to start in parallel.

Data type	Integer
Default	1
Range	0 - 2147483647

Start components as needed

Select this property if you want the server components started as they are needed by an application that is running on this server.

When this property is selected, server components are dynamically started as they are needed. When this property is not selected, all of the server components are started during the server startup process. Therefore, selecting this option can improve startup time, and reduce the memory footprint of the server, because fewer components are started during the startup process.

Starting components as they are needed is most effective if all of the applications, that are deployed on the server, are of the same type. For example, using this option works better if all of your applications are Web applications that use servlets, and JavaServer Pages (JSP). This option works less effectively if your applications use servlets, JSPs and Enterprise JavaBeans™ (EJB).

Note: To ensure compatibility with other WebSphere products, the default setting for this option is deselected. Before selecting this option, verify that any other WebSphere products, that you are running in conjunction with this product, support this functionality.

Access to internal server classes

Specifies whether the applications that are running on this server can access multiple server implementation classes.

If you select `Allow`, then applications can access many of the server implementation classes. If you select `Restrict`, then applications cannot access multiple server implementation classes. The applications get a ClassNotFoundException error if they attempt to access those classes.

Usually you should select `Restrict` for this property, because most applications use the supported APIs and do not need to access any of the internal classes. However, if your application requires the use of one or more of the internal server classes, select `Allow` as the value for this property.

The default value for this property is `Allow`.

Class loader policy
Select whether there is a single class loader to load all applications or a different class loader for each application.

Class loading mode
Specifies whether the class loader searches in the parent class loader or in the application class loader first to load a class. The standard for Developer Kit class loaders and the product class loaders is `Parent first`.

This field only applies if you set the Class loader policy field to `S*ingle`.

If you select `Application first`, your application can override classes contained in the parent class loader, but this action can potentially result in ClassCastException or linkage errors if you have mixed use of overridden classes and non-overridden classes.

Process ID
The process ID for this server on the native operating system.

This property is read only. The system automatically generates the value.

Cell name
The name of the cell in which this server is running.

This property is read only.

Node name
The name of the node in which this server is running.

This property is read only.

State
The runtime start state for this server.

This property is read only.

Product information
This link under Additional properties, displays the product information for your installation of the product. This information includes the product name, ID, version, build date, and build level.

From the Product Information page, you can click on the following links for additional product information:

- Components, for a list of all of the components that are installed.
- e-Fixes, for a list of all of the service updates that are installed.
- Extensions, for a list of the extensions that are installed.
- History report, for a detailed report of all installation events that have occurred since the product was installed, such as the installation of a specific service level.
- Product report, for a detailed report of the versions of the product that are installed.
- PTFs, for a list of all of PTFs that are installed.

Ports collection
Use this page to view and manage communication ports used by run-time components running within a process. Communication ports provide host and port specifications for a server.

To view this administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Communications > Ports**.

This page displays only when you are working with ports for application servers.

Port Name:

Specifies the name of a port. Each name must be unique within the server.

Host:

Specifies the IP address, domain name server (DNS) host name with domain name suffix, or just the DNS host name, used by a client to request a resource (such as the naming service, or administrative service).

Port:

Specifies the port for which the service is configured to accept client requests. The port value is used in conjunction with the host name.

Transport Details:

Provides a link to the transport chains associated with this port. If no transport chains are associated with this port, the string ″No associated transports″ appears in this column.

Ports settings:

Use this to view and change the configuration for a communication port used by run-time components running within a process. A communication port provides host and port specifications for a server.

You can view this administrative console page by clicking one of the following paths:
- **Servers > Server Types > WebSphere application servers >** *server_name* **> Ports >** *port_name*
- **Servers > Server Types > JMS servers >** *server_name* **> Security Port Endpoint**
- **Servers > Server Types > WebSphere JMS servers >** *server_name* **> Ports >** *port_name*

Port Name:

Specifies the name of the port. The name must be unique within the server.

Note that this field displays only when you are defining a port for an application server. You can select either:

Well-known Port
When you select this option , you can select a previously defined port from the drop down list

User-defined Port
When you select this option, you must create a port with a new name by entering the name of the new port in the text box

Data type String

Host:

Specifies the IP address, domain name server (DNS) host name with domain name suffix, or just the DNS host name, used by a client to request a resource (such as the naming service, administrative service, or JMS broker).

For example, if the host name is `myhost`, the fully qualified DNS name can be `myhost.myco.com` and the IP address can be `155.123.88.201`.

Host names on the ports can be resolvable names or IP addresses. The server will bind to the specific host name or IP address that is supplied. That port will only be accessible through the IP address that is

resolved from the given host name or IP address. The IP address may be of the IPv4 (Internet Protocol Version 4) format for all platforms, and IPv6 (Internet Protocol Version 6) format on specific operating systems where the server supports IPv6.

Note: If your TCP/IP network is set up to use distributed dynamic virtual IP addresses (DVIPAs), and if the node agent is in the process of starting the application server, TCP/IP waits until the JVM TCP/IP timeout period expires before notifying the node agent that the target application server is not responsive.

Data type String
Default * (asterisk)

Port:

Specifies the port for which the service is configured to accept client requests. The port value is used in conjunction with the host name.

Port numbers in the server can be reused among multiple ports as long as they have host names that resolve to unique IP addresses and there is not a port with the same port number and a wildcard (*) host name. A port number is valid in the range of 0 and 65535. 0 specifies that the server should bind to any ephemeral port available. Specifying the wildcard value is equivalent to specifying the loopback address or 127.0.0.1.

Note: Port sharing cannot be created using the administrative console. If you need to share a port, you must use wsadmin commands to define that port. You must also make sure that the same discrimination weights are defined for all of the transport channels associated with that port.

Protocol channels only accept their own protocol. However, application channels usually accept anything that reaches them. Therefore, for application channels, such as WebContainer, or Proxy, you should specify larger discrimination weights when sharing levels with protocol channels, such as HTTP or SSL. The one exception to this rule is if you have application channels that perform discrimination tests faster than the protocol channels. For example, a JFAP channel is faster at deciding on a request than the SSL protocol channel, and should go first for performance reasons. However, the WebContainer and Proxy channels must always be last because they accept everything that is handed to them.

Data type Integer
Default None

Note: The following table lists server endpoints and their respective port ranges. In contrast to the z/OS environment, for a distributed platform or the i5/OS environment, the ORB_LISTENER_ADDRESS and the BOOTSTRAP_ADDRESS endpoints must not specify the same port.

Endpoint (port)	Acceptable values for the port field
BOOTSTRAP_ADDRESS	1 - 65536
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	1 - 65535
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	1 - 65535
DATAPOWERMGR_INBOUND_SECURE	1 - 65536
DCS_UNICAST_ADDRESS	1 - 65536
DRS_CLIENT_ADDRESS	1 - 65536
ORB_LISTENER_ADDRESS	0 - 65535 (If 0 is specified, the server starts on any available port.)

Endpoint (port)	Acceptable values for the port field
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	1 - 65535
SIB_ENDPOINT_ADDRESS	1 - 65536
SIB_ENDPOINT_SECURE_ADDRESS	1 - 65536
SIB_MQ_ENDPOINT_ADDRESS	1 - 65536
SIB_MQ_ENDPOINT_SECURE_ ADDRESS	1 - 65536
SOAP_CONNECTOR_ ADDRESS	1 - 65536
WC_adminhost	1 - 65536
WC_adminhost_secure	1 - 65536
WC_defaulthost	1 - 65536
WC_defaulthost_secure	1 - 65536
ORB_SSL_LISTENER_ADDRESS	Not supported for the distributed and iSeries environments

Custom property collection

Use this page to view and manage arbitrary name-value pairs of data, where the name is a property key and the value is a string value that can be used to set internal system configuration properties.

The administrative console contains several Custom Properties pages that work similarly. To view one of these administrative pages, click one of the **Custom properties** links.

Name:

Specifies the name (or key) for the property.

Each property name must be unique. If the same name is used for multiple properties, the value specified for the first property is used.

Do not start your property names with `was.` because this prefix is reserved for properties that are predefined in the application server.

Value:

Specifies the value paired with the specified name.

Description:

Provides information about the name-value pair.

Custom property settings:

Use this page to configure arbitrary name-value pairs of data, where the name is a property key and the value is a string value that can be used to set internal system configuration properties. Defining a new property enables you to configure a setting beyond that which is available in the administrative console.

Note: Setting custom properties at the server level is deprecated. However, you can specify a custom property for a server or the deployment manager as a WebSphere variable. Server scoped WebSphere variables still override any settings specified at the node scope, or higher, and are added to the was.env file.

To set a custom property for either the deployment manager, or an application server, as an environment variable, in the administrative console, click **Environment > WebSphere variables**, select the deployment manager or server from the pull-down list of available servers, nodes and cells, and then click **New**.

To view this administrative console page, click one of the following paths:
- For an application server, click **Servers > Server Types > WebSphere application servers >** *server_name*. Then, in the Server Infrastructure section, click **Administration > Custom properties**.
- For a JMS server, click **Servers > Server Types > JMS servers >** *server_name*. Then, in the Server Infrastructure section , click **Administration > Custom properties**.
- For a deployment manager, click **System Administration > Deployment manager**, and then in the Server Infrastructure section, click **Java and process management > Process definition > Java virtual machine > Custom properties**.

You can then click **New** to, create a new custom property, click on the name of an existing property to change its settings, or click **Delete** to delete an existing property.

Name:

Specifies the name (or key) for the property.

Each property name must be unique. If the same name is used for multiple properties, the value specified for the first property is used.

Do not start your property names with `was.` because this prefix is reserved for properties that are predefined in the product.

Data type String

Value:

Specifies the value paired with the specified name.

Data type String

Description:

Provides information about the name and value pair.

Data type String

Server component collection

Use this page to view information about and manage the types of server components that a specific application server uses during application processing. The list of server components varies according to the type of applications a specific application server processes.

For example, SIP Container might be listed as a server component for an application server that handles Session Initiation Protocol (SIP) requests, while EJB Container might be listed as a server component for an application server that handles Enterprise JavaBeans (EJB) requests. However, Messaging Server might be listed as a server component for both application servers.

You can also use this page to manage the settings for these server component, as they relate to request processing. In particular, you can specify either started or stopped as the initial state for the server component when the server process starts.

To view this administrative console page, click **System administration > Deployment Manager***server_name*. Then, in the Server Infrastructure section, click **Administration > Server components**.

To view this administrative console page for a node agent, click **System administration > Node agents >** *node_agent_name*. Then, in the Server Infrastructure section, click **Administration > Server components**.

Type:

Specifies the server component type, such as Name Server or Messaging Server.

Server component settings:

Use this page to view or configure a server component instance.

To view this administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name*. Then, in the Server Infrastructure section, click **Administration > Server components >** *server_component_name*.

Name:

Specifies the name of the component.

Data type	String

Initial State:

Specifies the desired state of the component when the server process starts. The options are: *Started* and *Stopped*. The default is *Started*.

Data type	String
Default	Started

Thread pool collection

Use this page to view and manage the thread pools that an application server uses. A thread pool enables components of the server to reuse threads, which eliminates the need to create new threads at run time. Creating new threads expends time and resources.

To view this administrative console page, you can choose more than one navigational route. For example, click **Servers** → **Server Types** → **WebSphere application servers** → **server** → **Thread pools**.

To view the settings for a specific thread pool, click the name of that thread pool.

To create a thread pool, click **New** and enter the information on the resulting panel.

To delete a thread pool, select the thread pool you want to delete, then click **Delete**.

Thread pool settings:

Use this page to configure a thread pool that an application server uses. A thread pool enables components of the server to reuse threads, which eliminates the need to create new threads at run time. Creating new threads expends time and resources.

To view this administrative console page, you can choose more than one navigational route. For example, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Thread pool**, and then select the thread pool you need to configure.

To configure the thread pool for the ORB Service, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Container services > ORB service**. Then, under Thread Pool Settings, either:

- Select Use the ORB.thread.pool settings associated with the Thread Pool Manager (recommended), and then click **ORB thread pool settings**, or
- Select Use the thread pool settings directly associated with the ORB service, and then click **Thread pool settings**.

Note: Because these console panels display information dynamically, you might not see all of the fields listed on any particular panel.

Name:

The name of the thread pool to create. The name must be unique within the server.

This field does not appear if you click **thread pool settings**.

Data type String

Description:

A text description of the thread pool.

This field does not appear if you click **thread pool settings**.

Data type String

Minimum size:

Specifies the minimum number of threads to allow in the pool. When an application server starts, no threads are initially assigned to the thread pool. Threads are added to the thread pool as the workload assigned to the application server requires them, until the number of threads in the pool equals the number specified in the Minimum size field. After this point in time, additional threads are added and removed as the workload changes. However the number of threads in the pool never decreases below the number specified in the Minimum size field, even if some of the threads are idle.

This field does not appear if you click **thread pool settings**.

Data type Integer
Default 10

Maximum size:

Specifies the maximum number of threads to maintain in the default thread pool.

If your Tivoli® Performance Viewer shows the Percent Maxed metric to remain consistently in the double digits, consider increasing the Maximum size. The Percent Maxed metric indicates the amount of time that the configured threads are used.

Data type Integer

Default	50
Recommended	50 (25 on Linux systems)

Thread inactivity timeout:

Specifies the number of milliseconds of inactivity that should elapse before a thread is reclaimed. A value of 0 indicates not to wait and a negative value (less than 0) means to wait forever.

Note: The administrative console does not allow you to set the inactivity timeout to a negative number. To do this you must modify the value directly in the server.xml file.

Data type	Integer
Units	Milliseconds
Default	3500

Allow thread allocation beyond maximum thread size:

Specifies whether the number of threads can increase beyond the maximum size configured for the thread pool.

The maximum number of threads that can be created is constrained only within the limits of the Java Virtual Machine and the operating system. When a thread pool, that is allowed to grow, expands beyond the maximum size, the additional threads are not reused and are discarded from the pool after processing of the work items for which they were created is completed. When additional threads are created, a message is logged in the SYSOUT file to let you know that you went beyond the maximum size set for the thread pool.

Data type	Boolean
Default	Not enabled (false)

Core group service settings

Use this page to set up the application server properties that relate to core groups.

To view this administrative console page, click **Servers > Server Types > Application servers >***server*. Then in the Additional Properties section, select **Core group service**.

Click **Save** to save and synchronize your changes with all managed nodes.

Enable service at server startup

Select if you want the core group service, also known as the high availability manager service, to start on this process when the server starts. The core group service must be started before high availability functions, such as routing, and failover, work properly.

Note: The default value for this setting is selected Before disabling the core group service for a server process, make sure that none of the components that this process uses require high availability functions.

Default	Core group service starts when the server starts.

Core group name

Specifies the name of the core group that contains this application server as a member. To move a server to a different core group, in the administrative console, click **Servers > Core groups > Core group settings >** *core_group* **> Core group servers**.

Data type String

Allow activation
Select if high availability group members can be activated on this application server.

Is alive timer
Specifies the time interval, in seconds, at which the high availability manager will check the health of all of the active high availability group members that are running in this application server process. An active group member is a member that is able to accept work. If a group member fails, the application server on which the group member resides is restarted. If -1 is specified, the timer is disabled. If 0 (zero) is specified, the default value of 120 seconds is used.

Note: The value specified for this property can be overridden for the high availability groups using a particular policy if the Is alive timer property for that policy specifies a different time interval. If the Is alive timer setting specified for a policy is greater than 0 (zero), the high availability manager uses that time interval, instead of the one specified at this level, when determining how frequently it should check the health of a high availability group member using that particular policy.

Data type Any integer between -1 and 600, inclusive
Default 120 seconds

Transport buffer size
Specifies the buffer size, in megabytes, of the underlying group communication transport. The minimum buffer size is 10 megabytes.

Data type String
Default 10 megabytes

Environment entries collection
Use this page to view and manage arbitrary name-value pairs of data, where the name is a environment entry key and the value is a string value that can be used to set internal system configuration environment entries.

To view this page, in the administrative console click **Servers > Server Types > WebSphere application servers >** *server_name*, and then under Server Infrastructure, click **Java and process management > Environment entries**.

Name
Specifies the name (or key) for the environment entry. The name is a string that is used to set an internal system configuration environment entry.

Each environment entry name must be unique. If the same name is used for multiple environment entries, the value specified for the first environment entry that has that name is used.

Do not start your environment entry names with `was.` because this prefix is reserved for environment entries that are predefined for WebSphere Application Server.

Value
Specifies the value paired with the specified name.

Description
Provides information about the name-value pair.

Environment entries settings

Use this page to configure arbitrary name-value pairs of data, where the name is an environment entry key and the value is a string value that can be used to set internal system configuration environment entries. Defining a new environment entry enables you to configure a setting beyond that which is available in the administrative console.

To view this page, in the administrative console click **Servers > Server Types > WebSphere application servers >** *server_name*. Under Server Infrastructure, click **Java and process management > Environment entries**. Then do one of the following:

- Click **New** to create a new environment entry.
- Click the name of an existing environment entry to change its settings,
- Select an existing environment entry and click **Delete** to delete that entry.

Name:

Specifies the name (or key) for the environment entry.

Each environment entry name must be unique. If the same name is used for multiple environment entries, the value specified for the first environment entry that has that name is used.

Do not start an environment entry name with `was.` because this prefix is reserved for environment entries that are predefined in WebSphere Application Server.

Data type String

Value:

Specifies the value paired with the specified name.

Data type String

Description:

Provides information about the name and value pair.

Data type String

Starting an application server

When you start an application server, a new server process starts. This new server process isis based on the process definition settings of the current server configuration.

Before you begin

Before you start an application server, verify that all of the application required resources are available. You must also start all prerequisite subsystems.

If you want server components to dynamically start as they are needed by the installed applications, verify that the **Start components as needed** option is selected in the configuration settings for the application server before you start the application server. Selecting this option can improve startup time, and reduce the memory footprint of the application server. Starting components as they are needed is most effective if all of the applications that are deployed on the server are of the same type. For example, using this option works better if all of your applications are Web applications that use servlets, and JavaServer Pages (JSP). This option works less effectively if your applications use servlets, JSPs and Enterprise JavaBeans (EJB).

Note: To ensure compatibility with other WebSphere products, the default setting for this option is deselected. Before selecting this option, verify that any other WebSphere products, that you are running in conjunction with this product, support this functionality.

About this task

The node agent for the node on which an application server resides must be running before you can start the application server.

This procedure for starting a server also typically applies to restarting a server. The one exception might be if a server fails and you want the recovery functions to complete their processing prior to new work being started on that server. In this situation, you must restart the server in recovery mode.

After you create a new application server definition, you can start, stop, or manage the new server using the administrative console, or you can use commands to complete these tasks for the new server.

After you start an application server, other processes might not immediately discover the running application server. Application servers are discovered by the node agent. However, node agents are discovered by the deployment manager. Even though node agents typically discover local application servers quickly, it might take a deployment manager up to 60 seconds to discover a node agent.

If you are using clusters, the **Initial State** property of the application server subcomponent is not intended to be used to control the state of individual servers in the cluster at the time the cluster is started. This property is intended only as a way to control the state of the subcomponent of a server. You should use the Server options on the administrative console, or the startServer and stopServer command line commands to start and stop the individual servers of a cluster

There are several options available for starting an application server.

- You can use the administrative console:
  1. Click **Servers > Server Types > WebSphere application servers** to determine the node agent on which the application server that you are starting resides.
  2. Click **System Administration > Node agents**, and verify that the node agent is running.

     If the node agent is not running, issue the `startNode` command. After a node agent completely stops running and remains stopped, you cannot remotely start the node agent from the Node Agents page. You must issue the `startNode` command to start the node agent on the node where it runs.
  3. Click **Servers > Server Types > WebSphere application servers** again and select the application server that you want to start.
  4. Click **Start**. You can view the status and any messages or logs to make sure the application server starts.
- Windows You can use the Start menu on a Microsoft Windows operating system. **Start > Programs > IBM WebSphere > Network Deployment > *n* > Profiles > *profile_name*Start the server**. You can check that the server has successfully started by checking the startServer.log file. If the server has successfully started, the last two lines of the startServer.log file reads:

  ```
  Server launched.  Waiting for initialization status.
  Server server1 open for e-business; process id is 1932.
  ```

  The startServer.log file is located in the *profile_root*\logs\server1 directory if you have installed your server with the default settings. The server name and process ID vary depending on your settings.
- Windows Solaris HP-UX Linux You can issue a startServer command.
- AIX You can issue a command from the command line. Use the startServer command from the *app_server_root*/bin directory: You can check that the server has successfully started by checking the startServer.log file. If the server has successfully started, the last two lines of the startServer.log file reads:

```
Server launched.  Waiting for initialization status.
Server server1 open for e-business; process id is 1932.
```

The startServer.log file is located in the *profile_root*/logs/server1/ directory.

Results

The specified server starts. To verify that the server is in start state, in the administrative console, click **Servers > Server Types > WebSphere application servers**.

What to do next

After the server starts, deploy the applications that you want to run on this server.

If you need to start an application server with standard Java debugging enabled:

1. In the administrative console, click **Servers > Server Types > WebSphere application servers**.
2. Click the name of the application server with the processes that you want to trace and debug.
3. Under Server Infrastructure, click **Java and process management > Process definition**.
4. Select **Java virtual machine**.
5. On the Java virtual machine page, select the **Debug mode** option to enable the standard Java debugger. Set **Debug mode** arguments, if they are needed.
6. Click **OK**.
7. Save the changes to a configuration file
8. Stop the application server.
9. Start the application server again as previously described.

Restarting an application server in recovery mode

When an application server instance with active transactions in progress restarts after a failure, the transaction service uses recovery logs to complete the recovery process. These logs, which each transactional resource maintains, are used to rerun any InDoubt transactions and return the overall system to a self-consistent state.

About this task

When you restart an application server in recovery mode:

- Transactional resources complete the actions in their recovery logs and then shut down. This action frees up any resource locks that the application server held prior to the failure.
- During the recovery period, only the subset of application server functions that are necessary for transactional recovery to proceed are available.
- The application server does not accept new work during the recovery process.
- The application server shuts down when the recovery is complete.

This recovery process begins as soon as all of the necessary subsystems within the application server are available. If the application server is not restarted in recovery mode, the application server can start accepting new work as soon as the server is ready, which might occur before the recovery work has completed.

Normally, this process is not a problem. However, situations exist when your operating procedures might not be compatible with supporting recovery work and new work simultaneously. For example, you might have a high availability environment where the work handled by the application server that failed is immediately moved to another application server. This backup application server then exclusively processes the work from the application server that failed until recovery has completed on the failed application server and the two application servers can be re-synchronized. In this situation, you might want

the failing application server to only perform its transactional recovery process and then shut down. You might not want this application server to start accepting new work while the recovery process is taking place.

To prevent the assignment of new work to an application server that is going through its transaction recovery process, restart the application server in recovery mode.

When you restart a failed application server, the node agent for the node on which the failed application server resides must be running before you can restart that application server.

If you want to be able restart an application server in recovery mode, you must perform the following steps before a failure occurs, and then restart the application server to enable your configuration changes:

- If the server is monitored by a node agent, you must clear the Automatic restart option for that server. Clearing this option prevents the node agent from automatically restarting the server in normal mode, before you have a chance to start it in recovery mode.
  1. In the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name*.
  2. In the Server Infrastructure section, click **Java and process management > Monitoring Policy**.
  3. Clear the Automatic restart option.
- If a catastrophic failure occurs that leaves InDoubt transactions, issue the **startServer** *server_name* **-recovery** command from the command line. This command restarts the server in recovery mode. You must issue the command from the *profile_root*/bin directory for the profile with which the server is associated.

Results

The application server restarts in recovery mode, performs transactional recovery, and shuts down. Any resource locks that the application server held prior to the failure are released.

Detecting and handling problems with runtime components

You must monitor the status of runtime components to ensure that, once started, they remain operational as needed.

1. Regularly examine the status of runtime components.

   Browse messages displayed under WebSphere Runtime Messages in the status area at the bottom of the console. The runtime event messages, marked with a red X, provide detailed information on event processing.
2. If an application stops running, examine the status of the application If an application stops running when it should be operational, examine the status of the application on an Applications page and try restarting the application. If messages indicate that a server has stopped running, use the Application servers page to try restarting the server. If a cluster of servers stops running, use the Server Cluster page to try to restart the cluster. If the status of an application server is Unavailable, the node agent is not running in that node and you must restart the node agent before you can start the server.
3. If the runtime components do not restart, reexamine the messages and read information on problem determination to help you to restart the components.

Stopping an application server

Stopping an application server ends a server process based on the process definition settings in the current application server configuration.

Before you begin

Make sure you understand the impact of stopping a particular server has on your ability to handle work requests, especially if you need to maintain a highly available environment.

About this task

There are times you need to stop an application server. For example, you might have to apply service to an application running on that server, or you might want to change one of the application server's configuration setting. Use one of the following options when you need to stop an application server.

- Windows You can use the Start menu to stop your application server. In the administrative console, click **Start > Programs > IBM WebSphere > Network Deployment v**. *n* **> Stop the server**. When the server stops successfully, the stopServer.log file contains the following in the last two lines:

  ```
  Server stop request issued.  Waiting for stop status.
  Server server1 stop completed.
  ```

  The server name varies depending on your settings.
- For the z/OS and distributed platforms, except AIX, you can issue the **stopServer** command from the command line to stop a single server or the **stopManager** command to stop the deployment manager.

  AIX You can issue the **stopServer** or the **stopManager** commands from the /usr/WebSphere/AppServer/bin directory:

  ```
  #  ./stopServer.sh server1
  #   ./stopManager.sh
  ```
- You can use the administrative console to stop an application server:
  1. In the administrative console, click **Servers > Server Types > WebSphere application servers**.
  2. Select the application server that you want stopped and click **Stop**.
  3. Confirm that you want to stop the application server.
  4. View the **Status** value and any messages or logs to see whether the application server stops.

Results

The specified server stops as soon as requests assigned to that server finish processing. To verify that the server is in stop state, in the administrative console, click **Servers > Server Types > WebSphere application servers**.

What to do next

If you experience any problems shutting down a server, see the *Troubleshooting and support* PDF.

Changing time zone settings

In some application environments, it is important that application server components use the same time zone. You can use the administrative console to ensure that your application components use the correct time zone.

Before you begin

Determine the scope at which you want to set the time zone value. You can set the time zone value such that is applies for an entire cell, for an entire node, or only for a specific server.

Remember that time zone IDs should include an offset and, in almost all cases, a daylight saving time zone name for consistent results. For example, specify EST5EDT for Eastern Standard Time, Daylight Savings Time.

HP-UX When the East African Time Zone (EAT) is specified as your time zone setting, the HP-UX operating system Java virtual machine (JVM) uses Greenwich Mean Time (GMT). Therefore, log file time stamps are based on GMT instead of EAT. The situation might also causes problems in server federation if you attempt to synchronize with servers that are running on an operating system whose JVM correctly handles the EAT.

HP-UX If you need to use East African Time Zone as the time zone setting for a specific function, instead of using the following procedure, add the -Duser.timezone=EAT parameter to the appropriate Java command. For example, to have an application server use EAT as its time zone setting, add the -Duser.timezone=EAT parameter to the startServer command.

About this task

To change the time zone setting for a single application server:

Complete one or more of the following actions to set appropriate time zone values for your environment.

- Set the time zone for all of your server processes.
  1. In the administrative console, click **Environment > WebSphere variables**\ .
  2. Select `All scopes` from the list of scope options.
  3. Set a value for the TZ variable.

     If the TZ variable is included in the list of defined variables, click **TZ**, and then specify a new time zone value in the **Variable** field.

     If the TZ variable is not included in the list of defined variables, click **New**, and then specify `TZ` in the **Name** field, and the appropriate time zone value in the **Value** field.

     For example, if you specify `TZ` in the **Name** field, and `EST5EDT` in the **Value** field, Eastern Daylight Savings is used as the time zone setting for all of your server processes.
- Set the time zone for all of the server processes in a particular cell.
  1. In the administrative console, click **Environment > WebSphere variables**\ .
  2. Select the cell for which you want to set the time zone value from the list of scope options.
  3. Set a value for the TZ variable.

     If the TZ variable is included in the list of defined variables, click **TZ**, and specify a new time zone value in the **Value** field.

     If the TZ variable is not included in the list of defined variables, click **New**, and then specify `TZ` in the **Name** field, and the appropriate time zone value in the **Value** field.

     For example, if you specify `TZ` in the **Name** field, and `EST5EDT` in the **Value** field, Eastern Daylight Savings is used as the time zone setting for all of your server processes that are running in that cell.
- Set the time zone for all of the server processes in a particular node.
  1. In the administrative console, click **Environment > WebSphere variables**\ .
  2. Select the node for which you want to set the time zone value from the list of scope options.
  3. Set a value for the TZ variable.

     If the TZ variable is not included in the list of defined variables, click **New**, and then specify `TZ` in the **Name** field, and the appropriate time zone value in the **Value** field.

     If the TZ variable is included in the list of defined variables, click **TZ**, and specify a new time zone value in the **Value** field.

     For example, if you specify `TZ` in the **Name** field, and `EST5EDT` in the **Value** field, Eastern Daylight Savings is used as the time zone setting for all of your server processes that are running in that node.
- Set the time zone for a specific server.
  1. In the administrative console, click **Environment > WebSphere variables**\ .
  2. Select the server for which you want to set the time zone value from the list of scope options.
  3. Set a value for the TZ variable.

     If the TZ variable is included in the list of defined variables, click **TZ**, and then specify a new time zone value in the **Value** field.

     If the TZ variable is not included in the list of defined variables, click **New**, and then specify `TZ` in the **Name** field, and the appropriate time zone value in the **Value** field.

For example, if you specify `TZ` in the **Name** field, and `EST5EDT` in the **Value** field, Eastern Daylight Savings is used as the time zone setting for all of your server processes.

- Click **Apply**, and then click **Save** to save your changes.
- Stop and restart all of the affected application server that were running when you made the time zone changes.

Results

Your new time zone setting are in affect for the designated servers.

Time zone IDs that can be specified for the user.timezone property

The following table lists the time zone IDs that you can specify for the user.timezone property.

- The **Time zone ID** column lists time zones, in boldface, and the locations within each time zone.
- The **Raw offset** column lists the difference, in hours and minutes, between Greenwich Mean Time (GMT) and the specified time zone.
- The **DST offset** column lists the offset, in minutes, for Daylight Savings Time (DST). If the field is blank, the time zone does not use DST.
- The **Display name** column lists the names of the time zones.
- The **QTIMZON variable** column only applies to the i5/OS operating system. The **QTIMZON variable** column lists the corresponding value for the QTIMZON system variable. If multiple values are specified in this column, either value is acceptable.

Note: The United States and Canada are making changes to the Daylight Saving Time start and end dates. The Technote Changes to Daylight Saving Time will affect IBM WebSphere Application Server and its associated Operating Systems, that is available on the Support Web site, provides the latest information on service updates that are being made to support these changes.

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Etc/GMT+12	-12 : 00		GMT-12:00	
Etc/GMT+11	-11 : 00		GMT-11:00	
MIT	-11 : 00		West Samoa Time	
Pacific/Apia	-11 : 00		West Samoa Time	QN1100UTCS
Pacific/Midway	-11 : 00		Samoa Standard Time	
Pacific/Niue	-11 : 00		Niue Time	
Pacific/Pago_Pago	-11 : 00		Samoa Standard Time	
Pacific/Samoa	-11 : 00		Samoa Standard Time	
US/Samoa	-11 : 00		Samoa Standard Time	
America/Adak	-10 : 00	60	Hawaii-Aleutian Standard Time	QN1000HAST
America/Atka	-10 : 00	60	Hawaii-Aleutian Standard Time	
Etc/GMT+10	-10 : 00		GMT-10:00	
HST	-10 : 00		Hawaii Standard Time	
Pacific/Fakaofo	-10 : 00		Tokelau Time	
Pacific/Honolulu	-10 : 00		Hawaii Standard Time	QN1000UTCS
Pacific/Johnston	-10 : 00		Hawaii Standard Time	
Pacific/Rarotonga	-10 : 00		Cook Is. Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Pacific/Tahiti	-10 : 00		Tahiti Time	
SystemV/HST10	-10 : 00		Hawaii Standard Time	
US/Aleutian	-10 : 00	60	Hawaii-Aleutian Standard Time	
US/Hawaii	-10 : 00		Hawaii Standard Time	
Pacific/Marquesas	-9 : 30		Marquesas Time	
AST	-9 : 00	60	Alaska Standard Time	QN0900AST
America/Anchorage	-9 : 00	60	Alaska Standard Time	
America/Juneau	-9 : 00	60	Alaska Standard Time	
America/Nome	-9 : 00	60	Alaska Standard Time	
America/Yakutat	-9 : 00	60	Alaska Standard Time	
Etc/GMT+9	-9 : 00		GMT-09:00	
Pacific/Gambier	-9 : 00		Gambier Time	QN0900UTCS
SystemV/YST9	-9 : 00	60	Alaska Standard Time	
US/Alaska	-9 : 00	60	Alaska Standard Time	
America/Dawson	-8 : 00	60	Pacific Standard Time	
America/Ensenada	-8 : 00	60	Pacific Standard Time	
America/Los_Angeles	-8 : 00	60	Pacific Standard Time	
America/Tiajuana	-8 : 00	60	Pacific Standard Time	
America/Vancouver	-8 : 00	60	Pacific Standard Time	
America/Whitehorse	-8 : 00	60	Pacific Standard Time	
Canada/Pacific	-8 : 00	60	Pacific Standard Time	
Canada/Yukon	-8 : 00	60	Pacific Standard Time	
Etc/GMT+8	-8 : 00		GMT-08:00	
Mexico/BajaNorte	-8 : 00	60	Pacific Standard Time	
PST	-8 : 00	60	Pacific Standard Time	QN0800PST, QN0800U
PST8PDT	-8 : 00	60	Pacific Standard Time	
Pacific/Pitcairn	-8 : 00		Pitcairn Standard Time	QN0800UTCS
SystemV/PST8	-8 : 00		Pitcairn Standard Time	
SystemV/PST8PDT	-8 : 00	60	Pacific Standard Time	
US/Pacific	-8 : 00	60	Pacific Standard Time	
US/Pacific-New	-8 : 00	60	Pacific Standard Time	
America/Boise	-7 : 00	60	Mountain Standard Time	
America/Cambridge_Bay	-7 : 00	60	Mountain Standard Time	
America/Chihuahua	-7 : 00	60	Mountain Standard Time	
America/Dawson_Creek	-7 : 00		Mountain Standard Time	
America/Denver	-7 : 00	60	Mountain Standard Time	
America/Edmonton	-7 : 00	60	Mountain Standard Time	
America/Hermosillo	-7 : 00		Mountain Standard Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
America/Inuvik	-7 : 00	60	Mountain Standard Time	
America/Mazatlan	-7 : 00	60	Mountain Standard Time	
America/Phoenix	-7 : 00		Mountain Standard Time	QN0700MST2, QN0700UTCS
America/Shiprock	-7 : 00	60	Mountain Standard Time	
America/Yellowknife	-7 : 00	60	Mountain Standard Time	
Canada/Mountain	-7 : 00	60	Mountain Standard Time	
Etc/GMT+7	-7 : 00		GMT-07:00	
MST	-7 : 00	60	Mountain Standard Time	QN0700MST, QN0700T
MST7MDT	-7 : 00	60	Mountain Standard Time	
Mexico/BajaSur	-7 : 00	60	Mountain Standard Time	
Navajo	-7 : 00	60	Mountain Standard Time	
PNT	-7 : 00	60	Mountain Standard Time	
SystemV/MST7	-7 : 00		Mountain Standard Time	
SystemV/MST7MDT	-7 : 00	60	Mountain Standard Time	
UA/Arizona	-7 : 00		Mountain Standard Time	
US/Mountain	-7 : 00	60	Mountain Standard Time	
America/Belize	-6 : 00		Central Standard Time	
America/Cancun	-6 : 00	60	Central Standard Time	
America/Chicago	-6 : 00	60	Central Standard Time	
America/Costa_Rica	-6 : 00		Central Standard Time	QN0600UTCS
America/El_Salvador	-6 : 00		Central Standard Time	
America/Guatemala	-6 : 00		Central Standard Time	
America/Managua	-6 : 00		Central Standard Time	
America/Menominee	-6 : 00	60	Central Standard Time	
America/Merida	-6 : 00	60	Central Standard Time	
America/Mexico_City	-6 : 00	60	Central Standard Time	
America/Monterrey	-6 : 00	60	Central Standard Time	
America/North_Dakota/Center	-6 : 00	60	Central Standard Time	
America/Rainy_River	-6 : 00	60	Central Standard Time	
America/Rankin_Inlet	-6 : 00	60	Central Standard Time	
America/Regina	-6 : 00		Central Standard Time	
America/Swift_Current	-6 : 00		Central Standard Time	
America/Tegucigalpa	-6 : 00		Central Standard Time	
America/Winnipeg	-6 : 00	60	Central Standard Time	
CST	-6 : 00	60	Central Standard Time	QN0600CST, QN600S
CST6CDT	-6 : 00	60	Central Standard Time	
Canada/Central	-6 : 00	60	Central Standard Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Canada/East-Saskatchewan	-6 : 00		Central Standard Time	
Canada/Saskatchewan	-6 : 00		Central Standard Time	
Chile/EasterIsland	-6 : 00	60	Easter Is.Time	
Etc/GMT+6	-6 : 00		GMT-06:00	
Mexico/General	-6 : 00	60	Central Standard Time	
Pacific/Easter	-6 : 00	60	Easter Is. Time	
Pacific/Galapagos	-6 : 00		Galapagos Time	
Pacific/Easter	-6 : 00	60	Easter Is. Time	
Pacific/Galapagos	-6 : 00		Galapagos Time	
SystemV/CST6	-6 : 00		Central Standard Time	
SystemV/CST6CDT	-6 : 00	60	Central Standard Time	
US/Central	-6 : 00	60	Central Standard Time	
America/Bogota	-5 : 00		Colombia Time	
America/Cayman	-5 : 00		Eastern Standard Time	
America/Detroit	-5 : 00	60	Eastern Standard Time	
America/Eirunepe	-5 : 00		Acre Time	
America/Fort_Wayne	-5 : 00		Eastern Standard Time	
America/Grand_Turk	-5 : 00	60	Eastern Standard Time	
America/Guayaquil	-5 : 00		Ecuador Time	
America/Havana	-5 : 00	60	Central Standard Time	
America/Indiana/Indianapolis	-5 : 00		Eastern Standard Time	
America/Indiana/Knox	-5 : 00		Eastern Standard Time	
America/Indiana/Marengo	-5 : 00		Eastern Standard Time	
America/Indiana/Vevay	-5 : 00		Eastern Standard Time	
America/Indianapolis	-5 : 00		Eastern Standard Time	QN0500UTCS
America/Iqaluit	-5 : 00	60	Eastern Standard Time	
America/Jamaica	-5 : 00		Eastern Standard Time	
America/Kentucky/Louisville	-5 : 00	60	Eastern Standard Time	
America/Kentucky/Monticello	-5 : 00	60	Eastern Standard Time	
America/Knox_IN	-5 : 00		Eastern Standard Time	
America/Lima	-5 : 00		Peru Time	
America/Louisville	-5 : 00	60	Eastern Standard Time	
America/Montreal	-5 : 00	60	Eastern Standard Time	
America/Nassau	-5 : 00	60	Eastern Standard Time	
America/New_York	-5 : 00	60	Eastern Standard Time	
America/Nipigon	-5 : 00	60	Eastern Standard Time	
America/Panama	-5 : 00		Eastern Standard Time	
America/Pangnirtung	-5 : 00	60	Eastern Standard Time	
America/Port-au-Prince	-5 : 00		Eastern Standard Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
America/Porto_Acre	-5 : 00		Acre Time	
America/Rio_Branco	-5 : 00		Acre Time	
America/Thunder_Bay	-5 : 00	60	Eastern Standard Time	
Brazil/Acre	-5 : 00		Acre Time	
Canada/Eastern	-5 : 00	60	Eastern Standard Time	
Cuba	-5 : 00	60	Central Standard Time	
EST	-5 : 00	60	Eastern Standard Time	QN0500EST
EST5EDT	-5 : 00	60	Eastern Standard Time	
Etc/GMT+5	-5 : 00		GMT-05:00	
IET	-5 : 00		Eastern Standard Time	QN0500EST2
Jamaica	-5 : 00		Eastern Standard Time	
SystemV/EST5	-5 : 00		Eastern Standard Time	
SystemV/EST5EDT	-5 : 00	60	Eastern Standard Time	
US/East-Indiana	-5 : 00		Eastern Standard Time	
US/Eastern	-5 : 00	60	Eastern Standard Time	
US/Indiana-Starke	-5 : 00		Eastern Standard Time	
US/Michigan	-5 : 00	60	Eastern Standard Time	
America/Anguilla	-4 : 00		Atlantic Standard Time	
America/Antigua	-4 : 00		Atlantic Standard Time	
America/Aruba	-4 : 00		Atlantic Standard Time	
America/Asuncion	-4 : 00	60	Paraguay Time	
America/Barbados	-4 : 00		Atlantic Standard Time	
America/Boa_Vista	-4 : 00		Amazon Standard Time	
America/Caracas	-4 : 00		Venezuela Time	QN0400UTC2
America/Cuiaba	-4 : 00	60	Amazon Standard Time	
America/Curacao	-4 : 00		Atlantic Standard Time	
America/Dominica	-4 : 00		Atlantic Standard Time	
America/Glace_Bay	-4 : 00	60	Atlantic Standard Time	
America/Goose_Bay	-4 : 00	60	Atlantic Standard Time	
America/Grenada	-4 : 00		Atlantic Standard Time	
America/Guadeloupe	-4 : 00		Atlantic Standard Time	
America/Guyana	-4 : 00		Guyana Time	
America/Halifax	-4 : 00	60	Atlantic Standard Time	
America/La_Paz	-4 : 00		Bolivia Time	
America/Manaus	-4 : 00		Amazon Standard Time	
America/Martinique	-4 : 00		Atlantic Standard Time	
America/Montserrat	-4 : 00		Atlantic Standard Time	
America/Port_of_Spain	-4 : 00		Atlantic Standard Time	
America/Porto_Velho	-4 : 00		Amazon Standard Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
America/Puerto_Rico	-4 : 00		Atlantic Standard Time	QN0400UTCS
America/Santiago	-4 : 00	60	Chile Time	
America/Santo_Domingo	-4 : 00		Atlantic Standard Time	
America/St_Kitts	-4 : 00		Atlantic Standard Time	
America/St_Lucia	-4 : 00		Atlantic Standard Time	
America/St_Thomas	-4 : 00		Atlantic Standard Time	
America/St_Vincent	-4 : 00		Atlantic Standard Time	
America/Thule	-4 : 00	60	Atlantic Standard Time	
America/Tortola	-4 : 00		Atlantic Standard Time	
America/Virgin	-4 : 00		Atlantic Standard Time	
Antarctica/Palmer	-4 : 00	60	Chile Time	
Atlantic/Bermuda	-4 : 00	60	Atlantic Standard Time	QN0400AST
Atlantic/Stanley	-4 : 00	60	Falkland Is. Time	
Brazil/West	-4 : 00		Amazon Standard Time	
Canada/Atlantic	-4 : 00	60	Atlantic Standard Time	
Chile/Continental	-4 : 00	60	Chile Time	
Etc/GMT+4	-4 : 00		GMT-04:00	
PRT	-4 : 00		Atlantic Standard Time	
SystemV/AST4	-4 : 00		Atlantic Standard Time	
SystemV/AST4ADT	-4 : 00	60	Atlantic Standard Time	
America/St_Johns	-3 : 30	60	Newfoundland Standard Time	
CNT	-3 : 30	60	Newfoundland Standard Time	QN0330NST
Canada/Newfoundland	-3 : 30	60	Newfoundland Standard Time	
AGT	-3 : 00		Argentine Time	
America/Araguaina	-3 : 00	60	Brazil Time	
America/Belem	-3 : 00		Brazil Time	
America/Buenos_Aires	-3 : 00		Argentine Time	QN0300UTCS
America/Catamarca	-3 : 00		Argentine Time	
America/Cayenne	-3 : 00		French Guiana Time	
America/Cordoba	-3 : 00		Argentine Time	
America/Fortaleza	-3 : 00		Brazil Time	
America/Godthab	-3 : 00	60	Western Greenland Time	
America/Jujuy	-3 : 00		Argentine Time	
America/Maceio	-3 : 00		Brazil Time	
America/Mendoza	-3 : 00		Argentine Time	
America/Miquelon	-3 : 00	60	Pierre & Miquelon Standard Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
America/Montevideo	-3 : 00		Uruguay Time	
America/Paramaribo	-3 : 00		Suriname Time	
America/Recife	-3 : 00		Brazil Time	
America/Rosario	-3 : 00		Argentine Time	
America/Sao_Paulo	-3 : 00	60	Brazil Time	
Antarctica/Rothera	-3 : 00		Rothera Time	
BET	-3 : 00	60	Brazil Time	QN0300UTC2
Brazil/East	-3 : 00	60	Brazil Time	
Etc/GMT+3	-3 : 00		GMT-03:00	
America/Noronha	-2 : 00		Fernando de Noronha Time	QN0200UTCS
Atlantic/South_Georgia	-2 : 00		South Georgia Standard Time	
Brazil/DeNoronha	-2 : 00		Fernando de Noronha Time	
Etc/GMT+2	-2 : 00		GMT-02:00	
America/Scoresbysund	-1 : 00	60	Eastern Greenland Time	
Atlantic/Azores	-1 : 00	60	Azores Time	
Atlantic/Cape_Verde	-1 : 00		Cape Verde Time	QN0100UTCS
Etc/GMT+1	-1 : 00		GMT-01:00	
Africa/Abidjan	0 : 00		Greenwich Mean Time	
Africa/Accra	0 : 00		Greenwich Mean Time	
Africa/Bamako	0 : 00		Greenwich Mean Time	
Africa/Banjul	0 : 00		Greenwich Mean Time	
Africa/Bissau	0 : 00		Greenwich Mean Time	
Africa/Casablanca	0 : 00		Western European Time	
Africa/Conakry	0 : 00		Greenwich Mean Time	
Africa/Dakar	0 : 00		Greenwich Mean Time	
Africa/El_Aaiun	0 : 00		Western European Time	
Africa/Freetown	0 : 00		Greenwich Mean Time	
Africa/Lome	0 : 00		Greenwich Mean Time	
Africa/Monrovia	0 : 00		Greenwich Mean Time	
Africa/Nouakchott	0 : 00		Greenwich Mean Time	
Africa/Ouagadougou	0 : 00		Greenwich Mean Time	
Africa/Sao_Tome	0 : 00		Greenwich Mean Time	
Africa/Timbuktu	0 : 00		Greenwich Mean Time	
America/Danmarkshavn	0 : 00		Greenwich Mean Time	
Atlantic/Canary	0 : 00	60	Western European Time	
Atlantic/Faeroe	0 : 00	60	Western European Time	
Atlantic/Madeira	0 : 00	60	Western European Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Atlantic/Reykjavik	0 : 00		Greenwich Mean Time	
Atlantic/St_Helena	0 : 00		Greenwich Mean Time	
Eire	0 : 00	60	Greenwich Mean Time	
Etc/GMT	0 : 00		GMT+00:00	
Etc/GMT+0	0 : 00		GMT+00:00	
Etc/GMT-0	0 : 00		GMT+00:00	
Etc/GMT0	0 : 00		GMT+00:00	
Etc/Greenwich	0 : 00		Greenwich Mean Time	
Etc/UCT	0 : 00		Coordinated Universal Time	
Etc/UTC	0 : 00		Coordinated Universal Time	
Etc/Universal	0 : 00		Coordinated Universal Time	
Etc/Zulu	0 : 00		Coordinated Universal Time	
Europe/Belfast	0 : 00	60	Greenwich Mean Time	
Europe/Dublin	0 : 00	60	Greenwich Mean Time	
Europe/Lisbon	0 : 00	60	Western European Time	
Europe/London	0 : 00	60	Greenwich Mean Time	Q0000GMT2
GB	0 : 00	60	Greenwich Mean Time	
GB-Eire	0 : 00	60	Greenwich Mean Time	
GMT	0 : 00		Greenwich Mean Time	Q0000GMT
GMT0	0 : 00		GMT+00:00	
Greenwich	0 : 00		Greenwich Mean Time	
Iceland	0 : 00		Greenwich Mean Time	
Portugal	0 : 00	60	Western European Time	
UCT	0 : 00		Coordinated Universal Time	
UTC	0 : 00		Coordinated Universal Time	Q0000UTC
Universal	0 : 00		Coordinated Universal Time	
WET	0 : 00	60	Western European Time	
Zulu	0 : 00		Coordinated Universal Time	
Africa/Algiers	1 : 00		Central European Time	QP0100CET, QP0100UTCS
Africa/Bangui	1 : 00		Western African Time	
Africa/Brazzaville	1 : 00		Western African Time	
Africa/Ceuta	1 : 00	60	Central European Time	
Africa/Douala	1 : 00		Western African Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Africa/Kinshasa	1 : 00		Western African Time	
Africa/Lagos	1 : 00		Western African Time	
Africa/Libreville	1 : 00		Western African Time	
Africa/Luanda	1 : 00		Western African Time	
Africa/Malabo	1 : 00		Western African Time	
Africa/Ndjamena	1 : 00		Western African Time	
Africa/Niamey	1 : 00		Western African Time	
Africa/Porto-Novo	1 : 00		Western African Time	
Africa/Tunis	1 : 00		Central European Time	
Africa/Windhoek	1 : 00	60	Western African Time	
Arctic/Longyearbyen	1 : 00	60	Central European Time	
Atlantic/Jan_Mayen	1 : 00	60	Eastern Greenland Time	
CET	1 : 00	60	Central European Time	
ECT	1 : 00	60	Central European Time	QP0100CET3
Etc/GMT-1	1 : 00		GMT+01:00	
Europe/Amsterdam	1 : 00	60	Central European Time	
Europe/Andorra	1 : 00	60	Central European Time	
Europe/Belgrade	1 : 00	60	Central European Time	
Europe/Berlin	1 : 00	60	Central European Time	
Europe/Bratislava	1 : 00	60	Central European Time	
Europe/Brussels	1 : 00	60	Central European Time	
Europe/Budapest	1 : 00	60	Central European Time	
Europe/Copenhagen	1 : 00	60	Central European Time	
Europe/Gibraltar	1 : 00	60	Central European Time	
Europe/Ljubljana	1 : 00	60	Central European Time	
Europe/Luxembourg	1 : 00	60	Central European Time	
Europe/Madrid	1 : 00	60	Central European Time	
Europe/Malta	1 : 00	60	Central European Time	
Europe/Monaco	1 : 00	60	Central European Time	
Europe/Oslo	1 : 00	60	Central European Time	
Europe/Paris	1 : 00	60	Central European Time	
Europe/Prague	1 : 00	60	Central European Time	
Europe/Rome	1 : 00	60	Central European Time	
Europe/San_Marino	1 : 00	60	Central European Time	
Europe/Sarajevo	1 : 00	60	Central European Time	
Europe/Skopje	1 : 00	60	Central European Time	
Europe/Stockholm	1 : 00	60	Central European Time	
Europe/Tirane	1 : 00	60	Central European Time	
Europe/Vaduz	1 : 00	60	Central European Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Europe/Vatican	1 : 00	60	Central European Time	
Europe/Vienna	1 : 00	60	Central European Time	
Europe/Warsaw	1 : 00	60	Central European Time	
Europe/Zagreb	1 : 00	60	Central European Time	
Europe/Zurich	1 : 00	60	Central European Time	QP0100CET2
MET	1 : 00	60	Middle Europe Time	
Poland	1 : 00	60	Central European Time	
ART	2 : 00	60	Eastern European Time	
Africa/Blantyre	2 : 00		Central African Time	
Africa/Bujumbura	2 : 00		Central African Time	
Africa/Cairo	2 : 00	60	Eastern European Time	
Africa/Gaborone	2 : 00		Central African Time	
Africa/Harare	2 : 00		Central African Time	
Africa/Johannesburg	2 : 00		South Africa Standard Time	QP0200SAST
Africa/Kigali	2 : 00		Central African Time	
Africa/Lubumbashi	2 : 00		Central African Time	
Africa/Lusaka	2 : 00		Central African Time	
Africa/Maputo	2 : 00		Central African Time	
Africa/Maseru	2 : 00		South Africa Standard Time	
Africa/Mbabane	2 : 00		South Africa Standard Time	
Africa/Tripoli	2 : 00		Eastern European Time	
Asia/Amman	2 : 00	60	Eastern European Time	
Asia/Beirut	2 : 00	60	Eastern European Time	
Asia/Damascus	2 : 00	60	Eastern European Time	
Asia/Gaza	2 : 00	60	Eastern European Time	
Asia/Istanbul	2 : 00	60	Eastern European Time	
Asia/Jerusalem	2 : 00	60	Israel Standard Time	
Asia/Nicosia	2 : 00	60	Eastern European Time	
Asia/Tel_Aviv	2 : 00	60	Israel Standard Time	
CAT	2 : 00		Central African Time	
EET	2 : 00	60	Eastern European Time	QP0200EET
Egypt	2 : 00	60	Eastern European Time	
Etc/GMT-2	2 : 00		GMT+02:00	
Europe/Athens	2 : 00	60	Eastern European Time	
Europe/Bucharest	2 : 00	60	Eastern European Time	
Europe/Chisinau	2 : 00	60	Eastern European Time	
Europe/Helsinki	2 : 00	60	Eastern European Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Europe/Istanbul	2 : 00	60	Eastern European Time	
Europe/Kaliningrad	2 : 00	60	Eastern European Time	
Europe/Kiev	2 : 00	60	Eastern European Time	
Europe/Minsk	2 : 00	60	Eastern European Time	
Europe/Nicosia	2 : 00	60	Eastern European Time	
Europe/Riga	2 : 00	60	Eastern European Time	
Europe/Simferopol	2 : 00	60	Eastern European Time	
Europe/Sofia	2 : 00	60	Eastern European Time	
Europe/Tallinn	2 : 00	60	Eastern European Time	QP0200EET2, QP0200UTCS
Europe/Tiraspol	2 : 00	60	Eastern European Time	
Europe/Uzhgorod	2 : 00	60	Eastern European Time	
Europe/Vilnius	2 : 00	60	Eastern European Time	
Europe/Zaporozhye	2 : 00	60	Eastern European Time	
Israel	2 : 00	60	Israel Standard Time	
Libya	2 : 00		Eastern European Time	
Turkey	2 : 00	60	Eastern European Time	
Africa/Addis_Ababa	3 : 00		Eastern African Time	QP0300UTCS
Africa/Asmera	3 : 00		Eastern African Time	
Africa/Dar_es_Salaam	3 : 00		Eastern African Time	
Africa/Djibouti	3 : 00		Eastern African Time	
Africa/Kampala	3 : 00		Eastern African Time	
Africa/Khartoum	3 : 00		Eastern African Time	
Africa/Mogadishu	3 : 00		Eastern African Time	
Africa/Nairobi	3 : 00		Eastern African Time	
Antarctica/Syowa	3 : 00		Syowa Time	
Asia/Aden	3 : 00		Arabia Standard Time	
Asia/Baghdad	3 : 00	60	Arabia Standard Time	
Asia/Bahrain	3 : 00		Arabia Standard Time	
Asia/Kuwait	3 : 00		Arabia Standard Time	
Asia/Qatar	3 : 00		Arabia Standard Time	
Asia/Riyadh	3 : 00		Arabia Standard Time	
EAT	3 : 00		Eastern African Time	
Etc/GMT-3	3 : 00		GMT+03:00	
Europe/Moscow	3 : 00	60	Moscow Standard Time	
Indian/Antananarivo	3 : 00		Eastern African Time	
Indian/Comoro	3 : 00		Eastern African Time	
Indian/Mayotte	3 : 00		Eastern African Time	
W-SU	3 : 00	60	Moscow Standard Time	
Asia/Riyadh87	3 : 07		GMT+03:07	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Asia/Riyadh88	3 : 07		GMT+03:07	
Asia/Riyadh89	3 : 07		GMT+03:07	
Mideast/Riyadh87	3 : 07		GMT+03:07	
Mideast/Riyadh88	3 : 07		GMT+03:07	
Mideast/Riyadh89	3 : 07		GMT+03:07	
Asia/Tehran	3 : 30	60	Iran Standard Time	
Iran	3 : 30	60	Iran Standard Time	
Asia/Aqtau	4 : 00	60	Aqtau Time	QP0400UTC2
Asia/Baku	4 : 00	60	Azerbaijan Time	
Asia/Dubai	4 : 00		Gulf Standard Time	QP0400UTCS
Asia/Muscat	4 : 00		Gulf Standard Time	
Asia/Oral	4 : 00	60	Oral Time	
Asia/Tbilisi	4 : 00	60	Georgia Time	
Asia/Yerevan	4 : 00	60	Armenia Time	
Etc/GMT-4	4 : 00		GMT+04:00	
Europe/Samara	4 : 00	60	Samara Time	
Indian/Mahe	4 : 00		Seychelles Time	
Indian/Mauritius	4 : 00		Mauritius Time	
Indian/Reunion	4 : 00		Reunion Time	
NET	4 : 00	60	Armenia Time	
Asia/Kabul	4 : 30		Afghanistan Time	
Asia/Aqtobe	5 : 00	60	Aqtobe Time	QP0500UTC2
Asia/Ashgabat	5 : 00		Turkmenistan Time	
Asia/Ashkhabad	5 : 00		Turkmenistan Time	
Asia/Bishkek	5 : 00	60	Kirgizstan Time	
Asia/Dushanbe	5 : 00		Tajikistan Time	
Asia/Karachi	5 : 00		Pakistan Time	QP0500UTCS
Asia/Samarkand	5 : 00		Turkmenistan Time	
Asia/Tashkent	5 : 00		Uzbekistan Time	
Asia/Yekaterinburg	5 : 00	60	Yekaterinburg Time	
Etc/GMT-5	5 : 00		GMT+05:00	
Indian/Kerguelen	5 : 00		French Southern & Antarctic Lands Time	
Indian/Maldives	5 : 00		Maldives Time	
PLT	5 : 00		Pakistan Time	
Asia/Calcutta	5 : 30		India Standard Time	
IST	5 : 30		India Standard Time	QP0530IST
Asia/Katmandu	5 : 45		Nepal Time	
Antarctica/Mawson	6 : 00		Mawson Time	
Antarctica/Vostok	6 : 00		Vostok Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Asia/Almaty	6 : 00	60	Alma-Ata Time	QP0600UTC2
Asia/Colombo	6 : 00		Sri Lanka Time	
Asia/Dacca	6 : 00		Bangladesh Time	
Asia/Dhaka	6 : 00		Bangladesh Time	QP0600UTCS
Asia/Novosibirsk	6 : 00	60	Novosibirsk Time	
Asia/Omsk	6 : 00	60	Omsk Time	
Asia/Qyzylorda	6 : 00	60	Qyzylorda Time	
Asia/Thimbu	6 : 00		Bhutan Time	
Asia/Thimphu	6 : 00		Bhutan Time	
BST	6 : 00		Bangladesh Time	
Etc/GMT-6	6 : 00		GMT+06:00	
Indian/Chagos	6 : 00		Indian Ocean Territory Time	
Asia/Rangoon	6 : 30		Myanmar Time	
Indian/Cocos	6 : 30		Cocos Islands Time	
Antarctica/Davis	7 : 00		Davis Time	
Asia/Bangkok	7 : 00		Indochina Time	
Asia/Hovd	7 : 00		Hovd Time	
Asia/Jakarta	7 : 00		West Indonesia Time	QP0700WIB
Asia/Krasnoyarsk	7 : 00	60	Krasnoyarsk Time	
Asia/Phnom_Penh	7 : 00		Indochina Time	
Asia/Pontianak	7 : 00		West Indonesia Time	
Asia/Saigon	7 : 00		Indochina Time	QP0700UTCS
Asia/Vientiane	7 : 00		Indochina Time	
Etc/GMT-7	7 : 00		GMT+07:00	
Indian/Christmas	7 : 00		Christmas Island Time	
VST	7 : 00		Indochina Time	
Antarctica/Casey	8 : 00		Western Standard Time (Australia)	
Asia/Brunei	8 : 00		Brunei Time	
Asia/Chongqing	8 : 00		China Standard Time	
Asia/Chungking	8 : 00		China Standard Time	
Asia/Harbin	8 : 00		China Standard Time	
Asia/Hong_Kong	8 : 00		Hong Kong Time	QP0800JIST, QP0800UTCS
Asia/Irkutsk	8 : 00	60	Irkutsk Time	
Asia/Kashgar	8 : 00		China Standard Time	
Asia/Kuala_Lumpur	8 : 00		Malaysia Time	
Asia/Kuching	8 : 00		Malaysia Time	
Asia/Macao	8 : 00		China Standard Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Asia/Macau	8 : 00		China Standard Time	
Asia/Makassar	8 : 00		Central Indonesia Time	
Asia/Manila	8 : 00		Philippines Time	
Asia/Shanghai	8 : 00		China Standard Time	
Asia/Singapore	8 : 00		Singapore Time	
Asia/Taipei	8 : 00		China Standard Time	
Asia/Ujung_Pandang	8 : 00		Central Indonesia Time	QP0800WITA
Asia/Ulaanbaatar	8 : 00		Ulaanbaatar Time	
Asia/Ulan_Bator	8 : 00		Ulaanbaatar Time	
Asia/Urumqi	8 : 00		China Standard Time	
Australia/Perth	8 : 00		Western Standard Time (Australia)	QP0800AWST
Australia/West	8 : 00		Western Standard Time (Australia)	
CTT	8 : 00		China Standard Time	QP0800BST
Etc/GMT-8	8 : 00		GMT+08:00	
Hongkong	8 : 00		Hong Kong Time	
PRC	8 : 00		China Standard Time	
Singapore	8 : 00		Singapore Time	
Asia/Choibalsan	9 : 00		Choibalsan Time	
Asia/Dili	9 : 00		East Timor Time	
Asia/Jayapura	9 : 00		East Indonesia Time	QP0900WIT
Asia/Pyongyang	9 : 00		Korea Standard Time	
Asia/Seoul	9 : 00		Korea Standard Time	QP0900KST
Asia/Tokyo	9 : 00		Japan Standard Time	QP0900UTCS
Asia/Yakutsk	9 : 00	60	Yakutsk Time	
Etc/GMT-9	9 : 00		GMT+09:00	
JST	9 : 00		Japan Standard Time	QP0900JST
Japan	9 : 00		Japan Standard Time	
Pacific/Palau	9 : 00		Palau Time	
ROK	9 : 00		Korea Standard Time	
ACT	9 : 30		Central Standard Time (Northern Territory)	
Australia/Adelaide	9 : 30	60	Central Standard Time (South Australia)	QP0930ACST
Australia/Broken_Hill	9 : 30	60	Central Standard Time (South Australia/New South Wales)	
Australia/Darwin	9 : 30		Central Standard Time (Northern Territory)	
Australia/North	9 : 30		Central Standard Time (Northern Territory)	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Australia/South	9 : 30	60	Central Standard Time (South Australia)	
Australia/Yancowinna	9 : 30	60	Central Standard Time (South Australia/New South Wales)	
AET	10 : 00	60	Eastern Standard Time (New South Wales)	QP1000AEST
Antarctica/DumontDUrville	10 : 00		Dumont-d'Urville Time	
Asia/Sakhalin	10 : 00	60	Sakhalin Time	
Asia/Vladivostok	10 : 00	60	Vladivostok Time	
Australia/ACT	10 : 00	60	Eastern Standard Time (New South Wales)	
Australia/Brisbane	10 : 00		Eastern Standard Time (Queensland)	
Australia/Canberra	10 : 00	60	Eastern Standard Time (New South Wales)	
Australia/Hobart	10 : 00	60	Eastern Standard Time (Tasmania)	
Australia/Lindeman	10 : 00		Eastern Standard Time (Queensland)	
Australia/Melbourne	10 : 00	60	Eastern Standard Time (Victoria)	
Australia/NSW	10 : 00	60	Eastern Standard Time (New South Wales)	
Australia/Queensland	10 : 00		Eastern Standard Time (Queensland)	
Australia/Sydney	10 : 00	60	Eastern Standard Time (New South Wales)	
Australia/Tasmania	10 : 00	60	Eastern Standard Time (Tasmania)	
Australia/Victoria	10 : 00	60	Eastern Standard Time (Victoria)	
Etc/GMT-10	10 : 00		GMT+10:00	
Pacific/Guam	10 : 00		Chamorro Standard Time	QP1000UTCS
Pacific/Port_Moresby	10 : 00		Papua New Guinea Time	
Pacific/Saipan	10 : 00		Chamorro Standard Time	
Pacific/Truk	10 : 00		Truk Time	
Pacific/Yap	10 : 00		Yap Time	
Australia/LHI	10 : 30	30	Load Howe Standard Time	
Australia/Lord_Howe	10 : 30	30	Load Howe Standard Time	
Asia/Magadan	11 : 00	60	Magadan Time	
Etc/GMT-11	11 : 00		GMT+11:00	
Pacific/Efate	11 : 00		Vanuatu Time	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
Pacific/Guadalcanal	11 : 00		Solomon Is. Time	QP1100UTCS
Pacific/Kosrae	11 : 00		Kosrae Time	
Pacific/Noumea	11 : 00		New Caledonia Time	
Pacific/Ponape	11 : 00		Ponape Time	
SST	11 : 00		Solomon Is. Time	
Pacific/Norfolk	11 : 30		Norfolk Time	
Antarctica/McMurdo	12 : 00	60	New Zealand Standard Time	
Antarctica/South_Pole	12 : 00	60	New Zealand Standard Time	
Asia/Anadyr	12 : 00	60	Anadyr Time	
Asia/Kamchatka	12 : 00	60	Petropavlovsk-Kamchatski Time	
Etc/GMT-12	12 : 00		GMT+12:00	
Kwajalein	12 : 00		Marshall Islands Time	
NST	12 : 00	60	New Zealand Standard Time	QP1200NZST
NZ	12 : 00	60	New Zealand Standard Time	
Pacific/Auckland	12 : 00	60	New Zealand Standard Time	
Pacific/Fiji	12 : 00		Fiji Time	QN1200UTCS, QP1200UTCS
Pacific/Funafuti	12 : 00		Tuvalu Time	
Pacific/Kwajalein	12 : 00		Marshall Islands Time	
Pacific/Majuro	12 : 00		Marshall Islands Time	
Pacific/Nauru	12 : 00		Nauru Time	
Pacific/Tarawa	12 : 00		Gilbert Is. Time	
Pacific/Wake	12 : 00		Wake Time	
Pacific/Wallis	12 : 00		Wallis & Futuna Time	
NZ-CHAT	12 : 45	60	Chatham Standard Time	
Pacific/Chatham	12 : 45	60	Chatham Standard Time	QP1245UTCS
Etc/GMT-13	13 : 00		GMT+13:00	
Pacific/Enderbury	13 : 00		Phoenix Is. Time	
Pacific/Tongatapu	13 : 00		Tonga Time	
Etc/GMT-14	14 : 00		GMT+14:00	
Pacific/Kiritimati	14 : 00		Line Is. Time	

Web module or application server stops processing requests

If an application server process spontaneously closes, or Web modules stop responding to new requests, it is important that you quickly determine why this stoppage is occurring. You can use some of the following techniques to determine whether the problem is a Web module problem or an application server environment problem.

If an application server process spontaneously closes, or Web modules running on the application server stop responding to new requests:

- Try to Isolate the problem by installing the Web modules on different servers, if possible.
- Check the product directory structure for a file with a name like javacore[number].txt. This file is a Java thread dump file that the JVM creates if an application server process spontaneously closes.
- Use the Tivoli performance viewer to determine if any of the application server resources, such as the Java heap, or database connections, have reached their maximum capacity. If there is a resource problem, review the application code for a possible cause:
  - If database connections are being assigned to a request but are not being released when the requests finish processing, ensure that the application code performs a **close()** on any opened **Connection** object within a **finally{}** block.
  - If there is a steady increase in servlet engine threads in use, review application **synchronized** code blocks for possible deadlock conditions.
  - If there is a steady increase in a JVM heap size, review application code for memory leak opportunities, such as static (class-level) collections, that can cause objects to never get garbage-collected.
- Enable verbose garbage collection on the application server to help you determine if you have a memory leak problems. This feature adds detailed statements about the amount of available and in-use memory to the JVM error log file.

  To enable up verbose garbage collection:
  1. In the administrative console, click **Servers > Server Types > Application servers >** *server_name*. . Then, under Server Infrastructure, click **Java and process management > Process definition > Java virtual machine**, and select **Verbose garbage collection**.
  2. Stop and restart the application server.
  3. Periodically, browse the log file for garbage collection statements. Look for statements beginning with ″allocation failure″. This string indicates that a need for memory allocation has triggered a JVM garbage collection, to release unused memory. Allocation failures are normal and do not necessarily indicate a problem. However, the statements that follow the allocation failure statement show how many bytes are needed and how many are allocated. If these bytes needed statements indicate that the JVM keeps allocating more memory for its own use, or that the JVM is unable to allocate as much memory as it needs, there might be a memory leak.

  You can also use the Tivoli performance viewer to detect memory leak problems.
- Determine if the application server is running out of memory. If you determine that the application server is running out of memory, one of the following situations might be occurring:
  - There is a memory leak in application code that you must address. To pinpoint the cause of a memory leak, enable the **RunHProf** property on the Java Virtual Machine page of the administrative console. *server_name* is the name of the problem application server. After you enable the **RunHProf** property, you must:
    - Set the **HProf Arguments** field to a value similar to depth=20,file=heapdmp.txt. This value shows exception stacks to a maximum of 20 levels, and saves the heapdump output to the *app_server_root*/bin/heapdmp.txt file.
    - Save the settings.
    - Stop and restart the application server.
    - If possible, reenact the scenario or access the resource that caused the application server's process to spontaneously close, or its Web modules to stop responding to new requests. Then stop the application server. If you cannot reenact the scenario or access the resource, wait until the problem reoccurs, and then stop the application server.
    - Examine the file into which the heap dump was saved. For example, examine the *app_server_root*/bin/heapdmp.txt file:

    - Search for the string, ″SITES BEGIN″. This finds the location of a list of Java objects in memory, which shows the amount of memory allocated to the objects.
    - The list of Java objects occurs each time there was a memory allocation in the JVM. There is a record of what type of object the memory instantiated and an identifier of a trace stack, listed elsewhere in the dump, that shows the Java method that made the allocation.
    - The list of Java object is in descending order by number of bytes allocated. Depending on the nature of the leak, the problem class should show up near the top of the list, but this is not always the case. Look throughout the list for large amounts of memory or frequent instances of the same class being instantiated. In the latter case, use the ID in the trace stack column to identify allocations occurring repeatedly in the same class and method.
    - Examine the source code indicated in the related trace stacks for the possibility of memory leaks.
  – The JVM is using the maximum heap size that it is allowed to use. In this situation, you should increase the maximum heap size setting for application server if you have enough storage available to do so.
  – The server runtime is experiencing a problem. If you determine that there is a problem with the server runtime, make sure that you have applied all of the service updates for the product. If, after you apply all of the service updates, the problem still exists, contact IBM Support.
- Browse the thread dump for clues:

  The JVM creates a thread dump whenever an application server process spontaneously closes. You can also force an application to create a thread dump. After a dump is created, you can check the dump for clues as to why new requests are not being processed.

  To force a thread dump:
  1. Using the wsadmin command prompt, get a handle to the problem application server:

     ```
     wsadmin>set jvm [$AdminControl completeObjectName type=JVM,process=server1,*]
     ```
  2. Generate the thread dump:

     ```
     wsadmin>$AdminControl invoke $jvm dumpThreads
     ```
  3. Look for an output file, in the installation root directory for the product, with a name like javacore.*date.time.id*.txt.

  After the application creates the dump, you can check for the following clues:
  – ″Error″ or ″exception information″ strings at the beginning of the file. These strings indicate the thread that caused the application server process to spontaneously close. These strings are not be present if you forced the dump.
  – Look at the snapshot of each thread in the process. The thread dump contains a snapshot of each thread in the process, starting in the section labeled ″Full thread dump.″
    - Look for threads with a description that contains ″state:R″. Such threads are active and running when the dump is forced, or the process exited.
    - Look for multiple threads in the same Java application code source location. Multiple threads from the same location might indicate a deadlock condition (multiple threads waiting on a monitor) or an infinite loop, and help identify the application code with the problem.

IBM Support has documents and tools that can save you time gathering information needed to resolve problems as described in Troubleshooting help from IBM. Before opening a problem report, see the Support page:

- http://www.ibm.com/software/webservers/appserv/was/support/

Creating generic servers

A generic server is a server that is managed in the WebSphere Application Server administrative domain even though the server is not a server that is supplied by WebSphere Application Server. The WebSphere Application Server generic servers function enables you to define a generic server as an application server instance within the WebSphere Application Server administration, and associate it with a non-WebSphere WebSphere Application server or process.

About this task

There are two basic types of generic application servers:
- Non-Java applications or processes.
- Java applications or processes

Therefore, a generic server can be any server or process that is necessary to support the Application Server environment, including:
- A Java server
- A C or C++ server or process
- A CORBA server
- A Remote Method Invocation (RMI) server

You can use the wsadmin tool or the administrative console to create a generic server.

- **Create a non-Java application as a generic server.** The following steps describe how to use the administrative console to create a non-Java application as a generic application server.
  1. Select **Servers** > **Generic servers**
  2. Click **New**.
  3. Type in a name for the generic server.

     The name must be unique within the node. It is recommended that you use a naming scheme that makes it easy to distinguish your generic application servers from regular WebSphere Application Server servers.
  4. Click **Next**
  5. Click **Finish**. The generic server now appears as an option on the **Generic servers** page in the administrative console.
  6. On the **Generic servers** page, click on the name of the generic server.
  7. Under Additional Properties, click **Process Definition**.
  8. In the Executable name field, enter the name of the non-java process that is launched when you start this generic server.

     For example, if you are using a perl script as a generic server, enter the path to the perl.exe module in the Executable name field.

     If you have additional arguments, such as the name of the perl script and its parameters, enter them in the Executable arguments field. Multiple arguments must be separated by carriage returns. Use the Enter key on your keyboard to create these carriage returns in the Executable arguments field. The following example illustrates how a perl script application that requires two arguments should appear in this field:

     ```
     perl_application.pl
     arg1
     arg2
     ```

     Note: The Executable target type and Executable target properties are not used for non-Java applications. Executable target type and Executable target properties are only used for Java applications.
  9. Click **OK**.
- **Create a Java application as a generic server:** The following steps describe how to use the administrative console to create a Java application as a generic application server.
  1. Select **Servers > Server Types > Generic servers**
  2. Click **New**.
  3. Type in a name for the generic server.

     The name must be unique within the node. It is highly recommended that you use a naming scheme that makes it easy to distinguish your generic application servers from regular WebSphere Application Server servers.

4. Click **Next**
5. Click **Finish**. The generic server now appears as an option on the **Application servers** page in the administrative console.
6. Click **Finish**. The generic server now appears as an option on the **Generic servers** page in the administrative console.
7. On the Generic servers page, click on the name of the generic server.
8. Under Additional Properties, click **Process definition**.
9. In the Executable name field under General Properties, enter the path for the WebSphere Application Server default JVM, ${JAVA_HOME}/bin/java, which is used to run the Java application when you start this generic server.
10. In the Executable target type field under General Properties, select whether a Java class name, **JAVA_CLASS**, or the name of an executable JAR file, **EXECUTABLE_JAR**, is used as the executable target of this Java process. The default value for the product is **JAVA_CLASS**.
11. In the Executable target field under General Properties, enter the name of the executable target. Depending on the executable target type, this is either a Java class containing a main() method, or the name of an executable JAR file.) The default value for WebSphere Application Server is com.ibm.ws.runtime.WsServer.
12. Click **OK**.

Note: If the generic server is to run on an application server other than a WebSphere Application Server server, leave the Executable name field set to the default value and specify the Java class containing the main function for your application serve in the Executable target field.

What to do next

After you define a generic server, use the Application Server administrative console to start, stop, and monitor the associated non-WebSphere Application Server server or process when stopping or starting the applications that rely on them.

Note: You can use either the **Terminate** or **Stop** buttons in the administrative console to stop any application server, including a generic application server.

Starting and terminating generic application servers

This topic describes how to start and terminate generic servers.

About this task

If you create a generic server on a base WebSphere Application Server, you cannot use the base Application Server administrative console to start or terminate this server. You must use the wsadmin tool to manage this server.

If you create a generic server in a Network Deployment environment, you can use the administrative console to start and terminate this server.

1. Start a generic application server.

   There are two ways to start a generic server in a Network Deployment environment. You can use the managed bean (MBean) NodeAgent launchProcess operation of the wsadmin tool, or you can use the administrative console. To use the administrative console:

   a. In the administrative console, click **Servers > Server Types > Generic servers**.
   b. Select the name of the generic server you want to start, and then click **Start**.
   a. View the **Status** value and any messages or logs to see whether the generic server starts.
2. Terminate generic servers.

There are two ways to terminate a generic server in a Network Deployment environment. You can use the MBean terminate launchProcess operation of the wsadmin tool, or you can use the administrative console. To use the administrative console:

a. In the administrative console, click **Servers > Server Types > Generic servers**.

b. Select the check box beside the name of the generic server, and then click **Terminate** or **Stop**.

c. View the **Status** value and any messages or logs to see whether the generic server terminates.

Generic server settings

Use this page to view or change the settings of a generic server.

A generic server is a server that is managed in the product administrative domain, although it is not a server that is provided with the product. The generic server can be any server or process that is necessary to support the Application Server environment, including a Java server, a C or C++ server or process, or a Remote Method Invocation (RMI) server.

To view this administrative console page, click **Servers > Server Types > Generic servers >** *server_name*.

On the **Configuration** tab, you can edit fields. On the **Runtime** tab, you can look at read-only information. The **Runtime** tab is available only when the server is running.

Name

Specifies a logical name for the generic server.

Generic server names must be unique within a node. For multiple nodes within a cluster, you can have different generic servers with the same server name as long as the server and node pair are unique. For example, a server named server1 in a node named node1 in the same cluster with a server named server1 in a node named node2 is allowed. Configuring two servers named server1 in the same node is not allowed. The product uses the server name for administrative actions, such as referencing the server in scripting.

It is highly recommended that you use a naming scheme that makes it easy to distinguish your generic application servers from regular product application servers. This will enable you to quickly determine whether to use the Terminate or Stop button in the administrative console to stop a specific application server.

You must use the Terminate button to stop a generic application server.

Data type String

Default

Configuring transport chains

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.

Before you begin

Ensure that a port is available for the new transport chain. If you need to set up a shared port, you must:

- Use wsadmin commands to create your transport chain.
- Make sure that all channels sharing that port have the same discrimination weight assigned to them.

About this task

You need to configure transport chains to provide networking services to such functions as the service integration bus component of IBM service integration technologies, the caching proxy, and the high availability manager core group bridge service.

You can use either the administrative console or wsadmin commands to create a transport chain. If you use the administrative console, complete the following steps:

1. In the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name* or **Servers > Server Types > WebSphere proxy servers >** *server_name*, and then select one of the following options, depending on the type of chain you are creating:

   For application servers, in the Container settings section select one of the following options:

   - Click **SIP Container Settings > SIP container transport chains**.
   - Click **Web container settings > Web container transport chains**.
   - In the Server messaging section, click either **Messaging engine inbound transports** or **WebSphere MQ link inbound transports**.

   For proxy servers, under HTTP proxy server settings, click **Proxy server transports** and select either **HTTPS_PROXY_CHAIN** or **HTTP_PROXY_CHAIN**. Then click **HTTP proxy inbound channel**.
2. Click **New**.

   The Create New Transport Chain wizard initializes. During the transport chain creation process, you are asked to:

   - Specify a name for the new chain.
   - Select a transport chain template
   - Select a port, if one is available to which the new transport chain is bound. If a port is not available or you want to define a new port, specify a port name, the host name or IP address for that port, and a valid port number.

     Note: If you are configuring a chain that contains a TCP channel, the wizard displays a list of configured TCP channels and a list of the ports that the listed TCP channels are not using. You must select one of the ports that none of the other TCP channels are using.

     Similarly, if you are configuring a transport chain that contains a UDP channel, the wizard displays a list of already configured UDP channels and a list of the ports that these UDP channels are not using. You must select one of the ports that none of the other UDP channels are using.

   When you click **Finish**, the new transport chain is added to the list of defined transport chains on the **Transport chain** panel.
3. Click the name of a transport chain to view the configuration settings that are in effect for the transport channels contained in that chain.

   To change any of these settings, complete the following actions:

   a. Click the name of the channel whose settings you need to change.
   b. Change the configuration settings.

      Some of the settings, such as the port number, are determined by what is specified for the transport chain when it is created and cannot be changed.
   c. Click on **Custom properties** to set any custom properties that are defined for your system.
4. When you your configuration changes, click **OK**.
5. Stop the application server and start it again.

   You must stop the application server and start it again before your changes take effect.

What to do next

Update any routines you have that issue a call to start transports during server startup. When a routine issues a call to start transports during server startup, the product converts the call to a transport channel call.

Transport chains

Transport chains represent a network protocol stack that is used for I/O operations within an application server environment.

Transport chains are part of the channel framework function that provides a common networking service for all components, including the service integration bus component of IBM service integration technologies, WebSphere Secure Caching Proxy, and the high availability manager core group bridge service.

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, DCS. 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.

Note: If you have a routine that issues a call to start transports during server startup, unless you have a mixed-node environment and that server is running in a Version 5.1 node, the product converts the call to a transport chain call.

The transport chain configuration settings determine which I/O protocols are supported for that chain. Following are some of the more common types of channels. Custom channels that support requirements unique to a particular customer or environment can also be added to a transport chain.

DCS channel
Used in a Network Deployment environment by the core group bridge service, the data replication service (DRS), and the high availability manager to transfer data, objects, or events among application servers.

HTTP inbound channel
Used to enable communication with remote servers. It implements the HTTP 1.0 and 1.1 standards and is used by other channels, such as the Web container channel, to serve HTTP requests and to send HTTP specific information to servlets expecting this type of information.

HTTP inbound channels are used instead of HTTP transports to establish the request queue between a Web server plug-in, and a Web container in which the Web modules of an application reside.

HTTP proxy inbound channel
Used to handle HTTP requests between a proxy server and application server nodes.

HTTP Tunnel channel
Used to provide client applications with persistent HTTP connections to remote hosts that are either blocked by firewalls or require an HTTP proxy server, including authentication, or both. An HTTP Tunnel channel enables the exchange of application data in the body of an HTTP request or response that is sent to or received from a remote server. An HTTP Tunnel channel also enables client-side applications to poll the remote host and to use HTTP requests to either send data from the client or to receive data from an application server. In either case, neither the client nor the application server is aware that HTTP is being used to exchange the data.

JFAP channel
Used by the Java Message Service (JMS) server to create connections to JMS resources on a service integration bus.

MQ channel
Used in combination with other channels, such as a TCP channel, within the confines of WebSphere MQ support to facilitate communications between a WebSphere System Integration Bus and a WebSphere MQ client or queue manager.

SIP channel
Used to create a bridge in the transport chain between a session initiation protocol (SIP) inbound channel, and a servlet and JavaServer Page engine.

SIP container inbound channel
Used to handle communication between the SIP inbound channel and the SIP servlet container.

SIP inbound channel
Used to handle inbound SIP requests from a remote client.

SSL channel
Used to associate an Secure Sockets Layer (SSL) configuration repertoire with the transport chain. This channel is only available when SSL support is enabled for the transport chain. An SSL configuration repertoire is defined in the administrative console, under security, on the **SSL configuration repertoires > SSL configuration repertoires** page.

TCP channel
Used to provide client applications with persistent connections within a Local Area Network (LAN) when a node uses transmission control protocol (TCP) to retrieve information from a network.

UDP channel
Used to provide client applications with persistent connections within a Local Area Network (LAN) when a node uses user datagram protocol (UDP) to retrieve information from a network.

Web container channel
Used to create a bridge in the transport chain between an HTTP inbound channel and a servlet and JavaServer Page (JSP) engine.

HTTP transport collection

Use this page to view or manage HTTP transports. Transports provide request queues between Web server plug-ins and Web containers in which the Web modules of applications reside. When you request an application in a Web browser, the request is passed to the Web server, then along the transport to the Web container.

Note: You can use HTTP transports only on a Version 5.1 node in a mixed cell environment. You must use HTTP transport channels instead of HTTP transports to handle your HTTP requests on all of your other nodes.

The use of IPv6 (Internet Protocol Version 6) and WS-AT (Web Services Atomic Transactions) are not supported on HTTP transports; they are only supported on HTTP transport channel chains.

To view the HTTP Transport administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Web container settings > Web container > HTTP transports**.

Host

Specifies the host IP address to bind for transport. If the application server is on a local machine, the host name might be `localhost`.

Port

Specifies the port to bind for transport. The port number can be any port that currently is not in use on the system. The port number must be unique for each application server instance on a given machine.

For i5/OS and distributed operating systems, there is no limit to the number of HTTP ports that are allowed per process.

SSL Enabled
Specifies whether to protect transport connections with Secure Sockets Layer (SSL). The default is not to use SSL.

HTTP transport settings
Use this page to view and configure an HTTP transport. The name of the page might be that of an SSL setting such as DefaultSSLSettings. This page is not available if you do not have an HTTP transport defined for your system.

Note: You can use HTTP transports only on a Version 5.1 node in a mixed cell environment. You must use HTTP transport channels instead of HTTP transports to handle your HTTP requests on all of your other nodes.

The use of IPv6 (Internet Protocol Version 6) and WS-AT (Web Services Atomic Transactions) are not supported on HTTP transports; they are only supported on HTTP transport channel chains.

If you have HTTP transports defined for your system, in the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name*, and then in the Container Settings section, click **Web container > HTTP transports >** *host_name* to view or change the settings for your HTTP transport.

Host
Specifies the host IP address to bind for transport.

If the application server is on a local machine, the host name might be `localhost`.

Data type String

Port
Specifies the port to bind for transport. Specify a port number between 1 and 65535. The port number must be unique for each application server on a given machine.

Data type Integer
Range 1 to 65535

SSL Enabled
Specifies whether to protect transport connections with Secure Sockets Layer (SSL). The default is not to use SSL.

Data type Boolean
Default false

SSL
Specifies the Secure Sockets Layer (SSL) settings type for SSL connections. The options include one or more SSL settings that are defined in the Security Center; for example, DefaultSSLSettings, ORBSSLSettings, or LDAPSSLSettings.

Data type String
Default An SSL setting defined in the Security Center

HTTP transport custom properties

You can use the administrative console to set custom properties for an HTTP transport. The HTTP transport custom properties administrative console page only appears if you have an HTTP transport defined for your system.

Note: You can use HTTP transports only on a V5.1 node in a mixed cell environment. You must use HTTP transport channels instead of HTTP transports to handle your HTTP requests on all of your other nodes. The topic *HTTP Tunnel transport channel custom property* describes the custom properties that you can specify for an HTTP transport channel.

The use of IPv6 (Internet Protocol Version 6) and WS-AT (Web Services Atomic Transactions) are not supported on HTTP transports; they are only supported on HTTP transport channel chains.

If you are using HTTP transports, you can set the following custom properties on either the Web container or HTTP transport custom properties page in the administrative console. When set on the Web container custom properties page, all transports inherit the properties. Setting the same properties on a transport overrides like settings defined for a Web container.

To specify custom properties for a specific transport on the HTTP transport:

1. In the administrative console click **Servers > Server Types > WebSphere application servers >** *server_name*.
2. Then in the Container Settings section, click **Web container > Web container settings > HTTP transport**.
3. Select a host.
4. In the Additional Properties section, select **Custom Properties**.
5. On the custom properties page, click **New**.
6. On the settings page, enter the property you want to configure in the **Name** field and the value you want to set it to in the **Value** field.
7. Click **Apply** or **OK**.
8. Click **Save** on the console task bar to save your configuration changes.
9. Restart the server.

Following is a list of custom properties provided with the product. These properties are not shown on the settings page for an HTTP transport.

ConnectionIOTimeOut:

Use the ConnectionIOTimeOut property to specify how long the J2EE server waits for an I/O operation to complete. Set this variable for each of the HTTP transport definitions on the server. You will need to set this variable for both SSL transport and non-SSL transport. Specifying a value of zero disables the time out function.

Data type	Integer
Default	5 seconds for the i5/OS and distributed platforms

ConnectionKeepAliveTimeout:

Use the ConnectionKeepAliveTimeout property to specify the maximum number of seconds to wait for the next request on a keep alive connection.

Data type	Integer
Default	5 seconds for the i5/OS and distributed platforms

MaxConnectBacklog: This property is only valid for i5/OS and distributed platforms. Use the MaxConnectBacklog property to specify the maximum number of outstanding connect requests that the operating system will buffer while it waits for the application server to accept the connections. If a client attempts to connect when this operating system buffer is full, the connect request will be rejected.

Keep in mind that a single client browser might need to open multiple concurrent connections (perhaps 4 or 5); however, also keep in mind that increasing this value consumes more kernel resources. The value of this property is specific to each transport.

Data type	Integer
Default	511

MaxKeepAliveConnections: This property is only valid for i5/OS and distributed platforms. It is ignored on the z/OS platform because asynchronous I/O sockets are used to maintain connections in that environment. Use the MaxKeepAliveConnections property to specify the maximum number of concurrent keep alive (persistent) connections across all HTTP transports. To make a particular transport close connections after a request, you can set MaxKeepAliveConnections to `0` (zero), or you can set KeepAliveEnabled to `false` on that transport.

The Web server plug-in keeps connections open to the application server as long as it can. However, if the value of this property is too small, performance is negatively impacted because the plug-in has to open a new connection for each request instead of sending multiple requests through one connection. The application server might not accept a new connection under a heavy load if there are too many sockets in TIME_WAIT state. If all client requests are going through the Web server plug-in and there are many TIME_WAIT state sockets for port 9080, the application server is closing connections prematurely, which decreases performance. The application server closes the connection from the plug-in, or from any client, for any of the following reasons:

- The client request was an HTTP 1.0 request when the Web server plug-in always sends HTTP 1.1 requests.
- The maximum number of concurrent keep-alives was reached. A keep-alive must be obtained only once for the life of a connection, that is, after the first request is completed, but before the second request can be read.
- The maximum number of requests for a connection was reached, preventing denial of service attacks in which a client tries to hold on to a keep-alive connection forever.
- A time out occurred while waiting to read the next request or to read the remainder of the current request.

Data type	Integer
Default	90% of the maximum number of threads in the Web container thread pool. This prevents all of the threads from being held by keep alive connections so that there are threads available to handle new incoming connect requests.

MaxKeepAliveRequests:

Use the MaxKeepAliveRequests property to specify the maximum number of requests which can be processed on a single keep alive connection. This parameter can help prevent denial of service attacks when a client tries to hold on to a keep-alive connection. The Web server plug-in keeps connections open to the application server as long as it can, providing optimum performance.

On the i5/OS and distributed platforms, when this property is set to 0 (zero), the connection stays open as long as the application server is running.

Data type	Integer

Default	100 requests for the i5/OS and distributed platforms

KeepAliveEnabled: This property is only valid for i5/OS and distributed platforms. Use the KeepAliveEnabled property to specify whether or not to keep connections alive

Data type	String
Value	true or false
Default	true

RemoveServerHeader: Use this property to specify whether an existing server header is removed before a response message is sent. If this property is set to `true`, the value specified for the ServerHeaderValue property is ignored.

Data type	String
Value	true or false
Default	false

ResponseBufferSize:

This property is used to specify, in bytes, the default size of the initial buffer allocation for the response buffer. When the buffer fills up, a flush for this buffer space will automatically occur. If a value is not specified for this property, the default response buffer size of 32K bytes is used.

The setBufferSize() API method can be used to override the value specified for this custom property at the individual servlet level.

Data type	Integer
Default	32000 bytes

ServerHeaderValue: Use this property to specify a server header this is added to outgoing response messages if server header is not already provided. This property is ignored if the RemoveServerHeader property is set to true.

Data type	string
Default	WebSphere Application Server/*x.x* *x.x* is the version of WebSphere Application Server that you are using.

SoLingerValue: Use this property to specify, in seconds, the amount, that the socket close operation waits for data contained in the TCP/IP send buffer to be sent. This property is ignored if the UseSoLinger property is set to `false`.

Data type	Integer
Default	20 seconds

TcpNoDelay: Use this property to set the socket TCP_NODELAY option which enables and disables the use of the TCP Nagle algorithm for connections received on this transport. When this property is set to `true`, use of the Nagle algorithm is disabled.

Data type	String
Value	true or false
Default	true

Trusted: Use the Trusted property to indicate that the application server can use the private headers that the Web server plug-in adds to requests.

Data type	String
Value	true or false
Default	false

Note: This property must be set to false for Secure Sockets Layer (SSL) client certificate authentication to work.

UseSoLinger: Use this property to set the socket SO_LINGER option. This property configures whether the socket close operation waits until all of the data contained in the TCP/IP send buffer is sent before closing a connection. If this property is set to `true`, and the time expires before the all of the content of the send buffer sent, any data remaining in the send buffer is lost.

The SoLingerValue property is ignored if this property is set to `false`.

Data type	String
Value	true or false
Default	true

Transport chains collection

Use this page to view or manage transport chains. Transport chains enable communication through transport channels, or protocol stacks, which are usually socket based.

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.

The **Transport chains** page lists the transport chains defined for the selected application server. Transport chains represent network protocol stacks operating within this application server.

To view this administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Ports**. Click on **View associated transports** for the port whose transport chains you want to view.

Name

Specifies a unique identifier for the transport chain. The name must consist of alphanumeric or national language characters and can start with a number. The name must be unique within the product configuration. Click on the name of a transport chain to change its configuration settings.

Enabled

When set to true, indicates that the transport chain is activated at application server startup.

Host

Specifies the host IP address to bind for the transport chain. If the application server is on a local machine, the host name might be localhost.

Port

Specifies the port to bind for the transport chain. The port number can be any port that currently is not in use on the system, might be localhost or the wildcard character * (an asterisk). The port number must be unique for each application server instance on a given machine

SSL Enabled
When enabled, users are notified that there is a channel that enables Secure Sockets Layer (SSL) in the listed transport chain. When SSL is enabled, all traffic going through this transport is encrypted and digitally secured.

Transport chain settings
Use this page to view a list of the types of transport channels configured for the selected transport chain. 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, HTTP, or DCS.

To view this administrative console page, click **Servers > Server Types** , and then click either **WebSphere application servers** or **WebSphere proxy servers**. Click a server name, and then click **Ports > View associated transports** for the port whose transport chains you want view, and then click the name of a specific chain.

Name
Specifies the name of the selected transport chain.

You can edit this field to rename this transport chain. However, remember that the name must be unique within the product configuration.

Enabled
When checked, this transport chain is activated at application server or proxy server startup.

Transport channels
Lists the transport channels configured for this transport chain and their configuration settings. Click the name of a transport channel to view the configuration settings for that channel.

HTTP tunnel transport channel settings
Use this page to view and configure an HTTP tunnel transport channels. Inbound connections sent through this channel are tunneled over HTTP, allowing intermediates to view this data as the body of an HTTP message instead of in its natural format. This type of channel is often used to circumvent firewalls with protocol restrictions.

To view this administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Ports**. Click on **View associated transports** for the port associated with the HTTP Tunnel transport channel whose settings you want to look at.

Transport channel name
Specifies the name of the HTTP tunnel transport channel.

The name field cannot contain the following characters: **# \ / , : ; " * ? < > | = + & %** '

This name must be unique across all channels within the product environment. For example, an HTTP tunnel transport channel and a TCP transport channel cannot have the same name if they reside within the same system.

Data type string

Discrimination weight
Specifies the priority this channel has in relation to the other channels in this transport chain. This property is only used when port sharing is enabled and the transport chain includes multiple channels to which it might forward data. The channel in the chain with the lowest discrimination weight is the first one given the opportunity to look at incoming data and determine whether or not it owns that data.

Data type Positive integer
Default 0

HTTP transport channel settings

Use this page to view and configure an HTTP transport channel. This type of transport channel handles HTTP requests from a remote client.

An HTTP transport channel parses HTTP requests and then finds an appropriate application channel to handle the request and send a response.

To view this administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Ports**. Locate the port for the HTTP channel whose settings you want to view or configure, and click **View associated transports**. Click the name of the transport chain that includes this HTTP transport, and then click the name of the HTTP transport channel.

Transport channel name

Specifies the name of the HTTP transport channel.

The name field cannot contain any of the following characters: **# \ / , : ; " * ? < > | = + & % '**

This name must be unique across all channels in your system. For example, an HTTP transport channel and a TCP transport channel cannot have the same name if they reside within the same system.

Data type String

Discrimination weight

Specifies the priority this channel has in relation to the other channels in this transport chain. This property is only used when port sharing is enabled, and the transport chain includes multiple channels to which it might forward data. The channel in the chain that has the lowest discrimination weight is the first channel that looks at incoming data and determines whether it owns that data.

Data type Positive integer
Default 0

Read timeout

Specifies the amount of time, in seconds, that the HTTP transport channel waits for a read request to complete on a socket after the first read occurs. The read being waited for could be part of the body of the read request, such as a POST, or part of the headers, if all of the headers are not read as part of the first read that occurs on the socket for this request.

Note: The value specified for this property, in conjunction with the value specified for the Write timeout property, provides the timeout functionality that the ConnectionIOTimeout custom property provided in previous releases.

Data type Integer
Default 60 seconds

Write timeout

Specifies the amount of time, in seconds, that the HTTP transport channel waits on a socket for each portion of the response data to be transmitted. This timeout typically only occurs in situations where the writes are lagging behind new requests. This situation can occur when a client has a low data rate or the network interface card (NIC) for the server is saturated with I/O.

Note: The value specified for this property, in conjunction with the value specified for the Read timeout property, provides the timeout functionality that the ConnectionIOTimeout custom property provided in previous releases.

If some of your clients require more than 300 seconds to receive data being written to them, change the value specified for the Write timeout parameter. Some clients are slow and require more than 300 seconds to receive data that is sent to them. To ensure they are able to obtain all of their data, change the value specified for this parameter to a length of time in seconds that is sufficient for all of the data to be received. Make sure that if you change the value of this setting, that the new value still protects the server from malicious clients.

Data type	Integer
Default	300 seconds

Persistent timeout

Specifies the amount of time, in seconds, that the HTTP transport channel allows a socket to remain idle between requests.

Note: The value specified for this property provides the timeout functionality that the ConnectionKeepAliveTimeout custom property provided in previous releases.

Data type	Integer
Default	30 seconds

Use persistent (keep-alive) connections

When selected, specifies that the HTTP transport channel connections are left open between requests. Leaving the connections open can save setup and tear down costs of sockets if your workload has clients that send multiple requests.

If your clients only send single requests over substantially long periods of time, it is probably better to disable this option and close the connections right away rather than to have the HTTP transport channel setup the timeouts to close the connection at some later time.

The default value is true, which is typically the optimal setting.

Note: If a value other than 0 is specified for the maximum persistent requests property, the Use persistent (keep-alive) connections property setting is ignored.

Unlimited persistent requests per connection

When selected, specifies that the number of persistent requests per connection is not limited.

Maximum persistent requests per connection

When selected, specifies that the number of persistent requests per connection is limited to the number specified for the Maximum number of persistent requests property. This property setting is ignored if the Use persistent (keep-alive) connections property is not enabled.

Change the value specified for the Maximum persistent requests parameter to increase the number of requests that can flow over a connection before it is closed. When the Use persistent connections option is enabled, the Maximum persistent requests parameter controls the number of requests that can flow over a connection before it is closed. The default value is 100. This value should be set to a value such that most, if not all, clients always have an open connection when they make multiple requests during the same session. A proper setting for this parameter helps to eliminate unnecessary setting up and tearing down of sockets.

For test scenarios in which the client will never close a socket or where sockets are always proxy or Web servers in front of your application server, a value of -1 disables the processing, which limits the number of requests over a single connection. The persistent timeout still shuts down some idle sockets and protect your server from running out of open sockets.

Maximum persistent requests per connection

Specifies the maximum number of persistent requests that are allowed on a single HTTP connection. You can add a value to this field only if the **Maximum persistent requests per connection** property is selected.

When the Use persistent connections option is enabled, the Maximum persistent requests parameter controls the number of requests that can flow over a connection before it is closed. The default value is 100. This value should be set to a value such that most, if not all, clients always have an open connection when they make multiple requests during the same session. A proper setting for this parameter helps to eliminate unnecessary setting up and tearing down of sockets.

For test scenarios in which the client will never close a socket or where sockets are always proxy or Web servers in front of your application server, a value of -1 will disable the processing which limits the number of requests over a single connection. The persistent timeout will still shutdown some idle sockets and protect your server from running out of open sockets.

If a value of `0` or `1` is specified, only one request is allowed per connection.

Data type	Integer
Default	100

Maximum header field size

Specifies, in bytes, the maximum size for a header that can be included on an HTTP request.

Setting this property to a realistic size for your applications helps you to prevent denial of service (DoS) attacks that use large headers within an HTTP request as an attempt to make a system resource, such as the applications that handle HTTP requests, essentially unavailable to intended users.

The default for this property is `32768` bytes.

Maximum headers

Specifies the maximum number of headers that can be included in a single HTTP request.

Setting this property to a realistic number for your applications helps you to prevent denial of service (DoS) attacks that use a large number of headers within an HTTP request as an attempt to make a system resource, such as the applications that process HTTP requests, essentially unavailable to their intended users.

The default for this property is `50`.

Limit request body buffer size

When selected, specifies that size of the body of an HTTP request is limited.

This property can be used to prevent denial of service attacks that use large HTTP requests as an attempt to make a system resource, such as the applications that process HTTP requests, essentially unavailable to their intended users.

Maximum request body buffer size

Specifies, in bytes, the maximum size limit for the body of an HTTP request. If this size is exceeded, the request is not processed.

A value can be added to this field only if the **Limit request body buffer size** property is selected.

Logging

You can use the settings in this section to configure and enable National Center for Supercomputing Applications (NCSA) access logging, or HTTP error logging. If you are running the product on z/OS, you can also use this section to configure and enable Fast Response Cache Accelerator (FRCA) logging. Enabling any of these logging services slows server performance.

If you want any of the enabled logging services to start when the server starts, click **Servers > Server Types > WebSphere application servers >** *server_name*. Then in the Troubleshooting section, click **HTTP error, NCSA access and FRCA logging**, and select **Enable logging service at server start-up**. When this option is selected, any HTTP error, NCSA or FRCA logging service that is enabled automatically starts when the server starts.

NCSA access logging

By default, the **Use global logging service** option is selected for NCSA access logging. This setting means that the NCSA access logging settings default to the settings specified for NCSA access logging on the **HTTP error, NCSA access and FRCA logging** page in the administrative console. If you want to change these settings for this specific HTTP transport channel, expand the **NCSA Access logging** section, and select the **Use chain-specific logging** option.

After you select the **Use chain-specific logging** option, you can make the following configuration changes:

- Explicitly enable or disable NCSA access logging.
- Specify an access log file path that is different from the default path.
- Specify a maximum size for the access log file that is different from the default maximum size.
- Explicitly select the format of the NCSA access log file.

Enable access logging

When selected, a record of inbound client requests that the HTTP transport channel handles is kept in the NCSA access log file.

Access log file path

Specifies the directory path and name of the NCSA access log file. Standard variable substitutions, such as $(SERVER_LOG_ROOT), can be used when specifying the directory path.

Access log maximum size

Specifies the maximum size, in megabytes, of the NCSA access log file. When this size is reached, the *logfile_name* archive log file is created. However, every time that the original log file overflows this archive file, the file is overwritten with the most current version of the original log file.

Maximum number of historical files

Specifies the maximum number of historical versions of the NCSA access log file that are kept for future reference.

NCSA access log format

Specifies in which format the client access information appears in the NCSA log file. If Common is selected, the log entries contain the requested resource and a few other pieces of information, but does not contain referral, user agent, and cookie information. If Combined is selected, referral, user agent, and cookie information is included.

Error logging

By default, the **Use global logging service** option is selected for Error logging. This setting means that the Error logging settings default to the settings that are specified for Error logging on the **HTTP error,**

NCSA access and FRCA logging page in the administrative console. If you want to change these settings for this specific HTTP transport channel, expand the **Error logging** section, and select the **Use chain-specific logging** option.

After you select the **Use chain-specific logging** option, you can make the following configuration changes:

- Explicitly enable or disable HTTP Error logging.
- Specify the access log file path. This path can be different from the default path.
- Specify a maximum size for the error log file. This value can be larger or smaller than the default maximum size.
- Specify the type of error messages that you want included in the HTTP error log file.

Enable error logging
When selected, HTTP errors that occur while the HTTP channel processes client requests are recorded in the HTTP error log file.

Error log file path
Indicates the directory path and the name of the HTTP error log file. Standard variable substitutions, such as $(SERVER_LOG_ROOT), can be used when specifying the directory path.

Error log maximum size
Indicates the maximum size, in megabytes, of the HTTP error log file. When this size is reached, the *logfile_name* archive log file is created. However, every time that the original log file overflows this archive file, this file is overwritten with the most current version of the original log file.

Maximum number of historical files
Specifies the maximum number of historical versions of the HTTP error log file that are kept for future reference.

Error log level
Specifies the type of error messages that are included in the HTTP error log file.

You can select:

Critical
Only critical failures that stop the Application Server from functioning properly are logged.

Error The errors that occur in response to clients are logged. These errors require Application Server administrator intervention if they result from server configuration settings.

Warning
Information on general errors, such as socket exceptions that occur while handling client requests, are logged. These errors do not typically require Application Server administrator intervention.

Information
The status of the various tasks that are performed while handling client requests is logged.

Debug
More verbose task status information is logged. This level of logging is not intended to replace RAS logging for debugging problems, but does provide a steady status report on the progress of individual client requests. If this level of logging is selected, you must specify a large enough log file size in the **Error log maximum size** field to contain all of the information that is logged.

TCP transport channel settings
Use this page to view and configure a TCP transport channels. This type of transport channel handles inbound TCP/IP requests from a remote client.

To view this administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Ports**. Click on **View associated transports** for the port associated with the TCP transport channel whose settings you want to view.

Transport channel name

Specifies the name of the TCP transport channel.

The name field cannot contain the following characters: **# \ / , : ; " * ? < > | = + & %** '

This name must be unique across all channels in a WebSphere Application Server environment. For example, an HTTP proxy inbound channel and a TCP transport channel cannot have the same name if they reside within the same system.

Data type string

Port

Specifies the TCP/IP port this transport channel uses to establish connections between a client and an application server. The TCP transport channel binds to the hostnames and ports listed for the Port property. You can specify the wildcard * (an asterisk), for the hostname if you want this channel to listen to all hosts that are available on this system. However, before specifying the wildcard value, make sure this TCP transport channel does not have to bind to a specific hostname.

Data type string

Thread pool

This field only applies for i5/OS and distributed platforms. Select from the drop-down list of available thread pools the thread pool you want the TCP transport channel to use when dispatching work.

Maximum open connections

Specifies the maximum number of connections that are available for a server to use.

Leave the Maximum open connections property set to the default value 20000, which is the maximum number of connections allowed. The transport channel service by default manages high client connection counts and requires no tuning.

Default 20,000

Inactivity timeout

Specifies the amount of time, in seconds, that the TCP transport channel waits for a read or write request to complete on a socket.

If client connections are being closed without data being written back to the client, change the value specified for the Inactivity timeout parameter. This parameter controls the maximum number of connections available for a server's use. Upon receiving a new connection, the TCP transport channel waits for enough data to arrive to dispatch the connection to the protocol specific channels above the TCP transport channel. If not enough data is received during the time period specified for the Inactivity timeout parameter, the TCP transport channel closes the connection.

The default value for this parameter is 60 seconds, which is adequate for most applications. You should increase the value specified for this parameter if your workload involves a lot of connections and all of these connections can not be serviced in 60 seconds.

Note: The value specified for this property might be overridden by the wait times established for channels above this channel. For example, the wait time established for an HTTP transport channel overrides the value specified for this property for every operation except the initial read on a new socket.

Data type	Integer
Default	60 seconds

Address exclude list

Lists the IP addresses that are not allowed to make inbound connections.

Use a comma to separate the IPv4 or IPv6 or both addresses to which you want to deny access on inbound TCP connection requests.

All four numeric values in an IPv4 address must be represented by a number or the wildcard character * (an asterisk).

Following are examples of valid IPv4 addresses that can be included in an Address exclude list:

```
*.1.255.0
254.*.*.9
1.*.*.*
```

All eight numeric values of an IPv6 address must be represented by a number or the wildcard character * (an asterisk). No shortened version of the IPv6 address should be used. Even though a shortened version is processed with no error given, it does not function correctly in this list. Each numeric entry should be a 1- 4 digit hexadecimal number.

Following are examples of valid IPv6 addresses that can be included in an Address exclude list:

```
0:*:*:0:007F:0:0001:0001
F:FF:FFF:FFFF:1:01:001:0001
1234:*:4321:*:9F9f:*:*:0000
```

Note: The **Address include list** and **Host name include list** are processed before the **Address exclude list** and the **Host name exclude list**. If all four lists are defined:

- An address that is defined on either inclusion list will be allowed access provided it is not included on either of the exclusion lists.
- If an address is included in both an inclusion list and in an exclusion list, it will not be allowed access.

Address include list

Lists the IP addresses that are allowed to make inbound connections. Use a comma to separate the IPv4 or IPv6 or both addresses to which you want to grant access on inbound TCP connection requests.

All four numeric values in an IPv4 address must be represented by a number or the wildcard character * (an asterisk).

Following are examples of valid IP addresses that can be included in an Address include list:

```
*.1.255.0
254.*.*.9
1.*.*.*
```

All eight numeric values of an IPv6 address must be represented by a number or the wildcard character * (an asterisk). No shortened version of the IPv6 address should be used. Even though a shortened version is processed with no error given, it does not function correctly in this list. Each numeric entry should be a 1- 4 digit hexadecimal number.

Following are examples of valid IPv6 addresses that can be included in an **Address include list**:

```
0:*:*:0:007F:0:0001:0001
F:FF:FFF:FFFF:1:01:001:0001
1234:*:4321:*:9F9f:*:*:0000
```

Note: The Address include list and the Host name include list are processed before the Address exclude list and the Host name exclude list. If all four lists are defined:

- An address that is defined on either inclusion list will be allowed access provided it is not included on either of the exclusion lists.
- If an address is included in both an inclusion list and in an exclusion list, it will not be allowed access.

Host name exclude list

List the host names that are not allowed to make connections. Use a comma to separate the URL addresses to which you want to deny access on inbound TCP connection requests.

A URL address can start with the wildcard character * (an asterisk) followed by a period; for example, `*.Rest.Of.Address`. If a period does not follow the wildcard character, the asterisk will be treated as a normal non-wildcard character. The wildcard character cannot appear any where else in the address. For example, `ibm.*.com` is not a valid hostname.

Following are examples of valid URL addresses that can be included in a Host name exclude list:

```
*.ibm.com
www.ibm.com
*.com
```

Note: The Address include list and Host name include list are processed before the Address exclude list and the Host name exclude list. If all four lists are defined:

- An address that is defined on either inclusion list will be allowed access provided it is not included on either of the exclusion lists.
- If an address is included in both an inclusion list and in an exclusion list, it is not allowed access.

Host name include list

Lists the host names that are allowed to make inbound connections. Use a comma to separate the URL addresses to which you want to grant access on inbound TCP connection requests.

A URL address can start with the wildcard character * (an asterisk) followed by a period; for example, `*.Rest.Of.Address`. If a period does not follow the wildcard character, the asterisk will be treated as a normal non-wildcard character. The wildcard character cannot appear any where else in the address. For example, `ibm.*.com` is not a valid hostname.

Following are examples of valid URL addresses that can be included in a hostname include list:

```
*.ibm.com
www.ibm.com
*.com
```

Note: The Address include list and Host name include list are processed before the Address exclude list and the Host name exclude list. If all four lists are defined:

- An address that is defined on either inclusion list will be allowed access provided it is not included on either of the exclusion lists.
- If an address is included in both an inclusion list and in an exclusion list, it is not allowed access.

DCS transport channel settings

Use this page to view and configure an DCS transport channels. This type of transport channel handles inbound Distribution and Consistency Services (DCS) messages.

By default, two channel transport chains are defined for an application server that contains a DCS channel:

- The chain named DCS contains a TCP and a DCS channel.

- The chain named DCS-Secure contains a TCP, an SSL, and a DCS channel.

Both of these chains terminate in, or use the same TCP channel instance. This TCP channel is associated with the DCS_UNICAST_ADDRESS port and is not used in any other transport chain. One instance of an SSL channel is reserved for use in the DCS-Secure chain. It also is not used in any other transport chains.

To view this administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Ports**. Click **View associated transports** for the port associated with the DCS transport channel whose settings you want to look at.

Transport channel name
Specifies the name of the DCS transport channel.

The name field cannot contain the following characters: `# \ / , : ; " * ? < > | = + & % '`

This name must be unique across all channels in the product environment. For example, a DCS transport channel and a TCP transport channel cannot have the same name if they reside within the same system.

Data type String

Discrimination weight
Specifies the priority this channel has in relation to the other channels in this transport chain. This property is only used when port sharing is enabled and the transport chain includes multiple channels to which it might forward data. The channel in the chain with the lowest discrimination weight is the first one given the opportunity to look at incoming data and determine whether or not it owns that data.

Data type Positive integer
Default 0

SSL inbound channel
Use this page to determine which SSL inbound channel options to specify for the application server.

To view this administrative console page:
1. Click **Servers** > **Server Types** > **WebSphere application servers** > *server_name*.
2. Under Container settings, click **Web container settings** > **Web container transport chains** > i*secure_transport_chain*.
3. Under Transport channels, click **SSL Inbound Channel (SSL_1)**.

Transport Channel Name
Specifies the name of the SSL inbound channel.

The name field cannot contain the following characters: `# \ / , : ; " * ? < > | = + & % '`

This name must be unique across all channels in an application server environment. For example, an SSL inbound channel and a TCP transport channel cannot have the same name if they reside within the same system.

Data type String

Discrimination weight
Specifies the priority this channel has in relation to the other channels in this transport chain. This property is only used when port sharing is enabled and the transport chain includes multiple channels to which it

might forward data. The channel in the chain with the lowest discrimination weight is the first one given the opportunity to look at incoming data and determine whether or not it owns that data.

Data type	Positive integer
Default	0

Centrally managed
Specifies that the selection of an SSL configuration is based upon the outbound topology view for the Java Naming and Directory Interface (JNDI) platform.

Centrally managed configurations support one location to maintain SSL configurations rather than spreading them across the configuration documents.

Default:	Enabled

Specific to this endpoint
Specifies the SSL configuration alias that you want to use for outbound SSL communications.

This option overrides the centrally managed configuration for the JNDI (LDAP) protocol.

Session Initiation Protocol (SIP) inbound channel settings
Use this page to configure the SIP inbound channel settings.

To view this administrative console page, click **Servers** > **Application servers** > *server_name* > **Ports** . Click on **View associated transports** for the port associated with the UDP transport channel whose settings you want to view.

Transport channel name
Specifies the name of the SIP inbound transport channel.

The name field cannot contain the following characters: **# \ / , : ; " * ? < > | = + & %** '

This name must be unique across all channels in a WebSphere Application Server environment. For example, a SIP transport channel and a TCP transport channel cannot have the same name if they reside within the same system.

Default	UDP_(n) where (n) represents the number of instances of this channel in the system

Discrimination weight
Specifies the priority this channel has in relation to the other channels in this transport chain. This property is only used when port sharing is enabled and the transport chain includes multiple channels to which it might forward data. The channel in the chain with the lowest discrimination weight is the first one given the opportunity to look at incoming data and determine whether or not it owns that data.

Data type	Positive integer
Default	10

Session Initiation Protocol (SIP) container inbound channel settings
Use this page to configure the SIP container inbound channel settings.

To view this administrative console page, click **Servers** > **Application servers** > *server_name* > **Ports** . Click on **View associated transports** for the port associated with the UDP transport channel whose settings you want to view.

Transport channel name
Specifies the name of the SIP container inbound transport channel.

The name field cannot contain the following characters: # \ / , : ; ″ * ? < > | = + & % ’

This name must be unique across all channels in a WebSphere Application Server environment. For example, a SIP container transport channel and a TCP transport channel cannot have the same name if they reside within the same system.

Default UDP_(n) where (n) represents the number of instances of this channel in the system

Discrimination weight
Specifies the priority this channel has in relation to the other channels in this transport chain. This property is only used when port sharing is enabled and the channel chain includes multiple channels to which it might forward data. The channel in the chain with the lowest discrimination weight is the first one given the opportunity to look at incoming data and determine whether or not it owns that data.

Data type Positive integer
Default 10

Creating a new port

To create a new port and set up a channel chain to listen on a new port:

1. Go to the **Proxy Servers** > **SIP Proxy 1** > **Transport Chain** > **UDP_SIP_PROXY CHAIN** panel and select **UDP inbound channel (UDP 1)**.
2. On the following panel, select the **Port** (i.e., PROXY SIP ADDRESS (*:5060).
3. On the following panel, select **New**.

Note: See the information topic on Tuning SIP servlets for Linux for additional setting information for Linux platforms.

User Datagram Protocol (UDP) Inbound channel settings
Use this page to configure the UDP Inbound channel settings.

To view this administrative console page, click **Servers** > **Application servers** > *server_name* > **Ports** . Click on **View associated transports** for the port associated with the UDP transport channel whose settings you want to view.

Transport channel name
Specifies the name of the UDP inbound transport channel.

The name field cannot contain the following characters: # \ / , : ; ″ * ? < > | = + & % ’

This name must be unique across all channels in a WebSphere Application Server environment. For example, a UDP transport channel and a TCP transport channel cannot have the same name if they reside within the same system.

Default UDP_(n) where (n) represents the number of instances of this channel in the system

Address exclude list

Specifies the IP addresses that are not allowed to make inbound connections. Use a comma to separate the IPv4 and/or IPv6 addresses to which you want to deny access on inbound UDP connection requests.

The address include list and host name include list are processed before the address exclude list and the host name exclude list. If all four lists are defined:

- An address that is defined on either inclusion list will be allowed access provided it is not included on either of the exclusion lists.
- If an address is included in both an inclusion list and in an exclusion list, it is not allowed access.

Data type
String

Range
Valid IPv4 and IPv6 addresses with a wildcard character (*), an asterisk. All four elements of an IPv4 address must be represented by a number or a wildcard character. All eight numeric values of an IPv6 address must be represented by a number or the wildcard character (*).

Example
The following examples are valid IPv4 addresses that can be included in an Address exclude list:

```
*.1.255.0
254.*.*.9
1.*.*.*
```

All eight numeric values of an IPv6 address must be represented by a number or the wildcard character (*), an asterisk. No shortened version of the IPv6 address should be used. Even though a shortened version is processed with no error given, it does not function correctly in this list. Each numeric entry should be a 1- 4 digit hexadecimal number. The following examples are valid IPv6 addresses that can be included in an Address exclude list:

```
0:*:*:0:007F:0:0001:0001
F:FF:FFF:FFFF:1:01:001:0001
1234:*:4321:*:9F9f:*:*:0000
```

Address include list

Specifies the IP addresses that are allowed to make inbound connections. Use a comma to separate the IPv4 and/or IPv6 addresses to which you want to allow access on inbound UDP connection requests.

Data type
String

Range
Valid IPv4 and IPv6 addresses with a wildcard character (*), an asterisk. All four elements of an IPv4 address must be represented by a number or a wildcard character (*). All eight numeric values of an IPv6 address must be represented by a number or the wildcard character (*).

Web container inbound transport channel settings

Use this page to view and configure a Web container inbound channel transport. This type of channel transport handles inbound Web container requests from a remote client.

To view this administrative console page, click **Servers** → **Server Types** → **WebSphere application servers** → ***server_name*** → **Web Container Settings** → **Web container** → **Web container transport chains** → ***transport_chain*** → **Web container inbound channel (*transport_channel_name*)** .

Transport Channel Name
Specifies the name of the Web container inbound transport channel.

The name field cannot contain the following characters: # \ / , : ; " * ? < > | = + & % '

This name must be unique across all channels in a WebSphere Application Server environment. For example, a Web container inbound transport channel and a TCP transport channel cannot have the same name if they reside within the same system.

Data type String

Discrimination weight
Specifies the priority this channel has in relation to the other channels in this transport chain. This property is only used when port sharing is enabled and the transport chain includes multiple channels to which it might forward data. The channel in the chain with the lowest discrimination weight is the first one given the opportunity to look at incoming data and determine whether or not it owns that data.

Data type Positive integer
Default 0

Write buffer size
Specifies the amount of content in bytes to buffer unless the servlet explicitly calls flush/close on the response/writer output stream.

Data type bytes
Default 32768 bytes

DataPower appliance manager transport channel settings
Use this page to view and configure a DataPower appliance manager transport channel. This type of transport channel handles events from managed DataPower appliances.

To view this administrative console page, click **System administration . Deployment manager > Ports**. Find DataPowerMgr_inbound_secure, in the list of channels, and click **View associated transports**. In the list of associated transports, click DataPowerManagerInboundSecure, and then, under Transport Channels, click **DataPower appliance manager inbound channel (DPMGRDPMGR_1)**.

Transport channel name
Specifies the name of the transport channel.

The name field cannot contain the following characters: # \ / , : ; " * ? < > | = + & % '

This name must be unique across all channels in a WebSphere Application Server environment. For example, an HTTP transport channel and a TCP transport channel cannot have the same name if they reside within the same system.

Data type String

Discrimination weight
Specifies the priority this channel has in relation to the other channels in this transport chain. This property is only used when port sharing is enabled and the tranport chain includes multiple channels to which it might forward data. The channel in the chain with the lowest discrimination weight is the first one given the opportunity to look at incoming data and determine whether or not it owns that data.

Data type	Positive integer
Default	0

HTTP transport channel custom properties

If you are using an HTTP transport channel, you can add any of the following custom properties to the configuration settings for that channel.

To add a custom property:

1. In the administrative console, click **Servers > Server Types**, and then select one of the following options, depending on the type of chain you are creating:
   - **Application servers >** *server_name*, **> Web container settings > Web container transport chains >** *chain_name* **> HTTP Inbound Channel > Custom Properties > New**.
   - **Proxy servers**, and then under HTTP Proxy Server Settings, click **Proxy server transports**. Then, select either **HTTPS_PROXY_CHAIN** or **HTTP_PROXY_CHAIN**, and then click **> HTTP Inbound Channel > Custom Properties > New**.
2. Under **General Properties** specify the name of the custom property in the Name field and a value for this property in the Value field. You can also specify a description of this property in the Description field.
3. Click **Apply** or **OK**.
4. Click **Save** to save your configuration changes.
5. Restart the server.

Following are the descriptions of the HTTP transport channel custom properties provided with the product. These properties are not shown on the settings page for an HTTP transport channel.

CookiesConfigureNoCache

Use the CookiesConfigureNoCache property to specify whether the presence of a Set-Cookie header in an HTTP response message triggers the addition of several cache related headers. If this property is set to true, an Expires header with a very old date, and a Cache-Control header that explicitly tells the client not to cache the Set-Cookie header are automatically added. These headers are not automatically added if this property is set to false.

Data type	Boolean
Default	True

localLogFilenamePrefix

Use the localLogFilenamePrefix property to specify a prefix for the filename of the network log file. Normally, when inprocess optimization is enabled, requests through the inprocess path are logged based on the logging attributes set up for the Web container's network channel chain. You can use this property to add a prefix to the filename of the network log file. This new filename is then used as the filename for the log file for inprocess requests. Requests sent through the inprocess path are logged to this file instead of to the network log file. For example, if the log file for a network transport chain is named .../httpaccess.log, and this property is set to local for the HTTP channel in that chain, the filename of the log file for inprocess requests to the host associated with that chain is .../localhttpaccess.log.

Note: If you specify a value for the localLogFilenamePrefix custom property, you must also set the accessLogFileName HTTP channel custom property to the fully qualified name of the log file you want to use for in process requests. You cannot specify a variable, such as $(SERVER_LOG_ROOT), as the value for this custom property.

Data type	String

limitFieldSize
Use the limitFieldSize property to enforce the size limits on various HTTP fields, such as request URLs, or individual header names or values. Enforcing the size limits of these fields guards against possible Denial of Service attacks. An error is returned to the remote client if a field exceeds the allowed size.

Data type	Integer
Default	32768
Range	50-32768

limitNumHeaders
Use the limitNumHeaders property to limit the number of HTTP headers that can be present in an incoming message. If this limit is exceeded, an error is returned to the client.

Data type	Integer
Default	500
Range	50 to 500

RemoveServerHeader
Use the RemoveServerHeader property to force the removal of any server header from HTTP responses that the application server sends, thereby hiding the identity of the server program.

Data type	Boolean
Default	False

ServerHeaderValue
Use the ServerHeaderValue property to specify a header that is added to all outgoing HTTP responses if a server header does not already exist.

Data type	String
Default	WebSphere Application Server v/*x.x*, where *x.x* is the version of WebSphere Application Server that is running on your system.

HTTP Tunnel transport channel custom property
If you are using an HTTP Tunnel transport channel, you can add the following custom property to the configuration settings for that channel.

To add a custom property:

1. In the administrative console, click **Servers** > **Server Types** > **Application servers** > *server_name* > **Ports**. Click on **View associated transports** for the HTTP Tunnel port to whose configuration settings you want to add this custom property.
2. Click **New**.
3. Under **General Properties** specify the name of the custom property in the Name field and a value for this property in the Value field. You can also specify a description of this property in the Description field.
4. Click **Apply** or **OK**.
5. Click **Save** to save your configuration changes.
6. Restart the server.

Following is a description of the HTTP Tunnel transport channel custom property that is provided with the product. This property is not shown on the settings page for an HTTP Tunnel transport channel.

pluginConfigurable
Indicates whether or not the configuration settings for the HTTP Tunnel transport channel are included in the plugin-cfg.xml file for the Web server associated with the application server that is using this channel.

Configuration settings for each of the Web container transport channels defined for an application server are automatically included in the plugin-cfg.xml file for the Web server associated with that application server.

Data type	Boolean
Default	False

TCP transport channel custom properties
If you are using a TCP transport channel, you can use TCP transport channel custom properties to configure internal TCP transport channel properties.

To add a TCP transport channel custom property, perform the following actions.

1. In the administrative console, click **Servers > Server Types**, and then follow one of the following paths:
   - **Application servers >** *server_name*, and then select one of the following options, depending on the type of chain you are creating:
     - Expand **SIP container settings**, and click **SIP container transport chains**.
     - Expand **Web container settings**, and click **Web container transport chains**.
     - Expand **Server messaging**, and click either **Messaging engine inbound transports** or **WebSphere MQ link inbound transports**.
   - **Proxy servers**, and then expand **HTTP proxy server settings**, and click **Proxy server transports** and select either **HTTPS_PROXY_CHAIN** or **HTTP_PROXY_CHAIN**. Then click **HTTP proxy inbound channel**
2. Select the transport chain that includes the TCP channel for which you want to specify the custom property.
3. Select the **TCP inbound channel**.
4. Click **Custom properties > New**, expand **General properties**, and specify the name of the custom property in the **Name** field and a value for this property in the **Value** field. You can also specify a description of this property in the **Description** field.
5. Click **Apply** or **OK**.
6. Click **Save** to save your configuration changes.
7. Restart the server.

The following TCP transport channel custom property is provided with the product. This property is not shown on the settings page for a TCP transport channel.

listenBacklog
Use the listenBacklog property to specify the maximum number of outstanding connection requests that the operating system can buffer while it waits for the application server to accept the connections. If a client attempts to connect when this operating system buffer is full, the connect request is rejected. The value of this property is specific to each transport.

If you need to control the number of concurrent connections, use the **Maximum open connections** field on the administrative console TCP transport channel settings page.

Data type	Integer
Default	511

Transport chain problems

Review the following topics if you encounter a transport chain problem.

TCP transport channel fails to bind to a specific host/port combination

If a TCP transport channel fails to bind to a specific port, one of the following situations might have occurred:

- You are trying to bind the channel to a port that is already bound to another application, such as another instance of an application server.
- You are trying to bind to a port that is in a transitional state waiting for closure. This socket must transition to closed before you restart the server. The port might be in TIME_WAIT, FIN_WAIT_2, or CLOSE_WAIT state. Issue the `netstat -a` command from a command prompt to display the state of the port to which you are trying to bind.

  If you need to change the amount of elapse time that must occur before TCP/IP can release a closed connection and reuse its resources, see the *Tuning guide* PDF.

HP-UX

Error message CHFW0030E indicates there is ″No such file or directory,″

If you receive an Error message CHFW0030E that indicates there is no such file or directory, and you are running on an HP-UX operating system, make sure you have the most current patches for that operating system installed.

Deleting a transport chain

Transport chains cannot be deleted the same way that HTTP transports can be deleted. Because you cannot have multiple HTTP transports associated with the same port, when you delete an HTTP transport, you effectively delete the associated port and stop all traffic on that port. However, the process is more complicated for a transport chain because multiple transport chains might be associated with the same port and you do not want to disrupt traffic on transport chains that you are not deleting.

Before you begin

Determine whether you want to delete a particular transport chain or all of the transport chains that are associated with a specific port.

About this task

You might have to delete one or more transport chains if you have to delete a port.

To delete a transport chain:

1. In the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Ports**.
2. In the list of available ports, locate the port that you want to delete and click **View associated transports** for that port.
3. Select the transport chain you want to delete, and click **Delete**. If you intend to delete the port that is associated with this transport chain, repeat this step for all of the transport chains associated with this port.
4. Click **Save** to save your changes.

What to do next

If you delete all of the transport chains associated with a port, you can delete the port.

Disabling ports and their associated transport chains

Transport chains cannot be disabled the same way that HTTP transports can be disabled. Because you cannot have multiple HTTP transports associated with the same port, when you disable an HTTP transport, you effectively disable the associated port and stop all traffic on that port. However, the process is more complicated for a port that has associated transport chains because multiple transport chains might be associated with the same port, and you might not want to disrupt traffic on all of the transport chains at the same time.

Before you begin

Determine whether you want to disable a particular transport chain or all of the transport chains that are associated with a specific port.

About this task

You might need to disable a transport chain if you want to temporarily stop all incoming traffic on a particular port or on a particular transport chain that is associated with that port.

To disable a specific transport chain:

1. In the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Ports**.
2. In the list of available ports, locate the port that you want to delete and click **View associated transports** for that port.
3. Click the transport chain you want to disable.
4. Unselect the **Enabled** field, and click **OK**. If you want to temporarily stop all of the incoming traffic on a port, repeat this step for all of the transport chains associated with this port.
5. Click **Save** to save your changes.

What to do next

When you want traffic to resume on these disabled transport chains, repeat the preceding steps for all of the transport chains you disabled, and select the **Enabled** field.

SIP UDP transport channel custom properties

You can add the following custom properties to the configuration settings for the Session Initiation Protocol (SIP) UDP transport channel.

To specify custom properties for a specific UDP transport channel, navigate to the custom properties page, and specify a value for the custom property.

Note: The custom properties are supported as the primary method of configuration. Therefore, if a custom property is set and then you set the corresponding setting in the administrative console, the custom property value is used.

1. In the administrative console, expand **Servers > Server Types > WebSphere application servers >** *server_name* or **Servers > Server Types > WebSphere proxy servers >** *proxy_server_name*, to open the configuration tab for the server.
2. From **SIP container settings** or **SIP proxy server settings**, expand **SIP container transport chains** or **SIP proxy server transports**, and click **UDP specific chain**, and select **UDP inbound channel**.
3. From **Additional properties**, select **Custom properties** → **New**.
4. On the settings page, type the custom property to configure in the **Name** field, and type the value of the custom property in the **Value** field.
5. Click **Apply** or **OK**.

6. Click **Save** on the console task bar to save your configuration changes.
7. Restart the server.

The following list of UDP transport channel custom properties is provided with the product. These properties are not shown on the settings page for a UDP transport channel.

receiveBufferSizeSocket
Specifies the value in bytes for the lower level datagram buffers. This is the size of the DatagramSocket receive buffer (SO_RCVBUF) on any inbound User Datagram Protocol (UDP) channel.

Use this property to buffer multiple packets in the DatagramSocket layer. If the value of the property is too small, then packets might be lost if the server is overloaded. If the value is too large, then the packets might be delayed.

Data type	String
Default	1024000

sendBufferSizeSocket
Specifies the value, in bytes, for the lower-level datagram buffers, which is the size of the DatagramSocket send buffer (SO_SNDBUF) on any inbound UDP channel.

Use this property to buffer multiple packets in the DatagramSocket layer. If the value of the property is too small, then packets might be lost if the server is overloaded. If the value is too large, then the packets might be delayed.

Data type	String
Default	1024000

receiveBufferSizeChannel
Specifies the value, in bytes, for the maximum size of an incoming UDP packet, which is the size of the receive buffer that is allocated by the UDP channel.

Data type	String
Default	65535

Creating custom services

You can create one or more custom services for an application server. Each custom services defines a class that is loaded and initialized whenever the server starts and shuts down. Each of these classes must implement the com.ibm.websphere.runtime.CustomService interface. After you create a custom service, use the administrative console to configure that custom service for your application servers.

About this task

To define a routine that runs whenever a server or node agent starts and shuts down, you develop a custom service class and then configure a custom service instance. When the application server or node agent starts, the custom service starts and initializes.

Following is a list of restrictions that apply to the product custom services implementation. Most of these restrictions apply only to the initialize method:

- The initialize and shutdown methods must return control to the runtime.
- No work is dispatched into the server instance until all custom service initialize methods return.

- The initialize and shutdown methods are called only once on each service, and once for each operating system process that makes up the server instance.
- Initialization of process level static data, without leaving the process, is supported.
- Only JDBC RMLT (resource manager local transaction) operations are supported. Every unit of work (UOW) must be completed before the methods return.
- Creation of threads is not supported.
- Creation of sockets and I/O, other than file I/O, is not supported.
- Running standard Java Platform, Enterprise Edition (Java EE) code, such as client code, servlets, and enterprise beans, is not supported.
- The Java Transaction API (JTA) interface is not available.
- This feature is available in Java EE server processes and distributed generic server processes only.
- While the runtime makes an effort to call shutdown, there is no guarantee that shutdown will be called prior to process termination.
- JNDI operations that request resources are not supported.

1. Develop a custom service class that implements the com.ibm.websphere.runtime.CustomService interface.

   The com.ibm.websphere.runtime.CustomService interface includes an initialize and shutdown methods. The application server uses the initialize method to pass properties to the custom service. These properties can include:

   - A property that provides the name of an external file that contains configuration information for the service. You can use the externalConfigURLKey property to retrieve this information.
   - Properties that contain name-value pairs that are stored for the service, along with the other system administration configuration data for the service.

   Both the initialize and shutdown methods declare that they might create an exception, although no specific exception subclass is defined. If either method creates an exception, the runtime logs the exception, disables the custom service, and continues to start the server.
2. Configure the custom service.

   In the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name*, and then under Server Infrastructure, click **Custom Services > New**. Then, on the settings page for a custom service instance, create a custom service configuration for an existing application server or node agent, supplying the name of the class implemented. If your custom service class requires a configuration file, specify the fully-qualified path name to that configuration file in the **externalConfigURL** field. This file name is passed into your custom service class.

   To invoke a native library from the custom service, provide the path name in the **Classpath** field in addition to the path names that are used to locate the classes and JAR files for the custom service. This procedure adds the path name to the extension classloader, which allows the custom service to locate and correctly load the native library.
3. Stop the application server, and then restart it.

   If you are developing a custom service for an application server, stop the application server, and then restart the server.

   If you are developing a custom service for a node agent, stop and then restart the processing of the node agent. In the administrative console, click **System Administration > Node agents**, select the node agent you want to stop, and then click **Stop**. To restart the node agent, select the node agent you want to restart, and click **Restart**.

Results

Each custom services defines a class that is loaded and initialized whenever the server starts and shuts down.

The custom service loads and initializes whenever the server or node agent starts and stops.

Example

As previously mentioned, your custom services class must implement the com.ibm.websphere.runtime.CustomService interface. In addition, your class must implement the initialize and shutdown methods. The following example, shows the code that declares the class *ServerInit* that implements your custom service. This code assumes that your custom service class needs a configuration file. This example also includes the code that accesses the external configuration file. If your class does not require a configuration file, you do not have to include the configProperties portion of this code.

```
public class ServerInit implements com.ibm.websphere.runtime.CustomService
{
/**
* The initialize method is called by the application server runtime when the
* server starts. The Properties object that the application server passes
* to this method must contain all of  the configuration information that this
* service needs to initialize properly.
*
* @param configProperties java.util.Properties
*/
    static final java.lang.String externalConfigURLKey =
       "com.ibm.websphere.runtime.CustomService.externalConfigURLKey";

    static String ConfigFileName="";

    public void initialize(java.util.Properties configProperties) throws Exception
    {
        if (configProperties.getProperty(externalConfigURLKey) != null)
        {
           ConfigFileName = configProperties.getProperty(externalConfigURLKey);
        }

       // Implement rest of initialize method
    }
/**
* The shutdown method is called by the application server runtime when the
* server begins its shutdown processing.
*
    public void shutdown() throws Exception
    {
        // Implement shutdown method
    }
```

What to do next

Check the application server or node agent to verify that the initialize and shutdown methods of the custom service run the way that you want them to run.

Custom service collection

Use this page to view a list of services available to the application server and to see whether the services are enabled. A custom service provides the ability to plug into an application server and define code that runs when the server starts or shuts down.

To view this administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name*. Then, in the Server Infrastructure section, click **Administration > Custom services**.

If you are developing a custom service for a node agent, click **System Administration > Node agents >** *node_agent_name*. Then, in the Additional Properties section, click **Custom services** to view this administrative console page.

External Configuration URL
Specifies the URL for a custom service configuration file.

If your custom services class requires a configuration file, the value provides a fully-qualified path name to that configuration file. This file name is passed into your custom service class.

Classname
Specifies the class name of the service implementation. This class must implement the Custom Service interface.

Display Name
Specifies the name of the service.

Enable service at server startup
Specifies whether the server attempts to start and initialize the service when its containing process (the server) starts. By default, the service is not enabled when its containing process starts.

Custom service settings
Use this page to configure a service that runs in an application server.

To view this administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name*. Then, in the Server Infrastructure section, click **Administration > Custom services >***custom_service_name*.

If you are developing a custom service for a node agent, click **System Administration > Node agents >** *node agent_name*. Then, in the Additional Properties section, click **Custom services >***custom_service_name* to view this administrative console page.

Enable service at server startup:

Specifies whether the server attempts to start and initialize the service when its containing process (the server) starts. By default, the service is not enabled when its containing process starts.

Data type	Boolean
Default	false

External Configuration URL:

Specifies the URL for a custom service configuration file.

If your custom services class requires a configuration file, specify the fully-qualified path name to that configuration file for the value. This file name is passed into your custom service class.

Data type	String
Units	URL

Classname:

Specifies the class name of the service implementation. This class must implement the Custom Service interface.

Data type	String
Units	Java class name

Display Name:

Specifies the name of the service.

Data type String

Description:

Describes the custom service.

Data type String

Classpath:

Specifies the class path used to locate the classes and JAR files for this service.

Data type String
Units Class path

Defining application server processes

To enhance the operation of an application server, you can define command-line information for starting or initializing an application server process. Such settings define runtime properties such as the program to run, arguments to run the program, and the working directory.

About this task

A process definition can include characteristics such as Java virtual machine (JVM) settings, standard in, error and output paths, and the user ID and password under which a server runs.

You can define application server processes using the administrative console or the wsadmin tool.

1. In the administrative console, click **Servers > Server Types > WebSphere application servers**, and then click on an application server name.
2. In the Server Infrastructure section, click **Java and process management > Process definition**.
3. On the settings page for a process definition, specify the name of the executable to run, any arguments to pass when the process starts running, and the working directory in which the process will run. Then click **OK**.
4. UNIX Specify process execution statements for starting or initializing a UNIX or i5/OS process.
5. Specify monitoring policies to track the performance of a process.
6. Specify process logs to which standard out and standard error streams write. Complete this step if you do not want to use the default file names.
7. Specify name-value pairs for properties needed by the process definition.

   Note: Each custom property name must be unique. If the same name is used for multiple properties, the process uses the value specified for the first property that has that name.
8. Optional: Prevent the application server from creating javacore dumps.

   A javacore dump, or a thread dump as it is also called, is one of the primary problem determination documents that an application server creates. Also, the performance impact of creating a javacore dump is usually ignorable. Therefore, in most product environments, you should not suppress the creation of a javacore dump.

   In certain circumstances, such as when there are security consideration, you might want to prevent the application server from creating javacore dumps. To disable the javacore dump function:

a. In the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name*, and then in the Server Infrastructure section, click **Java and process management > Process definition > Java virtual machine > Custom properties > New**
b. In the Name field enter `DISABLE_JAVADUMP` and in the Value field, enter `true` to prevent the application server from creating javacore dumps.

9. Stop the application server, and then have the executable, that the process definition specifies, restart the server. If the executable cannot restart the application server, the executable should use the generic server.
10. Check the server to verify that the process definition runs and operates as intended.

Process definition settings

Use this page to configure a process definition. A process definition includes the command line information necessary to start or initialize a process.

To view this administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name*. Then, in the Server Infrastructure section, click **Java and process management > Process definition**.

Executable name

This command line information specifies the executable name that is invoked to start the process.

For example, if you are using a perl script as a generic server, enter the path to the perl.exe module in the Executable name field, and then enter the name of the perl script, along with any arguments, in the Executable arguments field.

Data type String

Executable arguments

This command line information specifies the arguments that are passed to the executable when starting the process.

You can enter multiple arguments in this field, but they must be separated by carriage returns. Use the Enter key on your keyboard to create these carriage returns. The following example illustrates how a perl script application that requires two arguments should appear in this field:

```
perl_application.pl
arg1
arg2
```

Data type String
Units Command-line arguments

Start command (startCommand)

This command line information specifies the platform-specific command to launch the server process.

Start command arguments (startCommandArgs)

This command line information specifies any additional arguments required by the start command.

Stop command (stopCommand)

This command line information specifies the platform-specific command to stop the server process

Specify two commands in the field, one for the Stop command, and one for the Immediate Stop (CANCEL) command.

Data type String

Format STOP *server_short_name*;CANCEL *server_short_name*

Stop command arguments (stopCommandArgs)
This command line information specifies any additional arguments required by the stop command.

Specify arguments for the Stop command and the Immediate Stop (CANCEL) command.

Data type String
Format *stop command arg string*;*immediate stop command arg string*

Terminate command (terminateCommand)
This command line information specifies the platform-specific command to terminate the server process.

Data type String
Format FORCE *server_short_name*

Terminate command arguments (terminateCommandArgs)
This command line information specifies any additional arguments required by the terminate command.

The default is an empty string.

Data type String
Format *terminate command arg string*

Working directory
Specifies the file system directory that the process uses as its current working directory. This setting only applies for i5/OS and distributed platforms. The process uses this directory to determine the locations of input and output files with relative path names.

Data type String

Executable target type
Specifies whether the executable target is a Java class or an executable JAR file.

Executable target
Specifies the name of the executable target. If the target type is a Java class name, this field contains the main() method. If the target type is an executable JAR file, this field contains the name of that JAR file.

Data type String

Process execution settings
Use this page to view or change the process execution settings for a server process.

A server process applies to either an application server, a node agent or a deployment manager.

If you are running on i5/OS or a distributed operating systems, to view this administrative console page for an application server, click **Servers > Server Types > WebSphere application servers >** *server_name*. Then, in the Server Infrastructure section, click **Java and process management > Process execution**.

To view this administrative console page for a node agent, click **System Administration > Node agents >***nodeagent_name*. Then, in the Server Infrastructure section, click **Java and process management > Process definition > Process execution**.

To view this administrative console page for a deployment manager, click **System Administration > Deployment manager**. Then, in the Server Infrastructure section, click **Java and process management > Process definition > Process execution**.

Process Priority:

Specifies the operating system priority for the process. The administrative process that launches the server must have root operating system authority in order to honor this setting.

Data type Integer
Default 20

UMASK:

Specifies the user mask under which the process runs (the file-mode permission mask).

The deployment manager and application servers must run with a 007 umask in order to support system management functions. Therefore, it is recommended that you do not change the default value of this setting for the deployment manager or the controller.

Data type Integer
Default 007

Run As User: AIX HP-UX Linux Solaris UNIX

Specifies the user that the process runs as. This user ID must be defined to the security system.

Windows This field is ignored if you are running on the Microsoft Windows operating system.

Data type String

Run As Group: AIX HP-UX Linux Solaris UNIX

Specifies the group that the process is a member of and runs as.

This field is ignored if you are running on the Microsoft Windows operating system.

Data type String

Run In Process Group: AIX HP-UX Linux Solaris UNIX

Specifies a specific process group for the process. A process group is a mechanism that the operating system uses to logically associate multiple processes and operate on them as a single unit. Usually, the operating system uses this mechanism for signal distribution.

Specific operating systems might allow other operations to be performed on a process group. Refer to your operating system documentation for more information on the operations that can be performed on a process group.

Windows This field is ignored if you are running on i5/OS, or on the Microsoft Windows operating systems.

Data type Integer

Default 0, which indicates that the process is not assigned to a specific process group.

Process logs settings

Use this page to view or change settings for specifying the files to which standard out and standard error streams write.

To view this administrative console page, in the administrative console:

For an application server that is running i5/OS or a distributed operating system, click **Servers > Server Types > WebSphere application servers >** *server_name*, and then, under Server Infrastructure, click **Java and process management > Process definition > Process logs**.
For a deployment manager that is running on i5/OS or distributed operating system, click **System Administration > Deployment manager**, and then under Server Infrastructure, click **Java and process management > Process definition > Process logs**.

Stdout File Name:

Specifies the file to which the standard output stream is directed. The file name can include a symbolic path name defined in the variable entries.

Use the field on the configuration tab to specify the file name. Use the field on the Runtime tab to select a file for viewing. View the file by clicking **View**.

Direct server output to the administrative console or to the process that launched the server, by either deleting the file name or specifying `console` on the configuration tab.

Data type String
Units File path name

Stderr File Name:

Specifies the file to which the standard error stream is directed. The file name can include a symbolic path name defined in the variable entries.

Use the field on the configuration tab to specify the file name. Use the field on the runtime tab to select a file for viewing. View the file by clicking **View**.

Data type String
Units File path name

Monitoring policy settings

Use this page to view or change settings that control how the node agent monitors and restarts a process.

To view this administrative console page, click **Servers > Server Types > WebSphere application servers >** *server_name*. Then, under Server Infrastructure, click **Java and process management > Monitoring policy**.

Maximum Startup Attempts:

Specifies the maximum number of times to attempt to start the application server before giving up.

Data type Integer

Ping Interval:

Specifies, in seconds the frequency of communication attempts between the parent process, such as the node agent, and the process it has spawned, such as an application server. Adjust this value based on your requirements for restarting failed servers. Decreasing the value detects failures sooner; increasing the value reduces the frequency of pings, reducing system overhead.

Data type	Integer
Range	Set the value greater than or equal to 0 (zero) and less than 2147483. If you specify a value greater than 2147483, the application server acts as though you set the value to 0. When you specify a value of 0, no checking is performed.

Ping Timeout:

When a parent process is spawning a child process, such as when a process manager spawns a server, the parent process pings the child process to see whether the child was spawned successfully. This value specifies the number of seconds that the parent process should wait (after pinging the child process) before assuming that the child process failed.

Data type	Integer
Units	Seconds
Range	Set the value greater than or equal to 0 (zero) and less than 2147483647. If you specify a value greater than 2147483647, the application server acts as though you set the value to 0.

Automatic Restart:

Specifies whether the process should restart automatically if it fails.

If you change the value specified for this field, you must restart the application server and the node agent before the new setting takes effect.

This setting does not affect what you specified for the Node Restart State setting. The two settings are mutually exclusive.

Data type	Boolean
Default	true for the distributed and i5/OS environments

Node Restart State:

The setting only displays for the Network Deployment product. It specifies the desired behavior of the servers after the node completely shuts down and restarts.

If a server is already running when the node agent stops, that server is still running after the node agent restarts. If a server is stopped when the node agent restarts, whether the node agent starts the server depends on the setting for this property:

- If this property is set to STOPPED, node agent does not start the server.
- If this property is set to RUNNING, the node agent always starts the server.
- If this property is set to PREVIOUS, the node agent starts the server only if the server was running when the node agent stopped.

This setting does not affect what you specified for the Automatic Restart setting. The two settings are mutually exclusive.

Data type	String
Default	STOPPED
Range	Valid values are STOPPED, RUNNING, or PREVIOUS. If you want the process to return to its current state after the node restarts, use PREVIOUS.

Automatically restarting server processes

There are several server processes that the operating system can monitor and automatically restart when the server processes stop abnormally.

Before you begin

AIX HP-UX Linux Solaris To set up this function on a Linux or supported UNIX operating system, you must have root authority to edit the inittab file.

Windows To set up this function on a Microsoft Windows operating system, you must belong to the Administrator group and have the following advanced user rights:

- Act as part of the operating system
- Log on as a service

The Installation wizard grants you the user rights if your user ID is part of the administrator group.

Windows If you are running on a Microsoft Windows Operating System, the Installation wizard displays a message that states that although the advanced user rights are now effective, they do not display as effective until the next time you log on to the Windows machine.

Windows You can also add the advanced user rights manually if you are performing a silent installation on a Windows operating system. For example, to grant the user rights to your administrator group user ID on a Windows operating system, perform the following procedure:

1. Click **Administrative Tools** in the Control Panel.
2. Click **Local Security Policy**.
3. Click **Local Policies**.
4. Click **User Rights Assignments**.
5. Right click **Act as part of the operating system**.
6. Click **Security**.
7. Click **Add**.
8. Click your user ID.
9. Click **Add**.
10. Click **OK**.
11. Click **OK**.
12. Right click **Log on as a service**.
13. Click **Security**.
14. Click **Add**.
15. Click **OK**.
16. Click **OK**.
17. Reboot your machine to make the settings effective.

Windows Consult your Windows help system for more information.

About this task

There are several environments where you might use this function of automatically restarting servers. You can restart the ***server1*** managed node process, for example. Here is a list of processes you might consider restarting:

- The **server1** managed node process
- The **server1** process on a stand-alone Application Server
- The **dmgr** process on a deployment manager node
- The **nodeagent** server process on any managed node
- The **IBM HTTP Server** process
- The **IBM HTTP Administration** process

Windows On a Windows operating system, you can create Windows services during installation, using the installation wizard. Each Windows service controls a single process, such as a stand-alone product instance. Multiple stand-alone Application Server processes require multiple Windows services, which you can define. The wizard lets you create services for these servers:

- The **server1** managed node process, defined as a manually started (versus automatic) service
- The **server1** stand-alone Application Server process, defined as a manually started service
- The **IBM HTTP Server** process and the **IBM HTTP Administration** process, defined as automatically started services when you choose to install the IBM HTTP Server feature
- The **dmgr** process on a deployment manager node, defined as a manually started service

The installation wizard does not provide a way to create a service for a node agent because the deployment manager instantiates each node agent after installation when you add an Application Server node to the deployment manager cell. For this reason, you must manually create a function that automatically starts a failed node agent server process.

AIX HP-UX Linux Solaris On a Linux or supported UNIX operating system, you must manually create a shell script that automatically starts any of the processes previously mentioned. Each UNIX shell script controls a single process, such as a stand-alone product instance. Multiple stand-alone Application Server processes require multiple UNIX scripts, which you can define.

Windows AIX HP-UX Linux Solaris In a Network Deployment environment, the **addNode** or **startNode** command starts a single unmonitored node agent only, the nodeagent process, and does not start all of the processes that you might define on the node. While running, the node agent monitors and restarts Application Server processes on that node, on either a Windows or a Linux and UNIX-based platform. Each Application Server process has MonitoringPolicy configuration settings that the node agent uses when monitoring and restarting the process.

It is recommended that you manually set up a monitored process for the deployment manager dmgr server and for any node agent defined for your system. To set up a monitored process:

- Windows On a Windows operating system, use a Windows service. You can install the Network Deployment version of the product as a Windows service during installation, or at a later time
- AIX HP-UX Linux Solaris On a Linux or supported UNIX operating system, use the rc.was example shell script that is provided with the appropriate version of the product.

1. Windows On a Windows operating system, **Use the Profile Management tool** to set up a Windows service to automatically monitor and restart processes related to the product.
   - Perform the following procedure from the Profile Management tool to select services that the installation wizard can set up:
     a. Click **Run WebSphere Application Server Network Deployment as a service**.

        If you select this option, the installation wizard creates the following service during installation:

IBMWAS6Service - *node_name*

IBMWAS6Service - *node_name* service controls the *node_name* process.

After you complete and verify the installation, use the Windows Services panel to change the **IBMWAS6Service -** *node_name* service to an automatic startup type.

1) Right click **IBMWAS6Service -** *node_name* and click **Properties**.
2) Click **Automatic** from the **Startup type** list box and click **OK**.

b. Click **Run IBM HTTP Server as a service**.

Select this option on the machine where you are installing the IBM HTTP Server.

If you select this option, the installation wizard creates the following services during the installation:

- **IBM HTTP Server 2.0.*x***
- **IBM HTTP Administration 2.0.*x***

The installation wizard defines the startup type of these services as **automatic**. It is not necessary for you to change the type from manual to automatic.

c. Enter your user ID and password and click **Next**.

In a coexistence environment, you can change the default service names to make them unique. In a same version coexistence scenario for IBM HTTP Server 2.0.x on a Windows platform, you cannot use the default service names created by the installer because they are common.

To work around this problem:

a. Install the first copy of IBM HTTP Server, either by itself or with the product, and select to install the services.
b. Customize the service names for the first install by running the following commands from the first install location:

```
apache -k install -n "IHS 2.0(1)"
apache -k install -f conf\admin.conf -n "IHS 2.0 Administration (1)"
```

c. Edit the AdminAlias directive in the *installLocation 1*\conf\admin.conf file to point to the new service name, such as **IHS 2.0(1)**.
d. Remove the default service names installed by the first install by running the following commands:

```
apache -k uninstall -n "IBM HTTP Server 2.0"
apache -k uninstall -n "IBM HTTP Administration 2.0"
```

e. Install the second copy of IBM HTTP Server, either by itself or with the product. The default service names correspond to the second install.

Note: Customized service names must be unique on your system.

2. AIX HP-UX Linux Solaris On a Linux or supported UNIX operating system, after you install the product, set up a shell script to automatically monitor and restart any related server processes.

a. Locate the rc.was example shell script, which is in the *app_server_root*/bin directory.
b. Create a new shell script for each process that the operating system is to monitor and restart.
c. Edit each shell script according to comments in its header, which provide instructions for identifying a product process.
d. Edit the inittab file of the operating system, to add an entry for each shell script you have created.

Comments in the header of the rc.was file include a sample inittab entry line for adding this script to the inittab table. Each inittab entry causes the operating system to call the specified shell script whenever the system initializes. As each shell script runs, it monitors and starts the server process you specified.

For example, if you create the following inittab entry for a process, the rc.was shell script is run whenever the system initializes, and if the process goes down while the system is initializing into a machine that is operating at a runlevel of 2, 3, or 5:

```
was:235:respawn:/usr/WebSphere/AppServer/bin/rc.was >/dev/console 2>&1
```

If you create the following inittab entry, the rc.was shell script only runs once when you initialize into a machine that is operating at a runlevel of 2,3, or 5:

```
was:235:once:/usr/WebSphere/AppServer/bin/rc.was >/dev/console 2>&1
```

Following is a list of the runlevels that can be specified. Runlevels usually default to either 3 or 5.

0-halt

1-Single user mode

2-Multiuser, without NFS (The same as 3, if you don't have networking)

3-Full multiuser mode

4-unused

5-X11

6-Reboot

If you don't know the runlevel into which your machine is booting, look at the following line in the inittab file:

```
id:x:initdefault
```

where *x* is the runlevel that the machine is booting into.

For example, if your machine is booting into a runlevel of 5, then all of the processes that are declared to run with a runlevel of 5 are started.

Note: Everything that is ran from the inittab file runs under the root user. Therefore, if you need the server to automatically start the process under a non-root user ID when the machine starts, you must also add the following line to the inittab file:

```
su user -c values
```

where *values* is the file path and arguments that are used to call the rc scripts you created, and *user* is the non-root user that you have configured the product to run as.

Each shell script monitors and restarts the following processes in a Network Deployment environment:

- A server process on a managed node
- A node agent process on a managed node
- A stand-alone Application Server process
- A deployment manager process

3. Windows On a Windows operating system, after installing the product, use the WASService.exe command to manually define the nodeagent server process as a Windows service.

   You can use the same command to manually define a Windows service for another installation instance or for another configuration instance of either the server1 process or the dmgr process.

4. Click **Apply** and then click **Save** to save the change directly to the master configuration.

Results

Windows On a Windows operating system, you can

- Use the **net start** and **net stop** commands to control the IBM HTTP Server services on a Windows system. For more information about these commands, see the Windows help file. Access these commands from the Start menu, clicking **Start > Programs > IBM HTTP Server**.
- Use the **Start the Server** and **Stop the Server** commands to control the product process. Access these commands from the Start menu, clicking **Start > Programs > IBM WebSphere > Application Server V6**.
- Use the **Start the Manager** and **Stop the Manager** commands to control the Network Deployment dmgr process. Access these commands from the Start menu, clicking **Start > Programs > IBM WebSphere > Application Server V6 > Deployment Manager**.

Processes started by a **startServer** command, a **startNode** command, or a **startManager**command are not running as monitored processes, regardless of how they are configured.

For example, you can configure a server1 process as a monitored process. However, if you start the server1 process using the **startServer** command, the operating system does not monitor or restart the server1 process because the operating system did not originally start the process as a monitored process.

What to do next

After the process is set up, the operating system can monitor each server process and restart the process if it stops.

Return to the Defining application server processes administrative console page to continue.

WASService command

The **WASService** command line tool enables you create a service for a product Java process on Linux and Windows operating systems.

You can create services for WebSphere Application Server Java processes. Potential services include the following server processes:

- The default server1 process on an application server node
- Application server processes that you create on an application server node
- The nodeagent process on an application server node that is part of a deployment manager cell
- The deployment manager process, dmgr

Note: Do not add an application server that is part of a federated or managed node as a Windows service. Use the node agent to manage federated nodes.

Windows To set up and run this function on a Microsoft Windows operating system, the user must belong to the administrator group and have the following advanced user rights:

- Act as part of the operating system
- Log on as a service

Location of the command file: Linux The wasservice.sh command file is located in the *app_server_root*\bin directory.

Windows The WASService.exe command file is located in the *app_server_root*\bin directory.

Command syntax:

Command syntax for starting an existing service

The command syntax is as follows: Linux

```
wasservice.sh -start service_name [optional startServer.bat parameters]
```

Windows

```
WASService.exe -start service_name [optional startServer.bat parameters]
```

Command syntax for creating a service or updating an existing service

The command syntax is as follows: Linux

```
wasservice.sh -add service_name
   -serverName server_name
   -profilePath server_profile_directory
```

```
   [-wasHome app_server_root]
   [-startArgs additional_start_arguments]
   [-stopArgs additional_stop_arguments]
   [-userid user_id -password password]
```

Windows

```
WASService.exe -add service_name
   -serverName server_name
   -profilePath server_profile_directory
   [-wasHome app_server_root]
   [-configRoot configuration_repository_directory]
   [-startArgs additional_start_arguments]
   [-stopArgs additional_stop_arguments]
   [-userid user_id -password password]
   [-logFile service_log_file]
   [-logRoot server_log_directory]
   [-restart true | false]
   [-startType automatic | manual | disabled]
```

Command syntax for deleting a service

The command syntax is as follows: Linux

```
wasservice.sh -remove service_name
```

Windows

```
WASService.exe -remove service_name
```

Command syntax for stopping a running service

The command syntax is as follows: Linux

```
wasservice.sh -stop service_name [optional stopServer.bat parameters]
```

Windows

```
WASService.exe -stop service_name [optional stopServer.bat parameters]
```

Command syntax for retrieving service status

The command syntax is as follows: Linux

```
wasservice.sh -status service_name
```

Windows

```
WASService.exe -status service_name
```

Parameters: Supported arguments include:

-add *service_name*
Creates a service named *service_name* or updates an existing service. The syntax is the same for both cases.

-configRoot *configuration_repository_directory*
Optional parameter that identifies the configuration directory of the installation root directory of a WebSphere Application Server product.

-encodeParams *service_name*
Optional parameter that forces the service to encode the -startArgs and -stopArgs so that the arguments cannot be determined by editing the registry. Use the parameter when creating a service with the -add parameter by adding -encodeParams to the command line with no arguments.

Windows Or encode the parameters of an existing service:

```
WASService -encodeParams service_name
```

-logFile *service_log_file*
Optional parameter that identifies a log file that the **WASService** command uses to record its activity.

-logRoot *server_log_directory*
Required parameter that identifies the server log directory for the profile. The **WASService** command looks for a file named *server_name*.pid to determine if the server is running.

-profilePath *server_profile_directory*
Specifies the directory path of the profile that defines the server process.

-remove *service_name*
Deletes the specified service.

-restart true | false
Restarts the existing service automatically if the service fails when set to true.

-serverName *server_name*
Identifies the server that the service controls.

-start *service_name* **[optional startServer.bat parameters]**
Starts the existing service.

-startArgs *additional_start_arguments*
Optional parameter that identifies additional parameters.

-startType automatic | manual | disabled
Defines the startup type of the new service. An automatic startup type starts automatically when the system starts or when the service is called for the first time. You must start a manual service before the operating system can load it and make it available. You cannot start a disabled service before changing the startup type.

-status *service_name*
Returns the current status of the service, which includes whether the service is running or stopped.

-stop *service_name* **[optional stopServer.bat parameters]**
Stops the specified service.

-stopArgs *additional_stop_arguments*
Optional parameter that identifies additional parameters.

-userid *user_id* **-password** *password*
Optional parameters that identify a privileged user ID and password that the Windows service will run as.

-wasHome *app_server_root*
Optional parameter that identifies the installation root directory of the product.

Default names for services that are created by the wizard: The names of the services that the Profile Management Tool can create are:

Deployment manager
IBM WebSphere Application Server V6.x - *node_name_of_the_deployment_manager_node*

Application server
IBM WebSphere Application Server V6.x - *node_name_of_the_server1_node*

Custom profile
After federating the node and creating an application server, a service can be created called IBM WebSphere Application Server V6.x - *node_name_of_the_managed_node*.

After creating a custom profile, you must federate the node to create a node agent server on the node. You can also use the administrative console of the deployment manager to create application server processes on the node. You can create a Windows service for the node agent server process.

A node agent server is also created after adding an application server node to a deployment manager cell. You can create a service for the node agent server process as described later.

Viewing the services panel: To view services, open the Control panel and click **Administrative Tools > Services**. Select a service to view information about it. Right click the service and click **Properties**. Four tabs provide information and functionality. For example, select the **Setup type** field on the **General** tab to change the setup type.

Examples: Windows

Creating a deployment manager service

This example creates a service called *IBM WebSphere Application Server V6.x - name_of_the_deployment_manager_service* that starts the dmgr process:

```
WASService -add name_of_the_deployment_manager_service
   -servername deployment_manager_server_name
   -profilePath profile_root
   -wasHome app_server_root
   -logFile profile_root\logs\WS_startManager.log
   -logRoot profile_root\logs\deployment_manager_server_name
   -restart true
```

where

- *name_of_the_deployment_manager_service* is the name that you want to give to the service
- *deployment_manager_server_name* is the name of your server

After entering the command, messages that are similar to those in the following example display in the command window:

```
Adding Service: name_of_the_deployment_manager_service
   Config Root: profile_root\config
   Server Name: deployment_manager_server_name
   Profile Path: profile_root
   Was Home: app_server_root
   Start Args:
   Restart: 1
IBM WebSphere Application Server V6.x - name_of_the_deployment_manager_service service successfully added.
```

Click **Start > Settings > Control Panel > Administrative Tools > Services** to work with the new service.

Creating a node agent service

This example creates a service called *IBM WebSphere Application Server V6.x - name_of_the_node_agent_service* that starts the nodeagent process:

```
WASService -add name_of_the_node_agent_service
   -servername node_agent_server_name
   -profilePath profile_root
   -wasHome app_server_root
   -logFile profile_root\logs\WS_startNode.log
   -logRoot profile_root\logs\node_agent_server_name
   -restart true
```

where

- *name_of_the_node_agent_service* is the name that you want to give to the service
- *node_agent_server_name* is the name of your server

After entering the command, messages that are similar to those in the following example display in the command window:

```
Adding Service: name_of_the_node_agent_service
   Config Root: profile_root\config
   Server Name: node_agent_server_name
   Profile Path: profile_root
   Was Home: app_server_root
   Start Args:
   Restart: 1
IBM WebSphere Application Server V6.x - name_of_the_node_agent_service service successfully added.
```

Creating an application server service

This example creates a service called *IBM WebSphere Application Server V6.x - name_of_the_application_server_service* that starts an application server process:

```
WASService -add name_of_the_application_server_service
   -servername application_server_name
   -profilePath profile_root
   -wasHome app_server_root
   -logFile profile_root\logs\WS_startServer.log
   -logRoot profile_root\logs\application_server_name
   -restart true
```

where

- *name_of_the_application_server_service* is the name that you want to give to the service
- *application_server_name* is the name of your server

After entering the command, messages that are similar to those in the following example display in the command window:

```
Adding Service: name_of_the_application_server_service
  Config Root: profile_root\config
  Server Name: application_server_name
  Profile Path: profile_root
  Was Home: app_server_root
  Start Args:
  Restart: 1
IBM WebSphere Application Server V6.x - name_of_the_application_server_service service successfully added.
```

Updating an existing application server service

This example for the Windows operating system updates an existing service called *IBM WebSphere Application Server V6.x - Server2 Service* with additional stop arguments, the user name and password. The parameters are automatically passed into the script that the Windows service uses to shutdown the system.

```
WASService -add "Server2 Service"
   -servername server2
   -profilePath profile_root
   -logRoot profile_root\logs\server2
   -stopArgs "-username user_name  -password password"
   -encodeParams
```

Starting and stopping a server process after creating a Windows service: Windows For this Windows operating system example, if you issue the startServer server1 command or the stopServer server1 after creating a Windows service for server1, a message that is similar to the following example displays:

```
Because server1 is registered to run as a Windows Service, the
request to start this server will be completed by starting the
associated Windows Service.
```

If you issue the startNode command or the stopNode command after creating a Windows service for the nodeagent process, a message that is similar to the following example displays:

```
Because nodeagent is registered to run as a Windows Service, the
request to start or stop this server will be completed by
starting or stopping the associated Windows Service. Examine
the log files to view messages related to this command.
```

If you issue the startManager command or the stopManager command after creating a Windows service for the deployment manager, a message that is similar to the following example displays:

```
Because dmgr is registered to run as a Windows Service, the
request to start or stop this server will be completed by
starting or stopping the associated Windows Service. Examine
the log files to view messages related to this command.
```

Configuring the JVM

As part of configuring an application server, you might define settings that enhance the way your operating system uses of the Java virtual machine (JVM).

About this task

The Java virtual machine (JVM) is an interpretive computing engine responsible for running the byte codes in a compiled Java program. The JVM translates the Java byte codes into the native instructions of the host machine. The application server, being a Java process, requires a JVM in order to run, and to support the Java applications running on it. JVM settings are part of an application server configuration.

To view and change the JVM configuration for an application server's process, use the Java virtual machine page of the administrative console or use wsadmin to change the configuration through scripting.

1. In the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name*. Then, under Server Infrastructure, click **Java and process management > Process definition**
2. Select **Java virtual machine**.
3. Specify values for the JVM settings as needed and click **OK**.
4. Click **Save** on the console task bar.
5. Restart the application server.

Example

"Configuring application servers for UCS Transformation Format" on page 302 provides an example that involves specifying a value for the **Generic JVM Arguments** property on the Java virtual machine page to enable UTF-8 encoding on an application server. Enabling UTF-8 allows multiple language encoding support to be used in the administrative console.

"Configuring JVM sendRedirect calls to use context root" on page 292 provides an example that involves defining a property for the JVM.

Java virtual machine settings

Use this page to view, and change the Java virtual machine (JVM) configuration settings of a process for an application server.

To view this administrative console page, connect to the administrative console and navigate to the Java virtual machine panel.

For i5/OS and distributed platforms, follow one of the following paths.

Application server	**Servers > Server Types > WebSphere application servers >** *server_name*. Then, in the Server Infrastructure section, click **Java and process management > Process definition > Java virtual machine**
Deployment manager	**System Administration > Deployment manager**. Then, in the Server Infrastructure section, click **Java and process management > Process definition > Java virtual machine**
Node agent	**System Administration > Node agent >** *node_agent*. Then, in the Server Infrastructure section, click **Java and process management > Process definition > Java virtual machine**

Classpath

Specifies the standard class path in which the Java virtual machine code looks for classes.

If you need to add a classpath to this field, enter each classpath entry into a separate table row. You do not have to add a colon or semicolon at the end of each entry.

The only classpaths that should be added to this field are the ones that specify the location of the following items:

- An inspection or monitoring tool to your system.
- JAR files for a product that runs on top of this product.
- JVM diagnostic patches or fixes.

Processing errors might occur if you add classpaths to this field that specify the location of the following items:

- JAR files for resource providers, such as DB2. The paths to these JAR files should be added to the relevant provider class paths.
- A user JAR file that is used by one or more of the applications that you are running on the product. The path to this type of JAR file should be specified within each application that requires that JAR file, or in server-associated shared libraries.
- An extension JAR file. If you need to add an extension JAR file to your system, you should use the ws.ext.dirs JVM custom property to specify the absolute path to this JAR file. You can also place the JAR file in the WAS_HOME/lib/ext/ directory, but using the ws.ext.dirs JVM custom property is the recommended approach for specifying the path to an extension JAR file.

Data type String

Boot classpath

Specifies bootstrap classes and resources for JVM code. This option is only available for JVM instructions that support bootstrap classes and resources.

If you need to add a classpath to this field, enter each classpath entry into a table row. You do not need to add the colon or semicolon at the end of each entry.

If you need to add multiple classpaths to this field, you can use either a colon (:) or semi-colon (;), depending on which operating system the node resides, to separate these classpaths.

The only classpaths that should be added to this field are the ones that specify the location of the following items:

- An inspection or monitoring tool to your system.
- JAR files for a product that runs on top of this product.
- JVM diagnostic patches or fixes.

Processing errors might occur if you add classpaths to this field that specify the location of the following items:

- JAR files for resource providers. such as DB2. The paths to these JAR files should be added to the relevant provider class paths.
- A user JAR file that is used by one or more of the applications that you are running on the product. The path to this type of JAR file should be specified within each application that requires that JAR file, or in server-associated shared libraries.
- An extension JAR file. If you need to add an extension JAR file to your system, you should use the ws.ext.dirs JVM custom property to specify the absolute path to this JAR file. You can also place the JAR file in the WAS_HOME/lib/ext/ directory, but using the ws.ext.dirs JVM custom property is the recommended approach for specifying the path to an extension JAR file.

Verbose class loading

Specifies whether to use verbose debug output for class loading. The default is to not enable verbose class loading.

If verbose class loading is enabled, the debug output is sent to one of the native process logs.

Data type	Boolean
Default	false

Verbose garbage collection

Specifies whether to use verbose debug output for garbage collection. The default is not to enable verbose garbage collection.

If verbose garbage collection is enabled, the debug output is sent to one of the native process logs.

,

Data type	Boolean
Default	false

When this field is enabled, a report is written to the output stream each time the garbage collector runs. This report should give you an indication of how the Java garbage collection process is functioning.

You can check the verboseGC report to determine:

- How much time the JVM is spending performing garbage collection.

  Ideally, you want the JVM to spend less than 5 percent of its processing time doing garbage collection. To determine the percentage of time the JVM spends in garbage collection, divide the time it took to complete the collection by the length of time since the last AF and multiply the result by 100. For example,

  ```
  83.29/3724.32 * 100 = 2.236 percent
  ```

  If you are spending more than 5 percent of your time in garbage collection and if garbage collection is occurring frequently, you might need to increase your Java heap size.
- If the allocated heap is growing with each garbage collection occurrence.

  To determine if the allocated heap is growing, look at the percentage of the heap that is remains unallocated after each garbage collection cycle, and verify that the percentage is not continuing to

decline. If the percentage of free space continues to decline you are experiencing a gradual growth in the heap size from garbage collection to garbage collection. This situation might indicate that your application has a memory leak.

Verbose JNI

Specifies whether to use verbose debug output for native method invocation. The default is not to enable verbose Java Native Interface (JNI) activity.

Data type	Boolean
Default	false

Initial heap size

Specifies, in megabytes, the initial heap size available to the JVM code. If this field is left blank, the default value is used.

For i5/OS and distributed platforms, the default initial heap size is 50 MB.

Note: These default values are sufficient for most applications.

Increasing this setting can improve startup. The number of garbage collection occurrences are reduced and a 10 percent gain in performance is achieved.

Increasing the size of the Java heap continues to improves throughput until the heap becomes too large to reside in physical memory. If the heap size exceeds the available physical memory, and paging occurs, there is a noticeable decrease in performance.

Maximum heap size

Specifies, in megabytes, the maximum heap size that is available to the JVM code. If this field is left blank, the default value is used.

Increasing the maximum heap size setting can improve startup. When you increase the maximum heap size, you reduce the number of garbage collection occurrences with a 10 percent gain in performance.

Increasing this setting usually improves throughput until the heap becomes too large to reside in physical memory. If the heap size exceeds the available physical memory, and paging occurs, there is a noticeable decrease in performance. Therefore, it is important that the value you specify for this property allows the heap to be contained within physical memory.

Note: These default values are appropriate for most applications. Enable the **Verbose garbage collection** property if you think garbage collection is occurring too frequently. If garbage collection is occurring too frequently, increase the maximum size of the JVM heap.

Run HProf

Specifies whether to use HProf profiler support. To use another profiler, specify the custom profiler settings using the **HProf Arguments** setting. The default is not to enable HProf profiler support.

If you set the **Run HProf** property to `true`, then you must specify command-line profiler arguments as values for the **HProf Arguments** property.

Data type	Boolean
Default	false

HProf arguments

Specifies command-line profiler arguments to pass to the JVM code that starts the application server process. You can specify arguments when HProf profiler support is enabled.

HProf arguments are only required if the Run HProf property is set to `true`.

Debug mode

Specifies whether to run the JVM in debug mode. The default is to not enable debug mode support.

If you set the **Debug mode** property to `true`, then you must specify command-line debug arguments as values for the **Debug arguments** property.

Data type	Boolean
Default	false

Debug arguments

Specifies command-line debug arguments to pass to the JVM code that starts the application server process. You can specify arguments when the **Debug mode** property is set to `true`.

If you enable debugging on multiple application servers on the same node, verify that the same value is not specified for the address argument. The address argument defines the port that is used for debugging. If two servers, for which debugging is enabled, are configured to use the same debug port, the servers might fail to start properly. For example, both servers might still be configured with the debug argument `address=7777`, which is the default value for the debug address argument.

Data type	String
Units	Java command-line arguments

Generic JVM arguments

Specifies command-line arguments to pass to the Java virtual machine code that starts the application server process.

You can enter the following optional command-line arguments in the **Generic JVM arguments** field. If you enter more than one argument, enter a space between each argument.

Note: If the argument states that it is only for the IBM Developer Kit only, you cannot use that argument with the JVM from another provider, such as the Microsoft or Hewlett-Packard

- **hotRestartSync:**

  Specify `hotRestartSync` if you want to enable the hot restart sync feature of the synchronization service. This feature indicates to the synchronization service that the installation is running in an environment where configuration updates are not made when the deployment manager is not active. Therefore, the service does not have to perform a complete repository comparison when the deployment manager or node agent servers restart. Enabling this feature improves the efficiency of the first synchronization operation after the deployment manager or a node agent restarts, especially for installations that include mixed release cells, use several nodes, and run several applications.
- **-Xquickstart**

  Specify`-Xquickstart` if you want the initial compilation to occur at a lower optimization level than in default mode. Later, depending on sampling results, you can recompile to the level of the initial compile in default mode.

  Note: Use`-Xquickstart` for applications where early moderate speed is more important than long run throughput. In some debug scenarios, test harnesses and short-running tools, you can improve startup time between 15-20 percent.
- **-Xverify:none**

  Specify`-Xverify:none` if you want to skip the class verification stage during class loading . Using **-Xverify:none** disables Java class verification, which can provide a 10-15 percent improvement in startup time. However corrupted or invalid class data is not detected when this argument is specified. If corrupt class data is loaded, the JVM might behave in an unexpected manner, or the JVM might fail.

Note:
  - Do not use this argument if you are making bytecode modifications, because the JVM might fail if any instrumentation error occurs.
  - If you experience a JVM failure or the JVM behaves in an unexpected manner while this argument is in affect, remove this argument as your first step in debugging your JVM problem.

- **-Xnoclassgc**

  Specify-`Xnoclassgc` if you want to disable class garbage collection. This argument results in more class reuse and slightly improved performance. However, the resources owned by these classes remain in use even when the classes are not being called. You can use the `verbose:gc` configuration setting if you want to monitor garbage collection. You can use the resulting output to determine the performance impact of reclaiming these resources. If the same set of classes are garbage collected repeatedly, you might want to disable class garbage collection. Class garbage collection is enabled by default.
- **-Xgcthreads**

  Specify `-Xgcthreads` if you want to use several garbage collection threads at one time. This garbage collection techniques is known as *parallel garbage collection*. This argument is valid only for the IBM Developer Kit.

  When entering this value in the **Generic JVM arguments** field, also enter the number of processors that are running on your machine. For example, if you have 3 processors running on your machine, enter `-Xgcthreads 3`. On a node with *n* processors, the default number of threads is *n*.

  Note: You should use parallel garbage collection if your machine has more than one processor.
- **-Xnocompactgc**

  Specify `-Xnocompactgc` if you want to disable heap compaction. Heap compaction is the most expensive garbage collection operation. If you are using the IBM Developer Kit, you should avoid heap compaction. If you disable heap compaction, you eliminate all associated overhead.
- **-Xgpolicy**

  Specify-`Xgpolicy` to set the garbage collection policy. This argument is valid only for the IBM Developer Kit.

  Set this argument to `optavgpause`, if you want concurrent marking used to track application threads starting from the stack before the heap becomes full. When this parameter is specified, the garbage collector pauses become uniform and long pauses are not apparent. However, using this policy reduces throughput because threads might have to do extra work.

  Set this argument to `optthruput` if you want to optimize throughput and it does not create a problem if long garbage collection pauses occur. This is the default parameter, recommended setting.
- **-XX**

  The Java Platform, Standard Edition 6 (Java SE 6) has generation garbage collection, which allows separate memory pools to contain objects with different ages. The garbage collection cycle collects the objects independently from one another depending on age. With additional parameters, you can set the size of the memory pools individually. To achieve better performance, set the size of the pool containing objects that have short life cycles, such that the objects in the pool are not kept through more then one garbage collection cycle. Use the `NewSize` and `MaxNewSize` parameters to specify the size of the new generation pool.

  Objects that survive the first garbage collection cycle are transferred to another pool. Use the`SurvivorRatio` parameter to specify the size of the survivor pool.`SurvivorRatio`. You can use the object statistics that the Tivoli Performance Viewer collects, or include the verbose:gc argument in your configuration setting to monitor garbage collection statistics. If garbage collection becomes a bottleneck, specify the following arguments to customize the generation pool settings to better fit your environment.

  ```
  -XX:NewSize=lower_bound
  -XX:MaxNewSize=upper_bound
   -XX:SurvivorRatio=new_ratio_size
  ```

  Note: The default values for these areguments are:`NewSize=2m MaxNewSize=32m SurvivorRatio=2`. However, if you have a JVM that is configured with a heap size that is greater than 1 GB, use

the values: -XX:newSize=640m -XX:MaxNewSize=640m -XX:SurvivorRatio=16, or set 50 to 60 percent of total heap size to a new generation pool.

- **-Xminf**

  Specify-Xminf if you want to change the minimum free heap size percentage. The heap grows if the free space is below the specified amount. In reset enabled mode, this argument specifies the minimum percentage of free space for the middleware and transient heaps. The valued specified for this argument is a floating point number, 0 through 1. The default is .3 (30 percent).

- **-server | -client**

  Java HotSpot Technology in Java SE 6 uses an adaptive JVM containing algorithms that, over time, optimize how the byte code performs. The JVM runs in two modes, **-server** and **-client**. In most cases, use **-server** mode, which produces more efficient run-time performance over extended lengths of time.

  If you use the default **-client** mode, the server startup time is quicker and a smaller memory footprint is created. However, this mode lowers extended performance. Use the **-server** mode, which improves performance, unless server startup time is of higher importance than performance. You can monitor the process size, and the server startup time to check the performance difference between using the **-client** and **-server** modes.

- **-Xshareclasses:none**

  Specify the-Xshareclasses:none argument to disable the share classes option for a process. The share classes option, which is available with Java SE 6, lets you share classes in a cache. Sharing classes in a cache can improve startup time and reduce memory footprint. Processes, such as application servers, node agents, and deployment managers, can use the share classes option.

  If you use this option, you should clear the cache when the process is not in use. To clear the cache, either call the app_server_root/bin/clearClassCache.bat/sh utility or stop the process and then restart the process.

  Note:

  – Solaris HP-UX The IBM JVM for J2SE 5 is not supported on Solaris, HP, and i5/OS.
  – J2EE application classes running in an application server process are not added to the shared class cache.

Data type	String
Units	Java command-line arguments

Executable JAR file name

Specifies a full path name for an executable JAR file that the JVM code uses.

Data type	String
Units	Path name

Disable JIT

Specifies whether to disable the just-in-time (JIT) compiler option of the JVM code.

If you disable the JIT compiler, throughput decreases noticeably. Therefore, for performance reasons, keep JIT enabled.

Data type	Boolean
Default	false (JIT enabled)
Recommended	JIT enabled

Operating system name

Specifies JVM settings for a given operating system.

When the process starts, the process uses the JVM settings that are specified for the node as the JVM settings for the operating system.

Configuring JVM sendRedirect calls to use context root

If the com.ibm.websphere.sendredirect.compatibility property is not set and your application servlet code has statements such as *sendRedirect("/home.html")*, your Web browser might display messages such as *Error 404: No target servlet configured for uri: /home.html*.

About this task

Note: The com.ibm.websphere.sendredirect.compatibility property is deprecated. You should modify your applications to redirect non-relative URLs (those starting with a "/") relative to the servlet container (*web_server_root*) instead of relative to the Web application context root.

To instruct the server to use the context root for that the application uses for sendRedirect() calls instead of using the document root for the Web server, configure the Java Virtual Machine (JVM) by setting the com.ibm.websphere.sendredirect.compatibility property to a `true` or `false` value.

1. Access the settings page for a property of the JVM.
   a. In the administrative console, click **Servers > Server Types > Application servers**.
   b. On the Application server page, click on the name of the server whose JVM settings you want to configure.
   c. On the settings page for the selected application server, in the Server Infrastructure section, click **Java and process management > Process definition**.
   d. On the Process definition page, click **Java virtual machine**.
   e. On the Java virtual machine page, click**Custom Properties**.
   f. On the Custom properties page, click **New**.
2. On the settings page for a property, specify `com.ibm.websphere.sendredirect.compatibility` in the **Name** field, and either `true` or `false` in the **Value** field. Then click **OK**.
3. Click **Save** on the console task bar.
4. Stop the application server, and then restart the application server.

Java virtual machine custom properties

You can use the administrative console to change the values of Java virtual machine (JVM) custom properties.

To set custom properties, connect to the administrative console and navigate to the appropriate Java virtual machine custom properties page.

Application server	**Servers > Server Types > WebSphere application servers >** *server_name*, and then, under Server Infrastructure, click **Java and process management > Process definition > Java virtual machine > Custom properties**
Deployment manager	**System Administration > Deployment manager > Java and process management > Process definition > Java virtual machine > Custom properties**
Node agent	**System Administration >Node agent >** *nodeagent_name* **> Java and process management > Process definition > Java virtual machine > Custom properties**

If the custom property is not present in the list of already defined custom properties, create a new property, and enter the property name in the Name field and a valid value in the Value field. Restart the server to complete your changes.

Note: Any custom property that begins with the string `was` is considered a system property. You can create a JVM custom property that starts with the string `was`, but you cannot use the administrative console to change the setting of such a custom property because any custom property that starts with the string `was` is not included in the list of available JVM custom properties that displays in the administrative console.

com.ibm.websphere.ejbcontainer.expandCMPCFJNDIName

The EJB container should allow for the expansion of the CMP Connection Factor JNDI Name when a user's JNDI name contains a user defined Application Server variable. The custom property, com.ibm.websphere.ejbcontainer.expandCMPCFJNDIName, makes it possible to expand the CMP Connection Fatory JNDI Name.

If the value is **true**, which is the default, the EJB Container expands a variable when found in the CMP Connection Factory JNDI Name. If the value is set to **false**, the EJB Container does not expand a variable.

com.ibm.websphere.sib.webservices.useTypeSoapArray

You can pass messages directly to a bus destination by overriding the JAX-RPC client binding namespace and endpoint address. However:

- The default RPC-encoded Web services string array message that is generated might not interoperate successfully with some target service providers.
- The string array message produced is not exactly the same as the standard JAX-RPC equivalent, which can interoperate successfully.

Here are examples of the two different messages:

- Service integration bus message:

```
<partname env:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/ xsi:type='ns1:ArrayOf_xsd_string'>
  <item xsi:type='xsd:anySimpleType'>namevalue</item>
</partname>
```

- JAX-RPC client message:

```
<partname xsi:type="soapenc:Array" soapenc:arrayType="xsd:string[1]">
  <item>namevalue</item>
</partname>
```

Set this property to `true` to modify the default behavior and send a string array message that is fully compatible with standard JAX-RPC. Setting this property modifies the default behavior for all outbound JMS Web services invocations sent from the service integration bus.

com.ibm.ws.sib.webservices.useSOAPJMSTextMessages

By default on WebSphere Application Server Version 6 or later, a SOAP over JMS Web service message sent by the Web services gateway is sent as a JmsBytesMessage, whereas on WebSphere Application Server Version 5.1 the Web services gateway sends a JmsTextMessage.

Set this property to `true` to modify the default behavior and send a compatible JmsTextMessage. Setting this property modifies the default behavior for all outbound JMS Web services invocations sent from the service integration bus.

com.ibm.websphere.ejbcontainer.expandCMPCFJNDIName

Use this Enterprise JavaBeans (EJB) custom property to expand the variables used in a container-managed persistence (CMP) connection factory Java™ Naming and Directory Interface (JNDI) name.

The EJB Container should allow for the expansion of the CMP connection factory JNDI name when a JNDI name contains a user-defined Application Server variable, although V6.1 does not support the expansion of variables. You need to use this property in order to expand the variables. You can enable or disable expansion.

To enable the expansion, the property value is `true`. To disable, use the value `false`.

The default is `true`.

If the value is `true`, the EJB container expands a variable found in the CMP connection factory JNDI name. If the value is `false`, the EJB container does not expand a variable.

com.ibm.websphere.application.updateapponcluster.waitforappsave

Specifies, in seconds, the amount of time that you want the deployment manager to wait for the extension tasks of the save operation to complete before starting the updated application.

Note: This property is only valid if it is specified for a deployment manager.

Usually during the save operation for an application update that is being performed using the rollout update process, the extension tasks of the save operation run as a background operation in a separate thread. If the main thread of the save operation completes before the synchronization portion of the rollout update process, the updated application fails to start properly.

When you add this custom property to your deployment manager settings, if the extension tasks of the save operation do not complete within the specified amount of time, the rollout update process stops the application update process, thereby preventing the application from becoming corrupted during the synchronization portion of the rollout update process.

The default value is 180.

com.ibm.ejs.sm.server.quiesceTimeout

Specifies, in seconds, the overall length of the quiesce timeout. If a request is still outstanding after this number of seconds, the server might start to shut down. For example, a value of 180 would be 3 minutes.

The default value is 180.

com.ibm.ejs.sm.server.quiesceInactiveRequestTime

Specifies, in milliseconds, how fast requests can come in and still be processed. For example, if you specify a value of 5000 for this property, the server does not attempt to shutdown until incoming requests are spaced at least 5 seconds apart. If the value specified for this property is too large, when the application server is stopped from the administrative console the following error message might be issued:

```
An error occurred while stopping Server1. Check the error logs for more information.
```

The default value is 5000 (5 seconds).

com.ibm.websphere.deletejspclasses

Use this property to indicate that you want to delete JavaServer Pages classes for all applications after those applications have been deleted or updated. The default value for this property is `false`.

com.ibm.websphere.deletejspclasses.delete

Use this property to indicate that you want to delete JavaServer Pages classes for all applications after those applications have been deleted, but not after they have been updated. The default value for this property is `false`.

com.ibm.websphere.deletejspclasses.update
Use this property to indicate that you want to delete JavaServer Pages classes for all applications after those applications have been updated, but not after they have been deleted. The default value for this property is `false`.

com.ibm.websphere.management.application.fullupdate
Use this property to specify that when any of your applications are updated, you want the binaries directory erased and the content of the updated EAR file completely extracted.

If this property is not specified, each changed file within an updated EAR file is individually updated and synchronized in the node. This process can be time consuming for large applications if a large number of files change.

Setting the com.ibm.websphere.management.application.fullupdate property to:

- `true` specifies that, when any of your applications are updated, you want the binaries directory erased and the content of the updated EAR file completely extracted.
- `false` specifies that, when any of your applications are updated, you only want the changed files within that EAR file updated on the node and then synchronized.

Note: Use the com.ibm.websphere.management.application.fullupdate.*application_name* property if you only want to do a full replacement for a specific application instead of all of your applications.

com.ibm.websphere.management.application.fullupdate.*application_name*
Use this property to specify that when the specified application is updated, you want the binaries directory for that application erased and the content of the updated EAR file completely extracted.

If this property is not specified, each changed file within the updated EAR file for the specified application is individually updated and synchronized in the node. This process can be time consuming for large applications if a large number of files change.

Setting the com.ibm.websphere.management.application.fullupdate.*application_name* property to:

- `true` specifies that when the specified application is updated, you want the binaries directory erased and the content of the updated EAR file completely extracted.
- `false` that when the specified application is updated, you only want the changed files updated on the node and then synchronized.

Note: Use the com.ibm.websphere.management.application.fullupdate property if you want the binaries directory erased and the content of the updated EAR file completely extracted whenever any of your applications are updated.

com.ibm.websphere.management.application.sync.recycleappasv5
Use this property to specify that you want your application recycling behavior to work the same way as this behavior worked in Version 5.x of the product.

In Version 6.x and higher, after an application update or edit operation occurs, depending on which files are modified, either the application or its modules are automatically recycled. This recycling process occurs for all application configuration file changes, and all non-static file changes.

However, in Version 5.x of the product, an application is recycled only if the Enterprise Archive (EAR) file itself is updated, or if the binaries URL attribute changes. An application is not recycled if there is a change to the application configuration file.

Setting the com.ibm.websphere.management.application.sync.recycleappasv5 property to:

- `true` specifies that you want your application recycling behavior to work the same way as this behavior worked in Version 5.x of the product.

- `false` specifies that you want your application recycling behavior to work according to the Version 6.x and higher behavior schema.

The default value for this custom property is`false`.

Note: You must define this property in the node agent JVM. However, when defining this property, you can specify a scope of cell if you want the setting to apply to all of the nodes within a specific cell. If this property is set at both the cell and node agent level, the node agent setting takes precedence for that particular node agent.

com.ibm.websphere.network.useMultiHome

Use this property in a multihomed environment to indicate on which IP addresses the application server listens. In a multihomed environment, there is normally a specific IP address that the application server is restricted to listening on for Discovery and SOAP messages. Setting the com.ibm.websphere.network.useMultiHome property to:

- `true` specifies that the product listens on all IP addresses on the host for Discovery and SOAP messages.
- `false` specifies that the product only listens on the configured host name for Discovery and SOAP messages. If you set this property to `false`, you should have a host name configured on the product that resolves to a specific IP address.
- `null` specifies that the product only listens on the default IP address only.

If you cannot contact the server, check the setting for com.ibm.websphere.network.useMultihome to ensure it is correct. You can change the value through the administrative console. Modify the defaults by setting the value for the server, deployment manager, and node agent. You must restart the server before these changes take effect.

com.ibm.websphere.webservices.attachements.maxMemCacheSize

Use this property to specify, in kilobytes, the maximum size of a Web services attachment that can be written to memory. For example, if your Web service needs to send 20 MB attachments, set the property to 20480.

When determining a value for this property, remember that the larger the maximum cache size, the more impact there is on performance, and, potentially, to the Java heap.

f you do not specify a value for this property, the maximum memory that is used to cache attachments is 32 KB, which is the default value for this property.

com.ibm.ws.pm.checkingDBconnection

Use this property to specify whether the persistence manager is to continue checking the availability of a database, that was previously marked as unavailable, until a connection with that database is successfully established.

If a database service is down when the persistent manager attempts to establish a connection to that database, the database is marked as unavailable. Typically, the persistent manager does not re-attempt to establish a connection after a database is marked as unavailable. If you sent this property to `true`, the persistence manager continues to check the availability of the database until it is able to successfully establish a connection to that database.

The default value for this property is `false`.

com.ibm.ws.webservices.contentTransferEncoding

Use this property to specify a range of bits for which .XML-encoding is disabled. Typically any integer that is greater than 127 is XML-encoded. When you specify this property:

- Web services disables encoding for integers that fall within the specified range.

- The HTTP transport message contains a ContentTransferEncoding header that is set to the value that is specified for this custom property.

Specify `7bit`, if you only want integers greater than 127 encoded. Specify `8bit`, if you only want integers greater than 255 encoded. Specify `binary`, if you want encoding disabled for all integers.

The default value is `7bit`.

com.ibm.ws.webservices.ignoreUnknownElements

Use this property to control whether clients can ignore extra XML elements that are sometimes found within literal SOAP operation responses.

Setting this property to `true` provides you with the flexibility of being able to update your server code to include additional response information, without having to immediately update your client code to process this additional information. However, when this functionality is enabled, the checking of SOAP message against the expected message structure is more relaxed than when this property is set to `false`.

com.ibm.ws.webservices.suppressHTTPRequestPortSuffix

Use this property to control whether a port number can be left in an HTTP POST request that sends a SOAP message.

Some Web service implementations do not properly tolerate the presence of a port number within the HTTP POST request that sends the SOAP message. If you have a Web service client that needs to inter-operate with Web service that cannot tolerate a port number within an HTTP POST request that sends a SOAP message, set this custom property to `true`.

When you set this property to `true`, the port number is removed from the HTTP POST request before it is sent.

Note: You must restart the server before this configuration setting takes affect.

The default value for this custom property is `false`.

com.ibm.websphere.ejb.UseEJB61FEPScanPolicy

Use this property to control whether the product scans pre-Java EE 5 modules for additional metadata during the application installation process or during server startup. By default, these legacy EJB modules are not scanned.

The default value for this custom property is `false`.

You must set this property to `true` for each server and administrative server that requires a change in the default value.

com.ibm.websphere.webservices.UseWSFEP61ScanPolicy

Use this property to control whether the product scans WAR 2.4 and earlier modules for JAXWS components and semi-managed service clients. By default, these legacy WAR modules are only scans for semi-managed service clients.

The default value for this custom property is `false`.

You must set this property to `true` for each server and administrative server that requires a change in the default value.

com.ibm.ws.ws.wsba.protocolmessages.twoway

Use this property to improve the performance of an application server that is handling requests for Web Services Business Activities (WS-BA). Specifying true for this custom property improves application server performance when WS-BA protocol messages are sent between two application servers. The default value for this property is true.

Note: If you decide to use this custom property, the property must be set on the application server that initiates the requests. It does not have to be set on the application server that receives the requests.

java.net.preferIPv4Stack

Use this property to disable IPv6 support. On operating systems where IPv6 support is available, the underlying native socket that the product uses is an IPv6 socket. On IPv6 network stacks that support IPv4-mapped addresses, you can use IPv6 sockets to connect to and accept connections from both IPv4 and IPv6 hosts.

Setting this property to `true` disables the dual mode support in the JVM which might, in turn, disrupt normal product functions. Therefore, it is important to understand the full implications before using this property. In general, setting this property is not recommended.

The default value for this custom property is false, except on the Microsoft Windows operating systems, where the default is true.

java.net.preferIPv6Addresses

Use this property to disable IPv4 support. Setting this property to `true` disables the dual mode support in the JVM which might, in turn, disrupt normal product functions. Therefore, it is important to understand the full implications before using this property. In general, setting this property is not recommended.

The default value for this custom property is false, except on the Windows operating system where the default is true.

ODCClearMessageAge

Use this property to establish a length of time, specified in milliseconds, after which an ODC message is removed from the bulletin board, even if the receiver has not acknowledged the message. Specifying a value for this property helps prevent the build up of messages that, for some reason, do not get acknowledged.

You can specify any positive integer as a value for this property, but a value of 300000 (5 minutes) or higher is recommended to avoid premature removal of messages.

The default value is 300000 milliseconds.

Preparing to host applications

Rather than use the default application server provided with the product, you can configure a new server and set of resources.

About this task

The default application server and a set of default resources are available to help you begin quickly. You can choose instead to configure a new server and set of resources. Here is what you need to do in order to set up a runtime environment to support applications.

1. Configure an application server.
2. Create a virtual host.

3. Configure a Web container. See the *Administering applications and their environment* PDF for more information.
4. Configure an EJB container. See the *Administering applications and their environment* PDF for more information.
5. Create resources for data access. See the *Administering applications and their environment* PDF for more information.
6. Create a JDBC provider and data source. See the *Administering applications and their environment* PDF for more information.
7. Create a URL and URL provider. See the *Administering applications and their environment* PDF for more information.
8. Create a mail session. See the *Administering applications and their environment* PDF for more information.
9. Create resources for session support. See the *Administering applications and their environment* PDF for more information.
10. Configure a Session Manager. See the *Administering applications and their environment* PDF for more information.

Configuring multiple network interface support

Application servers, by default, are configured to use all of the network interfaces that are available for them to use. You can change this configuration such that an application server only uses a specific network interface. However, you cannot configure it to use a subgroup of interfaces. For example, if you have three ethernet adapters, you cannot configure an application server to use two of the three adapters.

About this task

When an application server is configured to use all network interfaces, if it opens a socket on port 9901 on a machine with two TCP/IP addresses, it opens port 9901 on both IP addresses.

Windows On a Microsoft Windows operating system, the netstat output displays *.9901 in the Local Address field, indicating that port 9901 is bound to all network interfaces in the system.

When an application server is configured to use a specific network interface, it only communicates on that one network interface. For example, on a Windows operating system, if an application server opens a socket on port 7842 on an ethernet adapter with an address of 192.168.1.150, the netstat output displays 192.168.1.150.7842 in the Local Address field, indicating that port 7842 is only bound to 192.168.1.150.

If you have more than one network interface and you want to use each one separately, you must have a separate configuration profile for each interface. When network interfaces are used separately, a separate node agent is required for each network interface that has an application server running on it. Two application servers bound to two separate network interfaces on the same machine cannot be in the same node because they have different TCP/IP addresses.

Note:

- If you want a specific application server to use a single network interface, perform the following steps for that application server.
- If you want an entire node to use a single network interface, perform the following steps for your node agent and all the application servers in that node.
- If you want an entire cell to use a single network interface, perform the following steps for the deployment manager, node agent, and all the application servers in the node.
- When performing the following steps, do not specify localhost, a loop back address, such as 127.0.0.1, or an * (asterisk) for the TCP/IP addresses.

1. Update the com.ibm.CORBA.LocalHost and com.ibm.ws.orb.transport.useMultiHome Object Request Broker (ORB) custom properties.
   a. In the administrative console, navigate to the indicated page.
      - For an application server, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Container Settings > Container services > ORB Service**. Then in the Additional Properties section, click **Custom properties**.
      - For a deployment manager, click **System Administration > Deployment manager**. In the Additional Properties section, click **ORB Service**. Then, under Additional properties on the **ORB Service** page, click **Custom properties**.
      - For a node agent, click **System Administration > Node agents** *node_agent*. In the Additional Properties section, click **ORB Service**. Then, under Additional properties on the **ORB Service** page, click **Custom properties**.
   b. Select the com.ibm.CORBA.LocalHost custom property and specify an IP address or hostname in the Value field. Do not set this property to either localhost or *.

      If the com.ibm.CORBA.LocalHost property is not in the list of already defined custom properties, click **New** and then enter `com.ibm.CORBA.LocalHost` in the Name field and specify an IP address or hostname in the Value field.
   c. Select the com.ibm.ws.orb.transport.useMultiHome custom property and specify `false` in the Value field. If the com.ibm.ws.orb.transport.useMultiHome property is not in the list of already defined custom properties, click **New** and then enter `com.ibm.ws.orb.transport.useMultiHome` in the Name field and specify `false` in the Value field.
2. Update the Java virtual machine (JVM) com.ibm.websphere.network.useMultiHome custom property for discovery and SOAP connections.
   a. In the administrative console, navigate to the indicated page.
      - For an application server, click **Servers > Server Types > WebSphere application servers >** *server_name* **> Java process management > Process definition > Java virtual machine > Custom properties**.
      - For a deployment manager, click **System Administration > Deployment manager > Java process management > Process definition > Java virtual machine > Custom properties**.
      - For a node agent, click **System Administration > Node agent >** *node_ agent* **> Java process management > Process definition > Java virtual machine > Custom properties**.
   b. Select the com.ibm.websphere.network.useMultiHome custom property and specify `false` in the Value field. If the com.ibm.websphere.network.useMultiHome property is not in the list of already defined custom properties, click **New** and then enter `com.ibm.websphere.network.useMultiHome` in the Name field and specify `false` in the Value field.
3. Update the host name for TCP/IP connections.
   a. In the administrative console, navigate to the indicated page.
      - For an application server, click **Servers > Server Types > WebSphere application servers >** *server_name*, and then, in the Additional Properties section, click **Ports**.
      - For a deployment manager, click **System Administration > Deployment manager**, and then, in the Additional Properties section, click **Ports**.
      - For a node agent, click **System Administration > Node agents >** *node_ agent*, and then, in the Additional Properties section, click **Ports**.
   b. Update the Host field for each of the listed ports to the value specified for the com.ibm.CORBA.LocalHost ORB custom property in the first step. When you finish, none of the entries listed in the Host column should contain an * (asterisk).
4. Change the Initial State setting for each of the JMS servers to `Stopped` .
   a. In the administrative console, click **Servers > Server Types > JMS servers >**
   b. Click one of the listed JMS servers and change the value specified for the Initial State field to Stopped.

c. Repeat the previous step until the Initial State setting for all of the listed JMS servers is Stopped.

5. Change the Initial State setting for each of the listener ports to `Stopped` .
   a. In the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name*.
   b. Under Communications, click **Messaging > Message Listener Service > Listener Ports**.
   c. Click one of the listed listener ports and change the value specified for the Initial State field to `Stopped`.
   d. Repeat the previous step until the Initial State setting for all of the listed listener ports is `Stopped`.
6. Save your changes.
   a. In the administrative console, click **System administration > Save Changes to Master Repository**.
   b. Select Synchronize changes with nodes, and then click **Save**.
7. Stop and restart all the affected servers, node agents, and the deployment manager.

Results

You have configured an installation of WebSphere Application Server to communicate on one, and only one network interface on a machine that has more than one network interface.

Example

This example creates two nodes, each using a separate network interface, on a machine that has at least two network interfaces:

1. Use the Profile Management tool to create an application server and federate it into the desired cell.
2. Use the Profile Management tool to create an application server profile, specifying a host name that is different than the host name used for the previously created application server. Federate this application server into the desired cell.
3. Start the node agent and application server that are configured to the first network interface. Follow the preceding steps for the node agent and application server to prepare this node to communicate on the network interface you specified when you configured this application server.
4. Start the second node agent and application server. Follow the preceding steps for the node agent and application server to prepare this node to communicate only on the network interface that you specified when you configured the second application server.
5. Stop all of the node agents and application servers that you created in this example.
6. Restart all of these node agents and application servers.

You have two separate nodes running on two different network interfaces.

What to do next

If you are using a standalone Java client or server to communicate with WebSphere Application Server, and you are using the WebSphere Application Server Software Development Kit (SDK), add the following properties to your Java command to enable the ORB for your application to communicate with a specific network interface.

```
-Dcom.ibm.ws.orb.transport.useMultiHome=false
-Dcom.ibm.CORBA.LocalHost=host_name
```

host_name is the TCP/IP address or *hostname* of the network interface for the ORB to use.

Note: Do not set *host_name* to localhost, a loop back address, such as 127.0.0.1, or an * (asterisk).

Configuring application servers for UCS Transformation Format

You can use the client.encoding.override=UTF-8 JVM argument to configure an application server for UCS Transformation Format. This format enables an application server to handle most character encodings, including specialized mathematical and technical symbols.

About this task

The client.encoding.override=UTF-8 argument is provided for backwards compatibility. You should only specify this argument if you require multiple language encoding support in the administrative console and there is no other way for you to set the request character encoding required to parse post and query strings.

Before configuring an application server for UCS Transformation Format, you should try to either:

- Explicitly set the ServletRequest Encoding inside of the JSP or Servlet that is receiving the POST and or query string data, which is the preferred J2EE solution, or
- Enable the autoRequestEncoding, option, which uses the client's browser settings to determine the appropriate character encoding. Older browsers might not support this option.

Note: If the client.encoding.override=UTF-8 JVM argument is specified, the autoRequestEncoding option does not work even if it is enabled. Therefore, when an application server receives a client request, it checks to see if the charset option is set on the content type header of the request:

1. If it is set, the application server uses the content type header for character encoding.
2. If it is not set, the application server uses the character encoding that is specified for the default.client.encoding system property.
3. If neither charset nor the default.client.encoding system property is set, the application server uses the ISO-8859-1 character set.

The application server never checks for an Accept-Language header. However, if the autoRequestEncoding option is working, the application server checks for an Accept-Language header before checking to see if a character encoding is specified for the default.client.encoding system property.

To configure an application server for UCS Transformation Format:

1. In the administrative console, click **Servers > Server Types > WebSphere application servers**, and select the server that you want to enable for UCS Transformation Format.
2. Then, in the Server Infrastructure section, click **Java and process management > Process definition > Java virtual machine**.
3. Specify `-Dclient.encoding.override=UTF-8` for the **Generic JVM Arguments** property, and click **OK**. When this argument is specified, UCS Transformation Format is used instead of the character encoding that would be used if the autoRequestEncoding option was in effect.
4. Click **Save** to save your changes.
5. Restart the application server.

Results

The application server uses UCS Transformation Format for encoding.

Tuning application servers

The product contains interrelated components that must be harmoniously tuned to support the custom needs of your end-to-end e-business application.

About this task

This group of interrelated components is known as the queuing network. The queuing network helps the system achieve maximum throughput while maintaining the overall stability of the system.

The following steps describe various tuning tasks that may improve your application server performance. You can choose to implement any of these application server settings. These steps can be performed in any order.

1. **Tune the object request broker.** An Object Request Broker (ORB) manages the interaction between clients and servers, using the Internet InterORB Protocol (IIOP). It supports client requests and responses received from servers in a network-distributed environment. You can use the following parameters to tune the ORB:
   - Set **Pass by reference (com.ibm.CORBA.iiop.noLocalCopies)** as described in the *Tuning guide* PDF.
   - Set the **Connection cache minimum (com.ibm.CORBA.MaxOpenConnections)** as described in the *Tuning guide* PDF.
   - Set **Maximum size** as described in "Thread pool settings" on page 208
   - Set **com.ibm.CORBA.ServerSocketQueueDepth** as described in the *Administering applications and their environment* PDF.
   - Set the **com.ibm.CORBA.FragmentSize** as described in the *Administering applications and their environment* PDF.
2. **Tune the XML parser definitions.**
   - **Description:** Facilitates server startup by adding XML parser definitions to the jaxp.properties and xerxes.properties files in the ${*app_server_root*}/jre/lib directory. The XMLParserConfiguration value might change as new versions of Xerces are provided.
   - **How to view or set:** Insert the following lines in both files:

     ```
     javax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryImpl
     javax.xml.parsers.DocumentBuildFactory=org.apache.xerces.jaxp.
                DocumentBuilderFactoryImpl
     org.apache.xerces.xni.parser.XMLParserConfiguration=org.apache.xerces.parsers.
              StandardParserConfiguration
     ```
   - **Default value:** None
   - **Recommended value:** None
3. **Tune the dynamic cache service.**

   Using the dynamic cache service can improve performance. See the *Administering applications and their environment* PDF for information about using the dynamic cache service and how it can affect your application server performance.
4. **Tune the Web container.** The product Web container manages all HTTP requests to servlets, JavaServer Pages and Web services. Requests flow through a transport chain to the Web container. The transport chain defines the important tuning parameters for performance for the Web container. There is a transport chain for each TCP port that the product is listening on for HTTP requests. For example, the default HTTP port 9080 is defined in Web container inbound channel chain. Use the following parameters to tune the Web container:
   - HTTP requests are processed by a pool of server threads. The minimum and maximum thread pool size for the Web container can be configured for optimal performance. Generally, 5 to 10 threads per server CPU provides the best throughput. The number of threads configured does not represent the number of requests that the product can process concurrently. Requests are queued in the transport chain when all threads are busy. To specify the thread pool settings:
     a. Click **Servers > Server Types > WebSphere application servers >***server_name* **Web container settings > Web container > Web container transport chains**.
     b. Select the normal inbound chain for serving requests. This chain is typically called WCInboundDefault, and listens on port 9080.
     c. Click **TCP Inbound Channel (TCP_2)**.
     d. Set **Thread Pools** under Related Items.
     e. Select **WebContainer**.

f. Enter values for **Minimum Size** and **Maximum Size**.

- The HTTP 1.1 protocol provides a keep-alive feature to enable the TCP connection between HTTP clients and the server to remain open between requests. By default the product closes a given client connection after a number of requests or a timeout period. After a connection is closed, it is recreated if the client issues another request. Early closure of connections can reduce performance. Enter a value for the maximum number of persistent requests to (keep-alive) to specify the number of requests that are allowed on a single HTTP connection. Enter a value for persistent timeouts to specify the amount of time, in seconds, that the HTTP transport channel allows a socket to remain idle between requests. To specify values for Maximum persistent requests and Persistent timeout:
  a. Click **Servers > Server Types > WebSphere application servers >***server_name*. Then in the Container Settings section, click **Web container > Web container transport chains**.
  b. Select the normal inbound chain for serving requests. This chain is typically called WCInboundDefault, and listens on port 9080.
  c. Click **HTTP Inbound Channel (HTTP_2)**.
  d. Enter values for **Maximum persistent requests** and **Persistent timeout**.

5. **Tune the EJB container.** An Enterprise JavaBeans (EJB) container is automatically created when you create an application server. After the EJB container is deployed, you can use the following parameters to make adjustments that improve performance.
   - Set the **Cleanup interval** and the **Cache size** as described in the *Administering applications and their environment* PDF.
   - **Break CMP enterprise beans into several enterprise bean modules** while assembling EJB modules.

   See also the *Tuning guide* PDF.
6. **Tune the session management.**

   The installed default settings for session management are optimal for performance. See the *Tuning guide* PDF for more information about tuning session management.
7. **Tune the data sources and associated connection pools.** A data source is used to access data from the database; it is associated with a pool of connections to that database.
8. **Tune the URL invocation cache.**

   Each JavaServer Page is a unique URL. If you have more than 50 unique URLs that are actively being used, increase the value specified for the invocationCacheSize JVM custom property. This property controls the size of the URL invocation cache.

   Each JavaServer Page is a unique URL. If you have more than 50 unique URLs that are actively being used, increase the value specified for the invocationCacheSize JVM custom property. This property controls the size of the URL invocation cache. See the *Administering applications and their environment* PDF for more information on how to change this property.

Web services client to Web container optimized communication

To improve performance, there is an optimized communication path between a Web services client application and a Web container that are located in the same application server process. Requests from the Web services client that are normally sent to the Web container using a network connection are delivered directly to the Web container using an optimized local path. The local path is available because the Web services client application and the Web container are running in the same process.

This direct communication eliminates the need for clients and web containers that are in the same process to communicate over the network. For example, a Web services client might be running in an application server. Instead of accessing the network to communicate with the Web container, the Web services client can communicate with the Web container using the optimized local path. This optimized local path improves the performance of the application server by enabling Web services clients and Web containers to communicate without using network transports.

In a clustered environment, there is typically an HTTP server (such as IBM HTTP server) that handles incoming client requests, distributing them to the correct application server in the cluster. The HTTP server

uses information about the requested application and the defined virtual hosts to determine which application server receives the request. The Web services client also uses the defined virtual host information to determine whether the request can be served by the local Web container. You must define unique values for the host and port on each application server. You cannot define the values of host and port as wild cards denoted by the asterisk symbol (*) when you enable the optimized communication between the Web services application and the Web container. Using wild cards indicate that the local Web container can handle Web services requests for all destinations.

The optimized local communication path is disabled by default. You can enable the local communication path with the enableInProcessConnections custom property. Before configuring this custom property, make sure that you are not using wild cards for host names in your Web container end points. Set this property to **true** in the Web container to enabled the optimized local communication path. When disabled, the Web services client and the Web container communicate using network transports.

For information about how to configure the enableInProcessConnections custom property, see the *Administering applications and their environment* PDF.

When the optimized local communication path is enabled, logging of requests through the local path uses the same log attributes as the network channel chain for the Web container. To use a different log file for in process requests than the log file for network requests, use a custom property on the HTTP Inbound Channel in the transport chain. Use the localLogFilenamePrefix custom property to specify a string that is added to the beginning of the network log file name to create a file name that is unique. Requests through the local process path are logged to this specified file. For example, if the log filename is ../httpaccess.log for a network chain, and the localLogFilenamePrefix custom property is set to “local” on the HTTP channel in that transport chain, the local log file name for requests to the host associated with that chain is /localhttpaccess.log.

Note: If you specify a value for the localLogFilenamePrefix custom property, you must also set the accessLogFileName HTTP channel custom property to the fully qualified name of the log file you want to use for in process requests. You cannot specify a variable, such as $(SERVER_LOG_ROOT), as the value for this custom property.

Chapter 6. Balancing workloads with clusters

You should use server clusters and cluster members to monitor and manage the workloads of application servers.

Before you begin

You should understand your options for configuring application servers. To assist you in understanding how to configure and use clusters for workload management, consider this scenario. Client requests are distributed among the cluster members on a single machine. A *client* refers to any servlet, Java application, or other program or component that connects the end user and the application server that is being accessed.

In more complex workload management scenarios, you can distribute cluster members to remote machines.

About this task

Perform the following steps if you decide to use clusters to balance your workload.

1. Decide which application server you want to cluster.
2. Decide whether you want to replicate data. Replication is a service that transfers data, objects, or events among application servers.

   You can create a replication domain when creating a cluster.
3. Deploy the application onto the application server.
4. Create a cluster.

   After configuring the application server and the application components exactly as you want them to be, create a cluster. The original server instance becomes a cluster member that is administered through the cluster.
5. Create one or more cluster members.
6. Configure a backup cluster.

   A backup cluster handles requests if the primary cluster fails.
7. Start the cluster.

   When you start the cluster, all of the application servers that are members of that cluster start. Workload management automatically begins after the cluster members start.
8. After the cluster is running, you can perform the following tasks:
   - Stop the cluster.
   - Upgrade the applications that are installed on the cluster members.
   - Detect and handle problems with server clusters and their workloads.
   - Tune the behavior of the workload management run time.

     The default timeout value for the com.ibm.CORBA.RequestTimeout JVM property is `0`, which means wait forever. This default value is not a good setting to have for failover situations. Therefore, if your application is experiencing problems with timeouts, or if you have configured your system for failover situations, use the -CCD option on the LaunchClient command to set an appropriate non-zero value for this property.

     If the workload management state of the client refreshes too soon or too late, change the interval setting of the com.ibm.websphere.wlm.unusable.interval property.

What to do next

For stand-alone Java clients, you must define a bootstrap host. Stand-alone Java clients are clients that are located on a different machine from the application server and have no administrative server. Add the following line to the Java virtual machine (JVM) arguments for the client:

`-Dcom.ibm.CORBA.BootstrapHost=`*`machine_name`*

where *machine_name* is the name of the machine on which the administrative server is running.

Clusters and workload management

Clusters are sets of servers that are managed together and participate in workload management. Clusters enable enterprise applications to scale beyond the amount of throughput capable of being achieved with a single application server. Clusters also enable enterprise applications to be highly available because requests are automatically routed to the running servers in the event of a failure. The servers that are members of a cluster can be on different host machines. In contrast, servers that are part of the same node must be located on the same host machine. A cell can include no clusters, one cluster, or multiple clusters.

Servers that belong to a cluster are *members* of that cluster set and must all have identical application components deployed on them. Other than the applications configured to run on them, cluster members do not have to share any other configuration data. One cluster member might be running on a huge multi-processor enterprise server system, while another member of that same cluster might be running on a smaller system. The server configuration settings for each of these two cluster members are very different, except in the area of application components assigned to them. In that area of configuration, they are identical. This allows client work to be distributed across all the members of a cluster instead of all workload being handled by a single application server.

When you create a cluster, you make copies of an existing application server template. The template is most likely an application server that you have previously configured. You are offered the option of making that server a member of the cluster. However, it is recommended that you keep the server available only as a template, because the only way to remove a cluster member is to delete the application server. When you delete a cluster, you also delete any application servers that were members of that cluster. There is no way to preserve any member of a cluster. Keeping the original template intact allows you to reuse the template if you need to rebuild the configuration.

A *vertical cluster* has cluster members on the same node, or physical machine. A *horizontal cluster* has cluster members on multiple nodes across many machines in a cell. You can configure either type of cluster, or have a combination of vertical and horizontal clusters.

Clustering application servers that host Web containers automatically enables plug-in workload management for the application servers and the servlets they host. The routing of servlet requests occurs between the Web server plug-in and clustered application servers using HTTP transports, or HTTP transport channels.

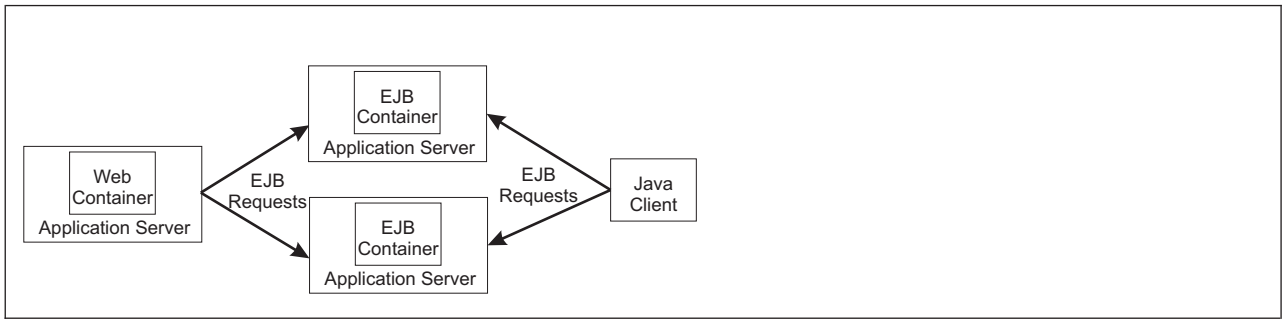
This routing is based on weights associated with the cluster members. If all cluster members have identical weights, the plug-in sends equal requests to all members of the cluster, assuming there are no strong affinity configurations. If the weights are scaled in the range from zero to twenty, the plug-in usually routes requests to those cluster members with the higher weight values.

You can use the administrative console to specify a weight for a cluster member. The weight you assign to a cluster member should be based on its approximate, proportional ability to do work. The weight value specified for a specific member is only meaningful in the context of the weights you specify for the other members within a cluster. The weight values do not indicate absolute capability. If a cluster member is unavailable, the Web server plug-in temporarily routes requests around that cluster member.

For example, if you have a cluster that consists of two members, assigning weights of 1 and 2 causes the first member to get approximately 1/3 of the workload and the second member to get approximately 2/3 of the workload. However, if you add a third member to the cluster, and assign the new member a weight of 1, approximately 1/4 of the workload now goes to the first member, approximately 1/2 of the workload goes to the second member, and approximately 1/4 of the workload goes to the third member. If the first cluster member becomes unavailable, the second member gets approximately 2/3 of the workload and third member gets approximately 1/3 of the workload.

The weight values only approximate your load balance objectives. There are other application dependencies, such as thread concurrency, local setting preferences, affinity, and resource availability that are also factors in determining where a specific request is sent. Therefore, do not use the exact pattern of requests to determine the weight assignment for specific cluster members.

Workload management for EJB containers can be performed by configuring the Web container and EJB containers on separate application servers. Multiple application servers can be clustered with the EJB containers, enabling the distribution of enterprise bean requests between EJB containers on different application servers.



In this configuration, EJB client requests are routed to available EJB containers in a round robin fashion based on assigned server weights. The EJB clients can be servlets operating within a Web container, stand-alone Java programs using RMI/IIOP, or other EJBs.

The server weighted round robin routing policy ensures a balanced routing distribution based on the set of server weights that have been assigned to the members of a cluster. For example, if all servers in the cluster have the same weight, the expected distribution for the cluster is that all servers receive the same number of requests. If the weights for the servers are not equal, the distribution mechanism sends more requests to the higher weight value servers than the lower weight value servers. The policy ensures the desired distribution, based on the weights assigned to the cluster members.

You can choose to have requests sent to the node on which the client resides as the preferred routing. In this case, only cluster members on that node are chosen (using the round robin weight method). Cluster members on remote nodes are chosen only if a local server is not available.

Multiple servers that can service the same client request form the basis for failover support. If a server fails while processing a client request, the failed request can be rerouted to any of the remaining cluster members. Even if several servers fail, as long as at least one cluster member is running, client requests continue to be serviced.

The backup cluster still functions even if all of the members of the primary cluster are not available.

Workload management for all platforms except z/OS

Workload management optimizes the distribution of client processing tasks. Incoming work requests are distributed to the application servers, enterprise beans, servlets, and other objects that can most effectively process the requests.

Workload management provides the following benefits to applications that are installed on the product:

- It balances client workloads, allowing processing tasks to be distributed according to the capacities of the different machines in the system.
- It provides failover capability by redirecting client requests if one or more servers is unable to process them. This improves the availability of applications and administrative services.
- It enables systems to be scaled up to serve a higher client load than provided by the basic configuration. With clustering, additional instances of servers, servlets, and other objects can easily be added to the configuration.
- It enables servers to be transparently maintained and upgraded while applications remain available for users.
- It centralizes the administration of servers and other objects.

In the product environment, you use clusters, transports, and replication domains to implement workload management.

Techniques for managing state

Multiple machine scaling techniques rely on using multiple copies of an application server; multiple consecutive requests from various clients can be serviced by different servers. If each client request is completely independent of every other client request, it does not matter if consecutive requests are processed on the same server. However, in practice, client requests are not independent. A client often makes a request, waits for the result, then makes one or more subsequent requests that depend on the results received from the earlier requests.

This sequence of operations on behalf of a client falls into two categories:

Stateless
: A server processes requests based solely on information provided with each request and does not rely on information from earlier requests. The server does not need to maintain state information between requests.

Stateful
: A server processes requests based on both the information provided with each request and information stored from earlier requests. The server needs to access and maintain state information generated during the processing of an earlier request.

For stateless interactions, it does not matter whether different requests are processed by different servers. However, for stateful interactions, the server that processes a request needs access to the state information necessary to service that request. Either the same server can process all requests that are associated with the same state information, or the state information can be shared by all servers that require it. In the latter case, accessing the shared state information from the same server minimizes the processing overhead associated with accessing the shared state information from multiple servers.

The load distribution facilities in the product use several different techniques for maintaining state information between client requests:

- Session affinity, where the load distribution facility recognizes the existence of a client session and attempts to direct all requests within that session to the same server.
- Transaction affinity, where the load distribution facility recognizes the existence of a transaction and attempts to direct all requests within the scope of that transaction to the same server.
- Server affinity, where the load distribution facility recognizes that although multiple servers might be acceptable for a given client request, a particular server is best suited for processing that request.
- The session manager, which is part of each application server, stores client session information and takes session affinity and server affinity into account when directing client requests to the cluster members of an application server. The workload management service considers server affinity and transaction affinity when directing client requests among the cluster members of an application server.

Creating clusters

A cluster is a set of application servers that you manage together as a way to balance workload.

Before you begin

Before you create a cluster:

- Review the content of the topic ″Clusters and workload management,″ especially the information about setting cluster weights.
- Decide if you want enterprise bean requests routed to the node on which the client resides.
- Decide if you want to use HTTP memory-to-memory replication.
- Determine the appropriate configuration settings for the first cluster member. 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.
- Decide on which node you want the first cluster member to reside.

About this task

You might want to create a cluster if you need to:

- Balance your client requests across multiple application servers.
- Provide a highly available environment for your applications.

A cluster enables you to manage a group of application servers as a single unit, and distribute client requests among the application servers that are members of the cluster.

To create a cluster:

1. In the administrative console, click **Servers > Clusters > WebSphere application server clusters > New**. The Create a new cluster wizard starts.
2. Specify a name for the cluster.
3. Select **Prefer local** if you want to enable host-scoped routing optimization. This option is enabled by default. When this option is enabled, if possible, EJB requests are routed to the client host. This option improves performance because client requests are sent to local enterprise beans.
4. Select **Configure HTTP session memory-to-memory replication** if you want a memory-to-memory replication domain created for this 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.

   To change these settings for the replication domain, click **Environment > Replication domains >** *replication_domain_name*. To modify the Web container settings, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Clusters members >** *cluster_member_name*. Then, in the Container settings section, click **Web container settings > Web**

container >Session management > Distributed environment settings in the administrative console. If you change these settings for one cluster member, you might also need to change them for the other members of this cluster.

5. Click **Next**.
6. Choose whether to create an empty cluster or to create the first member of the cluster.

   If you decide to create an empty cluster, to add members to this cluster, in the administrative console, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Clusters members > New**.

   To create an empty cluster:

   a. Select **None. Create an empty cluster.**
   b. Click **Next** to display a summary of the defined cluster.
   c. Click **Finish** to create the cluster, or click **Cancel** if you decide not to create this cluster.

   When you create the first cluster member, remember that 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.

   a. Specify the name of the first cluster member.
   b. Select the node on which you want this cluster member to reside.
   c. Specify the weight value for the 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`.
   d. Select **Generate unique HTTP ports** if you want to generate unique port numbers for every HTTP transport that is defined in the source server. When this option is selected, which is the default setting, this cluster member does not have HTTP transports or HTTP transport channels that conflict with any of the other servers that are defined on the same node. If you unselect this option, all of the cluster members will share the same HTTP ports.
   e. Select the core group to which you want this cluster member to belong. You are prompted for the core group only if you have more than one core group defined for this cluster.
   f. Select one of the following options as the basis for the first cluster member.
      - Create the member using an application server template.
      - Create the member using an existing application server as a template.
      - Create the member by converting an existing application server.

      Note: You can only add an existing application server to the cluster if you select that server as the first cluster member. You cannot add other existing application servers to that cluster after you create the first cluster member. If you add an existing server to a cluster, the only way to remove that server from the cluster is to delete the server. Therefore, you might want to use the existing server as a template for the first cluster member instead of as the cluster member. If you keep the original application server out of the cluster, you can reuse that server as the template if you need to rebuild the configuration.
7. Click **Next**.
8. Create additional cluster members. Before you create additional cluster members, check the configuration settings of the first cluster member. These settings are displayed at the bottom of the Create additional cluster members panel of the Create a new cluster wizard. For each additional member that you want to create:

   a. Specify a unique name for the member. The name must be unique within the node.
   b. Select the node to which you want to assign the cluster member.
   c. Specify the weight you want given to this member. The weight value controls the amount of work that is directed to the application server. If the weight value for the server is greater than the

weight values that are assigned to other servers in the cluster, then the server receives a larger share of the workload. The value can range from 0 to 20.

d. Select **Generate unique HTTP ports** if you want to generate unique port numbers for every HTTP transport that is defined in the source server.

e. Click **Add member**. You can edit the configuration settings of any of the newly created cluster members other than the first cluster member, or you can create additional cluster members. Click **Previous** to edit the properties of the first cluster member. The settings for the first cluster member become the settings for the cluster member template that is automatically created when you create the first cluster member.

9. When you finish creating cluster members, click **Next**.
10. View the summary of the cluster and then click **Finish** to create the cluster, click **Previous** to return to the previous wizard panel and change the cluster, or click **Cancel** to exit the wizard without creating the cluster.
11. To further configure a cluster, click **Servers > Clusters > WebSphere application server clusters >** , and then click the name of the cluster. Only the **Configuration** and **Local Topology** tabs appear until you save your changes.
12. Click **Review** to review your cluster configuration settings. Repeat the previous step if you need to make additional configuration changes.
13. If you do not want to make any additional configuration changes, select Synchronize changes with Nodes and then click **Save**. Your changes are saved and synchronized across all of your nodes.

    Note: If you click **Save**, but do not select Synchronize changes with Nodes, when you restart the cluster, the product does not start the cluster servers because it cannot find them on the node. If you want to always synchronize your configuration changes across your nodes, you can select Synchronize changes with Nodes as one of your console preferences.
14. Restart the cluster.

Results

You have created a cluster to which you can assign work requests. The **Runtime** and **Local Topology** tabs appear the next time you access this page.

What to do next

- You can click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Clusters members** in the administrative console, and then click the name of a cluster member to view all of the configuration settings for this cluster member. You can then use this page to change some of the configuration settings for the selected cluster member.

  For example, if you do not need to have all of the cluster member components start during the cluster startup process, you might want to reconfigure the cluster members, such that the **Start components as needed** is selected. This option is not selected when a new cluster member is created. Selecting this option can improve cluster startup time, and reduce the memory footprint of the cluster members.

  Note: Before selecting this option, verify that any other WebSphere products, that you are running in conjunction with this product, support this functionality.
- Use the administrative console to view or change the configuration settings for a cluster. For example, if you are running in a high availability environment, you can click **Servers > Clusters > WebSphere application server clusters >** *cluster_name*, and then select the **Enable failover of transaction log recovery** option for this cluster. This option allows the recovery of transactions to failover from one cluster member to another.
- If you create the cluster member by converting an existing application server and that server is a member of a bus, you need to migrate the messaging engine in the server to the scope of a cluster. To

do this, use the wsadmin command migrateServerMEtoCluster. Do not delete the messaging engine at server scope and re-create it a cluster scope, because this would prevent the messaging engine from working with previously configured destinations.

- Create additional cluster members.
- Start the cluster.
- Use scripting to automate the task of creating clusters.
- Create a static routing table to temporarily handle IIOP routing for the cluster if your high availability infrastructure is disabled.

Creating a cluster: Basic cluster settings

Use this page to enter the basic settings for a cluster.

To view this administrative console page, click **Servers > Clusters > WebSphere application server clusters > New**.

Cluster name

Specifies the name of the cluster. The cluster name must be unique within the cell.

Prefer local

Specifies that the host scoped routing optimization is enabled or disabled. The default is enabled, which means that, when possible, enterprise bean requests are routed to the client host. Enabling this setting improves performance because client requests are sent to local enterprise beans.

Data type	Boolean
Default	true

Configure HTTP session memory-to-memory replication

Specifies that when the cluster is created, a memory-to-memory replication domain is created for each of the members of this cluster.

If a replication domain is created, it 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, if a replication domain is created, the SIP container and Web container for each cluster member is configured for memory-to-memory replication.

To modify the replication domain settings, in the administrative console, click **Environment > Replication domains >** *replication_domain_name*.

The default mode setting for the replication domain is `Both client and server`. In this mode, all data sent to either the client or the server is replicated. This setting is good for an environment that has a middle to low traffic load. However, if your environment has a high traffic load, you should change the replication domain mode setting to either `Client only` , or `Server only`, because these settings provide better scaling. In `Client only` mode, only data sent to the client is replicated. In `Server only` mode, only data sent to the server is replicated.

To modify the mode setting for a replication domain:

1. In the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name*.
2. Under Container settings, click either **SIP container settings** or **Web container settings**, and then click **Session management > Distributed environment settings > Memory-to-memory replication**.

3. Select a different mode. It does not matter whether you change the mode under the SIP container or the Web container because the same replication domain settings apply to both containers.

Note: If you change any of the replication domain settings for one cluster member, including the mode setting, you should change them for all of the other members of the cluster.

Creating a cluster: Create first cluster member

Use this page to specify settings for the first cluster member.

There are two ways to create the first member of a cluster:

- You can create the first member when you create a new cluster.

  To create a new cluster, in the administrative console, click **Servers > Clusters > WebSphere application server clusters > New**.
- You can create an empty cluster and then add a first member after you finish creating the cluster.

  To create a cluster member for an existing cluster, in the administrative console, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Clusters members > New**.

When you create the first cluster member, a copy of that member is stored as part of the cluster data and becomes the template for all additional cluster members that you create.

When adding servers to a cluster, remember that the only way to remove an application server from a cluster is to delete the application server from the list of cluster members.

Member name

Specifies the name of the application server that is created for the cluster.

The member name must be unique on the selected node.

Select node

Specifies the node on which the application server resides.

Weight

Specifies the amount of work that is directed to the application server.

If the weight value for the server is greater than the weight values that are assigned to other servers in the cluster, the server receives a larger share of the cluster workload. The value can range from 0 to 20. Enter zero to indicate that you do not want requests to route to this application server unless this server is the only server that is available to receive requests.

Core group

Specifies the core group in which the application server resides. This field displays only if you have multiple core groups configured. You can change this value only for the first cluster member.

Generate unique HTTP ports

Specifies that a unique HTTP port is generated for the application server. By generating unique HTTP ports for the application server, you avoid potential port collisions and configurations that are not valid.

Select basis for first cluster member:

Specifies the basis you want to use for the 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.

- 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.
- If you select **None. Create an empty cluster** , a new cluster is created but it does not contain any cluster members.

Note: The basis options are available only for the first cluster member. All other members of a cluster are based on the cluster member template which is created from the first cluster member.

Creating a cluster: Summary settings

Use this administrative console page to view and save settings when you create a cluster or cluster member.

You can view this administrative console page whenever you create a new cluster or a new cluster member. This summary page displays your configuration changes before you commit the changes and the new cluster or cluster member is created.

To create a cluster, in the administrative console, click **Servers > Clusters > WebSphere application server clusters > New**.

To create a cluster member for an existing cluster, in the administrative console, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Clusters members > New**

The bounding node group of the cluster is based on the first application server that is added as a member of the cluster. The settings for this first member become the settings for the cluster member template that is then used to create additional cluster members. To select a different bounding node group, in the administrative console, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Clusters members > New**, and then select the appropriate node group for the new cluster member. When you select a different node group, another cluster member template is automatically created for that node group, if one does not already exist.

Review the changes to your configuration, and then click **Finish** to complete and save your work.

Creating a cluster: 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 members to a cluster, in the administrative console, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Cluster members > New**. After you enter the required information about the new cluster member, click **Add Member** to add this member to the cluster member list.

After adding a cluster member, you might need to change one or more of the property settings for this cluster member, or another cluster member that you just added. To change one or more property settings for any cluster member that you just added, other than the first cluster member, select that cluster member, and then click **Edit**. When you finish changing the property settings, click **Update Member** to save your changes.

If you decide not to create a particular cluster member, select the member and then click **Delete**.

You cannot edit or delete the first cluster member or an already existing cluster member.

If you create additional cluster members immediately after you create the first cluster member, the list of cluster members includes a checklist in front of the names of these additional cluster members. However,

a check box does not appear in front of the name of the first cluster member because you cannot delete this member or edit its settings. To modify the first cluster member, click **Previous**.

Similarly, if you are adding cluster members to a cluster that already has existing members, the existing members appear in the list of cluster members but a check box does not appear in front of the names of these cluster members. To delete one of these existing members or to change the settings of one of these cluster members, in the administrative console click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Cluster members**, and then select the member that you want to delete or whose configuration settings you want to change.

Member name
Specifies the name of the application server that is created for the cluster.

The member name must be unique on the selected node.

Select node
Specifies the node on which the application server resides.

In a mixed cell environment, you can use any server from within the node group to create a new cluster member. For example, if the node group to which the cluster belongs consists of a Version 7.0 node and a Version 6.1 node, you can use a server from either the Version 7.0 node or the Version 6.1 node to create a new cluster member. Similarly if the node group to which the cluster belongs consists of a Version 7.0 node and a Version 5.1 node, you can use a server from either the Version 7.0 node or the Version 5.1 node to create a new cluster member.

Weight
Specifies the amount of work that is directed to the application server.

If the weight value for the server is greater than the weight values that are assigned to other servers in the cluster, the server receives a larger share of the cluster workload. The value can range from 0 to 20. Enter zero to indicate that you do not want requests to route to this application server unless this server is the only server that is available to receive requests.

Generate unique HTTP ports
Specifies that a unique HTTP port is generated for the application server. By generating unique HTTP ports for the application server, you avoid potential port collisions and configurations that are not valid.

Server cluster collection
Use this page to view information about and change configuration settings for a cluster. A cluster consists of a group of application servers. If one of the application servers fails, requests are routed to other members of the cluster.

To view this administrative console page, click **Servers > Clusters > WebSphere application server clusters**.

To define a new cluster, click **New** to start the Create a new cluster wizard.

Name
Specifies a logical name for the cluster. The name must be unique among clusters within the containing cell.

Status
This field indicates whether the cluster is partially started, started, partially stopped, stopped, or unavailable. If the status is unavailable, the node agent is not running in that node and you must restart the node agent before you can start the server.

After you click **Start** or **Ripplestart** to start a cluster, each server that is a member of that cluster launches if it is not already running. When the first member launches, the state changes to `partially started`. The state remains `partially started` until all cluster members are running. When all cluster members are running, the state changes to `running` and the status changes to `started`. Similarly, when you click **Stop** or **ImmediateStop** to stop a cluster, the state changes to `partially stopped` when the first member stops, and then changes to `stopped` when all cluster members are not running.

	**Started**	The server is running.
	**Partially stopped**	The server is in the process of changing from a started state to a stopped state.
	**Stopped**	The server is not running.
	**Unknown**	The server status cannot be determined.

Server cluster settings

Use this page to view or change the configuration of a server cluster instance, and to view the local topology of a server cluster instance.

To change the configuration and local topology of a server cluster, in the administrative console click **Servers > Clusters > Clusters >***cluster_name*.

To view runtime information, such as the state of the server cluster, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name*, and then click the **Runtime** tab.

To display the topology of a specific cluster, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name*, and then click the **Local Topology** tab.

If the high availability infrastructure is disabled and you require IIOP routing capabilities, follow the instructions contained in the ″Enabling static routing for a cluster″ topic to create a static route table. This table enables the cluster to handle IIOP requests.

Note:

- Because the information contained in the static route table does not account for server runtime state, you should only use this option if the high availability infrastructure is disabled.

  Use of a static route table preempts the use of the dynamic routing table that is contained in cluster members. After the static file is transferred to a node, whenever a cluster member residing in that node starts, that cluster member uses the static table instead of the dynamic table to handle IIOP routing. If a cluster member is running when you create the static route table, then you must restart that cluster member to give that cluster member access to the static route table information, because the table content is loaded at run time.
- After the table is created, an informational message, similar to the following message, is issued that indicates the name of the file that contains the table and where that file is located:

  ```
  The route table for cluster MyCluster was exported to file
  /home/myInstall/was/server/profiles/dmgrProfile/config/cells/
     MyCell/clusters/Myfile.wsrttbl.
  ```

  As this message indicates, the file containing the static route table is placed in the config directory of the deployment manager for this cluster. Keep a record of this location so that you can delete this file when you are ready to start using dynamic routing again.
- If you set up a static route table, then you must statically set the ORB_LISTENER_ADDRESS port on each of the cluster members because the route table is static, and the cluster members do not communicate during state changes. If this port is not assigned, then the cluster members restart on different ports, and the static routing information is not able to route requests to the cluster members.

Cluster name:

Specifies a logical name for the cluster. The name must be unique among clusters within the containing cell.

Short name:

Specifies the short name for this cluster. This field displays only if you are running on z/OS.

The short name is used as the WLM APPLENV name for all servers that are part of this cluster.

If you specify a short name for a cluster member, the name:
- Must be one to eight characters in length
- Must contain only uppercase alphanumeric characters
- Cannot start with a number
- Must be unique in the cell

If you do not specify a short name, then the system assigns a default short name that is automatically unique within the cell. You can change the generated short name to conform with your naming conventions.

Bounding node group name:

Specifies the node group that forms the boundaries for this cluster. All application servers that are members of a cluster must be on nodes that are members of the same node group.

A node group is a collection of application server nodes. A node is a logical grouping of managed servers, usually on a system that has a distinct IP host address. All application servers that are members of a cluster must be on nodes that are members of the same node group. Nodes that are organized into a node group need enough capabilities in common to ensure that clusters formed across the nodes in the node group can host the same application in each cluster member. A node must be a member of at least one node group and can be a member of more than one node group.

Create and manage node groups by clicking **System administration > Node groups** in the administrative console.

Prefer local:

Specifies that the host scoped routing optimization is enabled or disabled. The default is enabled, which means that, when possible, enterprise bean requests are routed to the client host. Enabling this setting improves performance because client requests are sent to local enterprise beans.

Data type	Boolean
Default	true

Enable failover of transaction log recovery:

Specifies that for the transaction service component, failover of the transaction log for recovery purposes is enabled or disabled. The default is disabled.

When this setting is enabled, and the transaction service properties required for peer recovery of failed application servers in a cluster are properly configured, failover recovery of the transaction log occurs if the server processing the transaction log fails. If the transaction services properties required for peer recovery of failed application servers in a cluster are not properly configured, then this setting is ignored.

wlcID:

Specifies the currently registered workload controller (WLC) identifier for the cluster. This setting might not display for all configurations.

Data type String

State:

Specifies whether the cluster is stopped, starting, or running.

If all cluster members are stopped, the cluster state is `stopped`. After you request to start a cluster, the cluster state briefly changes to `starting` and each server that is a member of that cluster launches, if it is not already running. When the first member launches, the state changes to `websphere.cluster.partial.start`. The state remains `partially started` until all cluster members are running, then the state changes to `running`. Similarly, when stopping a cluster, the state changes to `partially stopped` as the first member stops and changes to `stopped` when all members are not running.

Valid values `starting, partially started, running, partially stopped, or stopped.`

Cluster topology

Use this page to display, in a tree format, a list of all of the application server clusters defined for your WebSphere Application Server environment. The list shows all of the nodes and cluster members that are included in each cluster contained in a cell.

To view this page, in the administrative console, click **Servers > Clusters > Cluster topology**.

Enabling static routing for a cluster

If your high availability infrastructure is disabled and you require IIOP routing capabilities, you can create a static routing table for the members of a cluster to use to handle enterprise bean requests. Because the information contained in this static routing table does not account for server runtime state, you should delete this table and return to using the dynamic routing table as soon as your high availability infrastructure is enabled.

Before you begin

Before you create a static route table, ensure that:

- The ORB_LISTENER_ADDRESS port is set to a non-zero value on each of the cluster members. Because the route table you create is static, and the cluster members do not communicate during state changes, if you do not set the ORB_LISTENER_ADDRESS port on each of the cluster members, the cluster members might restart on different ports, and IIOP requests will not be routed correctly.

  To change the value specified for the ORB_LISTENER_ADDRESS port:
  1. In the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name*, and then under Communications, click **Ports** .
  2. Click **ORB_LISTENER_ADDRESS** in the Port name field.
  3. Change the value specified for the Port field to a value that is greater than 0.
- Each cluster member is started and can use these new non-zero ORB_LISTENER_ADDRESS port values to correctly route IIOP requests.

About this task

You should only create a static route table if your high availability infrastructure is disabled and you require IIOP routing capabilities. To create a static route table:

1. Start the wsadmin tool if it is not already running.

2. Identify the cluster managed bean (MBean) for the cluster for which you are creating the route table, and assign that MBean to a variable.
   - Using Jacl:

     ```
     cluster = AdminControl.completeObjectName('cell=
       cell_name,type=Cluster,name=cluster_name,*')
     print cluster
     ```

   - Using Jython:

     ```
     set cluster [$AdminControl completeObjectName cell=
        cell_name,type=Cluster,name=cluster_name,*]
     puts $cluster
     ```

   These commands return the name of the cluster MBean for the specified cluster. For example, for cluster cluster1, the output from these commands will be similar to the following message:

   ```
   WebSphere:cell=mycell,name=cluster1,mbeanIdentifier=Cluster,type=
      Cluster,process=cluster1
   ```

3. Export the route table.
   - Using Jacl:

     ```
     $AdminControl invoke $cluster exportRouteTable
     ```

   - Using Jython:

     ```
     AdminControl.invoke(cluster, 'exportRouteTable')
     ```

   After the table is created, the name of the route table file, is displayed in a message similar to the following message:

   ```
   /home/myInstall/was/server/profiles/dmgrProfile/config/cells/mycell/
      clusters/cluster1/cluster1.wsrttbl
   ```

   As this message illustrates, the file containing the table is placed in the config directory of the deployment manager for that cluster. You should keep a record of this location so that you can delete this file when you are ready to start using dynamic routing again.
4. Synchronize the configuration changes across nodes.
   a. Clear the configuration repository Epoch. If you do not clear the configuration repository Epoch, the synchronization only updates the files that the configure service component edited, which does not include the file that contains the static routing table.

      Using Jacl:

      ```
      set configRepository [$AdminControl completeObjectName
           node=node_name,type=ConfigRepository,*]
      $AdminControl invoke $configRepository refreshRepositoryEpoch
      ```

      Using Jython:

      ```
      configRepository = AdminControl.completeObjectName('node=node_name,
          type=ConfigRepository,*')
      AdminControl.invoke(configRepository, 'refreshRepositoryEpoch')
      ```

   b. Repeat this process for each node that you want to synchronize.
5. Stop the cluster. Follow the instructions specified either the *Stopping clusters* or *Stopping clusters using scripting* topic.
6. Exit the wsadmin tool.
7. Use the following Debug flag appended to the startServer command to manually start each member of this cluster.

   ```
   -Dcom.ibm.websphere.management.registerServerIORWithLSD=false
   ```

   For example, to start server1 on a Windows operating system with static routing enabled, issue the following command from the server profile's bin directory:

   ```
   startServer.bat server1 -Dcom.ibm.websphere.management.registerServerIORWithLSD=false
   ```

Results

The cluster members use the static route table to perform IIOP routes.

What to do next

When your high availability infrastructure is enabled, follow the instructions in the topic *Disabling static routing for a cluster* to disable static routing. When static routing is disabled, the cluster members resume using dynamic routing.

Disabling static routing for a cluster

Because the information contained in a static routing table does not account for server runtime state, you should delete this table and return to using the dynamic routing table as soon as your high availability infrastructure is enabled. When you delete the static routing table, cluster members automatically resume using dynamic routing to handle enterprise bean requests.

About this task

Perform the following steps to delete the static routing table.

1. For each member of the cluster, set the ORB_LISTENER_ADDRESS port to 0 (zero).
   a. In the administrative console, click **Servers > Server Types > WebSphere application servers >** *server_name*, and then in the Communications section, click **Ports**.
   b. Click **ORB_LISTENER_ADDRESS** in the Port name field.
   c. Change the value specified for the Port field to 0.
2. Manually delete the static route table file from the config directory of the deployment manager for the cluster.

   The path to this config directory was included in the message that you received when you originally exported this file. If you did not retain this information, you can do a search in the deployment manager config directory for the file cluster_name.wsrttbl.
3. Synchronize the configuration changes across nodes.
   a. Clear the configuration repository Epoch. If you do not clear the configuration repository Epoch, the synchronization only updates the files that the configure service component edited, which does not include the file that contains the static routing table.

      Using Jacl:

      ```
      set configRepository [$AdminControl completeObjectName
           node=node_name,type=ConfigRepository,*]
      $AdminControl invoke $configRepository refreshRepositoryEpoch
      ```

      Using Jython:

      ```
      configRepository = AdminControl.completeObjectName('node=node_name,
          type=ConfigRepository,*')
      AdminControl.invoke(configRepository, 'refreshRepositoryEpoch')
      ```
   b. Repeat this process for each node that you want to synchronize.
4. Stop the cluster. Follow the instructions specified either the *Stopping clusters* or *Stopping clusters using scripting* topic.
5. Start the cluster again. Follow the instructions specified either the *Starting clusters* or *Starting clusters using scripting* topic.
6. Exit the wsadmin tool.

Results

The cluster members resume using the dynamic routing table to handle IIOP requests.

Adding members to a cluster

You can use clusters to balance workload in an environment containing multiple application servers.

Before you begin

Create a cluster if you do not already have a cluster defined for your environment.

About this task

If you are migrating from a previous version of the product, you can upgrade a portion of the nodes in a cell, while leaving others at the previous release level. For a time, you might be managing servers that are at a previous release level, and servers that are running at the current release level in the same cell.

When you create a cluster, you specify the node on which the first cluster member resides. In a mixed cell environment, you can use any server from within that node group to create a new cluster member. For example, if the node group to which the cluster belongs consists of a Version 7.0 node, and a Version 6.1 node, you can use a server from either the Version 6.1 or the Version 7.0 node to create a new cluster member.

Use the following procedure to create a new cluster member, view information about existing cluster members, or manage existing cluster members.

1. In the administrative console, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Cluster members**. The Cluster members page lists members of a cluster, and for each member indicates:
   - The node on which the member resides.
   - The version of the application server. This information specifies whether the cluster is a mixed cluster.
   - The configured weight for the member.
   - The runtime weight for the member. This weight indicates the proportionate workload that is currently directed to this cluster member.
   - Whether the member is started, stopped, or encountering problems.
2. Click **New** to create a new cluster member.

   Clicking **New** starts the Create a new cluster member wizard. Use this wizard to add new members to an already configured 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. Usually, only one template is available for you to use to create additional cluster members for a cluster. However, if a cluster includes nodes that are at different versions of the product, there is a different template for each version. For example, if a cluster has cluster members that reside on both a Version 6.1 node and a Version 7.0 node, the cluster has two templates. The Version 6.1 template is used when you create an additional cluster member on the Version 6.1 node, and the Version 7.0 template is used when you create an additional cluster member on the Version 7.0 node.

   To view the cluster member templates that are available for creating a new member of a cluster, in the administrative console, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Cluster members > Templates**.

   a. Specify a name for the application server that you are defining as a cluster member. The name must be unique within the node.

   b. Select the node for the cluster member.

   c. Specify the server weight.

      The weight value you specify controls the number of requests that are directed to the application server. Even though you specify a value of `0` to `20` as the weight of a server, the weight that is given to the server as a member of a cluster is a proportion that is based on the weight assigned to the server, and the sum of the weights of all members of the cluster. In this proportion, the weight that is assigned to the server is the numerator, and the sum of the weights of all members of the cluster is the denominator.

When you add a new member to a cluster, the number of requests that are sent to each server in the cluster decreases, assuming that the number of requests coming into the cluster stays the same. Similarly when you remove a new member from a cluster, the number of requests that are sent to each server in the cluster increases, assuming that the number of requests coming into the cluster stays the same.

For example, if you have a cluster that consists of members A, B, and C with weights 2, 3, and 4, respectively, then 2/9 of the requests are assigned to member A, 3/9 are assigned to member B, and 4/9 are assigned to member C. If a new member, member D, is added to the cluster and member D has a weight of 5, then member A now gets 2/14 of the requests, member B gets 3/14 of the requests, member C gets 4/14 of the requests, and member D gets 5/14 of the requests.

d. Specify whether to generate unique HTTP ports.

e. Click **Add member** to finish defining the cluster member. The first cluster member for this cluster is used as the template for this cluster member. You can repeat these steps to define other cluster members.

f. When you finish defining additional cluster members, review the summary information for the new cluster members. If you have to change any of the property settings for any of the new members, select that cluster member, and then click **Edit**. When you finish changing the property settings, click **Update member** to save your changes.

g. When you finish defining new cluster members, click **Next** to view the summary page for the cluster, and then click **Finish** to create these new cluster members.

3. Click **Review**, select **Synchronize changes with nodes**, and then click **Save** to save your changes.

Results

You created application servers that are members of an existing server cluster.

What to do next

If, when you created the new members, you chose to generate unique ports, update the alias list for the virtual host that you plan to use with the new servers.

You can also perform the following actions:

- On the Cluster members page in the administrative console, click the name of one of the cluster members, and examine the configuration settings for that cluster member. You can change any of the settings that are not appropriate.

  For example, if you do not need to have all of the cluster member components start during the cluster member startup process, you might want to select **Start components as needed**, which is not automatically selected when a new cluster member is created. When this property is selected, cluster member components are dynamically started as they are needed. When this property is not selected, all of the cluster member components are started during the startup process. Therefore, selecting this property usually results in improved startup performance because fewer components are started during the startup process.
- Click **Servers > Sever Types > WebSphere application servers >***sever_name* or click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Cluster members >** *cluster_member_name* to perform either of the following tasks:
  - Specify additional application server properties for this cluster member.
  - Click **Installed applications**.
- Create a backup cluster.
- Start the cluster.
- Use scripting to automate the task of adding cluster members.

Cluster member collection

Use this page to view and manage application servers that belong to a cluster. You can also use this page to change the weight of any of the listed application servers.

Application servers that are part of a cluster are referred to as cluster members. 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.

You can use this page to perform the following actions for the listed cluster members:

- Start a cluster member. To start a cluster member, select the server that you want to start. Then click **Start**.
- Restart a cluster member. To restart a cluster member, select the server that you want to restart. Then click **Restart**. When you restart a cluster member, the selected cluster member stops, following the normal server quiesce process, and then starts again.
- Stop a cluster member. To stop a cluster member, select the server that you want to stop, and then click one of the following buttons:

  Stop When you click this button, the normal server quiesce process is followed. This process allows in-flight requests to complete before the entire server process shuts down.

  Immediate Stop
  When you click this button, the selected sever stops but the normal server quiesce process is not followed. This shutdown mode is faster than the normal server stop processing, but some application clients might receive exceptions if an in-flight request does not complete before the server process shuts down.

  Terminate
  You should only click **Terminate** if the cluster member does not respond when you click **Stop** or **Immediate Stop** or when you issue the Stop or ImmediateStop commands. Some application clients can receive exceptions. Therefore, you should always attempt an immediate stop before clicking **Terminate**.

  Make Idle
  When you click this button, the cluster member moves to the idle state.
- Delete a cluster member. To delete a cluster member, select the cluster member that you want to delete. Then click **Delete**.

Any individual configuration change that you make to a cluster member does not affect the configuration settings of the cluster member template. You must use wsadmin commands to modify this template. Similarly, any changes that you make to the template do not affect existing cluster members.

See the *Using the administrative clients* PDF for more information on how to modify this template.

To view this administrative console page, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Clusters members**.

Member name

Specifies the name of the server in the cluster. On most platforms, the name of the server is the process name. The name must match the (object) name of the application server.

Node

Specifies the name of the node for the cluster member.

Host Name

Specifies the IP address, the full domain name system (DNS) host name with a domain name suffix, or the short DNS host name for the cluster member.

Version
Specifies the version of the product on which the cluster member runs.

Configured weight
Specifies the weight that is currently configured for the cluster member. The weight determines the amount of work that is directed to the cluster member. If the weight value for the server is greater than the weight values assigned to other servers in the cluster, the server receives a larger share of the cluster workload.

To change the configured weight for a cluster member you can either specify a new weight in the Configured weight field and click the **Update** button for the Configured weight column, or click on the name of the cluster member. Clicking the name of the cluster member navigates you to the page where you can change any of the configuration settings for that cluster member.

Runtime weight
Specifies the proportionate workload that is currently directed to the cluster member in comparison to the runtime weights of the rest of the cluster members. The runtime weight only applies for Remote Method Invocation over Internet Inter-ORB Protocol (RMI-IIOP) requests, Enterprise JavaBeans (EJB) requests, and HTTP requests received through a WebSphere proxy server.

You can use the **Runtime weight** property to dynamically adjust the weight that is given to a particular cluster member without having to stop and then restart that cluster member. To change the proportion, specify a new weight for the **Runtime weight** property, and then click the **Update** button for the Runtime weight column. The runtime weight change takes effect immediately.

Note: Changing the runtime weight for a cluster member does not affect the configured weight setting for that cluster member. The configured weight setting still applies for HTTP requests that do not come through a WebSphere proxy server.

Status
This field indicates whether the cluster member is started, stopped, partially stopped, or unavailable. If the status is unavailable, the node agent is not running in that node and you must restart the node agent before you can start the server.

	**Started**	The server is running.
	**Partially stopped**	The server is in the process of changing from a started state to a stopped state.
	**Stopped**	The server is not running.
	**Unknown**	The server status cannot be determined.

Cluster member settings
Use this page to manage the members of a cluster. A cluster of application servers are managed together and participate in workload management.

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.

Any individual configuration change that you make to a cluster member does not affect the configuration settings of the cluster member template. You can use wsadmin commands to modify the cluster member template, or you can click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Clusters members > Templates**. Any change that you make to the template does not affect existing cluster members. See the *Using the administrative clients* PDF for more information on how to use wsadmin commands to modify this template.

To view this administrative console page, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name*.

On the **Configuration** tab, you can edit fields. You can also click **Installed applications.** to view the status of applications that are running on this server. On the **Runtime** tab, which only appears when the cluster member is running, you can look at information about this cluster member. However, the information that displays on this page is read-only. You must return to the **Configuration** tab to change any of the settings that display.

Member name:

Specifies the name of the application server in the cluster. On most platforms, the name of the server is the process name. The member name must match the name of one of the servers that are listed on the application servers page.

Node name:

Specifies the name of the node on which the cluster member is running.

Weight:

Controls the number of requests that are directed to the application server. Even though you specify a value of `0` to `20` as the weight of a server, the weight that is assigned to the server is a proportion, in which, the weight assigned to the server is the numerator, and the sum of the weights of all members of the cluster is the denominator.

When you add a new member to a cluster, the number of requests that are sent to each server in the cluster decreases, assuming that the number of requests coming into the cluster stays the same. Similarly when you remove a new member from a cluster, the number of requests that are sent to each server in the cluster increases, assuming that the number of requests coming into the cluster stays the same.

For example, if you have a cluster that consists of members A, B, and C with weights 2, 3, and 4, respectively, then 2/9 of the requests are assigned to member A, 3/9 are assigned to member B, and 4/9 are assigned to member C. If a new member, member D, is added to the cluster and member D has a weight of 5, then member A now gets 2/14 of the requests, member B gets 3/14 of the requests, member C gets 4/14 of the requests, and member D gets 5/14 of the requests.

Data type	Integer
Range	0 to 20

Unique ID:

Specifies a numerical identifier for the application server that is unique within the cluster. The ID is used for affinity.

Data type	Integer

Run in development mode:

Enabling this option may reduce the startup time of an application server. This might include Java virtual machine (JVM) settings, such as disabling bytecode verification and reducing Just in Time (JIT) compiler compilation costs. Do not enable this setting on production servers. This setting is only available on application servers that are running in Version 6.0 and or higher cells.

Specifies that you want to use the JVM settings, **-Xverify** and **-Xquickstart**, on startup. After selecting this option, save the configuration and restart the server to activate development mode.

The default setting for this option is `false`, which indicates that the server is not started in development mode. Setting this option to `true` specifies that the server is started in development mode, using settings that decrease server startup time.

Data type	Boolean
Default	false

Parallel start:

Specifies whether to start the server on multiple threads. When you start the server on multiple threads, the server components, services, and applications start in parallel rather than sequentially, which might shorten the startup time.

The default setting for this option is `true`, which indicates that the server uses multiple threads when it starts. Setting this option to `false` specifies that the server uses a single thread when it starts, which might lengthen startup time.

The order in which applications start depends on the weight you assign to each application. The application with the lowest starting weight starts first. Applications with the same starting weight start in parallel. Use the `Starting weight` field on the **Applications > Application Types > WebSphere enterprise applications >** *application_name* **> Startup behavior** page of the administrative console to set the starting weight for an application.

Data type	Boolean
Default	true

Start components as needed:

Select this field if you want the cluster member components started as they are needed by an application that is running on this cluster member.

When this property is selected, cluster member components are dynamically started as they are needed. When this property is not selected, all of the cluster member components are started during the cluster startup process. Therefore, selecting this option can improve startup time, and reduce the memory footprint of the cluster members, because fewer components are started during the startup process.

Starting components as they are needed is most effective if all of the applications, that are deployed on the cluster, are of the same type. For example, using this option works better if all of your applications are Web applications that use servlets, and JavaServer Pages (JSP). This option works less effectively if your applications use servlets, JSPs and Enterprise JavaBeans (EJB).

Note: To ensure compatibility with other WebSphere products, the default setting for this option is deselected. Before selecting this option, verify that any other WebSphere products, that you are running in conjunction with this product, support this functionality.

Access to internal server classes:

Specifies whether the applications that are running on this server can access many of the server implementation classes.

If you select `Allow`, then, applications can access most of the server implementation classes. If you select `Restrict`, then applications cannot access server implementation classes. The applications get a ClassNotFoundException error if they attempt to access these classes.

Usually, you should select `Restrict` for this property, because most applications use the supported APIs, and do not need to access any of the internal classes. However, if your application requires the use of one or more of the internal server classes, then select `Allow` as the value for this property.

The default value for this property is `Allow`.

Class loader policy:

Specifies whether there is a single class loader that loads all of the applications, or a different class loader loads each application.

Class loading mode:

Specifies whether the class loader searches in the parent class loader, or in the application class loader first to load a class. The standard for Developer Kit class loaders and product class loaders is `Classes loaded with parent class loader first`.

This field only applies if you set the Class loader policy field to `Single`.

If you select `Classes loaded with local class loader first (parent last)`, your application can override classes that are contained in the parent class loader, but this action can potentially result in ClassCastException, or linkage errors, if you have mixed use of overridden classes and non-overridden classes.

Process ID:

Specifies the native operating system process ID for this server.

The process ID property is read only. The system automatically generates the value.

Cell name:

Specifies the name of the cell in which this server is running.

The Cell name property is read only.

Node name:

Specifies the name of the node in which this server is running.

The Node name property is read only.

State:

Specifies the runtime state for this server.

The State property is read only.

Cluster member templates collection

Use this page to view the list of cluster member templates that exist for this cluster. To edit the server properties of a template, click the name of that template.

Usually, only one template exists that you can use to create additional members for a cluster. However, if a cluster includes nodes that are at different versions of the product, a different template exists for each of these versions. For example, if a cluster has cluster members residing on both a Version 6.1 node and a Version 7.0 node, the cluster has two templates. The Version 6.1. template is used when you create an additional cluster member on the Version 6.1 node, and the Version 7.0 template is used when you create an additional cluster member on the Version 7.0 node.

If you modify a template, all new cluster members are created with the server property settings of the modified template. However, the property settings of existing cluster members do not change. If you make any change to a cluster member template, you should make the same change to all of the existing cluster members.

To change the server attributes of an existing cluster member, in the administrative console, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Clusters members**, and then click the name of the existing cluster member.

To view the cluster member templates that are available for creating a new member of a cluster, in the administrative console, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Cluster members >** *cluster_member_name*, and then click **Templates**.

Name
Specifies the name of the cluster member template.

Platform
Specifies the operating system platform to which this template applies.

Version
Specifies the version of the product to which the template applies.

Description
Specifies a description of this cluster member template. This field is optional and might be blank.

Creating backup clusters

Use this task to configure a backup cluster that handles Enterprise JavaBeans (EJB) requests if the primary cluster fails.

Before you begin

Before you begin, create two clusters that are able to provide backup for each other. The objects and resources available in the primary cluster must also be available in the backup cluster. You must use the same cluster name, install the same applications, use the same application names, and define the same resources in the backup cluster as in the primary cluster.

The primary cluster and the backup cluster must reside in separate cells because a cluster must have a unique name within a cell.

About this task

Perform this task to create a backup cluster for your EJB clusters. When all the servers in the primary cluster fail, work is not halted because the backup cluster can continue serving requests for EJB work.

To configure a backup cluster, specify a name and a port. The port is called a domain bootstrap address and consists of a bootstrap host and port. The bootstrap host is the host that contains the deployment manager in which the backup cluster is configured. The bootstrap port is equal to the bootstrap port for the same deployment manager.

The primary cluster and the backup cluster must reside in separate cells. The bootstrap host and port for the backup cluster determine which cell contains the backup cluster.

1. Determine the bootstrap host and port of the backup cluster.
   a. Connect the administrative console for the deployment manager that contains the backup cluster.
   b. Click **System Administration > Deployment manager > Ports > BOOTSTRAP_ADDRESS**. The host and port for the BOOTSTRAP_ADDRESS instance is the host and port that the backup cluster uses. Remember these values for when you configure the primary cluster.
2. Connect the administrative console to the deployment manager that contains the primary cluster. Click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Backup cluster**.
3. Ensure that the name of the backup cluster is the same as the primary cluster.
4. Click **Domain bootstrap address**. Specify the backup cluster deployment manager bootstrap host and port in the **Host** and **Port** fields. Click **OK**. The bootstrap host and port combined define a bootstrap address for the deployment manager. On the Domain Bootstrap Address page, use the **Configuration** tab to statically define the backup cluster; the static value is consumed each time the deployment manager starts. You can use the **Runtime** tab to define the backup cluster during run time only; when the deployment manager stops, the run-time backup cluster information is discarded.
5. Click **OK**.
6. Configure a core group bridge between each of the cluster core groups. Use an access point group to join the two core groups. In the deployment manager for the primary cell, configure an access point group that has a peer access point that refers to the core group access point in the backup cell. In the deployment manager for the backup cell, create an access point group that has the same name as the access point group that you created in the primary cell. Add a peer access point that refers to the core group access point in the primary cell. See Configuring the core group bridge service for more information.

   Note: If you are configuring a V5.x cluster to back up a cluster that is on the current release, do not configure the core group bridge service. Core groups are not supported in V5.x. Therefore, the V5.x cluster does not belong to a core group. The backup cluster still functions using only the domain bootstrap address.
7. Save the configuration.

Results

The backup cluster completes EJB requests when the primary cluster fails.

What to do next

If you experience problems when configuring your backup cluster, see the *Troubleshooting and support* PDF.

Backup clusters

Backup clusters mirror the primary server clusters. Mirrored cluster support is for Enterprise JavaBeans (EJB) requests only.

Overview and prerequisites

When all the members of a cluster are no longer available to service Enterprise JavaBeans (EJB) requests, any clients that must interact with one of the Enterprise JavaBeans (EJB) application servers in the cluster do not function. Mirrored clusters enable an EJB cluster (primary cluster) to failover to another Enterprise JavaBeans (EJB) cluster (backup cluster) when none of the Enterprise JavaBeans (EJB) application servers in the primary cluster are able to service a request. The backup cluster allows the client to continue functioning when all of cluster members in the primary cluster are not available.

The fail back is automatic. You do not have to initiate fail back to the primary cluster after restarting the servers in the primary cluster. The backup cluster stops servicing requests as soon as the primary cluster becomes available. However, all of the deployment managers must be functional for backup cluster support, and the primary cluster must be defined as the backup for the backup cluster.

For the backup cluster to take over servicing requests successfully:

- The objects and resources available in the primary cluster must also be available in the backup cluster.
- You must use the same cluster name, install the same applications, use the same application names, and define the same resources in the backup cluster as in the primary cluster.
- The primary cluster and the backup cluster must reside in separate cells because a cluster must have a unique name within a cell.
- Both the primary and backup clusters must have a backup cluster configured, and each cluster must specify the opposite cluster as its backup cluster.

Because the primary and backup clusters reside in different cells, with the current version of the product, the clusters also reside in different core groups. You must configure the core group bridge service to allow communication between core groups. The core group bridge service eliminates the requirement of a running deployment manager and a node agent for the backup cluster support. In the previous release, if the deployment manager stopped, new requests could not be forwarded to the backup cluster after the primary cluster failed. Any core group bridge server that is configured in the cell that contains the primary cluster can provide information about the backup cluster. The backup cluster support fails only if all of the core group bridge servers in a cell are not running.

For cluster failover and fail back to function as expected, all of the servers, including the deployment manager, node agents, and application servers, for both the primary cluster and the backup cluster must be at a release and level that provides mirrored cluster support. However, if you are using a Version 5.x cluster to back up a cluster that is on the current release, you do not have the core group bridge service functionality available to you on the Version 5.x cluster.

Configuration

Mirrored cluster support is not configured by default. To use the mirrored cluster support, you must specify backup clusters in your configuration. Each cluster can have only one backup cluster, which must be configured before it is specified as a backup cluster.

To configure a backup cluster in a cluster, you must specify a name and a domain bootstrap address. The bootstrap host is the host that contains the deployment manager in which the backup cluster is configured. The bootstrap port is the bootstrap port for this deployment manager.

The primary cluster and the backup cluster must reside in separate cells. To place mirrored clusters in separate cells, configure the appropriate backup cluster domain bootstrap address. The backup cluster bootstrap host and port determine which cell contains the backup cluster.

You can configure a backup cluster using the administrative console or the ExtendedCluster managed bean (MBean). To configure a backup cluster using the administrative console:

- Use the **Configuration** tab on the Domain Bootstrap Address page to statically define the backup cluster; the static value is consumed each time the deployment manager starts.
- Use the **Runtime** tab on the Domain Bootstrap Address page to define the backup cluster when the cluster is running. When the deployment manager stops, the information defining the run-time backup cluster is discarded.

Because the primary and backup clusters reside in different cells, with the current version of the product, the clusters also reside in different core groups. You must configure the core group bridge service to allow communication between core groups. Use an access point group to join the two core groups. In the

deployment manager for the primary cell, configure an access point group that has the core group access point for the backup cell as a peer access point. In the deployment manager for the backup cell, create an access point group that has the same name as the access point group you created in the primary cell. Add a peer access point that refers to the core group access point in the primary cell. If you are configuring a Version 5.x cluster to back up a cluster that is on the current release, you do not have to configure the core group bridge service because the Version 5.x cluster does not belong to a core group. The backup cluster still functions using only the domain bootstrap address.

If you are configuring a backup cluster using the ExtendedCluster MBean, you can change the runtime configuration only. The MBean change does not affect the static configuration. You can use the setBackup operation on the ExtendedCluster MBean to change the run-time configuration. For example, you can use the following Java code to set the primary cluster's backup cluster:

```
ac.invoke(extendedCluster, "setBackup", new Object[] {
  backupClusterName, backupBootstrapHost, backupBootstrapPort},
  new String[] {
    "java.lang.String", "java.lang.String", "java.lang.Integer"});
```

In this sample, *ac* is the AdminClient, and *extendedCluster* is the ExtendedClusterObjectName for the primary cluster.

Fail back support scenarios

There are two scenarios that affect the cluster fail back support.

In the first scenario, requests are made by the client to the primary cluster, which eventually stops accepting requests. The requests are then routed to the backup cluster. The client initially sent requests to the primary cluster and therefore has information about the primary cluster. As a result, when the primary cluster is available again, the requests fail back to the primary cluster.

In the second scenario, the client does not start sending requests until after the primary cluster is down, and the requests go directly to the backup cluster. In this case, the client has information about the backup cluster only. Because the client knows about the backup cluster only, when the primary cluster becomes available, the requests from this client continue to route to the backup cluster and do not fail back to the primary cluster when it becomes available. This scenario occurs when an object is created on the backup cluster. In this case, the backup cluster becomes the primary cluster for this object.

Both of these scenarios can occur within the same client at the same time, if the client is sending requests to existing objects and creating new objects after the primary cluster stops processing.

Backup cluster settings

Use this page to configure a backup server cluster. The backup server cluster is used if the primary server cluster fails.

Configuration of a backup cluster is only useful if the cluster contains an Enterprise JavaBeans (EJB) module and a client outside of the cluster uses the EJB module.

To view this administrative console page, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Backup cluster**.

Backup cluster name

Specifies the name of the backup cluster. The backup cluster must have the same name as the server cluster that is containing the backup cluster. The backup cluster and its containing server cluster can have identical names because they must reside in different cells.

Data type String

Domain bootstrap address settings

Use this page to specify the bootstrap address host and port of the deployment manager that contains the backup cluster. You must specify a value for the Host and Port fields and configure a core group bridge before you can use the backup cluster function.

When you start the deployment manager, the operating system uses the host and port values specified on the **Configuration** tab for the bootstrap address. If you need to change the host or port while the deployment manager is running, specify new values on the **Runtime** tab. Any values specified on this tab take affect as soon as they are saved. However, if you stop the deployment manager and then restart it, the values on the **Runtime** tab revert to the values that are specified on the **Configuration** tab when the deployment manager restarts.

To view this administrative console page, click **Servers > Clusters > WebSphere application server clusters >** *cluster_name* **> Backup cluster > Domain bootstrap address**.

Host:

Specifies the IP address, domain name server (DNS) host name with domain name suffix, or just the DNS host name, of the bootstrap host for the deployment manager of the backup cluster.

For example, if the host name is `myhost`, the fully qualified DNS name can be `myhost.myco.com` and the IP address can be `155.123.88.201`.

Data type String

Port:

Specifies the bootstrap port number for the deployment manager of the backup cluster. The port value is used in conjunction with the host name.

Data type Integer

Starting clusters

You can start all members of a cluster at the same time by requesting that the state of a cluster change to *running*. That is, you can start all application servers in a server cluster at the same time.

Before you begin

Make sure that the members of your cluster have the debug port properly set. If multiple servers on the same node have the same debug port set, the cluster could fail to start. See “Java virtual machine settings” on page 285 for more information on how to change the debug port.

Windows If you use the Windows Services facility to start and stop application servers that are part of a cluster, remember that the cluster state does not always update correctly. For example, if a cluster is running and you stop a cluster member through the Services GUI, the cluster state remains as started even though the server is no longer running.

If you want cluster member components to dynamically start as they are needed by the installed applications, verify that the **Start components as needed** option is selected in the configuration settings for each of the cluster members before you start the cluster. Selecting this option can improve cluster startup time, and reduce the memory footprint of the cluster members. Starting components as they are needed is most effective if all of the applications that are deployed on the cluster are of the same type. For

example, using this option works better if all of your applications are Web applications that use servlets, and JavaServer Pages (JSP). This option works less effectively if your applications use servlets, JSPs and Enterprise JavaBeans (EJB).

Note: To ensure compatibility with other WebSphere products, the default setting for this option is deselected. Before selecting this option, verify that any other WebSphere products, that you are running in conjunction with this product, support this functionality.

About this task

When you request that all members of a cluster start, the cluster state changes to partially started and each server that is a member of that cluster launches, if it is not already running. After all members of the cluster are running, the cluster state becomes running.

1. In the administrative console, click **Servers > Clusters > WebSphere application server clusters >** .
2. Select the clusters whose members you want started.
3. Click **Start** or **Ripplestart**.
   - **Start** launches the server process of each member of the cluster by calling the node agent for each server to start the servers. After all servers are running, the state of the cluster changes to *running*. If the call to a node agent for a server fails, the server does not start.
   - **Ripplestart** combines stopping and starting operations. It first stops and then restarts each member of the cluster. For example, your cluster contains 3 cluster members named server_1, server_2 and server_3. When you click Ripplestart, server_1 stops and restarts, then server_2 stops and restarts, and finally server_3 stops and restarts. Use the Ripplestart option instead of manually stopping and then starting all of the application servers in the cluster.

   Note: If recently added cluster members do not start, you might not have selected **Synchronize changes with nodes** when you added the members to the cluster. To determine if this is the problem:

   a. In the administrative console click **Servers > Clusters > WebSphere application server clusters >** , select the cluster whose members did not start, and click **Stop**.
   b. Click the name of the cluster, click **OK**, and then click **Review**.
   c. Select **Synchronize changes with Nodes**, and then click **Save**.
   d. Start the cluster and verify that all of the cluster members now start.

Results

When you start the members of a cluster, you automatically enable workload management.

Stopping clusters

Use this task to stop a cluster and any application servers that are members of that cluster.

Before you begin

Windows If you use the Windows Services facility to start and stop application servers that are part of a cluster, remember that the cluster state does not always update correctly. For example, if a cluster is running and you stop a cluster member through the Services GUI, the cluster state remains as Started even though the server is no longer running.

About this task

You can stop all application servers that are members of the same cluster at the same time by stopping the cluster.

1. Click **Servers > Clusters > WebSphere application server clusters >** in the console navigation tree to access the Server Cluster page.
2. Select those clusters whose members you want stopped.
3. Click **Stop** or **Immediate Stop**.
   - **Stop** halts each server in a manner that allows the server to finish existing requests and allows failover to another member of the cluster. When the stop operation begins the cluster state changes to *partially stopped*. After all servers stop, the cluster state becomes Stopped.
   - **Immediate Stop** brings down the server quickly without regard to existing requests. The server ignores any current or pending tasks. When the stop operation begins, the cluster state changes to *partially stopped*. After all servers stop, the cluster state becomes Stopped.

What to do next

See Chapter 6, “Balancing workloads with clusters,” on page 307 for more information about the tasks you can complete with clustering.

Replicating data across application servers in a cluster

Use this task to configure a data replication domain to transfer data, objects, or events for session manager, dynamic cache, or stateful session beans. Data replication domains use the data replication service (DRS), which is an internal component that performs replication services, including replicating data, objects, and events among application servers.

Before you begin

Determine if you are using a multi-broker replication domain. If you configured a data replication domain with a previous version of the product, you might be using a multi-broker replication domain. Any replication domains that you create with the current version of the product are data replication domains. You should migrate any multi-broker replication domains to data replication domains.

About this task

Use this task to configure *replication*, a service that transfers data, objects, or events among the application servers in a cluster. Use replication to prevent loss of session data with session manager, to further improve the performance of the dynamic cache service, and to provide failover in stateful session beans.

Note: If you select the **Configure HTTP memory-to-memory replication** option when you create a cluster, the replication domain is automatically created for you.

Similarly if, instead of WAS01Network , the cell name is simply WAS1, you have to pad the high level qualifier with the first three characters of the string DRSSTREAM. The high level qualifier then becomes WAS1DRS.

Complete the following steps to define two logstreams for an application on which DRS uses RLS for data replication.

1. Create a replication domain. Use one of the following methods to create a replication domain:
   - **Create a replication domain manually.**

     To create a replication domain manually without creating a new cluster, click **Environment > Replication domains > New** in the administrative console.

     On this page you can specify the properties for the replication domain, including timeout, encryption, and number of replicas.
   - **Create a replication domain when you create a cluster.**

To create a replication domain when you create a cluster, click **Servers > Clusters > Clusters > New** in the administrative console. Then click **Configure HTTP memory-to-memory replication**. The replication domain that is created has the same name as the cluster and has the default settings for a replication domain. The default settings for a replication domain are to create a single replica of each piece of data and to have encryption disabled. To modify the replication domain properties, click **Environment > Replication domains > New** *replication_domain_name* in the administrative console.

2. Configure the consumers, or the components that use the replication domains. Dynamic cache, session manager, and stateful session beans are the three types of replication domain consumers. Each type of consumer must be configured with a different replication domain. For example, session manager uses one replication domain and dynamic cache uses a different replication domain. However, use one replication domain if you are configuring HTTP session memory-to-memory replication and stateful session bean replication. Using one replication domain in this case ensures that the backup state information of HTTP sessions and stateful session beans are on the same application servers.
3. Determine whether your configuration requires additional thread resources.

   The replication service uses threads obtained from the Default thread pool for various tasks, including processing messages. Other application server components also use this thread pool. Therefore, during application server startup the default maximum thread pool size of 20 might not be sufficient to allow the replication service to obtain enough threads from the pool to process all of the incoming replication messages. The number of incoming messages is influenced by the number of application servers in the domain and the number of replication domain consumers on each application server. The number of messages to be processed increases as the number of application servers in the domain increases and/or the number of replication consumers increases.

   Persistent data not being replicated to the application servers during server startup might be an indication that you need to increase the setting for the maximum thread pool size. In larger configurations, doubling the maximum size of the Default thread pool to 40 is usually sufficient. However, if the number of application servers in a replication domain is greater ten and the number of replication domain consumers in each application server is greater than two, it might have to set the maximum thread pool size to a value greater than 40.

Results

Data is replicating among the application servers in a configured replication domain.

What to do next

If you select DES or 3DES as the encryption type for a replication domain, an encryption key is used for the encryption of messages. At regular intervals, for example once a month, you should go to the **Environment > Replication domains > New** page in the administrative console, and click **Regenerate encryption key** to regenerate the key. After the key is regenerated, you must restart all of the application servers that are configured as part of the replication domain. Periodically regenerating the key improves data security.

Replication

Replication is a service that transfers data, objects, or events among application servers. Data replication service (DRS) is the internal WebSphere Application Server component that replicates data.

Use data replication to make data for session manager, dynamic cache, and stateful session beans available across many application servers in a cluster. The benefits of using replication vary depending on the component that you configure to use replication.

- Session manager uses the data replication service when configured to do memory-to-memory replication. When memory-to-memory replication is configured, session manager maintains data about

sessions across multiple application servers, preventing the loss of session data if a single application server fails. For more information about memory-to-memory replication, see the *Administering applications and their environment* PDF.

- Dynamic cache uses the data replication service to further improve performance by copying cache information across application servers in the cluster, preventing the need to repeatedly perform the same tasks and queries in different application servers. For more information about replication in the dynamic cache, see the *Administering applications and their environment* PDF.
- Stateful session beans use the replication service so that applications using stateful session beans are not limited by unexpected server failures. For more information about stateful session bean failover, see the *Developing and deploying applications* PDF.

You can define the number of *replicas* that DRS creates on remote application servers. A replica is a copy of the data that copies from one application server to another. The number of replicas that you configure affects the performance of your configuration. Smaller numbers of replicas result in better performance because the data does not have to copy many times. However, if you create more replicas, you have more redundancy in your system. By configuring more replicas, your system becomes more tolerant to possible failures of application servers in the system because the data is backed up in several locations.

Defining a single replica configuration helps you to avoid a single point of failure in the system. However, if your system must be tolerant to more failure, introduce extra redundancy in the system. Increase the number of replicas that you create for any HTTP session that is replicated with DRS. The **Number of replicas** property for any replication domain that is used by the dynamic cache service must be set to `Entire domain`.

Session manager, dynamic cache, and stateful session beans are the three *consumers* of replication. A consumer is a component that uses the replication service. When you configure replication, the same types of consumers belong to the same *replication domain*. For example, if you are configuring both session manager and dynamic cache to use DRS to replicate objects, create separate replication domains for each consumer. Create one replication domain for all the session managers on all the application servers and one replication domain for the dynamic cache on all the application servers. The only exception to this rule is to create one replication domain if you are configuring replication for HTTP sessions and stateful session beans. Configuring one replication domain in this case ensures that the backup state information is located on the same backup application servers.

See the *Administering applications and their environment* PDF for more information on how to configure replication.

Replication domain collection

Use this page to view the configured replication domains that are used for replication by the HTTP session manager, dynamic cache service, and stateful session bean failover components. All components that need to share information must be in the same replication domain. Data replication domains replace multi-broker replication domains that were available for replication in prior releases. Migrated application servers use multi-broker replication domains which are collections of replicators. You should migrate any multi-broker replication domains to be data replication domains.

To view this administrative console page, click **Environment > Replication domains**.

Name

Specifies a name for the replication domain. The name of the replication domain must be unique within the cell.

Domain type

Following are the two types of replication domains:

Multi-broker domain	Specifies a replication domain that was created with a previous version of WebSphere Application Server. This type of replication domain consists of replicator entries. Support of this type of domain remains for backward compatibility, but is deprecated. Multi-broker and data replication domains do not communicate with each other, so migrate any multi-broker replication domains to the new data replication domains. You cannot create a multi-broker domain or replicator entries in the administrative console after the deployment manager is upgraded to the current version of WebSphere Application Server.
Data replication domain	Specifies a replication domain created with the latest version of WebSphere Application Server. If the deployment manager has been upgraded to the latest version of WebSphere Application Server, you can create data replication domains only. With the data replication domain, you can specify a number of replicas instead of statically partitioning your replication settings. Specify a data replication domain for each consumer of the domain, for example, two separate domains for dynamic cache and session manager.

Data replication domain settings

Use this page to configure a data replication domain. Use data replication domains to transfer data, objects, or events for session manager, dynamic cache, or stateful session beans among the application servers in a cluster.

To view this administrative console page, click **Environment > Replication domains >** *replication_domain_name*.

Name:

Specifies a name for the replication domain. The name must be unique within the cell.

Request timeout:

Specifies how long a replication domain consumer waits when requesting information from another replication domain consumer before it gives up and assumes the information does not exist.

Units seconds
Default 5 seconds

Encryption type:

Specifies the type of encryption to use when transferring replicated data to another area of the network. Select NONE if you don't want to use encryption, DES if you want to use data encryption standard, or 3DES if you want to use triple DES. The default is NONE. The DES and 3DES options encrypt data sent between application server processes (for example, session manager and dynamic caching). Encrypting data improves the security of the network that joins the processes.

If you select DES or 3DES, after you click **Apply** or **OK**, a key for global data replication is generated. At regular intervals, for example once each month, you should navigate to this page in the administrative console and click **Regenerate encryption key** to regenerate this key. Periodically regenerating the key enhances security.

Data type	String
Default	NONE

Number of replicas:

Specifies the number of replicas that are created for every entry or piece of data that is replicated in the replication domain.

Single replica	When you select this option, every HTTP session is replicated to exactly one other application server. This is the default value.
Entire domain	When you select this option, each object is replicated to every application server that is configured as a user of the replication domain.
Specify	When you select this option, you must specify, in the Number of replicas field, the number of replicas that you want created for each HTTP session.

Migrating servers from multi-broker replication domains to data replication domains

You can migrate multi-broker replication domains to data replication domains. Any multi-broker domains that exist in your application server environment were created with a previous version of the product.

Before you begin

Determine if the application server configuration you are migrating:

1. Uses an instance of data replication service in peer-to-peer mode or in client/server mode.

   Before you begin migrating a client/server mode replication domain, consider if migrating your replication domains might cause a single point of failure. Because you migrate the servers to the new type of replication domain one at a time, you risk a single point of failure if there are 3 or fewer application servers. Before migrating, configure at least 4 servers that use multi-broker replication domains. Perform the following steps to migrate the multi-broker domains to data replication domains:

   Dynamic cache replication domains use the peer-to-peer topology.
2. Uses HTTP session memory-to-memory replication that is overloaded at the application or web module level.

   If the application server configuration you are migrating uses HTTP session memory-to-memory replication that is overloaded at the application or web module level, you must upgrade your deployment manager to the current version of the product before you start the migration process.

About this task

For HTTP session affinity to continue working correctly when migrating Version 5.x application servers to Version 7.0 application servers, you must upgrade all of the Web server plug-ins for the product to the latest version before upgrading the application servers that perform replication.

After you upgrade your deployment manager to the latest version of the product, you can only create data replication domains. Any multi-broker domains that you created with a previous version of the product are still functional, however, you cannot use the administrative console to create new multi-broker domains or replicators.

The different versions of application servers cannot communicate with each other. When migrating your servers to the current version of the product, keep at least two application servers running on the previous version so that replication remains functional.

Make sure that all of your application servers that are using this multi-broker domain have been migrated to the current version of the product before you start to migrate any multi-broker domains that exist in your configuration.

To migrate the multi-broker domains that exist in your configuration:

1. Migrate two or more of your existing servers to the current version of the product. The remaining servers on the previous version of the product can still communicate with each other, but not with the migrated servers. The migrated servers can also communicate with each other.
2. In the administrative console, create an empty data replication domain. Click **Environment > > Replication domains > New** to create an empty data replication domain.
3. Add two of your migrated servers to the new data replication domain.

   For example, if you are migrated four servers, only add two of them to the new replication domain.
4. Configure the two servers as consumers of the replication domain.

   Configuring the servers as consumers of the replication domain enables them to use the new domain to share data.
5. Add some of the clients to the new data replication domain.

   Perform this step only if the application server configuration you are migrating uses an instance of data replication service in client/server mode.
6. Configure these clients as consumers of the replication domain.
7. Verify that the new data replication domain are successfully sharing data.

   Only the servers and clients that are added to the data replication domain and are configured as consumers of this domain can use the data replication domain functions.
8. Add the rest of your migrated servers to the new data replication domain.

   When the servers can use the new data replication domain to successfully share data, migrate the rest of the servers that are using the multi-broker replication domain to the new data replication domain.

   For example, if you are migrated four servers, add the remaining two servers to the new replication domain.
9. Configure these servers as consumers of the replication domain.
10. Add the rest of the clients to the new data replication domain.

    Perform this step only if the application server configuration you are migrating uses an instance of data replication service in client/server mode.
11. Configure these clients as consumers of the replication domain.
12. Restart all of the application servers and clients.
13. Delete the empty multi-broker replication domain.

What to do next

During this process, you might lose existing sessions. However, the application remains active through the entire process, so users do not experience down time during the migration. Create a new replication domain for each type of consumer. For example, create one replication domain for the session manager and another replication domain for dynamic cache.

Data replication domains

Data replication domains and multi-broker domains both perform the same function, which is to provide data replication between application servers in a cluster. Even though you can still configure existing multi-broker domains with the current version of the product, after you upgrade your deployment manager, you can only create data replication domains in the administrative console.

Note: A replication domain that was created with a previous version of the product might be a multi-broker domain. You should migrate these multi-broker domains to data replication domains. Data replication domains enable you to:

- Configure all of the instances of replication that need to communicate in the same replication domain.
- Configure the session manager with both types of replication domains to use topologies such as peer-to-peer, and client-to-server to isolate the function of creating and storing replicas on separate application servers.
- Control the redundancy of replication for each type of replication domain, because with a data replication domain, you can specify a specific number of replicas.

If you used multi-broker domains with earlier releases of the product, use the following comparison chart to learn the differences between how Version 5.x and Version 6.x and Version 7.0 application servers use the two types of replication domains:

	Version 5.x application servers using replication domains	**V6.x and Version 7.0 application servers using replication domains**
Replication domain types	Uses only multi-broker replication domains for replication.	Servers that are using the current version of the product can be configured to use both multi-broker replication domains and data replication domains for replication. The two types of domains provide backward compatibility with multi-broker domains that were created with a Version 5.x server. You should migrate any multi-broker domains to data replication domains.
Data transport method	Uses multi-broker domain objects that contain configuration information for the internal Java Message Service (JMS) provider, which uses JMS brokers as replicators.	Uses data replication domain objects that contain configuration information to configure the high availability framework on the product. The transport is no longer based on the JMS API. Therefore, no replicators and no JMS brokers exist. You do not have to perform the complex task of configuring local, remote, and alternate replicators. The earlier version of the product did not support data replication domains. The current version of the product can be configured to perform replication using old multi-broker domains by ignoring any JMS-specific configuration and by using the other parameters to configure replication through the high availability framework.

	Version 5.x application servers using replication domains	V6.x and Version 7.0 application servers using replication domains
Replication domain configuration	The earlier version of the product encourages the sharing of replication domains between different consumers, such as session manager and dynamic cache.	The current version of the product encourages creating a separate replication domain for each consumer. For example, create one replication domain for session manager and another replication domain for dynamic cache. The only situation where you should configure one replication domain is when configuring session manager replication and stateful session bean failover. Using one replication domain in this case ensures that the backup state information of HTTP sessions and stateful session beans are on the same application servers.
Partial partitioning	You can configure partial partitioning. Partition the replication domain to filter the number of processes to send data.	Partial partitioning is deprecated. When using data replication domains, you can specify a specific number of replicas for each entry. However, if you specify a number of replicas larger than the number of backup application servers that are running, the number of replicas is the number of application servers that are running. After the number of application servers increases above your configured number of replicas, the number of replicas that are created is equal to the number that you specified.
Domain sharing	Multiple data replication service (DRS) instances share multi-broker domains. A limitation exists on the number of multi-broker domains that you can create because every multi-broker domain contains at least one replicator. A maximum of one replicator can be on each application server.	All DRS instances in a replication domain use the same mode. Each replication domain must contain either client only and server only instances, or client and server instances only. For example, if one instance is configured to client and server, all other instances must be client and server. If one instance in a replication domain is configured to be a client only, you can add client only and server only instances, but not a client and server instance.

Deleting replication domains

You can manually create a replication domain, or have a replication domain automatically generated when you create a cluster. When you no longer need a previously defined replication domain, you can use either the wsadmin AdminConfig delete command or the administrative console to delete the replication domain from your application server environment. Deleting a cluster does not automatically delete a replication domain that is associated with that cluster.

Before you begin

- Verify that none of the applications that you are running require the replication domain you are deleting.

- Verify that neither dynamic caching nor the session manager are configured to use the replication domain that you are deleting. Deleting a replication domain that either dynamic caching or the session manager has been configured to use might cause processing errors.

About this task

Perform the following steps if you want to use the administrative console to delete a replication domain.

1. In the administrative console. click **Environment > > Replication domains**.
2. Select the replication domain that you want to delete.
3. Click **Delete**.
4. Select Synchronize changes with Nodes, and then click **Save** to save your workspace changes to the master configuration.

Results

Data is not replicated amongst the application servers that were part of the configured replication domain that you deleted, and all session data that is associated with the deleted replication domain is deleted.

Replicating data with a multi-broker replication domain

Use this task to mange replication domains that you migrated from a Version 5.x product environment.

Before you begin

Note: Multi-broker replication domains are not created in Version 7.0 product environments. However they can be migrated from existing Version 5.x product environments. If you migrate Version 5.x multi-broker replication domains, you can use the Multi-broker domain panel in the Version 7.0 administrative console to managed these domains.

Although you can manage migrated multi-broker domains with the current version of the product, after you upgrade your deployment manager, you can create only data replication domains in the administrative console. Consider migrating any existing multi-broker domains to the new data replication domains.

About this task

If you are performing this task, it is assumed that you configured replication with a previous version of the product, and defined replication domains that list connected replicator entries, residing in managed servers in the cell,) that can exchange data. You can manage these existing replication domains and replicator entries, but you cannot create new multi-broker replication domains or new replicator entries in the administrative console.

A replicator does not need to run in the same process as the application server that uses it. However, it might be easier to manage replicators and replication domains if a one-to-one relationship exists between replicators and application servers. During configuration, you can select the local replicator as the default replicator.

1. Manage multi-broker replication domain configuration settings. In the administrative console, click **Environment > Replication domains**.
2. Click **Multi-broker domain >***multi-broker domain_name*, and update the values for that particular multi-broker replication domain. The default values are generally sufficient, especially for the pooling and timeout properties.
   a. Name the replication domain.
   b. Specify the timeout interval.
   c. Specify the encryption type. The DES and TRIPLE_DES options encrypt data sent between application server processes and better secure the network joining the processes.

d. Partition the replication domain to filter the number of processes to which data is sent. Partitioning the replication domain is most often done if you are replicating data to support retrieval of an HTTP session if the process maintaining the HTTP session fails. Partitioning is not supported for sharing of cached data that is maintained by Web container dynamic caching.

e. Specify whether you want a single replication of data to be made. Enable the option if you are replicating data to support retrieval of an HTTP session if the process maintaining the HTTP session fails.

f. Specify whether processes should receive data in objects or bytes. Processes receiving data in objects receive the data and class definitions. Processes receiving data in bytes receive the data only.

g. Configure a pool of replication resources. Pooling replication resources can enhance the performance of the replication service.

3. Maintain the replicators that you have already defined. You cannot create any new replicators. The default convention is to define a replicator in each application server that uses replication. However, you can define a pool of replicators, separate from the servers hosting applications.

   a. In the administrative console, click **Environment > Replication domains** *replication_domain_name* **> Replicator entries >** *replicator_entry_name*.

   b. Specify a replicator name and select a server available within the cell to which you can assign a replicator. Also specify a host name and ports. Note that a replicator has two ports (replicator and client ports) that use the same host name but have different ports.

4. Click **OK** to save your changes.

What to do next

If you use the DES or TRIPLE_DES encryption type for a replicator, at regular intervals, such as monthly, you should click **Regenerate encryption key** on the Replication domains settings page. Periodically changing the encryption key enhances security.

Multi-broker replication domains

A multi-broker replication domain is a collection of replication entries, or *replicator* instances, used by clusters or individual servers within a cell. Multi-broker replication domains were created with a previous release of the product.

Note: After you upgrade your deployment manager to the latest version of the product, you can create data replication domains only. Any multi-broker domains that you created with a previous release of the product are still functional, however, you cannot create new multi-broker domains or replicator instances with the administrative console.

A replication entry, or replicator, is a run-time component that handles the transfer of internal product data. All replicators within a replication domain connect with each other, forming a network of replicators.

Components such as session manager and dynamic cache can connect to any replicator within a domain to receive data from their peer components on other application servers that are connected other replicators in the same domain. If the replicator that a component is connected to fails, the component automatically attempts to reconnect to another replicator in the domain and recover any data that was missed while the component was not connected to a replicator.

The default is to define a replication domain for a cluster when creating the cluster. However, replication domains can span across clusters.

Global default settings apply to a given replication domain across a cell. Most default settings tune and control the behavior of replicator entries that are in managed servers across the cell. Such default settings

control the use of encryption or the serialization and transferring of objects. Some default settings tune and control how specific product functions, such as session manager and dynamic caching, leverage replication, such as session use of partitions.

For situations that require settings values other than the default, change the values for a given replication domain. Settings include various resource allocation, replication strategies, such as grouping or partitioning, and methods, as well as some security related items.

If you are using replication for HTTP session failover, you might also need to filter where the session replicates. For example, only replicate to two places out of many. The global default settings define the partition size or number of groups and the session manager settings define the groups to which a particular instance belongs.

Filtering is less important if you are using replication to distribute information on data that is no longer valid and actual cached data maintained by dynamic caching. Replication does not occur for failover as much as for data synchronization across a cluster or cell when you likely want to avoid expensive costs for generating data potentially needed across those various servers.

Note that you can filter or segment by using multiple replication domains.

Multi-broker replication domain settings

Use this page to configure a multi-broker replication domain. This administrative console page applies only to replication domains that were created with a previous version of the product. Replication domains use the data replication service (DRS).

To view this administrative console page, click **Environment > Replication domains >** *multibroker_replication_domain_name*.

An application server that is connected to a replicator within a domain can access the same set of data sent out by any application server connected to any other replicator, including the same replicator. Data is not shared across replication domains.

Name:

Specifies a name for the replication domain. The name must be unique within the cell.

Request timeout:

Specifies the number of seconds that a replication domain consumer waits when requesting information from another replication domain consumer before giving up and assuming the information does not exist. The default is 5 seconds.

Data type	Integer
Units	Seconds
Default	5

Encryption type:

Specifies the type of encryption used before the object transfers over the network. The options include NONE, DES, TRIPLE_DES. The default is NONE. The DES and TRIPLE_DES options encrypt data sent between application server processes and secure the network joining the processes.

If you specify DES or TRIPLE_DES, a key for global data replication is generated after you click **Apply** or **OK**. When you use the DES or TRIPLE_DES encryption type, click **Regenerate encryption key** at regular intervals such as monthly because periodically changing the key enhances security.

DRS partition size:

Specifies the number of groups into which a replication domain is partitioned. By default, data sent by an application server process to a replication domain is transferred to all other application server processes connected to that replication domain. To filter or reduce the number of destinations for the data being sent, partition the replication domain. There should be at least one server listening to every partition. If there are no servers listening on a partition, all the replicas created in that partition are lost because there is no server to cache the objects. The default partition size is 10, and the partition size should be 10 or more to enhance performance.

Partitioning the replication domain is only applicable if you are replicating data to support retrieval of an HTTP session if the process maintaining the HTTP session fails. Partitioning is not supported for sharing of cached data maintained by Web container dynamic caching. As to dynamic caching, all partitions or groups are always active and used for data replication.

When you partition a replication domain, you define the total number of groups or partitions. Use this setting to define the number of groups. Then, when you configure a specific session manager under a Web container or as part of an enterprise application or Web module, select the partition to which that session manager instance listens and from which it accepts data. To specify the groups to which an application server listens, change the settings for affected servers on a session manager page. In addition, you can set a role or runtime mode for a server. This role or mode affects whether an application server process sends data to the replication domain, receives data, or does both. The default is both to receive and send data.

Data type	Integer
Default	10

Single replica:

Specifies that a single replication of data is made. Use this option only if you are using session manager with memory to memory replication. Enable this option if you are replicating data to support retrieval of an HTTP session if the process maintaining the HTTP session fails. This option restricts the recipient of the data to a single instance.

Note: Do not enable this option on a domain that is using dynamic cache replication.
This setting provides filtering beyond grouping or partitioning. Using this setting, you can choose to have data only sent to one other listening instance in the replication domain.

Default	false

Serialization method:

Specifies the object serialization method to use when replicating data. An administrative concern with replicating Java objects is locating the class definition, especially in a Java Platform, Enterprise Edition (Java EE) environment where class definitions might reside only in certain web modules or enterprise applications. Object serialization methods define whether the processes receiving data also need the class definition.

The options for this setting are OBJECT and BYTES. The default is BYTES.

OBJECT instructs a replicator to write the object directly to the stream. With OBJECT, a replicator must instantiate the object on the receiving side so it must have the class definition.

BYTES instructs a replicator to break down the object into bytes and then send only the bytes across the stream. With BYTES, a replicator does not need to instantiate the object on the receiving side. The

BYTES option is useful for failover, where the data is not used at the receiving side and the class definitions do not need to be stored on the receiving side. Or, the option requires that you move class definitions from the Web application class path to the system class path.

DRS pool size:

Specifies the size of the pool of resources allocated for communication with its Java Message Service (JMS) transport. You must configure this number to be the same as the DRS partition size. The default is 10.

Pooling replication resources can enhance the performance of the internal data replication service.

DRS pool connections:

Specifies that the domain replication service should create a pool of connections with its Java Message Service (JMS) transport rather than reusing a single connection. You can pool connections when using a single replica or client server environment. You should not pool connections in a peer to peer environment.

The default is to not create a pool of connections for replication.

Replicator entry collection:

Use this page to view and manage replicator entries. Replicator entries are for use only with multi-broker replication domains. Each multi-broker replication domain consists of one or more replicator entries.

To view this administrative console page, click **Environment > Replication domains >** *replication_domain_name* **> Replicator entries**.

Replicator entries are only valid for multi-broker domains, which are replication domains created with a previous version of the product. When you migrate your deployment manager to the current version of the product, you are no longer be able to create new replicator entries in the administrative console. You can only view and modify settings for replicator entries that were created with the previous version of the product.

Replicator name:

Specifies a name for the replicator entry.

Replicator entry settings:

Use this page to view and configure a replicator entry, or *replicator*. Replicators are used with multi-broker replication domains.

To view this administrative console page, click **Environment > Replication domains >** *replication_domain_name* **> Replicator entries >** *replicator_entry_name*.

Replicators communicate using Transmission Control Protocol/Internet Protocol (TCP/IP). Therefore, you must allocate an IP address and ports for replicators. Use this page to name a replicator and then to allocate an IP address and two ports (replicator and client ports) for the replicator.

Replicator name:

Specifies a name for the replicator entry.

Server:

Specifies the server for which you are defining a replicator. You can view the names of servers that do not already have replicators. You can create a maximum of one replicator on any application server.

Replicator and client host name:

Specifies the IP address, domain name service (DNS) host name with domain name suffix, or just the DNS host name, used by a client to request a Web application resource (such as a servlet, JavaServer Pages (JSP) file, or HTML page).

A replicator port and client port share the same host name.

Replicator Port:

Specifies the port for which the replicator is configured to accept messages from other replicators. The port value is used in conjunction with the host name.

The replicator port enables communication among replicators. It provides replicator port to replicator communication. The usual value specified is 7874.

Client Port:

Specifies the port for which the Web server is configured to accept client requests. The port value is used in conjunction with the host name.

The client port enables communication between an application server process and a replicator. It provides client port to application server communication. The usual value specified is 7873.

Deleting clusters

Use this task to remove a cluster and all of its cluster members.

Before you begin

Removing a cluster deletes the cluster and all associated cluster members. When you delete a cluster, there is no option to keep certain cluster members or applications that you have installed on any part of the cluster.

Note: If the cluster you are removing has applications or modules mapped to it, remap the modules to another cluster, or create a new cluster and remap the modules to the new cluster, before removing the old cluster. After a cluster to which modules are mapped is deleted, the modules cannot be remapped to another cluster. Therefore, if you do not remap the modules to another cluster before deleting the old one, you must uninstall all of the modules that were mapped to the old cluster, and then reinstall them on a different cluster.

1. In the administrative console, click **Servers > Clusters > WebSphere application server clusters**.
2. Make sure the cluster you want to remove is stopped.
   If the cluster is started, stop the cluster.
3. Delete the cluster. Select the cluster you want to delete, and click **Delete**.
4. Click **OK** and then click **Review** to preview your changes.
5. Select **Synchronize changes with Nodes**, and then click **Save** to save your changes.

Results

The cluster and all of the cluster members are deleted.

Deleting specific cluster members

Use this task to remove a cluster member from an existing cluster. Removing a cluster member deletes the associated application server.

About this task

You must delete an application server to remove it from a cluster.

If, in the administrative console, you select **Include cluster members in the collection** as one of your console page preferences for the Application servers page, you can use either the Application servers page or the Cluster members page to delete an application server.

To use the Cluster members page to remove an application server from a cluster:

1. In the administrative console, click **Servers > Clusters > WebSphere application server clusters >** .
2. Click the name of the cluster that contains the cluster member that you are removing from the cluster, and then click **Cluster members**.
3. Check the status of the cluster member that you are removing. If the cluster member is started, select the cluster member, and click **Stop**, and then view the Status of this cluster member again, along with any messages or logs to make sure the cluster member stops. You cannot remove a cluster member while it is running.
4. Delete the cluster member. Select the cluster member you want to delete, and click **Delete**.
5. Click **OK** and then click **Review** to preview your changes.
6. Select **Synchronize changes with Nodes**, and then click **Save** to save your changes.

Results

The cluster member is deleted.

Tuning a workload management configuration

You can set values for several workload management client properties to tune the behavior of the workload management runtime.

About this task

You set the properties as command-line arguments for the Java virtual machine (JVM) process in which the workload management client is running.

Caution: Set the values of these properties only in response to problems that you encounter. In most cases, you do not need to change the values. If workload management is functioning correctly, changing the values can produce undesirable results.

To change the property values, you can use the Java virtual machine page of the administrative console or wsadmin commands. In cases such as where a servlet is a client to an enterprise bean, use the administrative console page for the application server where the servlet is running to configure the properties. The steps below describe how to change the values using the console.

1. Access the Java Virtual Machine page.

   Servers > Server Types > WebSphere application servers > *server_name* **> Java and process management > Process definition > Java virtual machine**
2. On the Java Virtual Machine page, specify one or more of the following command-line arguments in the Generic JVM arguments field:
   - **-Dcom.ibm.CORBA.RequestTimeout=*timeout_interval***

If your application is experiencing problems with timeouts, this argument changes the value for the com.ibm.CORBA.RequestTimeout property, which specifies the timeout period for responding to requests sent from the client. This argument uses the -D option. *timeout_interval* is the timeout period in seconds. If your network experiences extreme latency, specify a large value to prevent timeouts. If you specify a value that is too small, an application server that participates in workload management can time out before it receives a response.

Note: Be careful specifying this property; it has no recommended value. Set it only if your application is experiencing problems with timeouts.

- **-Dcom.ibm.websphere.wlm.unusable.interval=*interval***

  If the workload management state of the client is refreshing too soon or too late, this argument changes the value for the com.ibm.websphere.wlm.unusable.interval property, which specifies the time interval that the workload management client runtime waits after it marks a server as unavailable before it attempts to contact the server again. This argument uses the -D option. *interval* is the time in seconds between attempts. The default value is 300 seconds. If the property is set to a large value, the server is marked as unavailable for a long period of time. This prevents the workload management refresh protocol from refreshing the workload management state of the client until after the time period has ended.

3. Click **OK** and **Save** to save your configuration changes.
4. Stop the application server and then restart the application server.

Workload management runtime exceptions

The product client can catch workload management runtime exceptions and implement strategies to handle the situation. For example, it can display an error message if no servers are available.

The workload management service might create the following exceptions if it encounters problems:

org.omg.CORBA.TRANSIENT with a minor code 1229066306 (0x40421042)
: This exception is created if the workload management routing service cannot retry a request and the failure resulted from a connection error. This exception indicates that the application should invoke some compensation logic and resubmit the request.

org.omg.CORBA.NO_IMPLEMENT with a minor code 1229066304 (0x49421040)
: This exception is created if the workload management service cannot contact any of the Enterprise JavaBeans (EJB) application servers that participate in workload management.

The workload management routing service can reroute a failed request to a different target transparently to the application if the application will not be adversely affected by a second attempt. Currently, the only way is to check if the request did not run in whole or part on the previous attempt. When a request runs in whole or in part, an *org.omg.CORBA.TRANSIENT with the minor code 1229066306 (0x49421042)* exception is created to signal that a request can be made again. This informs the application that another target might be available to satisfy the request, but the request could not be failed over transparently to the application. Thus, the application can resubmit the request. The routing service creates an *org.omg.CORBA.NO_IMPLEMENT with the minor code 1229066304 (0x49421040)* exception if it cannot locate a suitable target for the request. The exception is created, for example, if the cluster is stopped or if the application does not have a path to any of the cluster members.

Related tasks

Chapter 6, “Balancing workloads with clusters,” on page 307
You should use server clusters and cluster members to monitor and manage the workloads of application servers.

Appendix. Directory conventions

References in product information to *app_server_root*, *profile_root*, and other directories infer specific default directory locations. This topic describes the conventions in use for WebSphere Application Server.

Default product locations (distributed)

The following file paths are default locations. You can install the product and other components or create profiles in any directory where you have write access. Multiple installations of WebSphere Application Server Network Deployment products or components require multiple locations. Default values for installation actions by root and non-root users are given. If no non-root values are specified, then the default directory values are applicable to both root and non-root users.

app_client_root

The following list shows default installation root directories for the WebSphere Application Client.

User	Directory
Root	AIX /usr/IBM/WebSphere/AppClient (Java EE Application client only) HP-UX Linux Solaris /opt/IBM/WebSphere/AppClient (Java EE Application client only) Windows C:\Program Files\IBM\WebSphere\AppClient
Non-root	AIX HP-UX Linux Solaris *user_home*/IBM/WebSphere/AppServer/AppClient (Java EE Application client only) Windows C:\IBM\WebSphere\AppClient

app_server_root

The following list shows the default installation directories for WebSphere Application Server Network Deployment.

User	Directory
Root	AIX /usr/IBM/WebSphere/AppServer HP-UX Linux Solaris /opt/IBM/WebSphere/AppServer Windows C:\Program Files\IBM\WebSphere\AppServer
Non-root	AIX HP-UX Linux Solaris *user_home*/IBM/WebSphere/AppServer Windows C:\IBM\WebSphere\AppServer

cip_app_server_root

A *customized installation package* (CIP) is an installation package created with IBM WebSphere Installation Factory that contains a WebSphere Application Server Network Deployment product bundled with one or more maintenance packages, an optional configuration archive, one or more optional enterprise archive files, and other optional files and scripts.

The following list shows the default installation root directories for a CIP where *cip_uid* is the CIP unique ID generated during creation of the build definition file.

User	Directory
Root	AIX /usr/IBM/WebSphere/AppServer/cip/*cip_uid* HP-UX Linux Solaris /opt/IBM/WebSphere/AppServer/cip/*cip_uid* Windows C:\Program Files\IBM\WebSphere\AppServer\cip*cip_uid*
Non-root	AIX HP-UX Linux Solaris *user_home*/IBM/WebSphere/AppServer/cip/*cip_uid* Windows C:\IBM\WebSphere\AppServer\cip*cip_uid*

component_root
The component installation root directory is any installation root directory described in this topic. Some programs are for use across multiple components. In particular, the Update Installer for WebSphere Software is for use with WebSphere Application Server Network Deployment, Web server plug-ins, the Application Client, and the IBM HTTP Server. All of these components are part of the product package.

gskit_root
IBM Global Security Kit (GSKit) can now be installed by any user. GSKit is installed locally inside the installing product's directory structure and is no longer installed in a global location on the target system. The following list shows the default installation root directory for Version 7 of the GSKit, where *product_root* is the root directory of the product that is installing GSKit, for example IBM HTTP Server or the Web server plug-in.

Directory
AIX HP-UX Linux Solaris *product_root*/gsk7 Windows *product_root*\gsk7

if_root This directory represents the root directory of the IBM WebSphere Installation Factory. Because you can download and unpack the Installation Factory to any directory on the file system to which you have write access, this directory's location varies by user. IBM WebSphere Installation Factory is an Eclipse-based tool which creates installation packages for installing WebSphere Application Server in a reliable and repeatable way, tailored to your specific needs.

iip_root
This directory represents the root directory of an *integrated installation package* (IIP) produced by theIBM WebSphere Installation Factory. Because you can create and save an IIP to any directory on the file system to which you have write access, this directory's location varies by user. An IIP is an aggregated installation package that can include one or more generally available installation packages, one or more customized installation packages (CIPs), and other user-specified files and directories.

profile_root
The following list shows the default directory for a profile named *profile_name* on each distributed operating system.

User	Directory
Root	AIX /usr/IBM/WebSphere/AppServer/profiles/*profile_name* HP-UX Linux Solaris /opt/IBM/WebSphere/AppServer/profiles/*profile_name* Windows C:\Program Files\IBM\WebSphere\AppServer\profiles*profile_name*

User	Directory
Non-root	AIX HP-UX Linux Solaris *user_home*/IBM/WebSphere/AppServer/profiles/ Windows C:\IBM\WebSphere\AppServer\profiles\

plugins_root

The following default installation root is for the Web server plug-ins for WebSphere Application Server.

User	Directory
Root	AIX /usr/IBM/WebSphere/Plugins HP-UX Linux Solaris /opt/IBM/WebSphere/Plugins Windows C:\Program Files\IBM\WebSphere\Plugins
Non-root	AIX HP-UX Linux Solaris *user_home*/IBM/WebSphere/Plugins Windows C:\IBM\WebSphere\Plugins

updi_root

The following list shows the default installation root directories for the Update Installer for WebSphere Software.

User	Directory
Root	AIX /usr/IBM/WebSphere/UpdateInstaller HP-UX Linux Solaris /opt/IBM/WebSphere/UpdateInstaller Windows C:\Program Files\IBM\WebSphere\UpdateInstaller
Non-root	AIX HP-UX Linux Solaris *user_home*/IBM/WebSphere/UpdateInstaller Windows C:\IBM\WebSphere\UpdateInstaller

web_server_root

The following default installation root directories are for the IBM HTTP Server.

User	Directory
Root	AIX /usr/IBM/HTTPServer HP-UX Linux Solaris /opt/IBM/HTTPServer Windows C:\Program Files\IBM\HTTPServer
Non-root	AIX HP-UX Linux Solaris *user_home*/IBM/HTTPServer Windows C:\IBM\HTTPServer

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter 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 Intellectual Property & Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Trademarks and service marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. 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. For a current list of IBM trademarks, visit the IBM Copyright and trademark information Web site (www.ibm.com/legal/copytrade.shtml).

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both.

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