WebSphere® Application Server - Express 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 191.

Compilation date: September 5, 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
	Application Server	Administrative Agent	Job Manager	Secure Proxy Server	Administrative Subsystem	
Administrative Console Port (WC_ adminhost)	9060	9060	9960	----	----	serverindex.xml and virtualhosts.xml
Administrative Console Secure Port (WC_ adminhost_ secure)	9043	9043	9943	----	----	
HTTP Transport Port (WC_ defaulthost)	9080	----	----	80	----	
HTTPS Transport Secure Port (WC_ defaulthost_ secure)	9443	----	----	443	----	
Bootstrap Port (BOOTSTRAP_ ADDRESS)	2809	9807	9808	----	----	serverindex.xml
Cell Discovery Address (CELL_ DISCOVERY_ ADDRESS)	----	----	----	----	----	
CSIV2 Client Authentication Listener Port (CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS)	9402	9402	9402	----	----	
CSIV2 Server Authentication Listener Port (CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS)	9403	9403	9403	----	----	
High Availability Manager Communication Port (DCS_ UNICAST_ ADDRESS)	9353	----	----	----	----	
Internal JMS Server Port (JMSSERVER_ SECURITY_ PORT)	5557	----	----	----	----	
IPC Connector Port (IPC_ CONNECTOR_ ADDRESS)	9633	9630	9631	9633	9634	
MQ Transport Port (SIB_ MQ_ ENDPOINT_ ADDRESS)	5558	----	----	----	----	
MQ Transport Secure Port (SIB_ MQ_ ENDPOINT_ SECURE_ ADDRESS)	5578	----	----	----	----	

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

Port Name	Default Value					Files
	Application Server	Administrative Agent	Job Manager	Secure Proxy Server	Administrative Subsystem	
ORB Listener Port (ORB_ LISTENER_ ADDRESS)	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	9401	9401	----	----	
Service Integration Port (SIB_ ENDPOINT_ ADDRESS)	7276	----	----	----	----	
Service Integration Secure Port (SIB_ ENDPOINT_ SECURE_ ADDRESS)	7286	----	----	----	----	
SIP Container Port (SIP_ DEFAULTHOST)	5060	----	----	5060	----	
SIP Container Secure Port (SIP_ DEFAULTHOST_ SECURE)	5061	----	----	5061	----	
SOAP Connector Port (SOAP_ CONNECTOR_ ADDRESS)	8880	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

Version 6.1 port numbers

Table 2. Port definitions for WebSphere Application Server Version 6.1

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

Table 2. Port definitions for WebSphere Application Server Version 6.1 (continued)

Port Name	Default Value	Files
Bootstrap Port (BOOTSTRAP_ ADDRESS)	2809	serverindex.xml
CSIV2 Server Authentication Listener Port (CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS)	9403	
CSIV2 Client Authentication Listener Port (CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS)	9402	
DRS_ CLIENT_ ADDRESS **Note:** This port is deprecated and is no longer used in the current version of WebSphere Application Server.	7873	
High Availability Manager Communication Port (DCS_ UNICAST_ ADDRESS)	9353	
Internal JMS Server (JMSSERVER_ SECURITY_ PORT)	5557	
MQ Transport (SIB_ MQ_ ENDPOINT_ ADDRESS)	5558	
MQ Transport secure (SIB_ MQ_ ENDPOINT_ SECURE_ ADDRESS)	5578	
ORB Listener Port (ORB_ LISTENER_ ADDRESS)	9100	
Proxy Server Port (PROXY_ HTTP_ ADDRESS)	80	
Proxy Server Secure Port (PROXY_ HTTPS_ ADDRESS)	443	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9401	
Service Integration Port (SIB_ ENDPOINT_ ADDRESS)	7276	
Service Integration Port Secure (SIB_ ENDPOINT_ SECURE_ ADDRESS)	7286	
SIP Container Port (SIP_ DEFAULTHOST)	5060	
SIP Container Secure Port (SIP_ DEFAULTHOST_ SECURE)	5061	
SOAP Connector Port (SOAP_ CONNECTOR_ ADDRESS)	8880	
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

Version 6.0.x port numbers

Table 3. Port definitions for WebSphere Application Server Version 6.0.x

Port Name	Default Value	Files
HTTP_ TRANSPORT	9080	serverindex.xml and virtualhosts.xml
HTTP Admin Console Port (HTTP_ TRANSPORT_ ADMIN)	9060	
HTTPS Transport Port (HTTPS_ TRANSPORT)	9443	
HTTPS Admin Console Secure Port (HTTPS_ TRANSPORT_ ADMIN)	9043	

Table 3. Port definitions for WebSphere Application Server Version 6.0.x (continued)

Port Name	Default Value	Files
BOOTSTRAP_ ADDRESS	2809	serverindex.xml
SOAP_ CONNECTOR_ ADDRESS	8880	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9401	
CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	9403	
CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS	9402	
ORB_ LISTENER_ ADDRESS	9100	
DCS_ UNICAST_ ADDRESS	9353	
SIB_ ENDPOINT_ ADDRESS	7276	
SIB_ ENDPOINT_ SECURE_ ADDRESS	7286	
SIB_ MQ_ ENDPOINT_ ADDRESS	5558	
SIB_ MQ_ ENDPOINT_ SECURE_ ADDRESS	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	
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
NODE_ MULTICAST_ IPV6_ DISCOVERY_ ADDRESS	5001	serverindex.xml

Version 5.1.x port numbers

Table 4. Port definitions for WebSphere Application Server Version 5.1.x

Port Name	Default Value	Files
HTTP_ TRANSPORT	9080	server.xml and virtualhosts.xml
HTTPS_ TRANSPORT	9443	
HTTP_ TRANSPORT_ ADMIN	9090	
HTTPS_ TRANSPORT_ ADMIN	9043	
JMSSERVER_ SECURITY_ PORT	5557	server.xml
JMSSERVER_ QUEUED_ ADDRESS	5558	serverindex.xml
JMSSERVER_ DIRECT_ ADDRESS	5559	
BOOTSTRAP_ ADDRESS	2809	
SOAP_ CONNECTOR_ ADDRESS	8880	
DRS_ CLIENT_ ADDRESS	7873	
SAS_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	0	
CSIV2_ SSL_ SERVERAUTH_ LISTENER_ ADDRESS	0	
CSIV2_ SSL_ MUTUALAUTH_ LISTENER_ ADDRESS	0	
ORB_ LISTENER_ ADDRESS	0	
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

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

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.

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.

Each profile has its own administrative console and administrative scripting interface.

The installation program for WebSphere Application Server uses the **manageprofiles** command to create a profile named default.

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 can create the following type of profile:

Management profile with an administrative agent server

The basic function of the administrative agent is to provide a single interface to administer multiple 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.

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.

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 5. Space requirements

Profile or server type	Space required
Application server	200 MB
Management	30 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.

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

The installation procedure creates one profile named `default` for an application server named `server1`. You can use the Profile Management Tool or the manageprofiles command to create more application server processes. For example, a second profile can allow two different teams in a department to test independently of one another using the same machine. You can also create an administrative agent profile so that you can administer multiple application servers from a single administrative interface.

- 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 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.

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 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.

   - 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.

   The Profile creation options panel is displayed.

4. Select **Administrative agent**. Click **Next**.

   The Profile creation options panel is displayed.

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

6. 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.

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, 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.

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. 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.

12. 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.

13. 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.

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 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 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

The Installation wizard creates an application server profile with a server named server1. You can create additional profiles. Each additional profile is an application server named server1.

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.

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.

   - 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.

   The Profile creation options panel is displayed.

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

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 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:

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.

8. 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**.

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

10. 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.

11. 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.

12. 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.

13. 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.

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 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.

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, 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. For a given profile, have the same non-root user ID manage the entire profile.

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

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. 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

   ```
   chmod  -R g+wr /opt/IBM/WebSphere/AppServer/profileTemplates/default/documents
   ```

   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. For a given profile, have the same non-root user ID manage the entire profile.

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

Before you delete a profile, stop its application server to ensure that the application server can be deleted.

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

If your system uses administrative services, you can specify settings for the service.

- Use the settings page for an administrative service to configure administrative services.
- Change the host name.
- Administer multiple application servers through an administrative agent.

Administrative agent

An administrative agent provides a single interface to administer multiple application servers with standalone 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.

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 and view or change application server nodes registered to the administrative agent. An administrative agent provides a single interface to administer application servers in, for example, development, unit test, or server farm environments.

Before you begin

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 application server nodes in, development, unit test, or server farm environments, for example. 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 and view and change properties for nodes registered to the administrative agent. Read the section on planning the installation for topologies that include administrative agents.

- View and change properties for the administrative agent.

1. Click **System Administration > Administrative agent** from the navigation of the administrative agent administrative console.
   - 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*** .

     The panel is read-only. To change properties for the node, click the links under Additional Properties.

Results

Depending on the steps that you completed, you might have configured the administrative agent, or viewed or changed properties for a node registered to the administrative agent.

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**.

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.

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.

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 servers and applications. These managed bean (MBean) operations provide administrative auditing:

Table 6.

MBean type	MBean operations
Server	stop, stopImmediate

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 for the application server environment. The audit examples are found in the application server `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)
   ...
[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.

Changing host names

After creating a profile, 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 names of the server ports using the administrative console and command line tools.

Before you begin

Create a profile. 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 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, similar to the following example:

```
cells/isthmusCell16/nodes/isthmusNode06|serverindex.xml#ServerIndex_1
```

4. Modify the host name for the application server, similar to the following example:

   Enter the following line of code:

```
AdminConfig.modify('(cells/isthmusCell16/nodes/isthmusNode06|serverindex.xml
#ServerIndex_1)',  "[[hostName new_host_name]]")
```

   The command is split on multiple lines for printing purposes.

5. Verify that the host name is correct, similar to the following example:

   Enter the following line of code:

```
AdminConfig.show('(cells/isthmusCell07/nodes/isthmusCellManager07|
serverindex.xml#ServerIndex_1)', 'hostName')
```

   The response is:

```
'[hostName isthmus]'
```

   The command is 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. Update the application server with the changes.
   a. Stop the application server.

      Enter the following command:

```
stopServer server1 -profileName AppSrv01
```

   b. Restart the application server.

      Enter the following command:

```
startServer server1 -profileName AppSrv01
```

- Correct the host names for the ports that an application server 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**.
  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. Update the application server with the changes.
     a. Stop the application server.
        - Select **Servers** > **Application servers**.
        - Select the server that you want to stop.
        - Click **Stop**.
     b. Restart the application server.

        Enter the following command:

```
startServer server1 -profileName AppSrv01
```

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 tasks such as deploying the applications that you want to run on this server.

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	true|false

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 **Servers > Application 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 **Servers > Application Servers >** *server_name* **> 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.

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. 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 your server, its configuration, and its contents.

- “Hierarchy of directories of documents”
- “Changing configuration documents” on page 54
- “Transformation of configuration files” on page 55

Hierarchy of directories of documents

Changes made to the configuration documents are stored in the cell repository.

At the top of the hierarchy is the **cells** directory. It holds a subdirectory for the cell. The name of the cell subdirectory matches the name of the cell. For example, a cell named *cell1* has its configuration documents in the subdirectory *cell1*.

The subdirectories under the cell contain the entire set of documents for the node and server 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 to the node in the cell
- The **nodes** subdirectory, which holds a subdirectory for the node in the cell. The names of the nodes subdirectory matches the name of the node.

  The 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, the server on the node uses the variable definition in the node document and ignores the definition in the cell document.

  The node subdirectory holds a subdirectory for the server defined on the node. The name of the subdirectory matches the name of the server. The server subdirectory holds a `server.xml` file, which provides configuration data specific to the 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
     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 55″ 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.	

`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 application server 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 **Servers** > **Application Servers** > ***server_name*** > **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. Run the backupConfig command to back up configuration files. See the backupConfig command topic in the *Using the administrative clients* PDF for information.
2. 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.

  During the installation process the product creates a default application server, named server1. You must issue either the createApplicationServer or createGenericServer wsadmin command from a command line to create an additional application server or generic server.

  You cannot use the administrative console that is associated with the original base server to manage any additional servers that you create. You must either use command-line tools to perform these tasks for the additional servers, set up an administrative console for each server, or configure an administrative agent to provide a single interface to all of your servers, including the original base server.

  Note: If you create additional application servers, only use one server to modify and save configurations. There is no coordination of configuration setting between the different servers and if you modify and save configurations on multiple servers, your data might become corrupted.
- 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 machine. It is a configuration, rather than a live object, explaining 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 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 stand-alone application servers, 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 stand-alone application servers 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.
- Certain customization values.

The variable scoping mechanism for WebSphere variables enables you to define a variable at the node 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. 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.
3. 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**.
4. 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.
5. Save your configuration.
6. 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.

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 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 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.

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 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 7.

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 8.

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.

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.

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.
   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**.

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.

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.

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.
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, 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, 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 87 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. 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.
4. Install the application on the server.
   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

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 the application server first. After you stop the server, change the system clock, and then restart the server. If you change the system clock on one system, you must

ensure the clocks on all systems that communicate with each other, and have the product installed, are synchronized. Otherwise, you might experience errors, such as security tokens no longer being valid.

About this task

During the installation process, the product creates a default application server, named server1. If you create any additional application servers, you cannot start, stop, or manage these servers using the administrative console that is associated with the original base server. You must either use command-line tools to perform these tasks for the additional servers, set up an administrative console for each server, or configure an administrative agent to provide a single interface to all of your servers, including the original base server.

Note: If you create additional application servers, only use one server to modify and save configurations. There is no coordination of configuration setting between the different servers and if you modify and save configurations on multiple servers, your data might become corrupted.

You can perform the following steps to view and manage the default 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 monitor the default server.
2. Click **server1** to view or change the configuration settings for the default 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 selected when a new server is created. When this property is selected, server components are dynamically started as they are needed. Therefore, selecting this option can improve server startup time, and reduce the memory footprint of the 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 reconfigure the server, such that the **Start components as needed** is selected. This option is not selected when a new server is created.

   Note: Before selecting this option, verify that any other WebSphere products, that you are running in conjunction with this product, support this functionality.
3. Click **Review**, select **Synchronize changes with Nodes**.
4. Click **Save** to save any configuration changes that you made.

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 perform the following tasks 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 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.

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 **Servers > Server Types > WebSphere application 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, 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 channel must always be last because it accepts everything that is handed to it.

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.)
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 **Servers > Server Types > WebSphere application servers >** *server_name*. Then, in the Server Infrastructure section, click **Administration > 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)

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

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.

If you create any additional application servers, you cannot start, stop, or manage these servers using the administrative console that is associated with the original base server. You must either use command-line tools to perform these tasks for the additional servers, set up an administrative console for each server, or configure an administrative agent to provide a single interface to all of your servers, including the original base server. An administrative agent makes it easier to more fully administer these unfederated application servers.

Note: If you create additional application servers, only use one server to modify and save configurations. There is no coordination of configuration setting between the different servers and if you modify and save configurations on multiple servers, your data might become corrupted.

There are several options available for starting an application server.

- You can use the administrative console to complete the following steps:
  1. Click **Servers > Server Types > WebSphere application servers**.
  2. Select `server1` and 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. For example, if you are using the Express version of the product, click **Start > Programs > IBM WebSphere > Express V > *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, as shown in the following example. To start a server that is associated with a non-default profile, issue the startServer command from the *profile_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.

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.

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 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 when it should be operational, examine the status of the application on an Applications page and try restarting the application.
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 > Express 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 distributed platforms, except AIX®, you can use the **stopServer** command to stop an application server from the command line.

  A warning message displays if you are stopping the application server that is running the administrative console application.

  AIX You can issue the **stopServer** or the **stopManager** commands from the /usr/WebSphere/AppServer/bin directory:

  ```
  #  ./stopServer.sh server1
  #   ./stopManager.sh
  ```

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	
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	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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)	
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)	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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	
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	

Time zone ID	Raw offset (Hours : Minutes)	DST offset (Minutes)	Display name	QTIMZON variable (i5/OS only)
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 product environment. 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 application server. 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.

1. Start a generic application server.

   Use the launchProcess operation of the wsadmin tool to start a generic application server.

   a. View the **Status** value and any messages or logs to see whether the generic server starts.
2. Terminate generic servers.

   Use the MBean terminate launchProcess operation of the wsadmin tool to terminate a generic server.

   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.

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 a common networking service for all components.

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.

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.

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

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.

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

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.

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 starts and shuts down.

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 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**.

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*.

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.

For the WebSphere Application Server and the WebSphere Application Server - Express products, only the command-line information for starting or initializing a process applies.

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 a specific application server.

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**.

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**.

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

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

You can use this function to automatically restart Express servers. You can restart the ***server1*** process, for example.

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 If you do not install the base product as a Windows service during installation, you can use the **WASService** command in the *app_server_root*/bin directory to do so at a later time. You can use this command to add any product process as a Windows service. The operating system can then monitor each server process and restart the process if it stops.

1. Windows On a Windows operating system, **Use the installation wizard** to set up a Windows service to automatically monitor and restart processes related to the product.
   - Perform the following procedure from the installation wizard to select services that the installation wizard can set up:

a. Click **Run WebSphere Application Server as a service**.

   If you select this option, the installation wizard creates the following service during the installation:

   IBMWAS6Service - *node_name*

   The **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 an Express server process.

3. Windows On a Windows operating system, after installing the product, you can use the WASService.exe command in the *app_server_root*\bin directory to manually define a Windows service for another installation instance or for another configuration instance of the server1 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**.

Processes started by a **startServer**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

When the installation wizard creates a service, the uninstaller program can remove the service. If you use the **WASService** command to create a service yourself, it is your responsibility to remove the service when it is no longer valid. The uninstaller program does not remove services that you create with the **WASService** command.

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 name of the service that is created by the installation wizard is *IBM WebSphere Application Server V6.x - DefaultNode*.

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 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.
```

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 182 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 176 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, click **Servers > Server Types > WebSphere application servers >** *server_name*. 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 JVM 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, 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

- **-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 server 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**

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.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`.

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. 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.

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 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 100
   - 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.

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 - Express 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 - Express.

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 - Express 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 - Express, 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.