



## Installing your application serving environment

**Note**

Before using this information, be sure to read the general information under Appendix C, "Notices," on page 363.

**Compilation date: November 30, 2004**

**© Copyright International Business Machines Corporation 2004. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

How to send your comments . . . . .	vii
<b>Part 1. Product overview and quick start . . . . .</b>	<b>1</b>
<b>Chapter 1. Overview of installing your application serving environment . . . . .</b>	<b>3</b>
<b>Chapter 2. What is new in this release . . . . .</b>	<b>5</b>
What is new for installers . . . . .	6
New: Improved installation and configuration, with profiles . . . . .	8
Fast paths for WebSphere Application Server . . . . .	9
Operating system central. . . . .	11
Tutorials . . . . .	11
<b>Chapter 3. How do I install an application serving environment? . . . . .</b>	<b>13</b>
<b>Part 2. Planning and preparing . . . . .</b>	<b>21</b>
<b>Chapter 4. Installing your application serving environment &gt; Distributed operating systems</b>	<b>23</b>
Overview and new features for installing an application serving environment. . . . .	24
<b>Chapter 5. Release notes . . . . .</b>	<b>27</b>
IBM WebSphere Application Server, Version 6.0 Release Notes . . . . .	27
<b>Chapter 6. Task overview: Installing . . . . .</b>	<b>29</b>
Packaging . . . . .	30
Product offerings for Version 6. . . . .	31
New installation and operating model . . . . .	32
New programming model extensions . . . . .	32
New and improved launchpad console. . . . .	33
Hardware and software requirements . . . . .	40
Required disk space . . . . .	41
Supported operating systems . . . . .	41
Using the launchpad to start the installation. . . . .	41
Product version information. . . . .	45
versionInfo command . . . . .	47
genVersionReport command . . . . .	49
Installation: Resources for learning . . . . .	50
<b>Chapter 7. Planning the installation (diagrams) . . . . .</b>	<b>53</b>
Planning to install Network Deployment . . . . .	54
Planning to install Web server plug-ins . . . . .	63
Planning to install WebSphere Application Server Clients . . . . .	69
Planning to create application server environments . . . . .	71
Example: Choosing a topology for better performance . . . . .	74
Queuing network. . . . .	74
Queuing and clustering . . . . .	75
Queue configuration tips . . . . .	76
<b>Chapter 8. Preparing the operating system for product installation . . . . .</b>	<b>81</b>
Preparing AIX systems for installation . . . . .	81
Preparing HP-UX systems for installation. . . . .	85
Preparing Linux systems for installation . . . . .	90

Preparing Solaris systems for installation . . . . .	94
Preparing Windows systems for installation . . . . .	97
Preparing to install Network Deployment on an AIX system . . . . .	99
Preparing to install Network Deployment on an HP-UX system . . . . .	100
Preparing to install Network Deployment on a Linux system . . . . .	101
Preparing to install Network Deployment on a Solaris system . . . . .	102
Preparing to install Network Deployment on a Windows system . . . . .	103

---

## Part 3. Installing . . . . . 105

<b>Chapter 9. Installing the product and additional software . . . . .</b>	<b>107</b>
Roadmap for installing the Network Deployment product . . . . .	108
Using the launchpad . . . . .	108
Roadmaps for installing the Network Deployment product . . . . .	109
Understanding what you can do with the Network Deployment product . . . . .	111
WebSphere Application Server uses ISMP to install products . . . . .	111
What to do when an installation problem occurs . . . . .	111
Getting the latest information . . . . .	111
What to do next. . . . .	111
Installing Network Deployment . . . . .	111
Installing additional features on an existing Network Deployment installation . . . . .	116
Installing silently . . . . .	120
Customizing the options response file for Network Deployment . . . . .	123
Mounting CD-ROMS on Linux and UNIX operating systems . . . . .	133
Installing IBM HTTP Server . . . . .	134
Installing Web server plug-ins . . . . .	136
Selecting a Web server topology diagram and roadmap . . . . .	139
Configuration behavior of the Plug-ins installation wizard . . . . .	145
Web server configuration . . . . .	153
Configuring a Web server and an application server on separate machines (remote) . . . . .	157
Configuring multiple Web servers and remote stand-alone Application Servers . . . . .	163
Configuring a Web server and an application server profile on the same machine . . . . .	170
Configuring a Web server and a custom profile on the same machine. . . . .	180
Configuring a Web server and a deployment manager profile on the same machine . . . . .	185
responsefile.txt . . . . .	191
Uninstalling the Web server plug-ins for WebSphere Application Server . . . . .	200
Editing Web server configuration files . . . . .	203
Allowing Web servers to access the administrative console . . . . .	216
Installing Application Client for WebSphere Application Server . . . . .	218
Best practices for installing application clients . . . . .	220
Installing application clients silently . . . . .	221
Uninstalling Application Client for WebSphere Application Server . . . . .	222

---

## Part 4. Configuring . . . . . 225

<b>Chapter 10. Configuring the product after installation . . . . .</b>	<b>227</b>
firststeps command . . . . .	228
Using the Profile creation wizard . . . . .	234
Using the Profile creation wizard to create a deployment manager . . . . .	238
Using the Profile creation wizard to create a custom profile . . . . .	249
Using the Profile creation wizard to create an application server. . . . .	263
Deleting a profile . . . . .	274
wasprofile command . . . . .	275
Introduction to terms that describe Version 6 profiles . . . . .	275
Location of the command file. . . . .	278

Logging . . . . .	278
Required disk space . . . . .	279
Concurrent profile creation. . . . .	279
Entering lengthy commands on more than one line . . . . .	279
wasprofile.sh command syntax . . . . .	280
wasprofile.bat command syntax . . . . .	281
Parameters . . . . .	282
Use case scenarios . . . . .	284
Using the installation verification test . . . . .	290
ivt command . . . . .	291
<b>Chapter 11. Communicating with Web servers . . . . .</b>	<b>293</b>
Web server plug-in properties settings . . . . .	295
Plug-in log file name . . . . .	295
Plug-in installation location . . . . .	295
Plug-in configuration file name . . . . .	296
Automatically generate plug-in configuration file . . . . .	296
Automatically propagate plug-in configuration file . . . . .	297
Ignore DNS failures during Web server startup . . . . .	297
Refresh configuration interval . . . . .	297
Plug-in logging . . . . .	297
Web server plug-in request and response optimization properties settings . . . . .	298
Web server plug-in caching properties settings . . . . .	300
Web server plug-in request routing properties settings . . . . .	301
Web server plug-in custom properties . . . . .	302
Web server plug-in configuration service properties settings . . . . .	303
Enable automated Web server configuration processing . . . . .	303
Application Server property settings for a Web server plug-in . . . . .	303
Server role . . . . .	303
Connect timeout . . . . .	304
Maximum number of connections that can be handled by the Application Server . . . . .	304
Use extended handshake to check whether Application Server is running . . . . .	305
Send the header "100 Continue" before sending the request content . . . . .	305
Web server plug-in configuration properties . . . . .	305
Web server plug-in connections. . . . .	307
Web server plug-in remote user information processing . . . . .	308
Web server plug-ins . . . . .	309
Checking your IBM HTTP Server version . . . . .	309
Web server tuning parameters . . . . .	309
Gskit install images files . . . . .	313
Plug-ins: Resources for learning . . . . .	314
Web server plug-in tuning tips . . . . .	314
Tuning Web servers . . . . .	316
<b>Appendix A. Uninstalling the product . . . . .</b>	<b>317</b>
uninstall command . . . . .	318
vpd.properties file . . . . .	319
Uninstalling Network Deployment . . . . .	322
Uninstalling manually. . . . .	325
Manually uninstalling on an AIX system . . . . .	325
Manually uninstalling on an HP-UX system . . . . .	328
Manually uninstalling on a Linux system. . . . .	330
Manually uninstalling on a Solaris system . . . . .	332
Manually uninstalling on a Windows system . . . . .	334
<b>Appendix B. Troubleshooting installation . . . . .</b>	<b>337</b>

Troubleshooting Web server plug-ins installation and removal . . . . .	346
Installation component troubleshooting tips . . . . .	355
Troubleshooting installation problems. . . . .	355
Installation either completes with errors or warnings, or hangs . . . . .	356
Problems installing or starting Apache or IBM HTTP Server . . . . .	356
Messages issued during installation and profile creation . . . . .	357
Error: The input line is too long . . . . .	357
Note: The following file systems will be expanded during the installation . . . . .	358
The disk space is nn Mbyte less than required. The installation cannot continue.. . . .	358
Specify a different directory or perform a manual uninstall . . . . .	358
Error: /usr/opt/ibm/gskta/bin/gsk7ikm: not found . . . . .	359
Error writing file = There may not be enough temporary disk space. . . . .	359
Error: localhost is not a valid host name for remote access. . . . .	359
Warning: Cannot convert string "<type_name>"to type FontStruct . . . . .	360
INFO: Created system preferences directory in java.home . . . . .	360
The installer could not successfully add the product information into the RPM database.. . . .	360
Error: java.io.IOException: konqueror: not found. . . . .	361
<b>Appendix C. Notices . . . . .</b>	<b>363</b>
<b>Appendix D. Trademarks and service marks . . . . .</b>	<b>365</b>

---

## 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 919-254-0206.

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.





---

## Part 1. Product overview and quick start



---

# Chapter 1. Overview of installing your application serving environment

This topic provides links to conceptual overviews of installing the application serving environment.



## **“What is new for installers” on page 6**

This topic provides an overview of new and changed features in installation, migration, coexistence, and interoperability support.

## **Chapter 7, “Planning the installation (diagrams),” on page 53**

This topic describes common product topologies that you can install with WebSphere Application Server, Version 6 products.



## **Presentations from Education on Demand**

- V6 topology and terminology
- WebSphere profiles
- Installation overview
- Install Network Deployment and Express
- IBM HTTP Server installation
- Web server plug-in installation



---

## Chapter 2. What is new in this release

This topic provides a high level overview of what is new and changed in Version 6. The audience is anyone who is evaluating or planning to use Version 6, particularly users of previous versions or editions.



Welcome to Version 6, which helps you:

- Rapidly understand and obtain value from new J2EE 1.4 technology
- Make the most of your existing resources and lower your total cost of ownership
- Work more productively, with easier to use installation and administration interfaces
- Support your growing business with a firm foundation to the WebSphere platform

There is now greater consistency in the WebSphere Application Server family from top to bottom, including in the Express version. Such features include support for J2EE 1.4 across the family, which makes it easier to develop and deploy applications using industry standard tools, and support for the latest Web Services standards, making it easier to integrate applications inside the enterprise as well as externally with customers, partners and supplier.

IBM WebSphere Application Server Version 6 provides customers better scaling so more concurrent users can access an application built on the software. This can help reduce administrative and licensing costs for companies and provides better flexibility to quickly bring additional user resources on board.

### What is new for administrators



The biggest improvements and changes in system administration, monitoring, and tuning can be summarized as follows:

- Changes to the default configuration
- Incremental cell upgrade for easier migration
- Fine grained application update and other deployment improvements
- Enhanced administrative infrastructure, with J2EE 1.4 related changes
- Growing set of administrative commands
- Improved installation and configuration, with profiles
- Improved monitoring and performance tuning
- Default messaging provider
- Even more administrative improvements!

For details about each item, including links to relevant documentation, see the *Administration: Operations* PDF.

### Improvements to this documentation

- New How do I?... feature gathers documentation, multimedia demonstrations, tutorials, cheat sheets, and presentations for commonly performed tasks
- Improved installation documentation
- Improved scripting documentation
- Conceptual overviews and what is new topics have better content and are easier to find
- Quick start shortcuts direct you quickly to key materials
- New *Learn about WebSphere applications* section for studying the programming model

- Improved, task-based table of contents
- Improvements to troubleshooting content and placement

---

## What is new for installers

This topic highlights what is new or changed in Version 6 for users who are going to install, upgrade, or migrate the product, including coexistence and interoperability concerns.



The biggest improvement to installation is that now you can install one copy of the core files (binary system files and such) on a machine, then use *profiles* to define multiple application server runtime environments -- each with its own administrative interfaces -- that share the core files. This has many positive ramifications for preparing, installing, maintaining, and removing installations, including a decreased footprint.

Another big change is the introduction of separate installation routines for the Application Server product, the Web server, the Web server plug-ins, and the clients so that you can install only what you need on a particular machine.

The ability to incrementally upgrade nodes in a cell is a new feature discussed in the *Migrating, coexisting, and interoperating* PDF.

### *Redesigned Launchpad to plan and start installation*

A redesigned launchpad on the product CD launches one of the installation programs. See “Using the launchpad to start the installation” on page 41. The Launchpad also provides product solution roadmap diagrams to let you decide on the fly what to install where.

### *Install only what you need*

New, more granular installation procedures simplify installation and improve installation reliability. Installation routines exist for the following components on the product disc.

- WebSphere Application Server products
- IBM HTTP Server
- Web Server plug-ins
- Application clients

Separate installation procedures for the IBM HTTP Server product and for the Web server plug-ins let you install only what you need on a particular machine.

### *Installing WebSphere application server products*

In Version 6.0, installing WebSphere Application Server Network Deployment is a two-step process. The first step is using the installation wizard to install a shared set of core product files. The second step is using the new Profile creation wizard to create a *deployment manager profile*, an *Application Server profile*, or a *custom profile*.

For more information, see:

- Chapter 9, “Installing the product and additional software,” on page 107
- “Using the Profile creation wizard” on page 234
- “wasprofile command” on page 275

### *Installing IBM HTTP Server*

The installation wizard for IBM HTTP Server is redesigned for Version 6. At the end of the Web server installation, launch the new Plug-ins installation wizard to configure the IBM HTTP Server and the Application Server.

See “Installing IBM HTTP Server” on page 134.

### *Installing WebSphere Application Server clients*

A client application processes on a client machine and a host WebSphere Application Server machine. A client might provide the GUI, but process data on the host, for example. The Clients installation wizard installs environments for running client applications on the client machine. Some environments perform all necessary handshaking and protocol. *Thin* client environments require client applications to have their own protocols for such things as JNDI lookups.

See “Installing Application Client for WebSphere Application Server” on page 218.

### *The product installation root has changed*

The installation root varies by operating system, as before, but now includes IBM in the path. The installation root is the same for all three product editions now -- there is no longer a separate installation root for the Network Deployment product edition.

The installation root is now:

- For AIX: /usr/IBM/WebSphere/AppServer
- For Linux and UNIX: /opt/IBM/WebSphere/AppServer
- For Windows: C:\Program Files\IBM\WebSphere\AppServer

Additional directory structure changes are introduced, as described in “New: Improved installation and configuration, with profiles” on page 8.

### *Improved installation documentation*

New installation documentation on the product disc helps you retrieve relevant information. For an overview of the installation section, see “Overview and new features for installing an application serving environment” on page 24.

### *Simplified setup and prerequisites*

Redesigned product features simplify operating system setup and reduce prerequisite requirements. For details, see Chapter 8, “Preparing the operating system for product installation,” on page 81 and “Hardware and software requirements” on page 40.

### *Smaller footprint*

A redesigned installer simplifies the installation of the core product files for WebSphere Application Server products. You do not have to install the core product files multiple times to create multiple standalone Application Servers. Instead, use the new Profile creation wizard to create an application server runtime environment that has its own administrative interface.

See Chapter 10, “Configuring the product after installation,” on page 227.

### *Simplified service*

The installation program installs the system files (shared binary files), which you do not update until you install a service fix. Creation and configuration of application servers do not change the system files.

### *Easier removal and reinstallation*

Removal and reinstallation of Application Server environments is simplified and faster. Application Server environments are much easier to install and much easier to uninstall than in Version 5. Uninstalling an Application Server does not change the shared system files of the base WebSphere Application Server product.

In Version 5, some scenarios required uninstalling all traces of the product to produce a clean machine, which then allowed you to reinstall the product. In Version 6, that is a thing of the past. Application servers uninstall cleanly, almost effortlessly.

See Appendix A, “Uninstalling the product,” on page 317.

### *Better indicators of installation status*

Improved logging and status indicators let you easily determine the success or failure of an installation.

### *Silent installation requires license acceptance*

You now need to set the License Acceptance property to “true” in the silent installation response file. The property is `silentInstallLicenseAcceptance.value`, which is described further in “responsefile.nd.txt” on page 128.

### *New migration wizard*

A new migration wizard collects data for the migration command line tools. The wizard also monitors and reports migration status. Note that the installation program no longer performs the migration.

---

## **New: Improved installation and configuration, with profiles**

This topic describes how the product directory structure and use of profiles separates the binary system files from the installation-specific configuration information, leading to easier installation, maintenance, and configuration management.

The version 6 product installation program places the files it creates into one of two separate environments. It installs the binary system files that may be read-only. It also creates an initial profile, which is a run-time execution environment that includes configuration files, the default location for deployed applications, logs, and other data. All profiles on a machine can share the same system files, but do not change the system files.

Administrators can use any of the administrative clients to administer the configuration of each profile. It is highly recommended that you browse “wasprofile command” on page 275 for a full introduction to profiles.



*Administrators must now address commands to particular profiles*

System administrators of multiple Application Server instances on a single machine should note the following change. System administration commands are now profile-specific. In previous releases, commands were executed from the bin directory of the product installation root. Now commands must be executed from a particular profile/bin directory. For example, use the `-profileName` parameter to identify which Application Server to start.

For various ways to address profiles, see “Using the Profile creation wizard to create a deployment manager” on page 238.

*Profiles can be exported*

You can propagate the configuration from one wasprofile to another by exporting a wasprofile to a configuration archive and importing it to another wasprofile. This mechanism works between wasprofiles of the same installation or different installations. A restriction is that this mechanism only works for an unfederated wasprofile.

*Profiles sharing the same system files may be upgraded (or rolled back) individually.*

Because the version 6 binary images are installed at a different location, it is possible to upgrade each Version 5 cell member individually.

*Directory structure changes*

Files pertaining to a particular application server runtime environment are in a directory path containing the word profiles, as well as the particular profile name.

Note also that the product installation root has changed to include IBM, as described in “What is new for installers” on page 6.

---

## Fast paths for WebSphere Application Server

This topic describes the most quick and simple paths to deploying an application, with links that pinpoint the relevant information for reaching your goals quickly. It also describes the audience roles and tasks assumed by this documentation. The fast path described in this topic is intended to help you gain a little experience. It does not showcase the advanced product features that some users need or want to use in their production environments.

**Overview.** Deploying any type of application involves the following tasks.



### **Step 1: Install the product.**

The simplest scenario is to perform a typical installation of a single application server onto a Windows machine. See “Operating system central” on page 11 for direct links to installation steps for Windows (or another operating system of your choice). Planning information is available in the installation section.



### **Step 2: Obtain or develop your application.**

In the simplest scenario, you already have a packaged, J2EE 1.4 compliant Web application, perhaps from a vendor with whom you work. Other than that, a Web application is the most simple type of application to develop, or refer to the many Samples in the Samples Gallery.



### Step 3: Deploy and test your application.

In the simplest scenario, you will use the application installation wizard available in the administrative console. See s PDF for a detailed walkthrough of this task. The task describes many contingencies, but many of these steps can be disregarded if:

- You are deploying a Web application
- You accept the default settings whenever possible
- Your application does not require data access
- Your application does not require security

Also, in many situations, you can get by without modifying the default application server configuration.


















### Step 4: Administer your deployed application.

In the simplest scenario, you will monitor your application with the Tivoli Performance Viewer functionality built into the administrative console. See "Monitoring performance with Tivoli Performance Viewer (TPV)" in the *Administration: Operations* PDF for an overview of this task.

Now you should have some insight into the task flow for successfully deploying your applications.

To solidify and expand your understanding, view the following table. It shows the user role and task assumptions by which this documentation is organized, for predictability. If you know what role or task you are performing, you can disregard (at least temporarily) documentation that is labeled for other roles or tasks. By their nature, role and task models are simplified compared to reality. Also, one person might perform many roles and tasks in the course of a day.






User role	User tasks
 Installer	 Install application serving environment   Migrate deployed applications and their environment
 Administrator	 Administer applications and environment   Monitor and tune applications and their environment   Deploy applications into production

User role	User tasks
 Developer	 Develop or migrate application code  Assemble applications for deployment  Deploy applications for testing
 Security expert	 Secure applications and their environment
 Troubleshooter	 Use tools to troubleshoot problems

---

## Operating system central

This topic provides links to documentation that is specific to particular operating systems.

	"Preparing to install Network Deployment on an AIX system" on page 99
	"Preparing to install Network Deployment on an HP-UX system" on page 100
	"Preparing to install Network Deployment on a Linux system" on page 101
	"Preparing to install Network Deployment on a Solaris system" on page 102
	"Preparing to install Network Deployment on a Windows system" on page 103

---

## Tutorials

This topic describes how to find tutorials and their accompanying Samples, for learning how to accomplish your goals with the product.

### Education on Demand tutorials

The Education on Demand site provides education resources that you can use at your convenience. You can find Education on Demand tutorials referenced throughout this documentation.

- Obtaining help with tasks

Some of the tutorials are presented according to the task that you are performing, such as administration. The topics tend to be generally based, such as how to use the administrative tools.

- Exploring particular WebSphere application types

Tutorials also are presented according to the WebSphere application type or technology that they demonstrate, such as how to do something with Web services.

## **Tutorials with Technology for Developers Version 6**

A specific set of tutorials on developerWorks pertain to an early release of Version 6: WebSphere Application Server Technology for Developers Version 6 in late 2003.






Many of the administrative details of the product have changed since these tutorials were issued. However, many of them focus on using new aspects of the J2EE 1.4 programming model, which remain valid for the product release that you are now using. Use these tutorials with care.

## Chapter 3. How do I install an application serving environment?

- “Install a development environment”
- “Plan and install a production environment” on page 14
- “Configure the product after installation (required)” on page 16
- “Communicate with Web servers” on page 17
- “Troubleshoot the installation” on page 19



### Legend for “How do I?...” links

Documentation 	Show me 	Tell me 	Guide me 	Teach me 
Refer to the detailed steps and reference	Watch a brief multimedia demonstration	View the presentation for an overview	Be led through the console pages	Perform the tutorial with sample code
<b>Approximate time:</b> Varies	<b>Approximate time:</b> 3 to 5 minutes	<b>Approximate time:</b> 10 minutes+	<b>Approximate time:</b> 1/2 hour+	<b>Approximate time:</b> 1 hour+

### Install a development environment

---



#### Plan your installation of WebSphere Application Server

IBM WebSphere Application Server, Version 6 is an integrated platform that contains an Application Server, Web development tools, a Web server, and additional supporting software and documentation. Diagrams are provided to show several common installation configuration.



Tell me

---



#### Review the Release Notes

Learn about late-breaking changes and limitations, particularly involving installation, migration, and configuration.

Documentation 



### **Review the software and hardware prerequisites**

Learn which product configurations are supported.

Documentation



### **Use the launchpad to explore options and start the installation**

The launchpad console is the starting point for installing WebSphere Application Server products and trials.

Documentation



### **Learn about WebSphere applications**

Use this section as a starting point to investigate the technologies used in and by applications that you deploy on the application server.

Documentation:

"Learn about WebSphere applications" in the *Developing and Deploying Applications*

PDF:



---

## **Plan and install a production environment**

Look at configuration diagrams to determine how you want your application serving environment to be. Perform the needed installations.

- Topology and terminology



### **Choose a product edition**

Learn about the family of WebSphere Application Server products, the package that ships for each package, and the installable components on each product compact disc.

Documentation 

Tell me 



### **Plan your application serving environment**

Use the diagrams to decide among several popular installation scenarios.

Documentation:  
“Planning to install  
Network Deployment”

on page 54 



### **Review the Release Notes**

Learn about late-breaking changes and limitations, particularly involving installation, migration, and configuration.

Documentation 



### **Review the software and hardware prerequisites**

Learn which product configurations are supported.


Documentation 



### **Obtain an overview of installation**

High level steps for installing the product. Tips and knowledge that will make installation easier.

“Roadmap for  
installing the Network  
Deployment product”

on page 108 

Tell me 



## Installing the product

Documentation:

- “Preparing to install Network Deployment on an HP-UX system” on page 100
- “Preparing to install Network Deployment on an HP-UX system” on page 100
- “Preparing to install Network Deployment on a Linux system” on page 101
- “Preparing to install Network Deployment on a Solaris system” on page 102
- “Preparing to install Network Deployment on a Windows system” on page 103

Show me:

- Network Deployment



Tell me



## Take your first steps in using the product

The First steps console is a post-installation ease-of-use tool for directing WebSphere Application Server elements from one place. You can verify the installation, start a server, view Samples, register the product, and other activities.

Documentation



## Configure the product after installation (required)

Perform required Network Deployment configuration. Take your first steps using the product.



## Create application server environments (required for Network Deployment)



In Version 6.0, installing WebSphere Application Server Network Deployment is a two-step process. The first step is using the installation wizard to install a set of shared product binaries (system files). The second step is using the Profile creation wizard to create a profile. Application Server profiles are the run-time environments for Application Server processes. This topic describes common scenarios for creating Application Server profiles and provides links to profile creation procedures for each scenario.

Documentation:  
"Planning to create  
application server  
environments" on

page 71 

Show me:

- Standalone node
- Deployment manager
- Managed node


Tell me 



### Take your first steps in using the product

The First steps console is a post-installation ease-of-use tool for directing WebSphere Application Server elements from one place. You can verify the installation of a profile, start a server, view Samples, register the product, and other activities.

Documentation:  
"firststeps command"

on page 228 

## Communicate with Web servers

Perform tasks related to HTTP serving, including installing a complementary Web server available with your Application Server, and configuring communication between the Application Server and any supported Web server.



### Install IBM HTTP Server as your Web server

Install IBM HTTP Server V6 and its plug-in, or install a plug-in for another supported Web server to enable the Web server to work with WebSphere Application Server, V6.

Documentation:  
"Installing IBM HTTP  
Server" on page 134



Show me 

Tell me 



### Plan the Web server plug-ins in your application serving environment

The primary production configuration is an Application Server on one machine and a Web server on a separate machine. This configuration is referred to as a remote configuration. Contrast the remote configuration to the local configuration, which consists of the Application Server and the Web server on the same machine. Either configuration requires the Web server plug-in modules on the Web server machine.

Documentation:  
"Selecting a Web  
server topology  
diagram and  
roadmap" on page



### Install plug-ins for communicating with Web servers

After installing the Application Server and a Web server, you can install a WebSphere Application Server plug-in for a supported Web server. The binary plug-in is unique per Web server. The purpose of the binary plug-in is to provide the communication protocol between the Web server and the Application Server.

Documentation:  
"Installing Web server  
plug-ins" on page 136

Show me



Tell me



### Configure communication with the Web server

The Web server plug-ins for Web servers running on a distributed platform are provided on a separate CD from the WebSphere Application Server product. A Web Server Plug-in Installation Wizard that is also provided on that CD.

Documentation:

Console



Show me



Tell me



Guide me. See  
"Cheat sheets for the  
administrative  
console" in the  
*Administration:  
Operations* PDF.



### Configure the HTTP channel within the application server

The HTTP channel provides client applications with persistent HTTP connections to remote hosts that are either blocked by firewalls and/or require an HTTP proxy server (including authentication). An HTTP channel is used to exchange application data in the body of an HTTP request and HTTP

response sent to and received from a remote server. HTTP channels enable 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 are unaware that HTTP is being used to exchange the data.

Documentation: See  
"Configuring transport  
chains" in the  
*Administration: Setting*

up PDF. 


---



### **Enable your Web server to use HTTPS in addition to HTTP**

If you intend to use the Secure Sockets Layer (SSL) transport (also known as HTTPS), in addition to the plug-in files, you must also install the Global Security Kit (GSKit) on the workstation hosting your Web server. This kit helps the Web server connect to your Application Server.

Documentation: "Gskit  
install images files" on

page 313 

---

## **Troubleshoot the installation**

Troubleshoot installation problems and apply fixes as needed.

---



### **Troubleshoot installation problems**

Obtain help with a variety of installation problems.

Documentation:  
Appendix B,  
"Troubleshooting  
installation," on page

337 

---



### **Review the software and hardware prerequisites**

Learn which product configurations are supported.

Documentation 

-----

---

## Part 2. Planning and preparing



---

## Chapter 4. Installing your application serving environment > Distributed operating systems

The installation uses InstallShield for Multiplatforms (ISMP) to perform the installation. You can use the Installation wizard or perform the installation from a command line, using the silent installation method.

### WebSphere Application Server Network Deployment overview

The Network Deployment product lets you create unlimited numbers of the same stand-alone application servers that you can create with the Express product or the base product. However, the real strength of the Network Deployment product is to create and manage a multimachine environment, where you have installed managed application server nodes on different machines. You can manage all of the application servers as a group, or *cell* by using the administrative console of the deployment manager. Use the Profile creation wizard to create deployment manager profiles, application server profiles, and custom profiles. A deployment manager manages the configuration for all of the managed nodes in its cell and deploys applications to any managed node in the cell. All of the profiles share command files and other product binaries that are created during the installation.

The main reason to use managed nodes in a cell versus using the same number of stand-alone application servers is the centralized administration that the deployment manager provides for the cell. Another advantage is the ability to create clusters in a managed node that the deployment manager can manage with simple workload balancing. You cannot create clusters using the Express product or the base product. The Network Deployment product installs application server nodes that can run in stand-alone mode. You can also add the nodes to the deployment manager cell to manage them as a group.

Custom profiles are empty nodes that you must add to a deployment manager cell to make operational. After adding the node, use the administrative console for the deployment manager to customize the node. For example, create application servers on the node or clusters. A custom node does not include a `server1` process as an application server node does. Nor does it include the default application or any Sample applications. It is truly an empty node that you must customize.

### Creating multiple application servers on a single machine

You can create any number of application servers on a single machine. Although each profile is unique, with its own configuration, applications, data, logs, and administrative console, all of the servers share one set of command files and other product binary files. Changes that you make to one server profile do not affect the other servers. Configuration documents are maintained within each profile. The **wasprofile** command and its graphical user interface, the Profile creation wizard create the multiple stand-alone application servers.

You can also create clusters on an application server node when the node is part of a deployment manager cell. Use the administrative console of the deployment manager to create the clusters and cluster members.

### Other installable components on the product disc

The product disc contains IBM HTTP Server, the Web server plug-ins for WebSphere Application Server, and the Application Clients for WebSphere Application Server. Each installation procedure is described in this information center and in the IBM HTTP Server information center.



### How to buy IBM WebSphere Application Server software

See your IBM representative for licensing information, or visit the How to buy WebSphere Application Server software Web site.

---

## Overview and new features for installing an application serving environment

This topic summarizes the contents and organization of the installation documentation, including links to conceptual overviews and descriptions of new features.

-  Chapter 1, “Overview of installing your application serving environment,” on page 3
-  “What is new for installers” on page 6

### Sections in the installation documentation:



#### **Chapter 5, “Release notes,” on page 27**

This topic highlights known problems, limitations, and workarounds as an addendum to the product documentation. Briefly review this document before installing the product.



#### **Chapter 6, “Task overview: Installing,” on page 29**

This topic describes installing and customizing all of the installable components that are on the product compact disc.



#### **Chapter 7, “Planning the installation (diagrams),” on page 53**

This topic presents installation topologies, with installation roadmaps summarizing the product components that are needed to install each topology.



#### **Chapter 8, “Preparing the operating system for product installation,” on page 81**

This topic shows how to prepare each supported operating system for installation. A series of operating system-specific topics describe what to do.



#### **Chapter 9, “Installing the product and additional software,” on page 107**

This topic describes how to install the product, as the root user on a Linux platform or on a UNIX-based platform, or from a user ID that belongs to the administrator group on a Windows platform.



#### **Chapter 10, “Configuring the product after installation,” on page 227**

This topic describes what you might do after installing the product. Activities include configuring additional product configuration profiles; migrating a previous product configuration; and securing certain product files.



#### **Chapter 11, “Communicating with Web servers,” on page 293**

This topic describes how to establish communication between Web servers and application servers, so that Web servers can route requests to application servers. The WebSphere Application Server plug-ins for Web servers running on a distributed platform are provided on a



separate CD from the WebSphere Application Server product. It describes how to use the Web Server Plug-in Installation wizard provided on that CD, and then how to administer the resulting plug-in configurations.



**Appendix A, “Uninstalling the product,” on page 317**

This topic describes how to uninstall the product from the root user on a Linux platform or a UNIX-based platform, or from a user ID that belongs to the administrator group on a Windows platform.



**Appendix B, “Troubleshooting installation,” on page 337**

This topic describes how to identify and handle a variety of problems that are encountered during installation activities.



---

## Chapter 5. Release notes

This section contains the **release notes** for distributed operating systems. Release notes summarize late-breaking news and product limitations.

---

### IBM WebSphere Application Server, Version 6.0 Release Notes

This document contains release notes for:

- WebSphere Application Server Express, Version 6
- WebSphere Application Server, Version 6
- WebSphere Application Server Network Deployment, Version 6

Links are provided to the product support Web site, to product documentation, and to last minute updates, limitations, and known problems for the WebSphere Application Server Version 6 products.

- “Accessing last-minute updates, limitations, and known problems”
- “Accessing hardware and software requirements”
- “Accessing product documentation”
- “Accessing the product support Web site” on page 28
- “Contacting IBM Software Support” on page 28

#### Accessing last-minute updates, limitations, and known problems

The Version 6 Release notes are available on the product support site as technotes. To see a list of the release notes for Version 6.0, go to the WebSphere Application Server support Web page.

#### Accessing hardware and software requirements

The hardware and software requirements for the WebSphere Application Server products are provided on the Supported hardware and software Web page.

#### Accessing product documentation

The following documentation is installed with WebSphere Application Server.

#### The readme file.

Several readme files are provided:

- The `readme_en.html` file in the root directory of the product CD-ROM is a shadow of the Launchpad application. It provides links to the installation procedures and documentation for all of the installable components on the product CD. Translated versions of the file are in the launchpad directory.
- The `readme.html` file in the WAS directory points to this more detailed readme in the WAS directory and to the Getting Started Guide in the WAS/docs directory. This file is not translated.
- The `readme_en.html` in the readme directory of each installable component, such as this file. The Launchpad links to this file for a description of installing the WebSphere Application Server - Express product. This file is also available from the readme directory in the root directory of the product CD-ROM.

#### Getting Started Guide

The `InstallGuide_en.pdf` and the `InstallGuide_en.html` in the WAS/docs directory provide detailed instructions for installing WebSphere Application Server. The information in the guide is derived from

information in the online information center. The guide is not translated on the product CD; however, translated versions of the information center are available online.

## **Help files**

The help files provide detailed instructions for completing tasks and for specifying settings in the graphical systems management tools. Use the help menus, links, or buttons provided in the tool interfaces to access the help files.

## **Documentation on the Web**

For the Version 6 information center in all supported languages, go to:  
<http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp>.

For the entire information center set for all WebSphere Application Server products, go to the Information Center library page.

For the entire documentation set for all WebSphere Application Server products, including Adobe Acrobat PDF versions of the information, go to the WebSphere Application Server library page.

All of the Version 6 product documentation is in the V6 information center, including versions of the installed help files.

## **Accessing the product support Web site**

To search for the latest technotes, downloads, fixes, and other support-related information, go to the WebSphere Application Server support page.

## **Contacting IBM Software Support**

If you encounter a problem with the WebSphere Application Server, first try the following actions:

- Following the steps described in the product documentation
- Looking for related documentation in the online help
- Looking up error messages in the message reference

If you cannot resolve your problem by any of the preceding methods, contact IBM Technical Support.

Purchase of WebSphere Application Server entitles you to one year of telephone support under the Passport Advantage program. For details about Passport Advantage, visit the Passport Advantage page.

The number for Passport Advantage members to call for WebSphere Application Server support is 1-800-237-5511. Have the following information available when you call:

- Your contract or Passport Advantage number.
- Your WebSphere Application Server version and revision level, plus any installed fixes.
- Your operating system name and version.
- Your database type and version.
- Basic topology data: how many machines are running, how many application servers, and so on.
- Any error or warning messages that are related to your problem.

---

## Chapter 6. Task overview: Installing

This topic describes installing Version 6 on AIX, HP-UX, Linux, Solaris, and Windows operating systems.

This topic describes installing and customizing all of the installable components that are on the IBM WebSphere Application Server Network Deployment, Version 6.0 product compact disc.

You can obtain the product code to install on distributed platforms in any of the following ways:

- From the product compact disc
- From the Passport Advantage site, where licensed customers can download installation images

The launchpad application is available on the product CD and on downloaded installation images. The launchpad is the recommended method of installing components that are on the product CD. See “Using the launchpad to start the installation” on page 41 for more information about the launchpad.

If you cannot use the launchpad for some reason, examine the Information center version of the installation solution diagrams in the launchpad. The planning diagrams in the information center help you to verify your product selection and to identify and select your installation path. See Chapter 7, “Planning the installation (diagrams),” on page 53 for more information.

If you are already familiar with the types of topologies that you can create with WebSphere Application Server products, you can use the fastpath installation path on the launchpad to begin installing components that are on the product CD. If you cannot use the launchpad for some reason, you can find the installation programs for installable components in the following locations:

Installable component	Directory	Program
WebSphere Application Server Network Deployment	WAS	install
IBM HTTP Server	IHS	install
Web server plug-ins	plugin	install
Application Clients	AppClient	install

See the description of the product packaging in “Packaging” on page 30 for more information about installable components.

The information center for WebSphere Application Server products also includes information about the IBM HTTP Server, the application clients, and the Web server plug-ins. The information center also includes some information about the Application Server Toolkit.

Perform the following procedure to learn about and to create a running version of the product components on your machine. Plan to read through the major topics in the information center Getting Started section before beginning the installation. If you are planning to migrate from an earlier version, you can install the WebSphere Application Server product before migrating.

A few migration considerations exist after installing the Network Deployment product. When migrating a deployment manager from Version 5, name the Version 6 cell with the same cell name as in the V5 cell. After migrating a deployment manager to V6, you can migrate the managed nodes in the deployment manager in an incremental migration. See the *Migration Guide* PDF for more information. for more information. Create V6 application servers for the incremental migration with the same node name as the V5 managed node. Aside from these considerations, migration is greatly simplified in Version 6.

New for Version 6 is the Migration wizard. See the *Migration Guide* PDF for more information. for more information.

1. Review the installation solution diagrams for help in planning a design for your topology.  
The planning diagrams in the Information center can help you identify and select your installation path. See Chapter 7, “Planning the installation (diagrams),” on page 53 for more information.  
Before you start installing products with the launchpad, verify that your operating system is supported and that you have installed all of the prerequisite patches.
2. Prepare your operating platform for installation.  
Use the information in Chapter 8, “Preparing the operating system for product installation,” on page 81 to verify that your operating system is able to run the WebSphere Application Server products.  
After preparing your operating system, you are ready to install the WebSphere Application Server products.
3. Install the WebSphere Application Server product that you have selected.  
See Chapter 9, “Installing the product and additional software,” on page 107 for specific installation procedures.  
After installing the Network Deployment product, you must create a deployment manager and application servers. In Version 6, the run-time environments for server processes are called *profiles*. Installing the product code establishes the core product files on a machine. Creating profiles creates separate run-time environments without having to reinstall the core product files.
4. Configure the product after installation, as described in Chapter 10, “Configuring the product after installation,” on page 227.  
Create server processes after installation by creating profiles. Create a deployment manager first. Then create an application server and add the node into the deployment manager cell.  
The Profile creation wizard configures each profile using the `wasprofile` command-line tool. See “Using the Profile creation wizard” on page 234 and “`wasprofile` command” on page 275 for more information.
5. **Optional:** Consider migrating a previous installation to Version 6, as described in the *Migration Guide* PDF.  
You can migrate the configuration and applications from a previous installation of another version of WebSphere Application Server. Version 6 has a new utility to help you perform the migration. The Migration wizard is described in the *Migration Guide* PDF.

---

## Packaging

This topic describes the family of WebSphere Application Server products and the ship package for each product.

The WebSphere software platform for e-business starts with a foundation that is formed from Web application serving and integration. IBM WebSphere Application Server software provides the core software to deploy, integrate, and manage your e-business applications.

WebSphere Application Server software supports custom-built applications that are either based on integrated WebSphere software platform products or on other third-party products. Such applications can range from dynamic Web presentations to sophisticated transaction processing systems.

IBM WebSphere Application Server - Express, WebSphere Application Server, and WebSphere Application Server Network Deployment are incrementally related. As you scale up your e-business demand, you can also scale up your e-business capability by moving from one product to the next.

The Express product is the entry point for a single machine. The base Application Server product differs from the Express product in licensing and tooling support. The Network Deployment product includes all of the Application Server functionality of the other two products and also offers multiple machine hosting and centralized management through a deployment manager process.

The management interface for all three products includes a scripting interface and the administrative console, which is a Web application. The scripting interface and the administrative console of the

deployment manager are enhanced for managing all of the Application Servers in a distributed network cell. From the single point of control of the deployment manager, you can define servers, IBM HTTP Server processes, and clusters, start and stop servers and cluster members, and deploy applications anywhere in the cell.

## Product offerings for Version 6

IBM WebSphere Application Server products provide a next-generation application server on an industry-standard foundation. Each product addresses a distinct set of scenarios and needs. WebSphere Application Server, Version 6 includes the following product offerings:

- WebSphere Application Server - Express
- WebSphere Application Server
- WebSphere Application Server Network Deployment

### WebSphere Application Server - Express

This product offering addresses the basic programming and run-time needs of desktop developers and single-server production scenarios. The run-time environment addresses standards-based programming for Web and component-based programming, as well as Web services.

The administration model for this offering is a single-server environment without clustering, and without centralized administration of multiple server instances.

The Express product package includes the Rational Web Developer tool, which is a fully integrated development environment (IDE). Rational Web Developer is an enhanced version of the WebSphere Studio Site Developer in Version 5. The Express package also includes the Application Server Toolkit.

Installing the Express product offering on the same machine as the Rational Web Developer is not necessary. The IDE contains a replica of the Version 6 Application Server as a test environment.

### WebSphere Application Server

This product offering addresses the basic programming and run-time needs of desktop developers and single-server production scenarios. The run-time environment addresses standards-based programming for Web and component-based programming, as well as Web services.

The administration model is a single-server environment without clustering, and without centralized administration of multiple server instances.

The development environment offering is the Application Server Toolkit and a trial version of the Rational Application Developer product. The Rational Application Developer is an enhanced version of the WebSphere Studio Application Developer in Version 5.

### WebSphere Application Server Network Deployment

This product offering addresses WebSphere Application Server processes that run in multiple-server production scenarios. The Network Deployment product can create:

- Deployment managers

The deployment manager provides centralized administration of managed Application Server nodes and custom nodes as a single cell. The deployment manager provides basic clustering and caching support, including failover support and workload balancing. Use the Edge components in the Network Deployment package to further extend the environment with sophisticated caching, load balancing, and enhanced security functions.

- Application Servers

The Application Server in the Network Deployment product can run in a deployment manager cell as a managed node or on its own as a stand-alone Application Server.

- Custom nodes

A custom node is an empty node until you add it to the deployment manager cell for customization. Use the administrative console of the deployment manager to create servers and clusters on the custom managed node. Consider the custom node as a production-ready shell, ready for customization to contain your production-level servers and applications.

See “Planning to install Network Deployment” on page 54 for examples of common topologies that you can create with the Network Deployment product.

## New installation and operating model

Version 6 separates the core product binary data from the data that you create as you define and customize your Application Server run-time environment. You can have multiple sets of customized run-time environments. Each environment is known as a *profile* in Version 6.

### Application Server environments that are created during product installation

Product	Default environments	Application Server instances created during installation
WebSphere Application Server - Express	One stand-alone Application Server profile and the Profile creation wizard for creating other Application Server profiles on the same machine.	server1
WebSphere Application Server	One stand-alone Application Server profile and the Profile creation wizard for creating other Application Server profiles on the same machine.	server1
Network Deployment	A default run-time environment is not created during installation of the core product files.	From the last installation panel, you can launch the Profile creation wizard to create one of the three available profiles: <ul style="list-style-type: none"> <li>• Deployment manager profile</li> <li>• Application Server profile</li> <li>• Custom profile</li> </ul>

Install the Network Deployment product on each machine in a multiple-machine, distributed environment to give each server process access to a local copy of the core product files.

## New programming model extensions

Many programming model extensions that were formerly part of the Enterprise (V5.0.x) product or the WebSphere Business Integration Server Foundation (V5.1.x) product are now integrated into Version 6.

Among the applications are:

- Object pool service
- Scheduler
- Last participant support
- Startup beans
- Application profiles
- Work area partition service
- Internationalization service
- Work manager
- ActivitySession service
- Timer manager
- Compensation service



## New and improved launchpad console

The launchpad is the focal point for installing from the product compact disc, with important guidance for creating typical topology scenarios. Diagrams help you identify the types of systems that you can create. The launchpad is in the root directory of the product CD.

The launchpad works on installable components in the primary packet of compact discs. Select and launch any of the components on the product disc or view their HTML documentation on the disc.

Use the **launchpad** command if the program does not start automatically when you insert the product disc:

-   ./launchpad.sh
-  launchpad.bat



See “Using the launchpad to start the installation” on page 41 for more information.

If you cannot start the launchpad, see the appropriate locale-specific readme file in the root directory on the disc. You can also launch the installation programs directly, using the information in the following table:

Table 1. Installable components on the product CD

Launch command	Documentation	Name of installable component
WAS/install	WAS/readme/ readme_en.html WAS/docs/ InstallGuide_en.html	The name of the component varies per product offering: <ul style="list-style-type: none"><li>• WebSphere Application Server – Express</li><li>• WebSphere Application Server</li><li>• WebSphere Application Server Network Deployment</li></ul>
IHS/install	IHS/readme/ readme_en.html IHS/docs/ InstallGuide_en.html	IBM HTTP Server
plugin/install	plugin/readme/ readme_en.html plugin/docs/ InstallGuide_en.html	Web server plug-ins for WebSphere Application Server
AppClient/install	AppClient/readme/ readme_en.html AppClient/docs/ InstallGuide_en.html	Application Client for WebSphere Application Server

Other directories on the product CD contain components that are automatically installed during one or more of the installation procedures for the installable components.

  WebSphere Application Server products require that you log on as root on a Linux or UNIX operating system to install. Set the umask to 022 for security purposes after logging on as root before you install:

```
umask 022
```

Set file permissions to 755 after installation, as in the following example for one file:

```
chmod 755 setupCmdLine.sh
```

The IBM HTTP Server tolerates a non-root installation, but you cannot install the Global Services Kit 7 (GSKit 7) that is required for HTTPS security. You can install IBM HTTP Server as a non-root user, then log on as root to install the GSKit program from the directory on the product disc.

## Version 6 package descriptions

Each product package contains three sets of compact discs, which are the primary packet and two supplemental packets. The product CD is always in the primary packet. The product packages and their installable components are described in detail in the following sections:

- Packaging information for IBM WebSphere Application Server - Express
- Packaging information for IBM WebSphere Application Server
- Packaging information for IBM WebSphere Application Server Network Deployment




<b>Media/CD (below):</b>	<b>IBM WebSphere Application Server-Express</b> 	<b>IBM WebSphere Application Server</b> 	<b>IBM WebSphere Application Server Network Deployment</b> 
<b>Product disc in Primary disc packet</b>	"WebSphere Application Server-Express"	"WebSphere Application Server"	"WebSphere Application Server Network Deployment"
<b>Tools disc in Primary disc packet</b>	"Rational Web Developer" (single use)	"Application Server Toolkit" (unlimited use)	"Application Server Toolkit" (unlimited use)
<b>Tools disc in Supplemental disc packet</b>	"Application Server Toolkit" (unlimited use)	"Rational Application Developer" (trial)	"Rational Application Developer" (trial)
<b>DB2 disks in Supplemental disc packet</b>	"DB2 Express"	"DB2 Express"	"DB2 Universal Database V8.2"

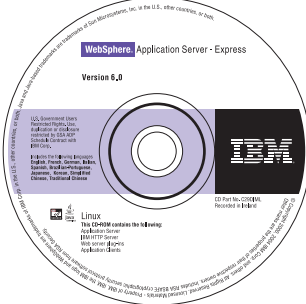
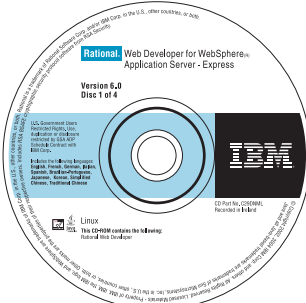
Figure 1. Installation images in WebSphere Application Server packages for Version 6. The graphic shows the packaging for Version 6, which supports the Java 2 Platform, Enterprise Edition (J2EE) 1.4 specification and the Java 2 SDK 1.4.2 specification.

**Packaging information for IBM WebSphere Application Server - Express:** The Express package contains three sets of compact discs:


- Compact discs in the primary Express packet
- Compact discs in the two supplemental packets



The product disc for the WebSphere Application Server - Express product is in the primary package.

**Compact discs in the primary Express packet:** The primary Express packet contains the compact discs that are listed in the following table:

Disc label	Versions of the disc	Comments
<p><b>WebSphere Application Server - Express</b></p> 	<p>Six different platform versions of this CD are available:</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• HP-UX</li> <li>• Linux</li> <li>• Linux on iSeries and pSeries</li> <li>• Solaris</li> <li>• Windows</li> </ul>	<p>Each platform-specific CD contains the following installable components:</p> <ul style="list-style-type: none"> <li>• WebSphere Application Server - Express in the WAS directory.</li> <li>• IBM HTTP Server in the IHS directory.</li> <li>• Web server plug-ins in the plugin directory.</li> <li>• Application Clients in the Application directory.</li> </ul> <p>In addition, the Windows platform CD has the Data Direct Java DataBase Connectivity (JDBC) Drivers.</p> <p>The launchpad application in the root directory lets you install and view information about any of the installable components in the primary packet.</p>
<p><b>Rational Web Developer for WebSphere Application Server</b></p> 	<p>Four compact discs are available for each of the following platforms:</p> <ul style="list-style-type: none"> <li>• Linux (Intel)</li> <li>• Windows</li> </ul>	<p>The license for Rational Web Developer is a single-use license.</p> <p>See the Application Server Toolkit in the supplemental package for an integrated development environment that includes an unlimited use license.</p>

*Compact discs in the two supplemental Express packets:* The supplemental Express packets contain the compact discs that are listed in the following table:

Disc label	Versions of the disc	Comments
<p><b>DB2 Universal Database Express Edition V8.2</b></p> 	<p>Three different platform versions of this CD are available:</p> <ul style="list-style-type: none"> <li>• Linux (Intel)</li> <li>• Windows 2000 Server</li> <li>• Windows 2003 Server</li> </ul>	<p>IBM DB2 Universal Database Express Edition is the lowest priced full-function relational database.</p> <p>DB2 UDB Express Edition is designed to meet the needs of small and medium businesses.</p> <p>DB2 UDB Express Edition features self-tuning and self-configuring capabilities for ease of manageability.</p> <p>Available for Linux and Windows platforms, DB2 supports transparent installation with business partner applications that you might develop.</p> <p>See IBM DB2 Universal Database Express Edition for more information.</p>

Disc label	Versions of the disc	Comments
<p>IBM Business Solutions for WebSphere Application Server - Express</p> 	<p>Six different platform versions of this CD are available:</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• HP-UX</li> <li>• Linux (Intel)</li> <li>• Linux on pSeries</li> <li>• Solaris</li> <li>• Windows</li> </ul>	<p>This disc contains a production-ready application, which is the IBM Telephone Directory application.</p>
<p>WebSphere Application Server - Express</p>  <p>Application Server Toolkit</p>	<p>Two different platform versions of this CD are available:</p> <ul style="list-style-type: none"> <li>• Linux (Intel)</li> <li>• Windows</li> </ul>	<p>The license for the Application Server Toolkit is an unlimited use license.</p>

**Packaging information for IBM WebSphere Application Server:** The base WebSphere Application Server package contains three sets of compact discs:


- Compact discs in the primary Application Server packet
- Compact discs in the two supplemental packets

**Compact discs in the primary Application Server packet:** The primary Application Server packet contains the compact discs that are listed in the following table:

Disc label	Versions of the disc	Comments
<b>WebSphere Application Server</b> 	Seven different platform versions of this CD are available: <ul style="list-style-type: none"> <li>• AIX</li> <li>• HP-UX</li> <li>• Linux</li> <li>• Linux on iSeries and pSeries</li> <li>• Linux on zSeries</li> <li>• Solaris</li> <li>• Windows 2000 Server and Windows 2003 Server</li> </ul>	Each platform-specific CD contains the following installable components: <ul style="list-style-type: none"> <li>• WebSphere Application Server in the WAS directory.</li> <li>• IBM HTTP Server in the IHS directory.</li> <li>• Web server plug-ins in the plugin directory.</li> <li>• Application Client in the AppClient directory.</li> </ul> <p><b>Windows</b> In addition, the Windows platform CD has the Data Direct Java Database Connectivity (JDBC) drivers.</p> <p>Use the launchpad application in the root directory to install and view information about any of the installable components in the primary packet.</p>
<b>WebSphere Application Server</b>  <b>Application Server Toolkit</b>	Two different platform versions of this CD are available: <ul style="list-style-type: none"> <li>• Linux (Intel)</li> <li>• Windows</li> </ul>	The license for the Application Server Toolkit is an unlimited use license.

**Compact discs in the two supplemental packets:** The supplemental Application Server packets contain the compact discs that are listed in the following table:

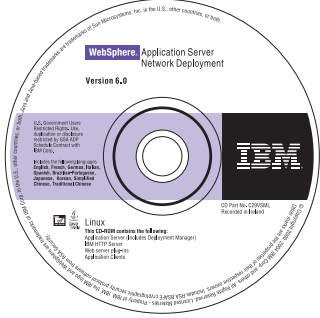
Disc label	Versions of the disc	Comments
<b>Rational Application Developer Trial for WebSphere Application Server</b> 	Three compact discs are available for Linux (Intel) platforms and four compact discs are available for Windows platforms.	Rational Application Developer is the next generation of the award-winning IBM WebSphere Studio Application Developer. <p>Rational Application Developer is a comprehensive integrated development environment for visually designing, constructing, testing, and deploying Web services, portals, and Java 2 Platform Enterprise Edition (J2EE) applications.</p> <p>Rational Application Developer accelerates J2EE development with a complete set of high-productivity tools, templates, and wizards.</p> <p>See the Application Server Toolkit in the primary package for an integrated development environment that includes an unlimited use license.</p>

Disc label	Versions of the disc	Comments
<p>DB2 Universal Database Express Edition V8.2</p> 	<p>Three different platform versions of this CD are available:</p> <ul style="list-style-type: none"> <li>• Linux (Intel)</li> <li>• Windows 2000 Server</li> <li>• Windows 2003 Server</li> </ul>	<p>IBM DB2 Universal Database Express Edition is the lowest-priced, full-function relational database.</p> <p>DB2 UDB Express Edition is designed to meet the needs of small and medium businesses.</p> <p>DB2 UDB Express Edition features self-tuning and self-configuring capabilities for ease of manageability, Linux or Windows platform choices, and support for transparent installation with business partner applications.</p> <p>See IBM DB2 Universal Database Express Edition for more information.</p> <p>This DB2 Express Edition is for use with WebSphere Application Server only.</p>

**Packaging information for IBM WebSphere Application Server Network Deployment:** The Network Deployment package contains three sets of compact discs:

- Compact discs in the primary Network Deployment packet
- Compact discs in the two supplemental packets

**Compact discs in the primary Network Deployment packet:** The primary Network Deployment packet contains the compact discs that are listed in the following table:

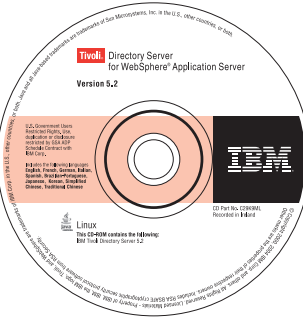

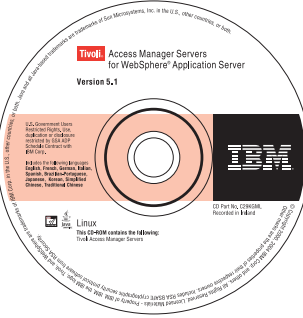
Disc label	Versions of the disc	Comments
<p>WebSphere Application Server Network Deployment</p> 	<p>Seven different platform versions of this CD are available:</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• HP-UX</li> <li>• Linux</li> <li>• Linux on iSeries and pSeries</li> <li>• Linux on zSeries</li> <li>• Solaris</li> <li>• Windows 2000 Server and Windows 2003 Server</li> </ul>	<p>Each platform-specific CD contains the following installable components:</p> <ul style="list-style-type: none"> <li>• Network Deployment in the WAS directory.</li> <li>• IBM HTTP Server in the IHS directory.</li> <li>• Web server plug-ins in the plugin directory.</li> <li>• Application Client in the AppClient directory.</li> </ul> <p>In addition, the Windows platform CD has the Data Direct Java Database Connectivity (JDBC) drivers.</p> <p>Use the launchpad application in the root directory to install and view information about any of the installable components in the primary packet.</p>

Disc label	Versions of the disc	Comments
<p>WebSphere Application Server Network Deployment</p> <p>Application Server Toolkit</p>	<p>One CD is available for each of the following platforms:</p> <ul style="list-style-type: none"> <li>Windows 2000</li> <li>Linux (Intel)</li> </ul>	<p>The license for the Application Server Toolkit is an unlimited use license.</p>

*Compact discs in the two supplemental packets:* The supplemental Network Deployment packets contain the compact discs that are listed in the following table:

Disc label	Versions of the disc	Comments
<p>WebSphere Application Server Network Deployment</p> <p>Edge Components</p>	<p>Seven different platform versions of this CD are available:</p> <ul style="list-style-type: none"> <li>AIX</li> <li>HP-UX</li> <li>Linux (Intel)</li> <li>Linux on pSeries</li> <li>Linux on zSeries</li> <li>Solaris</li> <li>Windows</li> </ul>	<p>Network Deployment addresses the needs of highly available, high-volume environments with the Edge components.</p> <p>The Edge components include sophisticated load balancing, caching, and centralized security capabilities.</p> <p>See WebSphere Application Server Network Deployment Edge components for more information.</p>
<p>DB2 Universal Database Enterprise Server Edition for WebSphere Application Server Network Deployment</p>	<p>Nine different platform versions of this CD are available:</p> <ul style="list-style-type: none"> <li>AIX 5.2</li> <li>AIX 5.3</li> <li>AIX for DBCS platforms</li> <li>HP-UX</li> <li>Linux (Intel)</li> <li>Linux on pSeries</li> <li>Linux on zSeries</li> <li>Solaris</li> <li>Windows</li> </ul>	<p>DB2 Universal Database V8.2 has numerous innovative enhancements.</p> <p>Improved integration with WebSphere Application Server Network Deployment, key development tools, and platforms help make programmers more efficient than ever.</p> <p>A broad array of autonomic or self-managing capabilities can free more administrator time to focus on driving business value.</p> <p>The ease of use in DB2 V8.2 and the self-managing characteristics might even eliminate the need for dedicated administrators in smaller implementations.</p>



Disc label	Versions of the disc	Comments
<p>Tivoli Directory Server for WebSphere Application Server</p> 	<p>Seven different platform versions of this CD are available:</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• HP-UX</li> <li>• Linux (Intel)</li> <li>• Linux on pSeries</li> <li>• Linux on zSeries</li> <li>• Solaris</li> <li>• Windows</li> </ul>	<p>The IBM Tivoli Directory Server product is a powerful Lightweight Directory Access Protocol (LDAP) infrastructure.</p> <p>Tivoli Directory Server provides a foundation for deploying comprehensive identity management applications and advanced software architectures.</p> <p>See IBM Tivoli Directory Server for more information.</p>
<p>Rational Application Developer Trial for WebSphere Application Server</p> 	<p>Three CDs for each of two different platform versions are available:</p> <ul style="list-style-type: none"> <li>• Linux</li> <li>• Windows</li> </ul>	<p>Rational Application Developer is the next generation of the award-winning IBM WebSphere Studio Application Developer.</p> <p>Rational Application Developer is a comprehensive integrated development environment for visually designing, constructing, testing, and deploying Web services, portals, and Java 2 Platform Enterprise Edition (J2EE) applications.</p> <p>Rational Application Developer accelerates J2EE development with a complete set of high-productivity tools, templates, and wizards.</p> <p>See the Application Server Toolkit in the primary package for an integrated development environment that includes an unlimited use license.</p>
<p>IBM Tivoli Access Manager Servers for WebSphere Application Server</p> 	<p>Six different platform versions of this CD are available:</p> <ul style="list-style-type: none"> <li>• AIX</li> <li>• HP-UX</li> <li>• Linux (Intel)</li> <li>• Linux on zSeries</li> <li>• Solaris</li> <li>• Windows</li> </ul>	<p>IBM Tivoli Access Manager for e-business integrates with e-business applications right out of the box, to deliver a secure, unified, and personalized e-business experience.</p> <p>By providing authentication and authorization APIs and integration, Tivoli Access Manager for e-business helps you secure access to business-critical applications and data that might be spread across the extended enterprise.</p> <p>See IBM Tivoli Access Manager for e-business for more information.</p>

## Hardware and software requirements

This topic describes prerequisites for installing WebSphere Application Server products.






The official statements of support for WebSphere Application Server products are provided on the Supported hardware and software Web page at the <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> IBM Web address. If there is a



conflict between the information provided in this document and the information on the Supported hardware and software Web address, the information at the Web site takes precedence. The information in this document is provided as a convenience only.

## Required disk space

Disk space requirements are platform-specific. See the following topics for information about required disk space and how to prepare your operating system for installation:

-  “Preparing AIX systems for installation” on page 81
-  “Preparing HP-UX systems for installation” on page 85
-  “Preparing Linux systems for installation” on page 90
-  “Preparing Solaris systems for installation” on page 94
-  “Preparing Windows systems for installation” on page 97

Additional space is required for the installable components in the secondary packet. Refer to the documentation for each installable component to determine exact space requirements.

## Supported operating systems

The official statements of support for the WebSphere Application Server products are provided on the Supported hardware and software Web page at the <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html> IBM Web address.

The installation programs for WebSphere Application Server products verify that a supported operating system is installed. The verification includes checking for required patches with the prereqChecker application.

Although the Installation wizard checks for prerequisite operating system patches, review the prerequisites on the IBM WebSphere Application Server supported hardware, software, and APIs Web site if you have not already done so. Refer to the documentation for non-IBM prerequisite and corequisite products to learn how to migrate to their supported versions.

Always consult the IBM WebSphere Application Server supported hardware, software, and APIs Web site to determine whether your operating system is supported when you receive a message from the prereqChecker program. The Web site lists all supported operating systems and the operating system fixes and patches that you must install to have a compliant operating system.

---

## Using the launchpad to start the installation

The launchpad console is the starting point for installing WebSphere Application Server products.

The launchpad is a Web application. Before using the launchpad, you must have a supported Web browser. The launchpad supports the following browsers:

- Mozilla
- Internet Explorer

The launchpad program is available on the root directory of the product CD in a program named:

-   1 launchpad.sh
-  1 launchpad.bat

Start the launchpad if it does not launch automatically.

Use this topic to learn how to use the launchpad.

WebSphere Application Server is an integrated platform that contains an Application Server, a set of Web development tools, a Web server, and additional supporting software and documentation. The launchpad is the single point of reference for installing the entire Application Server environment.

The launchpad opens in the language of the machine locale setting. The launchpad identifies components on the product disc that you can launch.

The launchpad can install the tools component in the primary packet of discs. The launchpad on the separate Application Server Toolkit disc can launch the installation of the tool on Windows 2000 and Linux (Intel) systems.

The launchpad has diagrams that show the possible application server topologies that you can create with all of the components in the primary packet.

A link exists for each installable component on the product compact disc and for the installable tools component that is on a separate disc in the primary packet of compact discs in the product package.

There is no launchpad for the compact discs in the secondary packet in the product package, such as the DB2 product. See “Packaging” on page 30 for more information about the installable components in each package.

For example, the installation links on the initial launchpad panel for the Network Deployment product are:








- The Welcome panel with its fastpath links that launch the installer programs for the following installable components:
  - Launch the installation wizard for WebSphere Application Server Network Deployment
  - Launch the installation wizard for IBM HTTP Server
  - Launch the installation wizard for Web server plug-ins
  - Launch the installation wizard for WebSphere Application Server Clients
  - Launch the installation wizard for the Application Server Toolkit
- A navigation frame on the left side of the launchpad has the following links:
  - Solution installation diagrams
  - WebSphere Application Server Network Deployment installation
  - IBM HTTP Server installation
  - Web server plug-ins installation
  - WebSphere Application Server Clients installation
  - Application Server Toolkit installation

The launchpad panel for each installable component in the primary packet includes a link to the installation program for the component and to documentation that describes the product, how to install it, and how to configure it for use.




For example, the links on the launchpad panel for the Network Deployment installable component are:

- Launch the Installation wizard for WebSphere Application Server Network Deployment
- View the product overview for WebSphere Application Server Network Deployment
- View the installation guide for WebSphere Application Server Network Deployment
- View the readme file for WebSphere Application Server Network Deployment

Perform the following procedure to use the launchpad.

1. Log on as root on a Linux or UNIX system, or as a member of the administrator group on a Windows system.
2. Install the Internet Explorer Web browser or the Mozilla browser if neither browser is installed.
  - a.  Install the two prerequisite packages that Mozilla on AIX requires and install Mozilla, as described in “Preparing AIX systems for installation” on page 81.
  - b.  Download the latest supported version of Internet Explorer. Download Internet Explorer from the following location:  
http://www.microsoft.com/windows/ie/default.mspx
  - c.    Download and install the Mozilla Web browser. If you do not have the Mozilla Web browser, download and install the browser from http://www.mozilla.org.
  - d.   Export the location of the supported browser. Export the location of the supported browser using the following command:  
EXPORT BROWSER=/usr/bin/mozilla
3. Start the launchpad.
 

You can start the launchpad manually using a fully qualified command instead of changing directories to the CD-ROM and running the command locally from the root directory:

  - Mount the CD-ROM drive on Linux or UNIX platforms, if necessary. This procedure varies per platform. See “Mounting CD-ROMS on Linux and UNIX operating systems” on page 133.
  - Open a shell window and issue a fully qualified command to start the launchpad:
    -   **launchpad.sh**
    -  **launchpad.bat**
4. Use the launchpad to access the product overview, the readme file, and the installation guide.
5. Click **Launch the installation wizard for ...** to launch the installation wizard for the installable component that you select.

You can use the launchpad to start the installation and to access information through a browser.

## Troubleshooting

If you can start the launchpad, but clicking a link does not resolve to a page in the launchpad, you might have the media for the wrong operating system in the CD drive. Check the validity of the media. Otherwise, use the following procedure. If the links do not work at the end of the procedure, open the readme file for reference and launch the installation programs directly.

If you cannot start the launchpad, use the following procedure to correct any error that is preventing the launchpad from displaying, then start the launchpad again:

1. Install Mozilla or Internet Explorer as described in the procedure.
2. If the product CD is no longer accessible, insert the CD.  
Mount the drive as necessary on Linux and UNIX platforms.
3. Enable the JavaScript function in your browser.

  Mozilla: Click **Edit > Preferences > Advanced > Scripts & Plugins**:

- Enable JavaScript for: Navigator.
- Allow scripts to select all boxes.

 Internet Explorer: Click **Tools > Internet Options > Security > Custom Level for Internet > Scripting > Active scripting > Enable**.

4. Restart the launchpad by issuing the following command:

-   `./launchpad.sh`
-  `launchpad.bat`

If you still cannot use the launchpad console, access the readme file in the root directory of the product disc or launch each installer program directly. You can launch the following links directly:

### Welcome page links

Fastpath links to installable components on the product disc are provided in the following table:

Installable component	Directory	Program
WebSphere Application Server Network Deployment	WAS	install
IBM HTTP Server	IHS	install
Web server plug-ins	plugin	install
Application clients	AppClient	install

The fastpath link to the installable component on the Application Server Toolkit disc is as follows:  
Application Server Toolkit: setup (in the root directory)

Mount the CD for the Application Server Toolkit and use the launchpad on that disc.

### Solution installation diagrams

See the installation scenarios in “Planning to install Network Deployment” on page 54.

### WebSphere Application Server Network Deployment installation

Links on the page include:

#### Launch the installation wizard...

Installable component	Directory	Program
WebSphere Application Server Network Deployment	WAS	install

#### View the product overview...

[/WAS/docs/InstallGuide\\_en.html#over](#)

This is a section of the *Getting Started* guide. Language versions of this information are available only in the online Information center in “Packaging” on page 30.

#### View the installation guide...

[/WAS/docs/InstallGuide\\_en.html#install](#)

This is the *Getting Started* guide. Language versions of the information in the *Getting Started* guide are available only in the online information center in Chapter 9, “Installing the product and additional software,” on page 107.

#### View the readme file...

[/WAS/readme/readme\\_en.html](#)

The *Getting Started* guide is also available in Adobe PDF format. To view this document, you must have Adobe Acrobat Reader, which is a free download from the Adobe Acrobat Web site.

Download the installation guide from:

[ftp://ftp.software.ibm.com/software/webserver/appserv/library/wasv60nd\\_gs.pdf](ftp://ftp.software.ibm.com/software/webserver/appserv/library/wasv60nd_gs.pdf)

### IBM HTTP Server installation

Links on the page include:

### Launch the installation wizard...

Installable component	Directory	Program
IBM HTTP Server	IHS	install

#### View the installation guide...

/IHS/docs/InstallGuide\_en.html

#### View the readme file...

/IHS/readme/readme\_en.html

### Web server plug-ins installation

Links on the page include:

### Launch the installation wizard...

Installable component	Directory	Program
Web server plug-ins	plugin	install

#### View the installation roadmap...

/plugin/index\_roadmap\_en.html

See “Selecting a Web server topology diagram and roadmap” on page 139.

#### View the installation guide...

/plugin/docs/InstallGuide\_en.html

See “Installing Web server plug-ins” on page 136.

#### View the readme file...

/plugin/readme/readme\_en.html

See “Configuration behavior of the Plug-ins installation wizard” on page 145 for an overview of how Web server configuration works.

Return to Chapter 9, “Installing the product and additional software,” on page 107 to continue.

---

## Product version information

The WebSphere Application Server product contains structural differences from previous versions. The `properties/version` directory in the `install_root` contains important data about the product and its installed components, such as the build version and build date. This information is included in `WAS.product` and `[component].component` files.

This topic describes the XML data files that store product information for Version 6 WebSphere Application Server products. By default, the document type declarations (DTDs) for these files are in the `properties/version/dtd` folder of the `install_root`, or the server root directory. See the “Directory locations” on page 46 section for more information.

This topic includes the following sections:

- A list of product information files
- A list of reports for displaying version information
- A list of directory locations

### Product information files

**XML files in the `properties/version` directory that store version information:**

## platform.websphere

One file whose existence indicates that a WebSphere Application Server product is installed. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE websphere PUBLIC "websphereId" "websphere.dtd">
<websphere name="IBM WebSphere Application Server" version="6.0"/>
```

The following XML files in the `properties/version` directory represent installed items and installation events such as product edition, version, component, and build information.

### WAS.product

One file whose existence indicates the particular WebSphere Application Server product that is installed. The type of product installed is indicated by the `<id>` tag. Data in the file indicates the version, build date, and build level.

For example, `<id>ND</id>.product` indicates that the installed product is WebSphere Application Server Network Deployment. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE product PUBLIC "productId" "product.dtd">
<product name="IBM WebSphere Application Server - ND">
  <id>ND</id>
  <version>6.0.0</version>
  <build-info date="02/03/05" level="s0461.18"/>
</product>
```

### component-name.component

Any number of component files that each indicate the presence of an installed component, which is part of the product. Data in the file indicates the component build date, build version, component name, and product version. For example, the file might be the `activity.component` file, which indicates that the **activity** component is installed. The activity component is part of the Network Deployment product. An example of the file follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE component PUBLIC "componentId" "component.dtd">
<component build-date="02/03/05" build-version="s0461.18"
  name="activity" spec-version="6.0"/>
```

## Reports

WebSphere Application Server provides the ability to generate **Version** reports from the data in the files. The following report-generation scripts are available in the `install_root/bin` directory.

### Product version reports

The following report generation scripts extract data from XML data files in the `properties/version` folder:

- `versionInfo` script  
Lets you use parameters to create a version report on Linux and UNIX platforms, or on Windows platforms.
- `genVersionReport` script  
Generates the `versionReport.html` report file in the `bin` directory on Linux and UNIX platforms, or on Windows platforms. The report includes the list of components.

### Directory locations

Product information files are located relative to the WebSphere Application Server product `install_root`, or the server root directory.

Default file paths are:

#### Version directory

`install_root/properties/version`

## DTD directory

*install\_root*/properties/version/dtd

## Temporary directory

Specified by the java.io.tmpdir Java system property

## versionInfo command

The **versionInfo** command generates reports from data it extracts from XML files in the *properties/version* folder.

### Product version information

The */properties/version* directory in the installation root contains important data about the product and its installed components, such as the build version and build date. This information is included in *WAS.product* and *component.component* files.

**Restriction:** There is one restriction. Do not use the *versionInfo.sh* or *versionInfo.bat* script while installing or uninstalling the product.

### Product version reports

The following report generation scripts extract data from XML data files in the *properties/version* folder:

- *versionInfo* script  
Lets you use parameters to create a version report on Linux and UNIX platforms, or on Windows platforms.
- *genVersionReport* script  
Generates the *versionReport.html* report file in the *bin* directory on Linux and UNIX platforms, or on Windows platforms. The report includes the list of components.

### Location of the command file

The command file is a script. On Linux and UNIX platforms, the command file is named *versionInfo.sh* in the *install\_root/bin* directory. On Windows platforms, the command file is named *versionInfo.bat* in the *install\_root\bin* directory.

### Syntax for the versionInfo command on a Linux or UNIX platform

The command syntax is:

```
versionInfo.sh [ -format text | html ]  
               [ -file output file ]  
               [ -long ]  
               [ -efixes ]  
               [ -efixDetail ]  
               [ -ptfs ]  
               [ -ptfDetail ]  
               [ -components ]  
               [ -componentDetail ]
```

```
versionInfo.sh [ -help | -? | /help | /? | -usage ]
```

Issue the command from the *install\_root/bin* directory.

### Syntax for the versionInfo command on a Windows platform

The command syntax is:

```
versionInfo [ -format text | html]
            [ -file output file]
            [ -long ]
            [ -efixes ]
            [ -efixDetail ]
            [ -ptfs ]
            [ -ptfDetail ]
            [ -components ]
            [ -componentDetail ]
```

```
versionInfo [ -help | -? | /help | /? | -usage ]
```

Issue the command from the *install\_root*\bin directory.

## Parameters

### **-? or /? (Windows only)**

Displays command syntax.

### **-components**

Adds a list of installed components to the report.

### **-componentDetail**

Adds details about installed components to the report.

### **-efixes**

Adds a list of applied interim fixes to the report.

### **-efixDetail**

Adds details about applied interim fixes to the report.

### **-file *fileName***

Specifies the output file name. The report goes to standard output (stdout) by default.

### **-format text | html**

Selects the format of the report. The default is "text".

### **-help or /help (Windows only)**

Displays command syntax.

### **-long**

Creates the long version of the report.

### **-ptfDetail**

Adds details about applied fix packs to the report.

### **-ptfs**

Adds a list of applied fix packs to the report.

### **-usage**

Displays command syntax.

## Logging

As the tool runs, it creates reports instead of log entries. Reports appear on the console unless directed to a file with the *-file filename* parameter. There is no default file name. You must specify a file name to generate a file.

## Creating the report

This example shows how to issue the command on a Windows platform, from the bin directory of the product:



```
D:\Program Files\IBM\WebSphere\AppServer\bin>versionInfo
WVER0010I: Copyright (c) IBM Corporation 2002; All rights reserved.
WVER0011I: WebSphere Application Server Release 6.0
WVER0012I: VersionInfo Reporter Version 1.15, Dated 9/20/03
```

## genVersionReport command

The **genVersionReport** command generates the `versionReport.html` report file in the `bin` directory on Linux and UNIX platforms, or on Windows platforms.

### Product version information

The `/properties/version` directory in the installation root contains important data about the product and its installed components, such as the build version and build date. This information is included in `WAS.product` and `[component].component` files.

### Product version reports

The following report generation scripts extract data from XML data files in the `properties/version` folder:

- `versionInfo` script  
Lets you use parameters to create a version report on Linux and UNIX platforms, or on Windows platforms.
- `genVersionReport` script  
Generates the `versionReport.html` report file in the `bin` directory on Linux and UNIX platforms, or on Windows platforms. The report includes the list of components, fixes, and fix packs.

### Location of the command file

The command file is a script. On Linux and UNIX platforms, the command file is named `genVersionReport.sh` in the `install_root/bin` directory. On Windows platforms, the command file is named `genVersionReport.bat` in the `install_root\bin` directory.

### Syntax for the versionInfo command on a Linux or UNIX platform

The command syntax is:

```
genVersionReport.sh
```

Issue the command from the `install_root/bin` directory.

### Syntax for the versionInfo command on a Windows platform

The command syntax is:

```
genVersionReport.bat
```

Issue the command from the `install_root\bin` directory.

### Logging

As the tool runs, it creates the `versionReport.html` report file instead of log entries.

### Creating the report

This example shows how to issue the command on a Windows platform, from the `bin` directory of the product:

D:\Program Files\IBM\WebSphere\AppServer\bin>genVersionReport  
WVER0010I: Copyright (c) IBM Corporation 2002; All rights reserved.  
WVER0011I: WebSphere Application Server Release 6.0  
WVER0012I: VersionInfo Reporter Version 1.15, Dated 9/20/03

---

## Installation: Resources for learning

Use the following links to find relevant supplemental information about installation and customization. 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 Redbooks that supplement the broad coverage of the release documentation with in-depth examinations of particular product areas.

One important link is:

How to buy WebSphere Application Server software

This IBM Web site describes pricing and technical details. If you have already purchased the software, view links to additional information about:

- Planning, business scenarios, and IT architecture
- Programming instructions and examples
- Programming specifications
- Administration
- Support

### Planning, business scenarios, and IT architecture

- IBM WebSphere Application Server supported hardware, software, and APIs

The official site for determining product prerequisites for hardware, software and APIs for all WebSphere Application Server products.

- IBM developerWorks WebSphere

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.

- IBM WebSphere Application Server library and information centers Web site

The IBM WebSphere Application Server Library Web site contains links to all WebSphere Application Server information centers, for all versions. It also lets you access each information center in your native language.

- IBM WebSphere Application Server home page

The IBM WebSphere Application Server home page contains useful information, including support links and downloads for fixes, APARs, tools, and trials.

- IBM WebSphere software platform home page

The IBM WebSphere software platform home page introduces WebSphere products and describes how companies can easily transform to an e-business, with software that can grow as fast as the business it supports.

- Information center for WebSphere Application Server Edge components

The information center for WebSphere Application Server Edge components contains complete documentation for the Caching Proxy and the Load Balancer in these PDF online books, *WebSphere Application Server Concepts, Planning, and Installation for Edge Components*, *WebSphere Application*

*Server Caching Proxy Administration Guide, WebSphere Application Server Load Balancer Administration Guide, and WebSphere Application Server Programming Guide for Edge Components.*

- developerWorks: IBM Patterns for e-business

The IBM developerWorks site is the source for IBM patterns for e-business, a set of tested, reusable intellectual assets that you can use to design and implement your e-business network and architecture.

- The User centered design (UCD) for different project types, part 2

This Web page is the latest of two articles that describes design activities that IBM scientists have found most useful in various types of projects. This article defines user interface design elements, including the design prototype, use case model, and design specification document.

### **Programming instructions and examples**

- IBM developerWorks

IBM developerWorks contains many excellent resources for developers, including tutorials on Web development-related topics. There is an excellent tutorial on the JDBC API.

- IBM Redbooks

The IBM Redbooks site contains many documents that are related to WebSphere Application Server.

### **Programming specifications**

- J2EE information

For more information about J2EE specifications, visit the Sun site.

### **Administration**

- Best Practices Zone on developerWorks

The WebSphere Best Practices Zone is a collection of best practices for administering WebSphere Application Server. Over time, the zone is intended to grow to include best practices for using other WebSphere software products, and to cover more topics. Use the feedback mechanism to submit your best practice suggestions.

- The IBM Glossary of Computing Terms

This glossary defines technical terms used in many IBM products. It is not a comprehensive resource of all IBM computing terms. This resource is provided for information purposes only and is updated periodically. IBM takes no responsibility for the accuracy of the information it contains.

### **Support**

- Ten Steps to Getting Support for WebSphere Application Server

If you are new to a product, you might have difficulty finding all the information you need. And if you come across a problem, where do you go for help? Whether you are a new user looking for introductory information, or an experienced user looking for a workaround for a specific defect, you can benefit immediately from extensive Web-based support from IBM. You can download fix packs, search on keywords, look up FAQs, Hints and Tips, and so forth. Always use this Web resource before contacting IBM Support directly.

- 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-3.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 displays.

- IBM e-server Support: Fix Central

A Web facility for downloading fixes for the AIX operating system and the OS/400 operating system.

- Adobe Acrobat Web site

This Adobe Web site offers a free download of the Adobe Acrobat Reader product.



---

## Chapter 7. Planning the installation (diagrams)

This topic describes common product topologies that you can install with the product.

Use this topic to understand the capabilities of your product package. Knowing what you can do with the product might influence how you install the product and other installable components on the product disc.

This topic describes topology diagrams and shows you how to create the topologies by showing what components to install for each topology.

### Phased installation roadmap

To install a Version 6 production environment, you install the following components:

- The WebSphere Application Server product on your product CD
- A supported Web server, such as the IBM HTTP Server V6 on the product CD
- A binary plug-in module for your Web server from the product CD

You can also use the product CD to install an application client environment on a client machine. Running Java 2 Platform, Enterprise Edition (J2EE) and thin application clients that communicate with WebSphere Application Server requires that elements of the Application Server are installed on the machine on which the client runs. However, if the machine does not have the Application Server installed, you can install Application Server clients to provide a stand-alone client run-time environment for your client applications.

You can use the Application Server Toolkit CD in the primary packet of discs to install a development environment. Or you can use the Rational Application Developer Trial CD in the supplemental packet of discs to install a fully integrated development environment that includes an exact replica of the Application Server for development testing.

### Installation features

Installation features in V6 include:

Feature	Description
The Network Deployment product includes Application Server and managed nodes.	You can use the Profile creation wizard to create deployment manager profiles, Application Server profiles, and custom profiles after installing the WebSphere Application Server Network Deployment product. You do not have to install another set of binary files to install an Application Server. All profiles on a machine share the core product files of the Profile creation wizard. (If you install again to create another set of core product files, there are two Profile creation wizards. One for each set of core product files.)
The product CD includes all of the installable components that are required to create an e-business environment.	You can use the product CD to install the IBM HTTP Server, the Web server plug-ins, and the WebSphere Application Server Clients. You do not have to use separate CDs. Separate installation programs exist within component directories on the product CD.
Each installable component has its own installation program.	You can use the V6 launchpad to install any installable component on the product CD. Or you can install each component directly using the <b>install</b> command in each component directory.
The launchpad can install any installable product in the primary packet of compact discs.	The launchpad can also install the Application Server Toolkit on Windows 2000 and Linux (Intel) systems. The Application Server Toolkit is on a separate disc, which requires you to change discs to launch the installation.

Review topology diagrams for each of the following installable components to determine which topology best fits your needs. The diagrams and their accompanying procedures can serve as a roadmap for installing a similar topology.

This topic describes installation scenarios for the following installable components:

- WebSphere Application Server Network Deployment
- Web server plug-ins
- Application clients

In addition to product installation diagrams for the installable components, this topic also links to a roadmap for using the Profile creation wizard, which is new for Version 6. The Profile creation wizard lets you create run-time environments for application server processes.

Each of the following installation scenarios includes topology diagrams and associated installation steps. Each step links to a specific procedure for installing a component or to a description of a command or tool.

1. Review the installation scenarios for the Network Deployment product, as described in “Planning to install Network Deployment.”
2. Review the installation scenarios for the WebSphere Application Server plug-ins, as described in “Planning to install Web server plug-ins” on page 63.
3. Review the installation scenarios for the application clients, as described in “Planning to install WebSphere Application Server Clients” on page 69.
4. Review the installation scenarios for the Profile creation wizard, as described in “Planning to create application server environments” on page 71.
5. **Optional:** Review interoperability and coexistence diagrams to know what is possible with Version 6. WebSphere Application Server V6 can interoperate with your other e-business systems, including other versions of WebSphere Application Server. *Interoperability* provides a communication mechanism for WebSphere Application Server nodes that are at different versions. *Coexistence* describes multiple versions or instances running on the same machine, at the same time.  
Interoperability support enhances migration scenarios with more configuration options. It often is convenient or practical to interoperate during the migration of a configuration from an earlier WebSphere Application Server version to a later one when some machines are at the earlier version and some machines are at the later version. The mixed environment of machines and application components at different software version levels requires interoperability and coexistence.  
It is often impractical, or even physically impossible, to migrate all the machines and applications within an enterprise at the same time. Understanding multiversion interoperability and coexistence is therefore an essential part of a migration between version levels.
6. **Optional:** Consider performance when designing your network, as described in “Example: Choosing a topology for better performance” on page 74 and “Queuing network” on page 74.

You can review installation scenarios to identify the specific steps to follow when installing more than one component on a single machine or on separate machines.

After determining an appropriate installation scenario, you are ready to install the necessary components and to configure the products for the system that you selected.

---

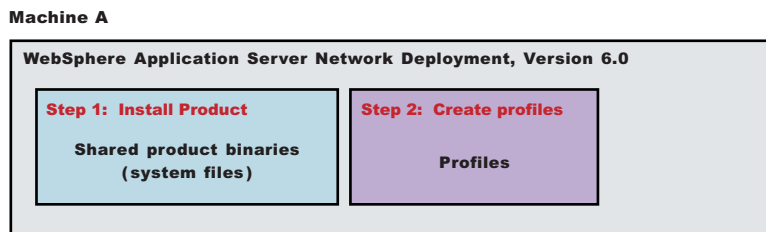
## Planning to install Network Deployment

This topic describes common installation scenarios and links to component installation procedures.

In Version 6.0, installing WebSphere Application Server Network Deployment is a two-step process. The first step is using the installation wizard to install a shared set of core product files. The second step is using the *Profile creation wizard* to create a *deployment manager profile*, an *Application server profile*, or a *custom profile*.

A profile is a separate data partition that includes the files that define a run-time environment for an application server process.

A running application server process, such as a deployment manager, can create, read, update, or delete the configuration files, data files, and log files in its profile. The application server process has read-only access to the system files, which include command files and other shared product binary files. System files are updated only by installing refresh packs or fix packs, or by products that extend WebSphere Application Server Network Deployment.



## Scenarios for installation

The following information describes scenarios for installing the product in various topologies on one or more machines. Two types of WebSphere Application Server topologies are possible using the Network Deployment product:

- Topologies for a stand-alone application server
- Topologies for a managed group of application servers

### Topologies for a stand-alone application server

Each stand-alone application server has its own administrative console and runs independently of other application servers.

The following topologies are described in this topic.

- **Scenario 1:** Single-machine installation of a stand-alone application server
- **Scenario 2:** Single-machine installation of a stand-alone application server and a Web server
- **Scenario 3:** Two-machine installation of a stand-alone application server and a Web server
- **Scenario 4:** Two-machine installation of multiple stand-alone application servers and a Web server

### Topologies for a managed group of application servers

A managed group of application servers is called a *cell*. A cell consists of one deployment manager and one or more federated application servers, which are called *managed nodes*.

A node becomes a managed node in either of two ways:

- Federating the node within an *Application Server profile* into the cell
- Federating the node within a *custom profile* into the cell

The deployment manager is the single point of administration for all of the managed nodes in the cell. The deployment manager maintains the configuration files for nodes that it manages and deploys applications to those managed nodes.

The following topologies for a cell are described in this topic.

- **Scenario 5:** Single-machine installation of a cell of application servers
- **Scenario 6:** Single-machine installation of a cell of application servers and a Web server
- **Scenario 7:** Two-machine installation of a cell of application servers and a Web server
- **Scenario 8:** Three-machine installation of a cell of application servers and a Web server

Each of the following scenarios includes a diagram and a list of detailed installation steps.

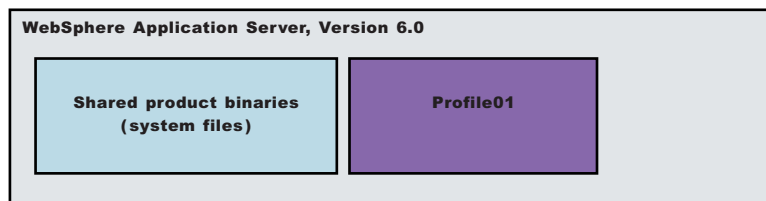
Some scenarios are more typical in production environments. For example, Scenario 1 supports a lighter workload than Scenario 3 or Scenario 4. However, Scenario 1 is a fully functional environment. Scenarios 3 and 4 are typical production environments for a stand-alone application server. Scenario 8 is a typical production scenario for a cell environment.

- **Scenario 1:** Install a stand-alone application server on a single machine.

Installing WebSphere Application Server Network Deployment by itself on a single machine lets you create a stand-alone Application Server profile. Each stand-alone application server profile includes a server1 application server process. Installing Network Deployment creates the set of system files. The Profile creation wizard creates the profile for the application server. The profile is a separate data partition with files that define the stand-alone application server environment.

In this scenario, the application server uses its internal HTTP transport chain for communication, which is suitable for handling an application with a relatively low request work load. For example, this type of installation can support a simple test environment or a departmental intranet environment.

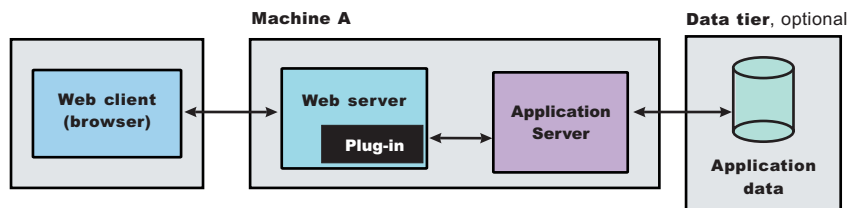
**Machine A**



1. Install WebSphere Application Server Network Deployment.
2. Create an Application Server profile using the Profile creation wizard.

- **Scenario 2:** Install a stand-alone application server and a Web server on a single machine.

Installing a Web server, such as IBM HTTP Server, on the same machine as the application server provides a more robust Web server environment. Installing a Web server plug-in is a requirement for the Web server to communicate with the application server. This type of installation supports rigorous testing environments or production environments that do not require a firewall. However, this is not a typical production environment.

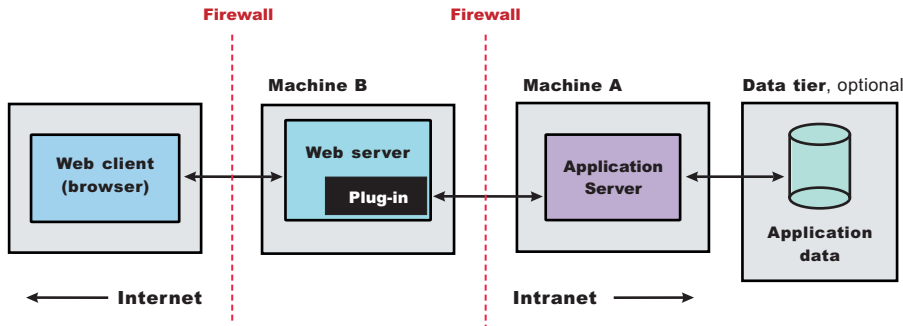


1. Install WebSphere Application Server Network Deployment.



2. Create an Application Server profile using the Profile creation wizard.
  3. Install IBM HTTP Server or another supported Web server.
  4. Install the Web server plug-ins and configure the Web server using the Plug-ins installation wizard.
- **Scenario 3:** Install a stand-alone application server and a Web server on separate machines.

In the typical production environment, the application server on one machine communicates with a Web server on a separate (remote) machine through the Web server plug-in. Optional firewalls can provide additional security for the application server machine.



1. Install WebSphere Application Server Network Deployment on Machine A.
  2. Create an Application Server profile using the Profile creation wizard on Machine A.
  3. Install IBM HTTP Server or another supported Web server on Machine B.
  4. On Machine B, install the Web server plug-ins and configure the Web server using the Plug-ins installation wizard.
  5. The Plug-ins installation wizard creates a script named `configureWeb_server_name` in the `plugins_install_root/bin` directory on Machine B. Copy the script to the `install_root/bin` directory on Machine A.
  6. Run the `configureWeb_server_name` script to create a Web server definition in the administrative console. You can then use the administrative console to manage the Web server.
  7. Propagate the `plugin-cfg.xml` file from the application server to the Web server using the administrative console. Click **Servers > Web server > Propagate Plug-in**. (Web servers other than IBM HTTP Server require manual propagation.)
- **Scenario 4:** Install multiple stand-alone application servers on one machine and a Web server on a separate machine.

The Profile creation wizard can create a deployment manager profile, an application server profile, or a custom profile. Each profile is a separate data partition containing the files that define the run-time environment. After creating a profile and installing a dedicated Web server, use the Plug-ins installation wizard to install a plug-in and to update the Web server configuration file. The Web server can then communicate with the application server.

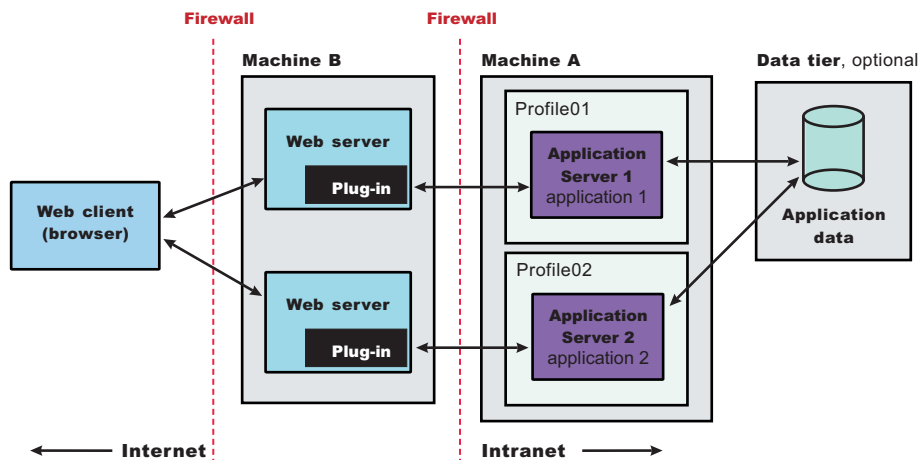
This topology lets each profile have unique applications, configuration settings, data, and log files, while sharing the same set of system files. Creating multiple profiles creates multiple application server environments that you can dedicate to different purposes.

For example, each application server on a Web site can serve a different application. In another example, each application server can be a separate test environment that you assign to a programmer or a development team.

### Updating the core product files

Another feature of having multiple profiles is enhanced serviceability. When a refresh pack or a fix pack updates the core product files on a machine, all of the application server profiles that were created from the core product files begin using the updated files. In some situations, you might prefer to not update

all of the application servers on a machine. In such situations, simply install the product a second time to create a second set of core product files. Create application server profiles from both installations to manage the product updates incrementally.



1. Install WebSphere Application Server Network Deployment on Machine A.
  2. Create the first Application Server profile using the Profile creation wizard on Machine A.
  3. Install IBM HTTP Server or another supported Web server on Machine B.
  4. On Machine B, install the Web server plug-ins and configure the first Web server using the Plug-ins installation wizard.
  5. The Plug-ins installation wizard creates a script named `configureWeb_server_name` in the `plugins_install_root/bin` directory on Machine B. Copy the script to the `install_root/bin` directory on Machine A.
  6. Run the `configureWeb_server_name` script to create a Web server definition in the administrative console. You can then use the administrative console to manage the Web server.
  7. Propagate the `plugin-cfg.xml` file from the application server to the Web server using the administrative console. Click **Servers > Web server > Propagate Plug-in**. (Web servers other than IBM HTTP Server require manual propagation.)
  8. Create the second Application Server profile using the Profile creation wizard on Machine A. Make the profile the default profile during the profile creation by selecting the check box on the appropriate panel.  
The script that the Plug-ins installation wizard creates works on the default profile only. So, this script can only create a Web server definition on the profile that is the default profile at the time that the script runs.
  9. Install IBM HTTP Server or another supported Web server on Machine B.
  10. On Machine B, install the Web server plug-ins to configure the second Web server using the Plug-ins installation wizard. Both Web servers share a single installation of the plug-in binaries but must be configured individually.
  11. The Plug-ins installation wizard creates a script named `configureWeb_server_name` for the second Web server. The script is in the `plugins_install_root/bin` directory on Machine B. Copy the script to the `install_root/bin` directory on Machine A.
  12. Run the `configureWeb_server_name` script to create a Web server definition in the administrative console. You can then use the administrative console to manage the Web server.
  13. Propagate the `plugin-cfg.xml` file from the second application server to the Web server using the administrative console. Click **Servers > Web server > Propagate Plug-in**. (Web servers other than IBM HTTP Server require manual propagation.)
- **Scenario 5:** Install a cell of managed application server nodes on one machine.

WebSphere Application Server Network Deployment can create a cell of managed application servers on a single machine from one installation of the core product files. The Profile creation wizard creates the deployment manager. After starting the deployment manager, return to the Profile creation wizard to create one or more application servers for the cell. Application server profiles have a default application server, called `server1`, and default applications. An Application Server node becomes a managed node after federating the node into the deployment manager cell.

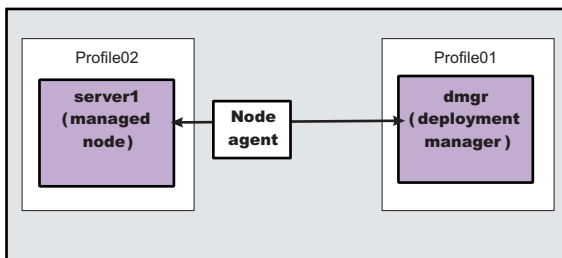
The deployment manager provides the administration for all managed nodes that are in its cell. Periodically the configuration and application files on a managed node refresh from the master copy of the files hosted on the deployment manager during *synchronization*.

In certain secure environments, the Profile creation wizard cannot federate a custom profile into a cell. Such cases require you to use the **addNode** command instead. If you have configured the deployment manager to use a JMX connector type other than the default SOAP connector, use the **addNode** command to add the node to the cell.

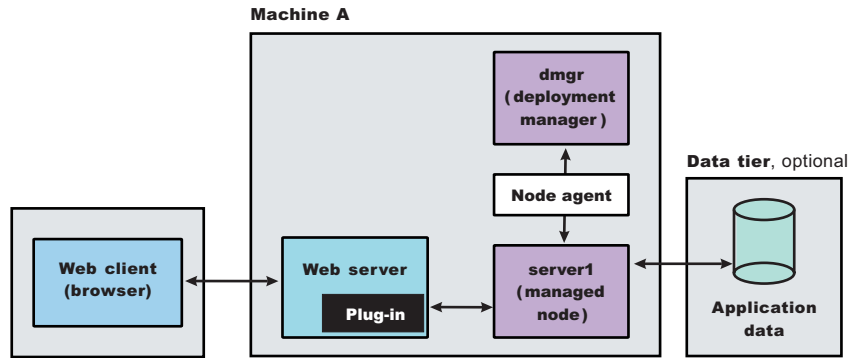
The deployment manager provides the administration for all managed nodes that are in its cell. Periodically the deployment manager refreshes the configuration files and application files on the managed node. Copying the master version of the files hosted on the deployment manager to the managed nodes is a process called *synchronization*.

In a cell environment, only the managed nodes serve applications, not the deployment manager. The managed node in this scenario uses its internal HTTP transport chain for communication, which is suitable for an application with a relatively low request work load. For example, this type of installation can support a simple test environment or a departmental intranet environment.

#### Machine A

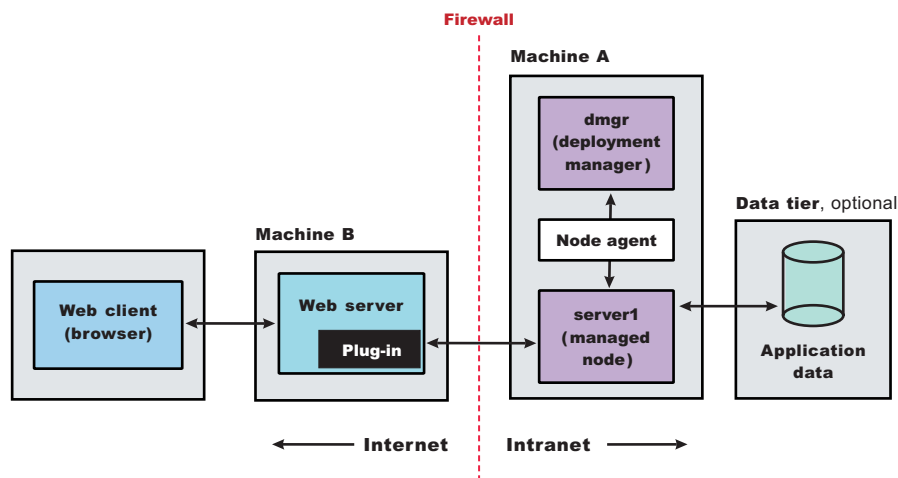


1. Install WebSphere Application Server Network Deployment.
  2. Create a deployment manager profile using the Profile creation wizard.
  3. Start the deployment manager using the First steps console or the **startManager** command.
  4. Create an Application Server profile using the Profile creation wizard.
  5. Start the application server using the First steps console or the **startServer server1** command.
  6. Add the application server node to the cell using the administrative console of the deployment manager. Click **System Administration > Nodes** to add the node.
- **Scenario 6:** Install a cell of managed application server nodes and a Web server on one machine. Installing a Web server, such as IBM HTTP Server, on the same machine as the application server provides a richer set of configuration options. Installing a Web server plug-in is required for the Web server to communicate with the server in the managed node. This type of installation can support either rigorous testing in a cell environment or production environments that do not require a firewall.



1. Install WebSphere Application Server Network Deployment.
  2. Create a deployment manager profile using the Profile creation wizard.
  3. Start the deployment manager using the First steps console or the **startManager** command.
  4. Create an Application Server profile using the Profile creation wizard.
  5. Start the application server using the First steps console or the **startServer server1** command.
  6. Add the application server node to the cell using the administrative console of the deployment manager. Click **System Administration > Nodes** to add the node.
  7. Install IBM HTTP Server or another supported Web server.
  8. Install the Web server plug-ins and configure the Web server using the Plug-ins installation wizard.
  9. The Plug-ins installation wizard creates a script named `configureWeb_server_name` in the `plugins_install_root/bin` directory. Run the `configureWeb_server_name` script to create a Web server definition in the administrative console. You can then use the administrative console to manage the Web server.
- **Scenario 7:** Install a cell of managed application server nodes on one machine and a Web server on a separate machine.

In a typical production environment, a managed node in a cell communicates with a Web server on a separate (remote) machine through the Web server plug-in. An optional firewall can provide additional security for the application server machine.



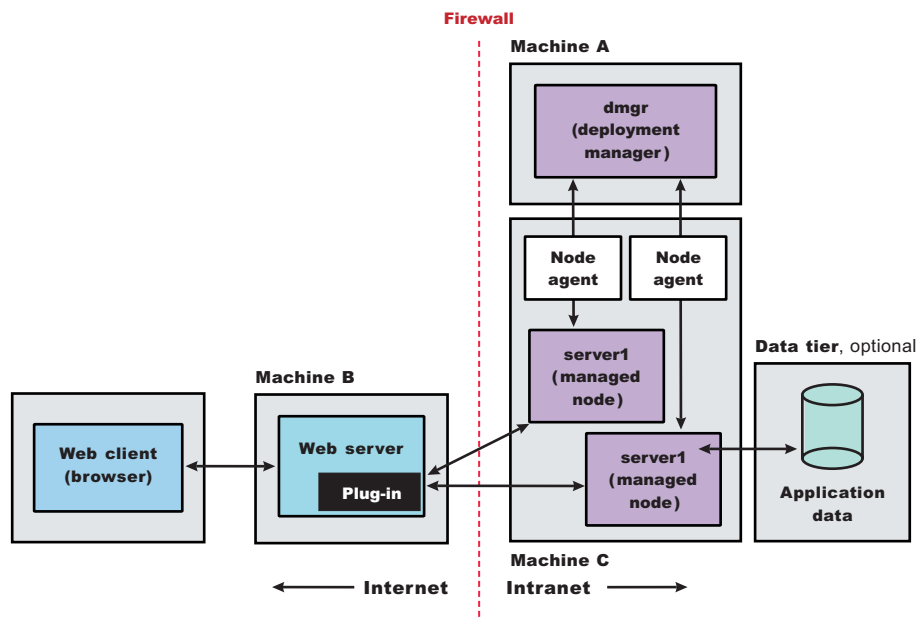
1. Install WebSphere Application Server Network Deployment on Machine A.
2. Create a deployment manager profile using the Profile creation wizard on Machine A.
3. Start the deployment manager using the First steps console or the **startManager** command on Machine A.

4. Create the Application Server profile and make this profile the default profile using the Profile creation wizard on Machine A.
  5. Start the application server using the First steps console or the **startServer server1** command on Machine A.
  6. Add the application server node to the cell using the administrative console of the deployment manager on Machine A. Click **System Administration > Nodes** to add the node.
  7. Install IBM HTTP Server or another supported Web server on Machine B.
  8. On Machine B, install the Web server plug-ins and configure the Web server using the Plug-ins installation wizard.
  9. The Plug-ins installation wizard creates a script named `configureWeb_server_name` in the `plugins_install_root/bin` directory on Machine B. Copy the script from Machine B to the `install_root/bin` directory on Machine A.  
 You have the option of using the script to create the Web server definition in the configuration of the deployment manager or using the administrative console of the deployment manager to create the Web server definition.
  10. Run the `configureWeb_server_name` script on Machine A to create a Web server definition or use the administrative console of the deployment manager to create the definition. You can then use the administrative console to manage the Web server.
  11. Propagate the `plugin-cfg.xml` file from the deployment manager on Machine A to the Web server on Machine B using the administrative console. Click **Servers > Web server > Propagate Plug-in**. (Web servers other than IBM HTTP Server require manual propagation.)
- **Scenario 8:** Install a deployment manager on one machine, multiple managed application server nodes on a second machine, and a Web server on a third machine.

The primary advantage of a cell over a stand-alone application server is its scalability. Managing a cell to keep it in proportion with workload levels is possible. In this scenario, managed nodes exist on Machine C. All of the managed nodes are federated into the same deployment manager. Depending on your needs, an application server in each managed node could serve the same or different applications. Having multiple machines and multiple application server profiles lets you use vertical and horizontal scaling:

- *Vertical scaling* creates multiple managed nodes on the same physical machine.
- *Horizontal scaling* creates cell members on multiple physical machines.

The managed nodes in this scenario communicate with the same Web server. However, the preferred strategy is to have a dedicated Web server for each managed node.



1. Install WebSphere Application Server Network Deployment on Machine A.
2. Create a deployment manager profile using the Profile creation wizard on Machine A.
3. Start the deployment manager using the First steps console or the **startManager** command on Machine A.
4. Install WebSphere Application Server Network Deployment on Machine C.
5. Create the first Application Server profile using the Profile creation wizard on Machine C.
6. Start the first application server using the First steps console or the **startServer server1** command on Machine C.
7. Create the second Application Server profile and make this profile the default profile using the Profile creation wizard on Machine C.
8. Start the second Application Server using the First steps console or the **startServer server1** command on Machine C.
9. Add both application server nodes to the cell using the administrative console of the deployment manager on Machine A. Click **System Administration > Nodes** to add the nodes.
10. Install IBM HTTP Server or another supported Web server on Machine B.
11. Install the Web server plug-ins and configure the Web server using the Plug-ins installation wizard on Machine B.
12. The Plug-ins installation wizard creates a script named *configureWeb\_server\_name* in the *plugins\_install\_root/bin* directory on Machine B. Copy the script from Machine B to the *install\_root/bin* directory on Machine A.  
You have the option of using the script to create the Web server definition in the configuration of the deployment manager or using the administrative console of the deployment manager to create the Web server definition.
13. Run the *configureWeb\_server\_name* script on Machine A to create a Web server definition or use the administrative console of the deployment manager to create the definition. You can then use the administrative console to manage the Web server.
14. Propagate the *plugin-cfg.xml* file from the deployment manager on Machine A to the Web server on Machine B using the administrative console. Click **Servers > Web server > Propagate Plug-in**. (Web servers other than IBM HTTP Server require manual propagation.)

You can review common installation scenarios to find a possible match for the topology that you intend to install. Each product installation diagram provides a high-level procedure for installing the components that comprise the topology.

After determining a possible topology, follow the steps in the overall procedure. Useful links to the installation procedures for each installable component are in the list of related topics.

---

## Planning to install Web server plug-ins

This topic describes common installation scenarios and links to component installation procedures for each scenario.

The primary production configuration is an application server on one machine and a Web server on a separate machine. This configuration is referred to as a *remote* configuration. Contrast the remote configuration to the local configuration, where the application server and the Web server are on the same machine.

The Plug-ins installation wizard has four main tasks:

- Installs the binary plug-in module on the Web server machine.
- Configures the Web server configuration file on the Web server machine to point to the binary plug-in module and to the XML configuration file for the binary module.
- Installs a temporary XML configuration file for the binary module (`plugin-cfg.xml`) on the Web server machine in remote scenarios.
- Creates the configuration for a Web server definition on the application server machine. The wizard processes the creation of the Web server definition differently depending on the scenario:

### Web server plug-in installation for stand-alone application server environments

- Recommended remote stand-alone Application Server installation:

Creates a configuration script that you run on the application server machine. Install the Web server and its plug-in on a different machine than the application server. This configuration is recommended for a production environment.

- Local stand-alone Application Server installation:

Detects the default profile on a local application server machine and creates the Web server definition for it directly. Install the Web server and its plug-in on the same machine with the application server. This configuration is for development and test environments.

### Web server plug-in installation for distributed environments (cells)

- Recommended remote distributed installation:

Creates a configuration script that you run on the application server machine. Install the Web server and its plug-in on a different machine than the deployment manager or managed node. This configuration is recommended for a production environment.

- Local distributed installation:

Creates a configuration script that you run when the deployment manager is running. Install the Web server and its plug-in on the same machine with the deployment manager or a managed node. This configuration is for development and test environments.

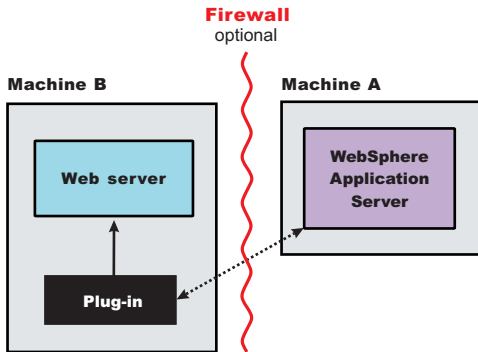
Select a link to go to the appropriate steps in the following procedure.

- **Set up a remote Web server installation.**

The remote Web server configuration is recommended for production environments.





The remote installation installs the Web server plug-in on the Web server machine when the application server is on a separate machine, such as shown in the following graphic:



### Remote installation scenario

Table 2. Installation and configuration

Step	Machine	Task
1	A	Install your WebSphere Application Server product. See Chapter 9, "Installing the product and additional software," on page 107.
2	A	Create an application server profile. See "Using the Profile creation wizard" on page 234.
3	B	Install IBM HTTP Server or another supported Web server. See "Installing IBM HTTP Server" on page 134.
4	B	Install the binary plug-in module using the Plug-ins installation wizard. See "Configuring a Web server and an application server on separate machines (remote)" on page 157.  The script for creating and configuring the Web server is created under the <code>plug-ins_install_root/bin</code> directory.
5	B	Copy the <code>configureWeb_server_name</code> script to Machine A. If one machine is running under Linux or UNIX and the other machine is running under Windows, copy the script from the <code>plug-ins_install_root/bin/crossPlatformScripts</code> directory.
6	A	Paste the <code>configureWeb_server_name</code> script from Machine B to the <code>was_install_root/bin</code> directory on Machine A.
7	A	Run the script from a command line.
8	A	Verify that the application server is running. Open the administrative console and save the changed configuration.
9	B	  Run the <code>plug-ins_install_root/setupPluginCfg.sh</code> script for a Domino Web Server before starting a Domino Web server. Otherwise, start the Web server.
10	B	Run the snoop servlet.  To verify with your own application, regenerate and propagate the <code>plugin-cfg.xml</code> file after installing the application.

### Regeneration of the plugin-cfg.xml file

During the installation of the plug-ins, the temporary `plugin-cfg.xml` file is installed on Machine B in the `plug-ins_install_root/config/web_server_name` directory.

The Web server plug-in configuration service regenerates the `plugin-cfg.xml` file automatically.

To use the real `plugin-cfg.xml` file from the application server, propagate the `plugin-cfg.xml` file as described in the next section.



### Propagation of the plugin-cfg.xml file

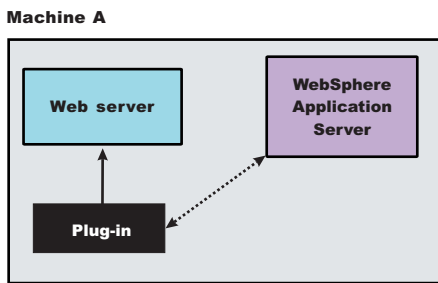
The Web server plug-in configuration service propagates the plugin-cfg.xml file automatically for IBM HTTP Server 6.0 only.

For all other Web servers, propagate the plug-in configuration file manually. Copy the plugin-cfg.xml file from the `profiles_install_root/config/cells/cell_name/nodes/Web_server_name_node/servers/web_server_name` directory on Machine A. Paste the file into the `plug-ins_install_root/config/web_server_name` directory on Machine B.

- **Set up a local Web server configuration.**



The local Web server configuration is recommended for a development or test environment.

A local installation includes the Web server plug-in, the Web server, and the application server on the same machine:



### Local installation scenario

Table 3. Installation and configuration

Step	Machine	Task
1	A	Install your WebSphere Application Server product. See Chapter 9, “Installing the product and additional software,” on page 107.
2	A	Create an application server profile. See “Using the Profile creation wizard” on page 234.
3	A	Install IBM HTTP Server or another supported Web server. See “Installing IBM HTTP Server” on page 134.
4	A	Install the binary plug-in module using the Plug-ins installation wizard. See “Configuring a Web server and an application server profile on the same machine” on page 170.  The Web server definition is automatically created and configured during the installation of the plug-ins.
5	A	Verify that the application server is running. Open the administrative console and save the changed configuration.
6	B	  Run the <code>plug-ins_install_root/setupPluginCfg.sh</code> script for a Domino Web Server before starting a Domino Web server. Otherwise, start the Web server.
7	B	Run the Snoop servlet.  To verify with your own application, regenerate and propagate the plugin-cfg.xml file after installing the application.

### Regeneration of the plugin-cfg.xml file

The Web server plug-in configuration service regenerates the plugin-cfg.xml file automatically.

The plugin-cfg.xml file is generated in the `profiles_install_root/profile_name/config/cells/cell_name/nodes/Web_server_name_node/servers/web_server_name` directory. The generation occurs when the Web server definition is created.

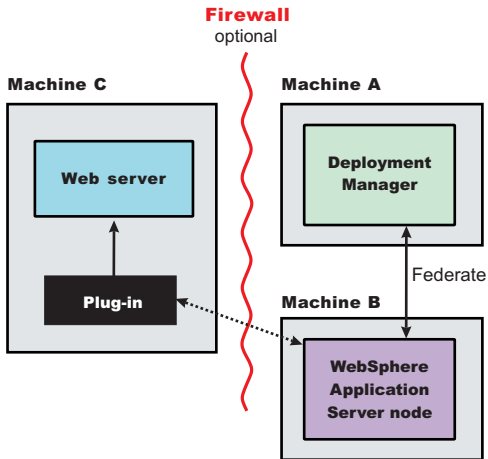
## Propagation of the plugin-cfg.xml file

The local file does not require propagation.

- **Set up a remote Web server installation in a cell.**

The remote Web server configuration is recommended for production environments.

The remote installation installs the Web server plug-in on the Web server machine when the application server is on a separate machine, such as shown in the following graphic:




## Remote installation scenario

Table 4. Installation and configuration

Step	Machine	Task
1	A	Install WebSphere Application Server Network Deployment. See Chapter 9, “Installing the product and additional software,” on page 107.
2	A	Create a deployment manager profile. See “Using the Profile creation wizard” on page 234.
3	A	Start the deployment manager with the <code>./profiles_install_root/profile_name/bin/startManager.sh</code> command or its Windows equivalent.
4	B	Install WebSphere Application Server Network Deployment. See Chapter 9, “Installing the product and additional software,” on page 107.
5	B	Create an application server profile. See “Using the Profile creation wizard” on page 234.
6	B	Federate the node with the <code>./profiles_install_root/profile_name/bin/addNode.sh dmgrhost 8879 -includeapps</code> command or its Windows equivalent.
7	C	Install IBM HTTP Server or another supported Web server. See “Installing IBM HTTP Server” on page 134.
8	C	Install the binary plug-in module using the Plug-ins installation wizard. See “Configuring a Web server and an application server on separate machines (remote)” on page 157.  The script for creating and configuring the Web server is created under the <code>plug-ins_install_root/bin</code> directory.
9	C	Copy the <code>configureWeb_server_name</code> script to Machine A.  If one machine is running under Linux or UNIX and the other machine is running under Windows, copy the script from the <code>plug-ins_install_root/bin/crossPlatformScripts</code> directory.
10	A	Paste the <code>configureWeb_server_name</code> script from Machine C to the <code>was_install_root/bin</code> directory on Machine A.

Table 4. Installation and configuration (continued)

Step	Machine	Task
11	A	Run the script from a command line after verifying that the deployment manager is running.  If you have enabled security or changed the default JMX connector type, edit the script and include the appropriate parameters on the <b>wsadmin</b> command.
12	A/B	Use the administrative console of the deployment manager on Machine A to start the application server on Machine B. Wait for synchronization to occur and save the new configuration.
13	C	 Run the <code>plug-ins_install_root/setupPluginCfg.sh</code> script for a Domino Web Server before starting a Domino Web server. Otherwise, start the Web server.
14	C	Run the Snoop servlet.  To verify with your own application, regenerate and propagate the <code>plugin-cfg.xml</code> file after installing the application.

### Regeneration of the plugin-cfg.xml file

During the installation of the plug-ins, the temporary `plugin-cfg.xml` file is installed on Machine C in the `plug-ins_install_root/config/web_server_name` directory.

The Web server plug-in configuration service regenerates the `plugin-cfg.xml` file automatically.

To use the real `plugin-cfg.xml` file from the application server, propagate the `plugin-cfg.xml` file as described in the next section.

### Propagation of the plugin-cfg.xml file

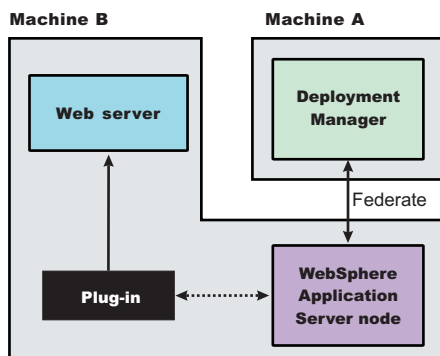
The Web server plug-in configuration service propagates the `plugin-cfg.xml` file automatically for IBM HTTP Server 6.0 only.

For all other Web servers, propagate the plug-in configuration file, by manually copying the `plugin-cfg.xml` file from the `profiles_install_root/profile_name/config/cells/cell_name/nodes/node_name/servers/web_server_name` directory on Machine A to the `plug-ins_install_root/config/web_server_name` directory on Machine C.

- **Set up a local distributed Web server configuration.**


The local Web server configuration is recommended for a development or test environment.

A local distributed installation includes the Web server plug-in, the Web server, and the managed application server on the same machine:



## Local distributed installation scenario

Table 5. Installation and configuration

Step	Machine	Task
1	A	Install WebSphere Application Server Network Deployment. See Chapter 9, “Installing the product and additional software,” on page 107.
2	A	Create a deployment manager profile. See “Using the Profile creation wizard” on page 234.
3	A	Start the deployment manager with the <code>profiles_install_root/profile_name/bin/startManager.sh</code> command or its Windows equivalent.
4	B	Install WebSphere Application Server Network Deployment. See Chapter 9, “Installing the product and additional software,” on page 107.
5	B	Create an application server profile. See “Using the Profile creation wizard” on page 234.
6	B	Federate the node with the <code>./profiles_install_root/profile_name/bin/addNode.sh dmgrhost 8879 -includeapps</code> command or its Windows equivalent.
7	B	Install IBM HTTP Server or another supported Web server. See “Installing IBM HTTP Server” on page 134.
8	B	Install the binary plug-in module using the Plug-ins installation wizard. See “Configuring a Web server and an application server profile on the same machine” on page 170.  The script for creating and configuring the Web server is created in the <code>plug-ins_install_root/bin</code> directory.
11	B	After verifying that the deployment manager is running on Machine A, run the <code>configureWeb_server_name</code> script from a command line in the <code>plug-ins_install_root/bin</code> directory on Machine B.  If you have enabled security or changed the default JMX connector type, edit the script and include the appropriate parameters.
12	A/B	Use the administrative console of the deployment manager on Machine A to start the application server on Machine B. Wait for synchronization to occur and save the new configuration.
13	B	 Run the <code>plug-ins_install_root/setupPluginCfg.sh</code> script for a Domino Web Server before starting a Domino Web server. Otherwise, start the Web server.
14	B	Run the Snoop servlet.

### Regeneration of the plugin-cfg.xml file

The Web server plug-in configuration service regenerates the `plugin-cfg.xml` file automatically.

The `plugin-cfg.xml` file is generated at the location `profiles_install_root/profile_name/config/cells/cell_name/nodes/node_name/servers/web_server_name` directory, when the Web server definition is created.

Regenerate the `plugin-cfg.xml` file in the Web server definition in the application server whenever the configuration changes. The Web server has immediate access to the file whenever it is regenerated.

When the Web server plug-in configuration service (an administration service) is enabled on Machine A, the `plugin-cfg.xml` file is automatically generated for all Web servers.

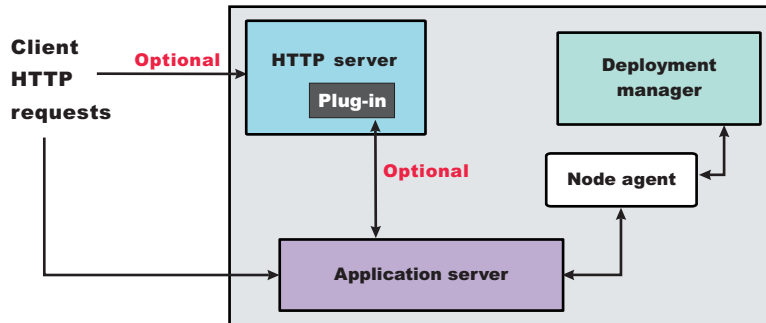
### Propagation of the plugin-cfg.xml file

Node synchronization is used to propagate the `plugin-cfg.xml` file from Machine A to Machine B.

When the Web server plug-in configuration service (an administration service) is enabled on Machine A, the `plugin-cfg.xml` file is automatically propagated for all Web servers.

### Alternate configuration

This procedure describes installing the plug-ins on two machines. However, you can perform this procedure on a single machine as shown in the following graphic. A local distributed installation also includes the Web server plug-in, the Web server, the Application Server, and the deployment manager on the same machine:



You can set up a remote or local Web server by installing Application Server, the Web server, and then the Web server plug-ins.

See “Web server configuration” on page 153 for more information about the files involved in configuring a Web server.

See “Configuration behavior of the Plug-ins installation wizard” on page 145 for information about the logic behind the processing scenarios for the Plug-ins installation wizard.

See “Editing Web server configuration files” on page 203 for information about how the Plug-ins installation wizard configures supported Web servers.

See “Installing Web server plug-ins” on page 136 for information about other installation scenarios for installing Web server plug-ins.

---

## Planning to install WebSphere Application Server Clients

This topic helps you examine typical topologies and uses for WebSphere Application Server Clients.

This topic is one in a series of topics described in Chapter 7, “Planning the installation (diagrams),” on page 53. Consider all of the planning scenarios that are mentioned in the parent article to determine the best approach to installing your e-business network. This topic describes installing and using the WebSphere Application Server Clients.

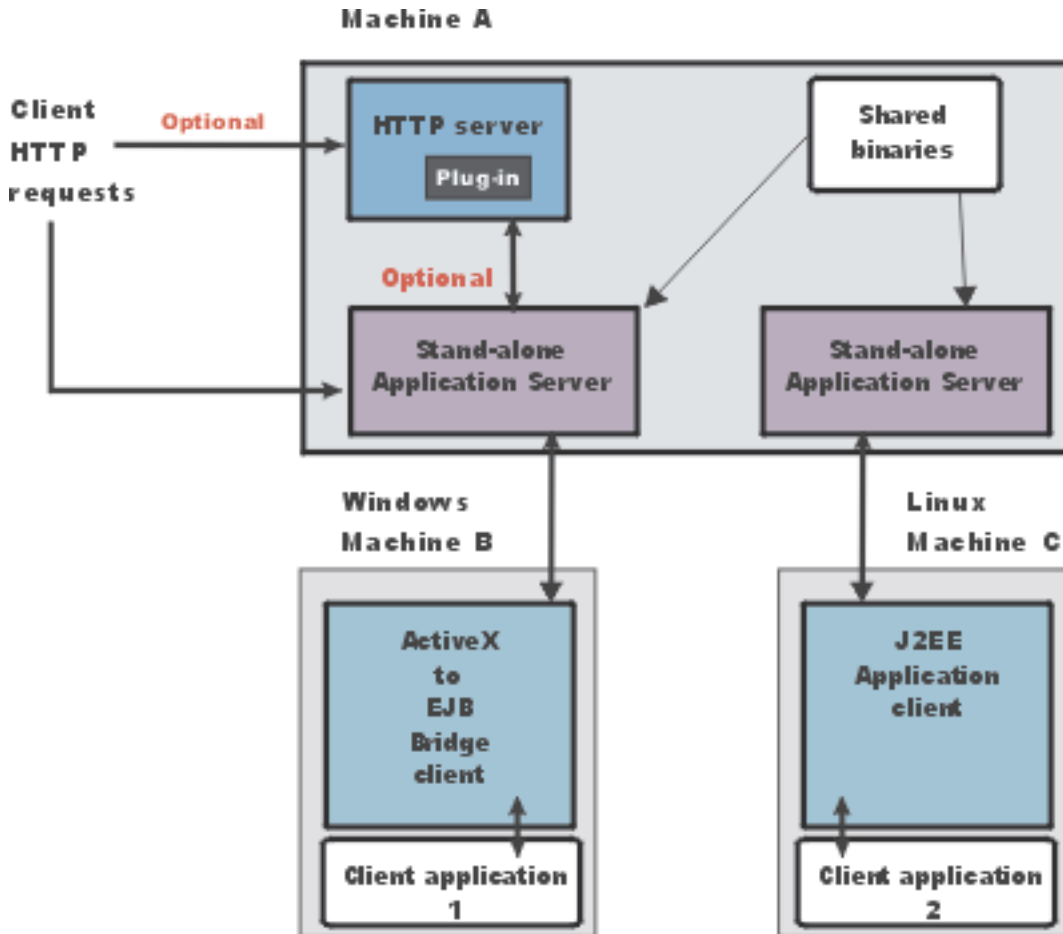
In a traditional client server environment, the client requests a service and the server fulfills the request. Multiple clients use a single server. Clients can also access several different servers. This model persists for Java clients except that now these requests use a client run-time environment.

In this model, the client application requires a servlet to communicate with the enterprise bean, and the servlet must reside on the same machine as the WebSphere Application Server.

The Application Client for WebSphere Application Server, Version 6 now consists of the following models:

- ActiveX application client
- Applet client
- J2EE application client
- Pluggable and thin application clients

The following graphic shows a topology for installing the Application Client and using client applications:



The example shows two types of application clients installed in a topology that uses client applications to access applications and data on Machine A:

- The ActiveX application client on Machine B is a Windows only client that uses the Java Native Interface (JNI) architecture to programmatically access the Java virtual machine (JVM) API. The JVM code exists in the same process space as the ActiveX application (Visual Basic, VBScript, or Active Server Pages (ASP) files) and remains attached to the process until that process terminates.
- The J2EE application client on Machine C is a Java application program that accesses enterprise beans, Java Database Connectivity (JDBC) APIs, and Java Message Service message queues. The application program must configure the execution environment of the J2EE application client and use the Java Naming and Directory Interface (JNDI) name space to access resources.

Use the following procedure as a roadmap for installing the Application Client.

1. Install the WebSphere Application Server product from your product CD on Machine A to establish the core product files.
2. Use the Profile creation wizard to create both stand-alone application server profiles.
3. Use the administrative console of each application server to deploy any user applications.
4. Use the administrative console of each application server to create a Web server configuration for the Web server.
5. Use the administrative console of each application server to regenerate each `plugin-cfg.xml` file in the local Web server configuration.
6. Install the IBM HTTP Server from the product CD on Machine A.
7. Use the Plug-ins installation wizard to install the plug-in for IBM HTTP Server on Machine A.

- The wizard automatically configures the HTTP Server to communicate with the first application server.
8. Install the Application Client from your product CD on Machine B.
    - a. Select the Custom install type.
    - b. Select the ActiveX to EJB Bridge feature.
    - c. Select to add the Java run time to the system path.
    - d. Select the Java run time as the default JRE, which adds the Java run time path to the beginning of the system path.
  9. Install the Application Client from your product CD on Machine C.
    - a. Select the Custom install type.
    - b. Select the J2EE application client feature.

---

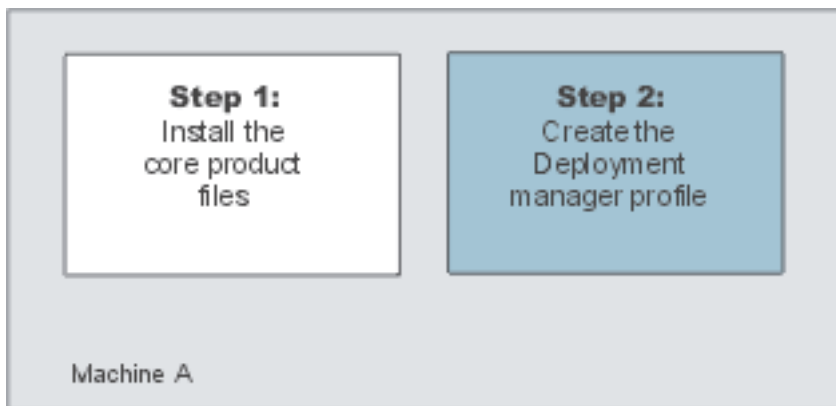
## Planning to create application server environments

Application server profiles are the run-time environments for application server processes. This topic describes common scenarios for creating application server profiles and provides links to profile creation procedures for each scenario.

Install the core product files for a WebSphere Application Server product before using the Profile creation wizard to create additional application server run-time environments.

This topic describes how to use the Profile creation wizard to install deployment managers, managed nodes, and stand-alone application servers. Each profile for a deployment manager or an application server is a run-time environment, with data files, configuration files, applications, and an administrative console. The profile for a managed node is a special case that is dependent on the deployment manager profile that owns the managed node.

1. Create a deployment manager, as described in “Using the Profile creation wizard to create a deployment manager” on page 238.

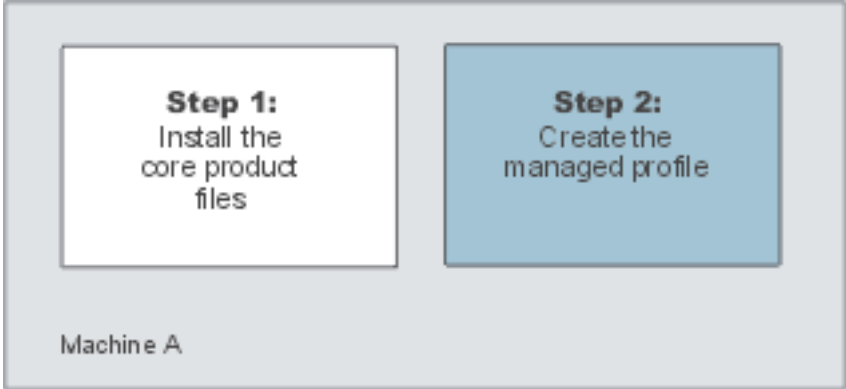


The installation procedure gives you the option of creating a deployment manager during installation. However, you can use the Profile creation wizard to create a deployment manager when one was not created during installation. Or, you can create another deployment manager on a machine where a deployment manager already exists.

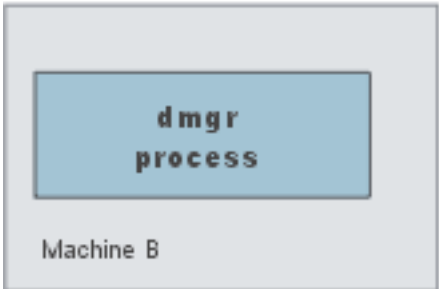
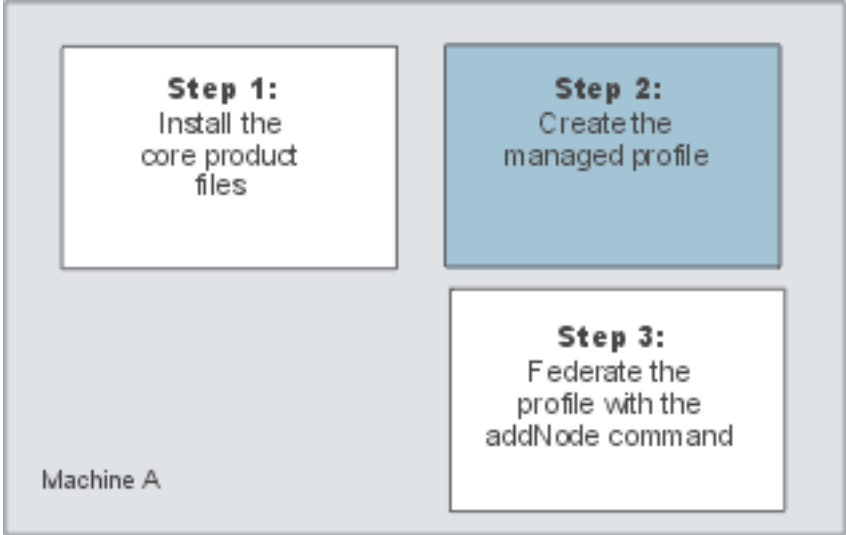
2. Create a managed node, as described in “Using the Profile creation wizard to create a custom profile” on page 249.

The installation procedure gives you the option of creating a managed node during installation. However, you can use the Profile creation wizard to create a managed node when one was not created during installation. Or, you can use the wizard to create more managed nodes on a machine where a managed node already exists.

The first part of the process is to install the Network Deployment product to create the core product files. Then you can use the Profile creation wizard to create a managed profile.



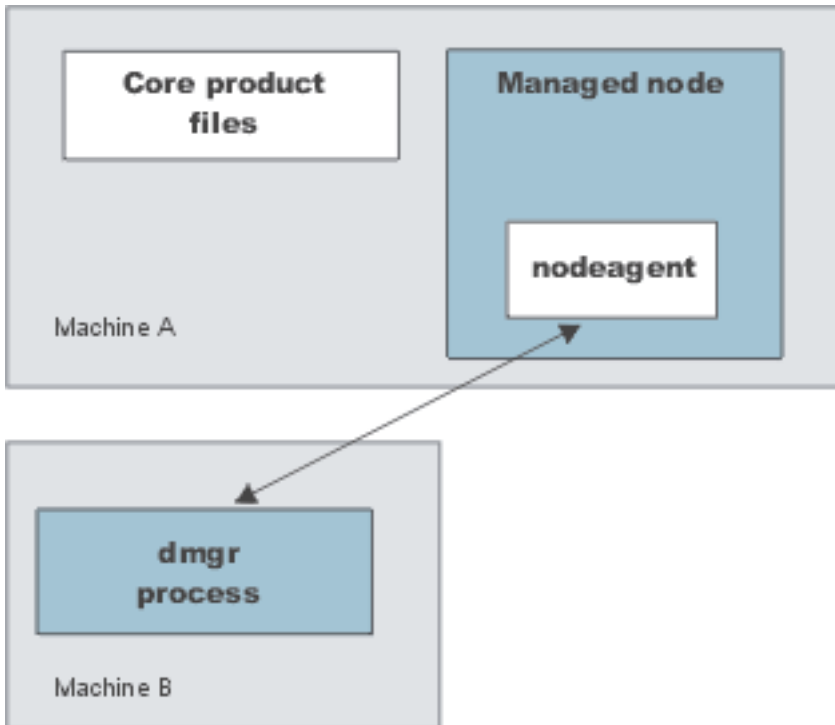
The next part of the process is to federate the managed profile into the deployment manager cell. This changes the managed profile into a managed node.



A managed node has a nodeagent process but does not have Sample applications or an application server process in contrast to a stand-alone application server, which has a server1 process and

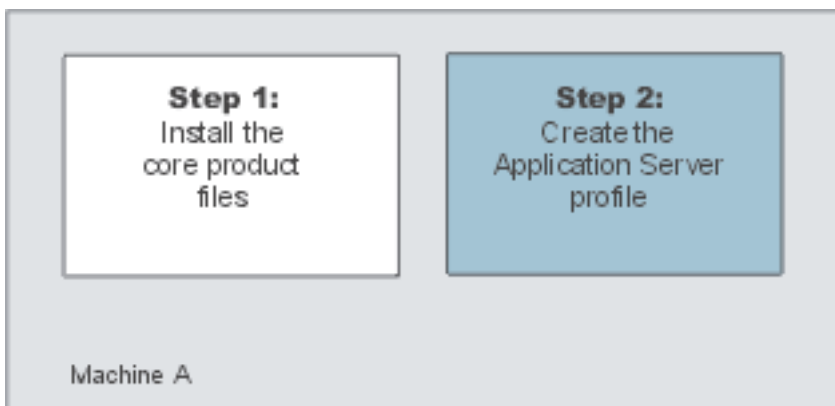


applications, but does not have a nodeagent process.



Start the nodeagent process to allow the administrative console of the deployment manager to create server processes on the managed node.

3. Create a stand-alone application server, as described in “Using the Profile creation wizard to create an application server” on page 263.



The installation procedure gives you the option of creating a stand-alone application server during installation. However, you can use the Profile creation wizard to create a stand-alone application server when one was not created during installation. Or, you can use the wizard to create more stand-alone application servers on a machine where an application server already exists.

4. Create a deployment manager and a managed node on the same machine.  
The installation procedure gives you the option of creating a deployment manager and managed node on the same machine during installation. However, you can also use the Profile creation wizard to create a deployment manager and a managed node at any time after installation of the core product files. Or, you can use the wizard to create a managed node on a machine where a deployment manager already exists.

Any time that you create two or more application server processes on one machine, verify that the machine is capable of hosting both processes. See the hardware prerequisites on the IBM WebSphere Application Server supported hardware, software, and APIs Web site.

- a. Create a deployment manager, as described in “Using the Profile creation wizard to create a deployment manager” on page 238.
- b. Create a managed node, as described in “Using the Profile creation wizard to create a custom profile” on page 249.

Use the Profile creation wizard after installing the core product files to create:

- A stand-alone application server environment
- A deployment manager environment
- Managed nodes for the deployment manager cell

After installing the core product files and using the Profile creation wizard to create your e-business environment, you are ready to deploy applications to test the environment.

---

## Example: Choosing a topology for better performance

WebSphere Application Server provides various Workload Management (WLM) topologies. Two topologies were compared to show how the type of topology you choose can affect performance.

In this comparison, Topology A contains a Web server and a WebSphere Application Server plug-in in front of a cluster of WebSphere Application Servers. Each cluster member contains a Web container and an Enterprise JavaBeans (EJB) container. Topology B includes a Web server, a plug-in, and a Web container in front of a cluster of EJB containers. In both topologies, Object Request Broker (ORB) pass by reference is selected and the backend database is on a dedicated machine.

Result: Topology A had 10% to 20% higher throughput than Topology B when running the Java 2 Platform, Enterprise Edition Benchmark sample for WebSphere (Trade).

**Note:** You can download the Benchmark sample for WebSphere (Trade) from the following Web site:

<http://www-306.ibm.com/software/webservers/appserv/was/performance.html>

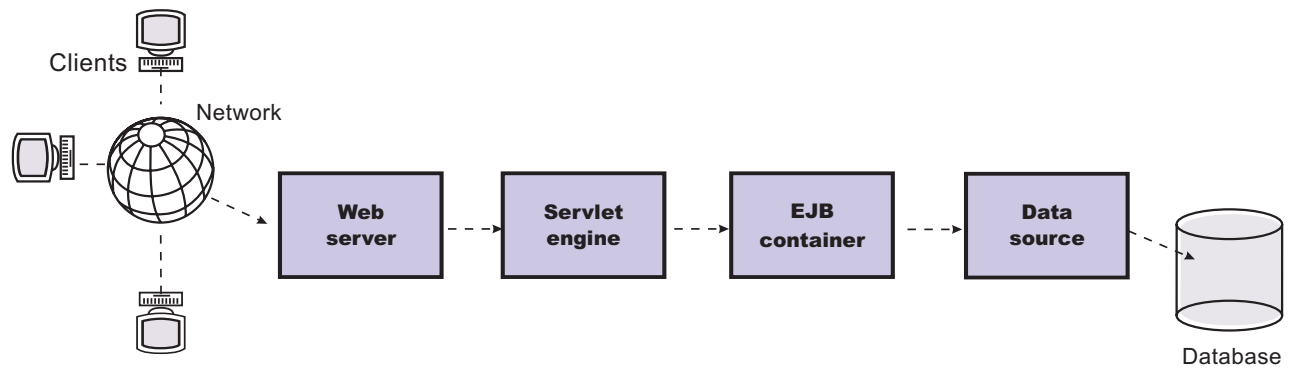
Topology A has an advantage because the Web container and EJB container are running in a single Java virtual machine (JVM). In Topology B, the ORB pass by reference option is ignored between the Web container cluster member and the EJB container member. In Topology A, the EJB container uses the same thread passed from the Web container. The request does not have to be passed from one thread in one JVM to another thread in another JVM.

In this test environment, Topology A had the advantage. However, many factors related to the application and environment can influence your results.

---

## Queuing network

WebSphere Application Server contains interrelated components that must be harmoniously tuned to support the custom needs of your end-to-end e-business application. These adjustments help the system achieve maximum throughput while maintaining the overall stability of the system. This group of interconnected components is known as a queuing network. These queues or components include the network, Web server, Web container, EJB container, data source, and possibly a connection manager to a custom back-end system. Each of these resources represents a queue of requests waiting to use that resource. Various queue settings are described in the *Tuning Guide* PDF.



**Figure Reference 1: WebSphere queuing network**

Most of the queues that make up the queuing network are closed queues. A closed queue places a limit on the maximum number of requests present in the queue, while an open queue has no limit. A closed queue supports tight management of system resources. For example, the Web container thread pool setting controls the size of the Web container queue. If the average servlet running in a Web container creates 10MB of objects during each request, a value of 100 for thread pools limits the memory consumed by the Web container to 1GB.

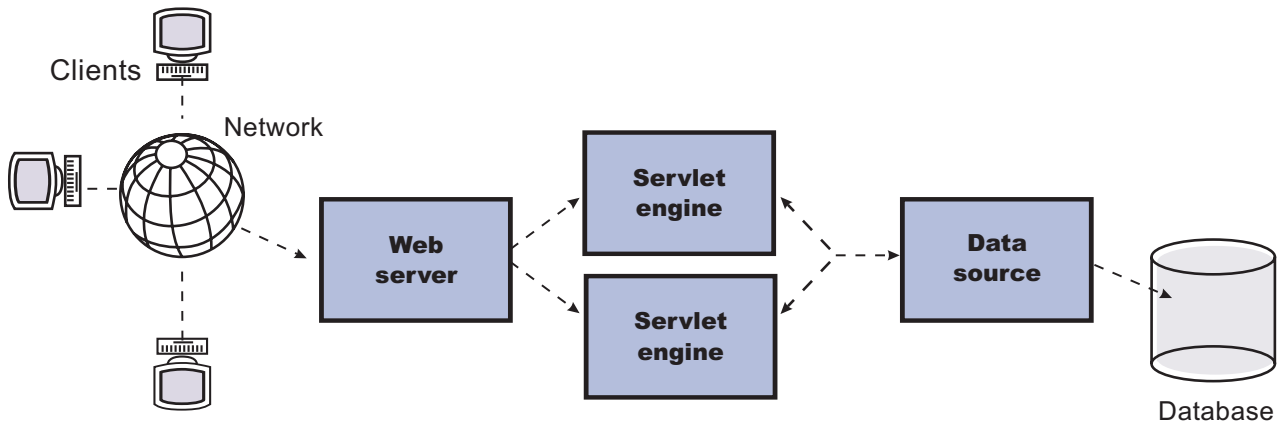
In a closed queue, requests can be active or waiting. An active request is doing work or waiting for a response from a downstream queue. For example, an active request in the Web server is doing work, such as retrieving static HTML, or waiting for a request to complete in the Web container. A waiting request is waiting to become active. The request remains in the waiting state until one of the active requests leaves the queue.

All Web servers supported by WebSphere Application Server are closed queues, as are WebSphere Application Server data sources. You can configure Web containers as open or closed queues. In general, it is best to make them closed queues. EJB containers are open queues. If there are no threads available in the pool, a new one is created for the duration of the request.

If enterprise beans are called by servlets, the Web container limits the number of total concurrent requests into an EJB container, because the Web container also has a limit. The Web container limits the number of total concurrent requests only if enterprise beans are called from the servlet thread of execution. Nothing prevents you from creating threads and bombarding the EJB container with requests. Therefore, servlets should not create their own work threads.

## Queuing and clustering

Cloning application servers can be a valuable asset in configuring highly-scalable production environments, especially when the application is experiencing bottlenecks that are preventing full CPU utilization of symmetric multiprocessing (SMP) servers. When adjusting the WebSphere Application Server system queues in clustered configurations, remember that when a server is added to a cluster, the server downstream receives twice the load.



**Figure Reference 1: Clustering and queuing**

Two servlet engines are located between a Web server and a data source. It is assumed that the Web server, servlet engines and data source, but not the database, are all running on a single SMP server. Given these constraints, the following queue considerations must be made:

- Double the Web server queue settings to ensure ample work is distributed to each Web container.
- Reduce the Web container thread pools to avoid saturating a system resource like CPU or another resource that the servlets are using.
- Reduce the data source to avoid saturating the database server.
- Reduce Java heap parameters for each instance of the application server. For versions of the Java virtual machine (JVM) shipped with WebSphere Application Server, it is crucial that the heap from all JVMs remain in physical memory. For example, if a cluster of four JVMs is running on a system, enough physical memory must be available for all four heaps.

## Queue configuration tips

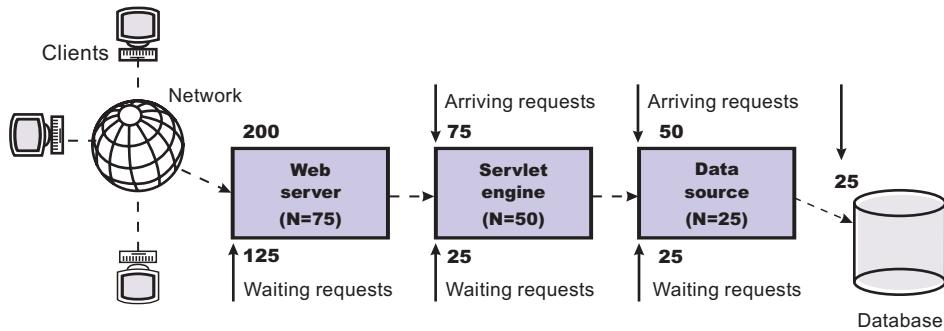
The following section outlines a methodology for configuring the WebSphere Application Server queues. Moving the database server onto another machine or providing more powerful resources, for example a faster set of CPUs with more memory, can dramatically change the dynamics of your system.

There are four tips for queuing:

- **Minimize the number of requests in WebSphere Application Server queues.**

In general, requests wait in the network in front of the Web server, rather than waiting in WebSphere Application Server. This configuration only supports those requests that are ready for processing to enter the queuing network. Specify that the queues furthest upstream or closest to the client are slightly

larger, and queues further downstream or furthest from the client are progressively smaller.



**Figure Reference 1: Upstream queuing network**

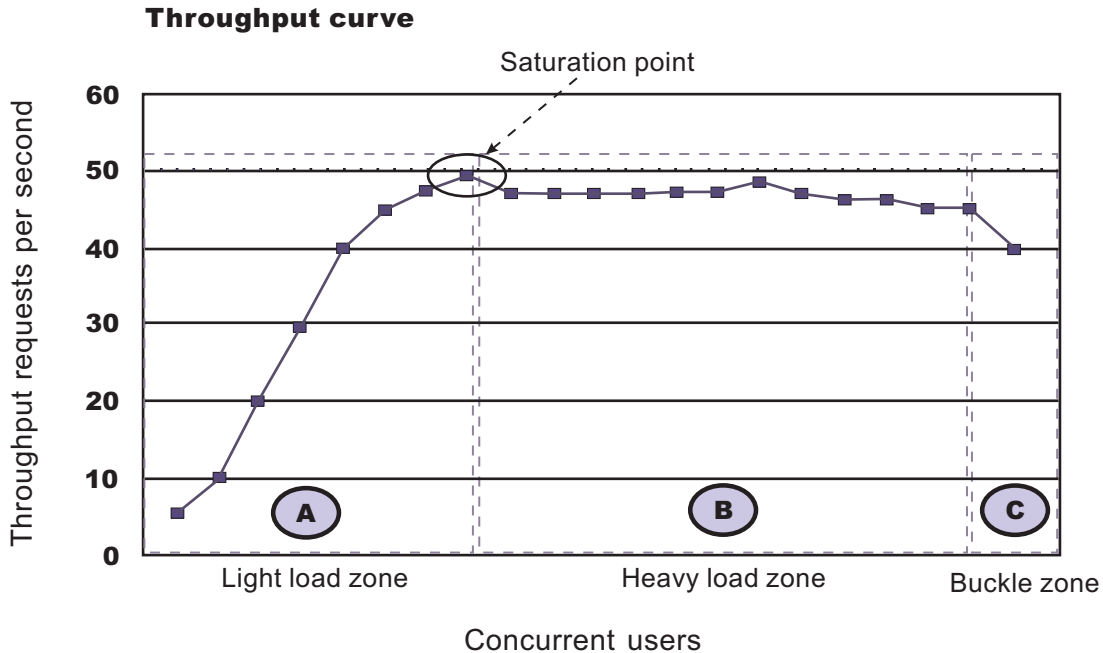
Queues in the queuing network become progressively smaller as work flows downstream. When 200 client requests arrive at the Web server, 125 requests remain queued in the network because the Web server is set to handle 75 concurrent clients. As the 75 requests pass from the Web server to the Web container, 25 requests remain queued in the Web server and the remaining 50 are handled by the Web container. This process progresses through the data source until 25 user requests arrive at the final destination, the database server. Because there is work waiting to enter a component at each point upstream, no component in this system must wait for work to arrive. The bulk of the requests wait in the network, outside of WebSphere Application Server. This type of configuration adds stability, because no component is overloaded.

You can then use the Edge Server to direct waiting users to other servers in a WebSphere Application Server cluster.

- **Draw throughput curves to determine when the system capabilities are maximized.**

You can use a test case that represents the full spirit of the production application by either exercising all meaningful code paths or using the production application. Run a set of experiments to determine when the system capabilities are fully stressed or when it has reached the saturation point. Conduct these tests after most of the bottlenecks are removed from the application. The goal of these tests is to drive CPUs to near 100% utilization. For maximum concurrency through the system, start the initial baseline experiment with large queues. For example, start the first experiment with a queue size of 100 at each of the servers in the queuing network: Web server, Web container and data source. Begin a series of experiments to plot a throughput curve, increasing the concurrent user load after each experiment. For example, perform experiments with one user, two users, five, 10, 25, 50, 100, 150 and 200 users. After each run, record the throughput requests per second, and response times in seconds per request. The curve resulting from the baseline experiments resembles the following typical

throughput curve shown as follows:



The WebSphere Application Server throughput is a function of the number of concurrent requests present in the total system. Section A, the light load zone, shows that the number of concurrent user requests increases, the throughput increases almost linearly with the number of requests. At light loads, concurrent requests face very little congestion within the WebSphere Application Server system queues. At some point, congestion starts to develop and throughput increases at a much lower rate until it reaches a saturation point that represents the maximum throughput value, as determined by some bottleneck in the WebSphere Application Server system. The most manageable type of bottleneck occurs when the WebSphere Application Server machine CPUs become fully utilized because adding CPUs or more powerful CPUs fixes the bottleneck.

In the heavy load zone or Section B, as the concurrent client load increases, throughput remains relatively constant. However, the response time increases proportionally to the user load. That is, if the user load is doubled in the heavy load zone, the response time doubles. At some point, represented by Section C, the buckle zone, one of the system components becomes exhausted. At this point, throughput starts to degrade. For example, the system might enter the buckle zone when the network connections at the Web server exhaust the limits of the network adapter or if the requests exceed operating system limits for file handles.

If the saturation point is reached by driving CPU utilization close to 100%, you can move on to the next step. If the saturation CPU occurs before system utilization reaches 100%, it is likely that another bottleneck is being aggravated by the application. For example, the application might be creating Java objects causing excessive garbage collection bottlenecks in the Java code.

There are two ways to manage application bottlenecks: remove the bottleneck or clone the bottleneck. The best way to manage a bottleneck is to remove it. You can use a Java-based application profiler, such as Rational Application Developer, Performance Trace Data Visualizer (PTDV), Borland's Optimizeit, JProbe or Jinsight to examine overall object utilization.

- **Decrease queue sizes while moving downstream from the client.**

The number of concurrent users at the throughput saturation point represents the maximum concurrency of the application. For example, if the application saturates WebSphere Application Server at 50 users, using 48 users might produce the best combination of throughput and response time. This value is called the Max Application Concurrency value. Max Application Concurrency becomes the preferred value for adjusting the WebSphere Application Server system queues. Remember, it is desirable for most users to wait in the network; therefore, queue sizes should increase when moving downstream farther from the client. For example, given a Max Application Concurrency value of 48, start

with system queues at the following values: Web server 75, Web container 50, data source 45. Perform a set of additional experiments adjusting these values slightly higher and lower to find the best settings. To help determine the number of concurrent users, view the Servlet Engine Thread Pool and Concurrently Active Threads metric in the Tivoli Performance Viewer.

- **Adjust queue settings to correspond to access patterns.**

In many cases, only a fraction of the requests passing through one queue enters the next queue downstream. In a site with many static pages, a number of requests are fulfilled at the Web server and are not passed to the Web container. In this circumstance, the Web server queue can be significantly larger than the Web container queue. In the previous example, the Web server queue was set to 75, rather than closer to the value of Max Application Concurrency. You can make similar adjustments when different components have different execution times.

For example, in an application that spends 90% of its time in a complex servlet and only 10% of its time making a short JDBC query, on average 10% of the servlets are using database connections at any time, so the database connection queue can be significantly smaller than the Web container queue. Conversely, if the majority of servlet execution time is spent making a complex query to a database, consider increasing the queue values at both the Web container and the data source. Always monitor the CPU and memory utilization for both the WebSphere Application Server and the database servers to verify that the CPU or memory are not saturating.





---

## Chapter 8. Preparing the operating system for product installation






This topic describes how to prepare your operating platform for installing one of the WebSphere Application Server products. Links in the topic exist for each operating system platform. Select your operating system to view the corresponding procedure.

Before preparing the installation environment, read “Planning to install Network Deployment” on page 54.

You must install the necessary prerequisites for your operating system. Use the following link to go to the procedure for your operating platform.

Prepare your operating system for installation.

Select the appropriate procedure:

-  “Preparing AIX systems for installation”
-  “Preparing HP-UX systems for installation” on page 85
-  “Preparing Linux systems for installation” on page 90
-  “Preparing Solaris systems for installation” on page 94
-  “Preparing Windows systems for installation” on page 97

You are now ready to install the WebSphere Application Server Network Deployment product.

---

### Preparing AIX systems for installation

This topic describes how to prepare an AIX system for the installation of IBM WebSphere Application Server, Version 6 products.

The installation uses a InstallShield for Multiplatforms (ISMP) wizard. You can use the graphical interface of the wizard. You can also use the wizard in silent mode.

Silent mode is a command line invocation with a parameter that identifies an options response file. Edit the options response file before installing.

If you encounter a problem such as needing more temporary space or missing prerequisite packages on your operating system, cancel the installation, make the required changes, and restart the installation.

Use the following procedure to prepare the operating system for the installation of WebSphere Application Server products.

1. Log on as root.

You cannot install the product correctly as a non-root user.

If you create a copy of the product CD-ROM, do so as root. Copies made from non-root users do not preserve the correct file attributes and do not work.

In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```


2. Stop all WebSphere Application Server-related Java processes on the machine where you are installing the product.
3. Stop any Web server process such as the IBM HTTP Server.
4. Use the System Management Interface Tool (SMIT) to display packages that are installed to determine whether you must update packages that are described in the following steps.
5. Download the most current version of the INFOZIP product to avoid problems with zipped files. Although zipped files are primarily used in the service stream, prepare your AIX operating system by downloading a current version of the INFOZIP package from the <http://www.info-zip.org> Web site.
6. Install the prerequisite x1C.rte 6.0 run-time code on AIX 5.2 maintenance level 1.  
You must install the x1C.rte 6.0 run-time code before you install the Global Security Kit 7 (GSKit7). The GSKit is installed as part of the installation of the IBM HTTP Server and also as part of the installation of Web server plug-ins for WebSphere Application Server. Download the run-time code as a fix from the AIX Support site.

If you have AIX 5.2, you can install the x1C.rte 6.0 run-time code from the AIX 5.2 CD.

7. Provide adequate disk space.

With the JFS file system on AIX, you can allocate expansion space for directories. If the Installation wizard does not have enough space, ISMP issues a system call for more space that increases the space allocation dynamically. The message you might see when this occurs for the /usr directory is similar to the following example:

NOTE: The following file systems will be expanded during the installation:  
/usr

 Manually verify that the required space for creating a profile is available on AIX. A known problem in the underlying InstallShield for Multiplatforms (ISMP) code prevents proper space checking on AIX systems at the time that the product disc was created.

### Attention:

The Network Deployment product requires the following disk space:

#### **730 MB for the /usr/IBM/WebSphere/AppServer directory before creating profiles**

The installation root directory includes the core product files. This size does not include space for profiles or applications. Profiles require 40 MB of temp space in addition to the sizes shown. Profiles have the following space requirements:

#### **30 MB for the Deployment manager profile**

The Tivoli Global Security Kit requires this space.

#### **200 MB for an Application Server profile with the Sample applications**

This size does not include space for applications that you might develop and install.

#### **10 MB for an unfederated custom profile**

This size does not include space for applications that you might develop and install. The requirement does include space for the node agent. However, you must federate a custom profile to create an operational managed node.

After federating a custom profile, the resulting managed node contains a functional node agent only. Use the deployment manager to create server processes on the managed node.

#### **100 MB for the /tmp directory**

The temporary directory is the working directory for the installation program.

### 830 MB total requirement

This amount is the total space requirement when installing the product from the CD and when not installing service. Installing profiles requires more space.

The following space is required for the IBM HTTP Server product:

#### 110 MB for the `/usr/IBMIHS` directory

The IBM HTTP Server product requires this space.

#### 35 MB for the `/usr/ibm/gsk7` directory

The Tivoli Global Security Kit requires this space.

The run-time module is `gskkm.rte`.

The following space is the maximum amount that is required for the Web server plug-ins for WebSphere Application Server:

#### 200 MB for the `/usr/IBM/WebSphere/plugins` directory

The Web server plug-ins require this disk space.

#### 25 MB for the `/usr/ibm/gsk7` directory

The Tivoli Global Security Kit requires this space.

The following space is required for the IBM HTTP Server product:

#### 110 MB for the `/opt/IBMIHS` directory

The IBM HTTP Server product requires this space.

#### 25 MB for the `/opt/ibm/gsk7` directory

The Tivoli Global Security Kit requires this space.

The following space is required for the WebSphere Application Server Clients:

#### 150 MB for the `/usr/IBM/WebSphere/AppClient` directory

The amount of space required to install the application clients is actually less than 150 MB.

The amount of space depends on the clients that you install as features.

The Installation wizard for each component displays required space on the confirmation panel before you install the product files and selected features. The Installation wizard also warns you if you do not have enough space to install the product.

If you plan to migrate applications and the configuration from a previous version, verify that the application objects have enough disk space. As a rough guideline, plan for space equal to 110 percent of the size of the application objects:

- **For Version 4.0.x:** The size of enterprise archive (EAR) files
- **For Version 5.0.x:** The size of EAR files

8. Unmount file systems with broken links to avoid `java.lang.NullPointerException` errors.

Unmount file systems with broken links before installing.

Installation can fail with the following error when broken links exist to file systems:

An error occurred during wizard bean change notification:

```
java.lang.NullPointerException
  at com.ibm.wizard.platform.aix.AixFileUtils.
    getFileSystemData(AixFileUtils.java:388)
  at com.ibm.wizard.platform.aix.AixFileUtils.
    getPartitionDataWithExecs(AixFileUtils.java:172)
  at com.ibm.wizard.platform.aix.AixFileUtils.
    getPartitionData(AixFileUtils.java:104)
  at com.ibm.wizard.platform.aix.AixFileServiceImpl.
    getPartitionNames(AixFileServiceImpl.java:397)
...
```

Use the **df -k** command to check for broken links to file systems. Look for file systems that list blank values in the *1024-blocks size* column. Columns with a value of "-" (dash) are not a problem. The following example shows a problem with the `/dev/lv00` file system:

```
> df -k
Filesystem      1024-blocks      Free %Used    Iused %Iused Mounted on
/dev/hd4         1048576      447924   58%     2497    1% /
/dev/hd3         4259840     2835816   34%      484    1% /tmp
/proc            -            -         -         -      - /proc
/dev/lv01        2097152     229276   90%     3982    1% /storage
/dev/lv00
/dev/hd2         2097152     458632   79%    42910    9% /usr
iw031864:/cdrom/db2_v72_eee_aix32_sbcs
```

The `/proc` file system is not a problem. The `iw031864:/cdrom/db2_v72_eee_aix32_sbcs` file system is a definite problem. The `/dev/lv00` file system is also a likely problem. Use one of the following commands to solve this problem:

```
> umount /cdrom/db2_v72_eee_aix32_sbcs
> umount /cdrom
```

Start the installation again. If the problem continues, unmount any file systems that have blank values, such as the `/dev/lv00` file system in the example. If you cannot solve the problem by unmounting file systems with broken links, reboot the machine and start the installation again.

9. Verify that prerequisites and corequisites are at the required release levels.

Although the Installation wizard checks for prerequisite operating system patches with the `prereqChecker` application, review the prerequisites on the IBM WebSphere Application Server supported hardware, software, and APIs Web site if you have not already done so.

Refer to the documentation for non-IBM prerequisite and corequisite products to learn how to migrate to their supported versions.

10. Verify the system **cp** command when using emacs or other freeware.

If you have emacs or other freeware installed on your operating system, verify that the system **cp** command is used.

- a. Type which `cp` at the command prompt before running the installation program for the WebSphere Application Server product.
- b. Remove the freeware directory from your `PATH` if the resulting directory output includes freeware. For example, assume that the output is similar to the following message: `.../freeware/bin/cp`. If so, remove the directory from the `PATH`.
- c. Install the WebSphere Application Server product.
- d. Add the freeware directory back to the `PATH`.

If you install with a **cp** command that is part of a freeware package, the installation might appear to complete successfully, but the Java 2 SDK that the product installs might have missing files in the `install_root/java` directory.

Missing files can destroy required symbolic links. If you remove the freeware **cp** command from the `PATH`, you can install the Application Server product successfully.

Perform the following step to verify that the Java 2 SDK is working correctly.

11. Verify that the Java 2 SDK on your copy of the product CD is functioning correctly.

If you created your own product CD from an ISO image or by copying the actual CD, perform the following steps to verify that the Java 2 SDK on the product CD-ROM is working correctly.

- a. Change directories to the `/mnt/JDK/repository/prereq.jdk/java/bin` directory on the product CD-ROM. For example:

```
cd /mnt/JDK/repository/prereq.jdk/java/bin
```

- b. Verify the Java 2 SDK version. Type the following command:

```
./java -version
```

The command completes successfully with no errors when the Java 2 SDK is intact.

12. **Optional:** Install the Mozilla browser if it is not already installed. The Mozilla browser supports the launchpad console.
  - a. Use `smit` to see if the Mozilla 1.4 or 1.7 or later package is already installed.
  - b. Download two prerequisites from the AIX Toolbox for Linux Applications (glib & gtk+). Download the packages from the following locations:
    - `ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/glib/glib-1.2.10-2.aix4.3.ppc.rpm`
    - `ftp://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/RPMS/ppc/gtkplus/gtkplus-1.2.10-4.aix5.1.ppc.rpm`
  - c. Install the packages after downloading them: Use the following command:

```
rpm -Uvh glib-1.2.10-2.aix4.3.ppc.rpm gtkplus-1.2.10-4.aix5.1.ppc.rpm
```
  - d. Download the latest supported version of Mozilla (1.7 or later) for AIX. Download Mozilla for AIX from the following location:  
`http://www14.software.ibm.com/webapp/download/search.jsp?go=y&rs=mozilla`  
Download the installp image and install it from `smit`.  
The Mozilla images distributed on the `http://www.mozilla.org` Web site are not provided by IBM. IBM has not tested and does not support the download from the non-IBM Web site. Download the Mozilla images from the IBM Web site to verify that the version that you download is tested and supported.
13. **Optional:** Export the location of the supported browser.

Export the location of the supported browser using a command that identifies the actual location of the browser.

For example, if the Mozilla package is in the `bin /mozilla` directory, use the following command:  
`EXPORT BROWSER=/usr/bin/mozilla`
14. **Silent installation only:** Make an allowance for a known ISMP problem that causes a call to the Xwindows service during a silent installation.

The `DISPLAY` environment variable on your AIX machine might point to an X server that is not logged in. Two common scenarios can cause this to occur:

  - Your AIX machine has an X server running, but the X server is stuck at the graphical login screen because you have not yet logged in.
  - Your AIX machine is configured to display X Windows applications on a remote X server that is not logged in.

A silent installation can hang in either case as ISMP calls Xwindows services.

Two solutions exist:

  - Login to the local X server through the graphical user interface before beginning the silent installation.
  - Export the `DISPLAY` environment variable to point to null or blank.  
`EXPORT DISPLAY=null`

---

## Preparing HP-UX systems for installation

This topic describes how to prepare an HP-UX system for the installation of IBM WebSphere Application Server, Version 6 products.

The installation uses a InstallShield for Multiplatforms (ISMP) wizard. You can use the graphical interface of the wizard. You can also use the wizard in silent mode.

Silent mode is a command line invocation with a parameter that identifies an options response file. Edit the options response file before installing.

If you encounter a problem such as needing more temporary space or missing prerequisite packages on your operating system, cancel the installation, make the required changes, and restart the installation.

Use the following procedure to prepare the operating system for the installation of WebSphere Application Server products.

1. Log on as root.

You cannot install the product correctly as a non-root user.

If you create a copy of the product CD-ROM, do so as root. Copies made from non-root users do not preserve the correct file attributes and do not work.

In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```

2. **Optional:** Download and install the Mozilla Web browser so that you can use the launchpad application on the product disc. If you do not have the Mozilla Web browser, download and install the browser from <http://www.mozilla.org>.

3. **Optional:** Export the location of the supported browser. Export the location of the supported browser using a command that identifies the actual location of the browser.

For example, if the Mozilla package is in the `bin /mozilla` directory, use the following command:

```
EXPORT BROWSER=/opt/bin/mozilla
```

4. Stop all WebSphere Application Server-related Java processes on the machine where you are installing the product.
5. Stop any Web server process such as the IBM HTTP Server.
6. Provide adequate disk space.

### **Attention:**

The Network Deployment product requires the following disk space:

#### **780 MB for the `/opt/IBM/WebSphere/AppServer` directory before creating profiles**

The installation root directory includes the core product files. This size does not include space for profiles or applications. Profiles require 40 MB of temp space in addition to the sizes shown. Profiles have the following space requirements:

#### **30 MB for the Deployment manager profile**

This size does not include space for Sample applications that you might install. The size also does not include space for applications that you might deploy.

#### **200 MB for an application server profile with the Sample applications**

This size does not include space for applications that you might develop and install.

#### **10 MB for an unfederated custom profile**

This size does not include space for applications that you might develop and install. The requirement does include space for the node agent. However, you must federate a custom profile to create an operational managed node.

After federating a custom profile, the resulting managed node contains a functional node agent only. Use the deployment manager to create server processes on the managed node.



### **100 MB for the /tmp directory**

The temporary directory is the working directory for the installation program.

### **880 MB total requirement**

This amount is the total space requirement when installing the product from the CD and when not installing service. Installing profiles requires more space.

The following space is required for the IBM HTTP Server product:

### **110 MB for the /opt/IBMIHS directory**

The IBM HTTP Server product requires this space.

### **25 MB for the /opt/ibm/gsk7 directory**

The Tivoli Global Security Kit requires this space.

The run-time module is gsk7bas for HP-UX platforms.

The following space is the maximum amount that is required for the Web server plug-ins for WebSphere Application Server:

### **280 MB for the /opt/IBM/WebSphere/plugins directory**

The Web server plug-ins require this disk space.

### **25 MB for the /opt/ibm/gsk7 directory**

The Tivoli Global Security Kit requires this space.

The following space is required for the IBM WebSphere Application Server Clients:

### **150 MB for the /opt/IBM/WebSphere/AppClient directory**

The amount of space required to install the application clients is actually less than 150 MB. The amount of space depends on the clients that you install as features.

The Installation wizard for each component displays required space on the confirmation panel before you install the product files and selected features. The Installation wizard also warns you if you do not have enough space to install the product.

If you plan to migrate applications and the configuration from a previous version, verify that the application objects have enough disk space. As a rough guideline, plan for space equal to 110 percent of the size of the application objects:

- **For Version 4.0.x:** The size of enterprise archive (EAR) files
- **For Version 5.0.x:** The size of EAR files

7. Set kernel values to support Application Server. Several HP-UX kernel values are typically too small for the product.

To set kernel parameters, perform the following steps:

- a. Log into the host machine as root.
- b. Determine the physical memory, which you must know to avoid setting certain kernel parameters above the physical capacity:
  - 1) Start the HP-UX System Administration Manager (SAM) utility.
  - 2) Select **Performance Monitors > System Properties > Memory**.
  - 3) Note the value for Physical Memory and click **OK**.
  - 4) Exit from the SAM utility.
- c. Set the maxfiles and maxfiles\_lim parameters to at least 4096. (The following table recommends 8000 and 8196, respectively. You must first edit the /usr/conf/master.d/core-hpux file, so the SAM utility can set values greater than 2048:
  - 1) Open the /usr/conf/master.d/core-hpux file in a text editor.
  - 2) Change the line, " \*range maxfiles<=2048" to " \*range maxfiles<=60000"
  - 3) Change the line, " \*range maxfiles\_lim<=2048" to " \*range maxfiles\_lim<=60000"
  - 4) Save and close the file. Old values might be stored in the /var/sam/boot.config file. Force the SAM utility to create a new boot.config file:

- a) Move the existing version of the `/var/sam/boot.config` file to another location, such as the `/tmp` directory.
- b) Start the SAM utility.
- c) Select **Kernel Configuration > Configurable Parameters**. When the Kernel Configuration window opens, a new `boot.config` file exists.

Alternatively, rebuild the `boot.config` file with the following command:

```
# /usr/sam/lbin/getkinfo -b
```

- d. Set new kernel parameter values:
  - 1) Start the SAM utility.
  - 2) Click **Kernel Configuration > Configurable Parameters**.
  - 3) For each of the parameters in the following table, perform this procedure:
    - a) Highlight the parameter to change.
    - b) Click **Actions > Modify Configurable Parameter**.
    - c) Type the new value in the **Formula/Value** field.
    - d) Click **OK**.

Typical kernel settings for running WebSphere Application Server display in the following table:

Parameter	Value
<code>dbc_max_pct</code>	25
<code>maxdsiz</code>	805306358
<code>maxdsiz</code>	2048000000 (when running multiple profiles on the same system)
<code>maxfiles_lim</code>	8196 (Change this one before maxfiles.)
<code>maxfiles</code>	8000
<code>maxssiz</code>	8388608
<code>maxswapchunks</code>	8192
<code>max_thread_proc</code>	3000
<code>maxuprc</code>	512
<code>maxusers</code>	512
<code>msgmap</code>	2048
<code>msgmax</code>	65535
<code>msgmax</code>	131070 (when running multiple profiles on the same system)
<code>msgmnb</code>	65535
<code>msgmnb</code>	131070 (when running multiple profiles on the same system)
<code>msgmni</code>	50
<code>msgseg</code>	32767
<code>msgssz</code>	32
<code>msgtql</code>	2046
<code>nfile</code>	58145
<code>nflocks</code>	3000
<code>ninode</code>	60000
<code>nkthread</code>	7219
<code>nproc</code>	4116
<code>npty</code>	2024
<code>nstrpty</code>	1024
<code>nstrtel</code>	60
<code>sema</code>	1



Parameter	Value
semaem	16384
semmap	514
semmni	2048
semmns	16384
semmnu	1024
semume	200
semvmx	32767
shmmax	2147483647
shmем	1
shmmni	1024
shmseg	1024
STRMSGSZ	65535

When WebSphere Application Server and IBM DB2 are on the same machine, some kernel values are higher than those shown in the preceding table.

See the Recommended HP-UX kernel configuration parameters for DB2 V8 Web page for more information.

e. Click **Actions > Process New Kernel**.

f. Click **Yes** on the information window to confirm your decision to restart the machine.

Follow the on-screen instructions to restart your machine and to enable the new settings.

g. If you plan to redirect displays to non-HP machines, do the following before running the WebSphere Application Server installation wizard:

1) Issue the following command to obtain information on all the public locales that are accessible to your application:

```
# locale -a
```

2) Choose a value for your system from the output that is displayed and set the LANG environment variable to this value. Here is an example command that sets the value of LANG to en\_US.iso88591

```
# export LANG=en_US.iso88591
```

8. Verify that prerequisites and corequisites are at the required release levels.

Although the Installation wizard checks for prerequisite operating system patches with the prereqChecker application, review the prerequisites on the IBM WebSphere Application Server supported hardware, software, and APIs Web site if you have not already done so.

Refer to the documentation for non-IBM prerequisite and corequisite products to learn how to migrate to their supported versions.

9. Verify the system **cp** command when using emacs or other freeware.

If you have emacs or other freeware installed on your operating system, verify that the system **cp** command is used.

a. Type which cp at the command prompt before running the installation program for the WebSphere Application Server product.

b. Remove the freeware directory from your PATH if the resulting directory output includes freeware. For example, assume that the output is similar to the following message: `.../freeware/bin/cp`. If so, remove the directory from the PATH.

c. Install the WebSphere Application Server product.

d. Add the freeware directory back to the PATH.

If you install with a **cp** command that is part of a freeware package, the installation might appear to complete successfully, but the Java 2 SDK that the product installs might have missing files in the *install\_root/java* directory.

Missing files can destroy required symbolic links. If you remove the freeware **cp** command from the PATH, you can install the Application Server product successfully.

Perform the following step to verify that the Java 2 SDK is working correctly.

10. Verify that the Java 2 SDK on your copy of the product CD is functioning correctly.

If you created your own product CD from an ISO image or by copying the actual CD, perform the following steps to verify that the Java 2 SDK on the product CD-ROM is working correctly.

- a. Change directories to the `/mnt/JDK/repository/prereq.jdk/java/bin` directory on the product CD-ROM. For example:

```
cd /mnt/JDK/repository/prereq.jdk/java/bin
```

- b. Verify the Java 2 SDK version. Type the following command:

```
./java -version
```

The command completes successfully with no errors when the Java 2 SDK is intact.

---

## Preparing Linux systems for installation

This topic describes how to prepare a Linux system for installing IBM WebSphere Application Server, Version 6 products.

The installation uses a InstallShield for Multiplatforms (ISMP) wizard. You can use the graphical interface of the wizard. You can also use the wizard in silent mode.

Silent mode is a command line invocation with a parameter that identifies an options response file. Edit the options response file before installing.

If you encounter a problem such as needing more temporary space or missing prerequisite packages on your operating system, cancel the installation, make the required changes, and restart the installation.

Use the following procedure to prepare the operating system for the installation of WebSphere Application Server products.

1. Log on as root.

You cannot install the product correctly as a non-root user.

If you create a copy of the product CD-ROM, do so as root. Copies made from non-root users do not preserve the correct file attributes and do not work.

In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```

2. **Optional:** Download and install the Mozilla Web browser so that you can use the launchpad application on the product disc. If you do not have the Mozilla Web browser, download and install the browser from <http://www.mozilla.org>.

3. **Optional:** Export the location of the supported browser. Export the location of the supported browser using a command that identifies the actual location of the browser.

For example, if the Mozilla package is in the `bin /mozilla` directory, use the following command:

```
EXPORT BROWSER=/opt/bin/mozilla
```

4. Stop all WebSphere Application Server-related Java processes on the machine where you are installing the product.
5. Stop any Web server process such as the IBM HTTP Server.

6. Provide adequate disk space.

**Attention:**

The Network Deployment product requires the following disk space:

**730 MB for the /opt/IBM/WebSphere/AppServer directory before creating profiles**

The installation root directory includes the core product files. This size does not include space for profiles or applications. Profiles require 40 MB of temp space in addition to the sizes shown. Profiles have the following space requirements:

**30 MB for the Deployment manager profile**

This size does not include space for Sample applications that you might install. The size also does not include space for applications that you might deploy.

**200 MB for an Application Server profile with the Sample applications**

This size does not include space for applications that you might develop and install.

**10 MB for an unfederated custom profile**

This size does not include space for applications that you might develop and install. The requirement does include space for the node agent. However, you must federate a custom profile to create an operational managed node.

After federating a custom profile, the resulting managed node contains a functional node agent only. Use the deployment manager to create server processes on the managed node.

**100 MB for the /tmp directory**

The temporary directory is the working directory for the installation program.

**830 MB total requirement**

This amount is the total space requirement when installing the product from the CD and when not installing service. Installing profiles requires more space.

The following space is the maximum amount that is required for the Web server plug-ins for WebSphere Application Server:

**200 MB for the /usr/IBM/WebSphere/plugins directory**

The Web server plug-ins require this disk space.

**25 MB for the /usr/ibm/gsk7 directory**

The Tivoli Global Security Kit requires this space.

The run-time module is gskkm.rte.

The following space is required for the IBM HTTP Server product:

**110 MB for the /opt/IBMIHS directory**

The IBM HTTP Server product requires this space.

**25 MB for the /opt/ibm/gsk7 directory**

The Tivoli Global Security Kit requires this space.

The run-time module is gsk7bas.i386.rpm for Linux distributed platforms and gsk7bas.s390.rpm for Linux for S/390 platforms.

The following space is required for the IBM WebSphere Application Server Clients:

## 150 MB for the /opt/IBM/WebSphere/AppClient directory

The amount of space required to install the application clients is actually less than 150 MB. The amount of space depends on the clients that you install as features.

### Attention:

The following space is required for the Application Server Toolkit:

## 550 MB maximum for the /opt/IBM/WebSphere/AST directory

The Application Server Toolkit does not include an integrated test environment.

Install the WebSphere Application Server Network Deployment product and configure an application server to create a test environment.

The Installation wizard for each component displays required space on the confirmation panel before you install the product files and selected features. The Installation wizard also warns you if you do not have enough space to install the product.

If you plan to migrate applications and the configuration from a previous version, verify that the application objects have enough disk space. As a rough guideline, plan for space equal to 110 percent of the size of the application objects:

- **For Version 4.0.x:** The size of enterprise archive (EAR) files
- **For Version 5.0.x:** The size of EAR files

7. Verify that prerequisites and corequisites are at the required release levels.

Although the Installation wizard checks for prerequisite operating system patches with the prereqChecker application, review the prerequisites on the IBM WebSphere Application Server supported hardware, software, and APIs Web site if you have not already done so.

Refer to the documentation for non-IBM prerequisite and corequisite products to learn how to migrate to their supported versions.

8. Provide necessary prerequisites for Red Hat Enterprise Linux V3.0. A known limitation exists in the prerequisites checker program when examining prerequisite packages on Linux systems. Although not all of the following packages are required for Version 6, the packages are required if you install Version 6 to coexist with a Version 5.x installation.

You must install the following packages on Red Hat Enterprise Linux V3.0 for xSeries platforms:

- compat-gcc-7.3-2.96.122
- compat-libstdc++-7.3-2.96.122
- compat-libstdc++-devel-7.3-2.96.122
- compat-glibc-7.x-2.2.4.32.5
- compat-gcc-c++-7.3-2.96.122
- compat-db-4.0.14-5
- rpm-build-4.2.1-4.2

The following package is required for Red Hat Enterprise Linux V3.0 for pSeries platforms:  
rpm-build-4.2.1-4.2.

You must install the following packages on Red Hat Enterprise Linux V3.0 for S/390 (z/VM and VM/ESA) platforms:

- compat-db-4.0.14-5
- compat-pwdb-0.62-3
- compat-libstdc++-7.2-2.95.3.77
- rpm-build-4.2.1-4.2

You can install a later release of any of these packages.

9. Upgrade Red Hat Enterprise Linux V3.0 to service level 2 (update 2) or service level 3 (update 3). Upgrade the RHEL 3.0 service level by downloading and installing the service updates from Red Hat.

If you do not upgrade the service level, certain national language issues in the released version of GLIBC included in Red Hat Enterprise Linux V3 might cause Java virtual machine (JVM) failures due to segmentation faults. These failures can occur during installation when the locale is set to anything other than an English locale.

Do not install, log off, and log back on when you are installing from the operator console attached to the machine. This can produce segmentation faults that cause the installation to fail.

See Appendix B, “Troubleshooting installation,” on page 337 for more information.

10. Prepare the SuSE Linux Enterprise Server 8.0 - Powered by UnitedLinux 1.0 operating platform for WebSphere Application Server installation.

- a. Install SP3 for the United Linux 1.0 operating platform to let you use the Launchpad.

It is your responsibility to install this service pack. The `prereqChecker` function of the installer cannot detect service pack versions definitively on United Linux. Kernel unames and versions between 8.0 and 8.0.3 are identical. No signature RPM denotes a service pack install.

- b. Use the IBM Developer Kit that WebSphere Application Server provides to support the Java 2 SDK on the SuSE SLES 8.0 operating system to avoid potential problems when uninstalling an interim fix or a fix pack. To use the IBM Developer Kit, remove the `java2-jre-1.3.1-524` and `java2-1.3.1-524` RPMs from the machine before installing WebSphere Application Server.

11. Correct font problems on SuSE Linux Enterprise Server 8.0 in Simplified Chinese and Traditional Chinese locales.

On the Linux for Power platform that SuSE Linux Enterprise Server 8.0 provides, a missing package causes a font problem. The `ttf-hanyi` package is not installed during the normal product installation of the SuSE 8.0 operating system. The missing package causes the Installation wizard for WebSphere Application Server products to display garbled characters in the Simplified Chinese locale and in the Traditional Chinese locale.

Copy the `ttf-hanyi-2021016-0.noarch.rpm` package on the SuSE 8.0 for i386 CD to the Power PC system. Install the package on the Power PC machine and reboot the machine to solve the problem.

12. Verify the system `cp` command when using emacs or other freeware.

If you have emacs or other freeware installed on your operating system, verify that the system `cp` command is used.

- a. Type which `cp` at the command prompt before running the installation program for the WebSphere Application Server product.
- b. Remove the freeware directory from your PATH if the resulting directory output includes freeware. For example, assume that the output is similar to the following message: `.../freeware/bin/cp`. If so, remove the directory from the PATH.
- c. Install the WebSphere Application Server product.
- d. Add the freeware directory back to the PATH.

If you install with a `cp` command that is part of a freeware package, the installation might appear to complete successfully, but the Java 2 SDK that the product installs might have missing files in the `install_root/java` directory.

Missing files can destroy required symbolic links. If you remove the freeware `cp` command from the PATH, you can install the Application Server product successfully.

Perform the following step to verify that the Java 2 SDK is working correctly.

13. Verify that the Java 2 SDK on your copy of the product CD is functioning correctly.

If you created your own product CD from an ISO image or by copying the actual CD, perform the following steps to verify that the Java 2 SDK on the product CD-ROM is working correctly.

- a. Change directories to the `/mnt/JDK/repository/prereq.jdk/java/bin` directory on the product CD-ROM. For example:

```
cd /mnt/JDK/repository/prereq.jdk/java/bin
```

- b. Verify the Java 2 SDK version. Type the following command:

```
./java -version
```

The command completes successfully with no errors when the Java 2 SDK is intact.

---

## Preparing Solaris systems for installation

This topic describes how to prepare Solaris systems for the installation of IBM WebSphere Application Server, Version 6 products.

The installation uses a InstallShield for Multiplatforms (ISMP) wizard. You can use the graphical interface of the wizard. You can also use the wizard in silent mode.

Silent mode is a command line invocation with a parameter that identifies an options response file. Edit the options response file before installing.

If you encounter a problem such as needing more temporary space or missing prerequisite packages on your operating system, cancel the installation, make the required changes, and restart the installation.

Use the following procedure to prepare the operating system for the installation of WebSphere Application Server products.

1. Log on as root.

You cannot install the product correctly as a non-root user.

If you create a copy of the product CD-ROM, do so as root. Copies made from non-root users do not preserve the correct file attributes and do not work.

In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```

2. **Optional:** Download and install the Mozilla Web browser so that you can use the launchpad application on the product disc. If you do not have the Mozilla Web browser, download and install the browser from <http://www.mozilla.org>.
3. **Optional:** Export the location of the supported browser. Export the location of the supported browser using a command that identifies the actual location of the browser.

For example, if the Mozilla package is in the `bin /mozilla` directory, use the following command:

```
EXPORT BROWSER=/opt/bin/mozilla
```

4. Stop all WebSphere Application Server-related Java processes on the machine where you are installing the product.
5. Stop any Web server process such as the IBM HTTP Server.
6. Provide adequate disk space.

### Attention:

The Network Deployment product requires the following disk space:

#### **730 MB for the `/opt/IBM/WebSphere/AppServer` directory before creating profiles**

The installation root directory includes the core product files. This size does not include space for profiles or applications. Profiles require 40 MB of temp space in addition to the sizes shown. Profiles have the following space requirements:

#### **30 MB for the Deployment manager profile**

This size does not include space for Sample applications that you might install. The size also does not include space for applications that you might deploy.

### **200 MB for an Application Server profile with the Sample applications**

This size does not include space for applications that you might develop and install.

### **10 MB for an unfederated custom profile**

This size does not include space for applications that you might develop and install. The requirement does include space for the node agent. However, you must federate a custom profile to create an operational managed node.

After federating a custom profile, the resulting managed node contains a functional node agent only. Use the deployment manager to create server processes on the managed node.

### **100 MB for the /tmp directory**

The temporary directory is the working directory for the installation program.

### **1030 MB total requirement**

This amount is the total space requirement when installing the product from the CD and when not installing service. Installing profiles requires more space.

The following space is required for the IBM HTTP Server product:

#### **110 MB for the /opt/IBMIHS directory**

The IBM HTTP Server product requires this space.

#### **25 MB for the /opt/ibm/gsk7 directory**

The Tivoli Global Security Kit requires this space.

The following space is the maximum amount that is required for the Web server plug-ins for WebSphere Application Server:

#### **200 MB for the /opt/IBM/WebSphere/plugins directory**

The Web server plug-ins require this disk space.

#### **35 MB for the /opt/ibm/gsk7 directory**

The Tivoli Global Security Kit requires this space.

The following space is required for the IBM WebSphere Application Server Clients:

#### **150 MB for the /opt/IBM/WebSphere/AppClient directory**

The amount of space required to install the application clients is actually less than 150 MB. The amount of space depends on the clients that you install as features.

The Installation wizard for each component displays required space on the confirmation panel before you install the product files and selected features. The Installation wizard also warns you if you do not have enough space to install the product.

If you plan to migrate applications and the configuration from a previous version, verify that the application objects have enough disk space. As a rough guideline, plan for space equal to 110 percent of the size of the application objects:

- **For Version 4.0.x:** The size of enterprise archive (EAR) files
- **For Version 5.0.x:** The size of EAR files

7. Set kernel values to support Application Server. Several Solaris kernel values are typically too small.

Before installing, review the machine configuration:

```
sysdef -i
```

The kernel values are set in the /etc/system file, as shown in the following example.



```

set shmsys:shminfo_shmmax = 4294967295
set shmsys:shminfo_shmseg = 1024
set shmsys:shminfo_shmmni = 1024
set semsys:seminfo_semaem = 16384
set semsys:seminfo_semmni = 1024
set semsys:seminfo_semmap = 1026
set semsys:seminfo_semmns = 16384
set semsys:seminfo_semmsl = 100
set semsys:seminfo_semopm = 100
set semsys:seminfo_semmnu = 2048
set semsys:seminfo_semume = 256
set msgsys:msginfo_msgmap = 1026
set msgsys:msginfo_msgmax = 65535
set rlim_fd_cur=1024

```

You can change kernel values by editing the `/etc/system` file then rebooting the operating system. For more information about setting up the Solaris system, see the Solaris System Administration documentation. For example, the *Solaris Tunable Parameters Reference Manual*.

Queue managers are generally independent of each other. Therefore system kernel parameters, for example `shmmni`, `semmni`, `semmns`, and `semmnu` need to allow for the number of queue managers in the system.

8. Verify that prerequisites and corequisites are at the required release levels.

Although the Installation wizard checks for prerequisite operating system patches with the `prereqChecker` application, review the prerequisites on the IBM WebSphere Application Server supported hardware, software, and APIs Web site if you have not already done so.

Refer to the documentation for non-IBM prerequisite and corequisite products to learn how to migrate to their supported versions.

Solaris 9 SPARC workstations require the following patches to fix certain :

- 112874-16 SunOS 5.9: patch `libc`
- 113319-12 SunOS 5.9: `libnsl nispaswdd` patch
- 115545-01 SunOS 5.9: `nss_files` patch
- 115542-01 SunOS 5.9: `nss_user` patch
- 115544-01 SunOS 5.9: `nss_compat` patch

Solaris 8 SPARC workstations require the following patches to fix certain :

- 115827-01 SunOS 5.8: `/sbin/sulogin` and `/sbin/netstrategy` patch
- 115583-01 SunOS 5.8: `/usr/lib/nss_user.so.1` patch
- 108820-02 SunOS 5.8: `nss_compat.so.1` patch
- 113648-02 SunOS 5.8: `/usr/sbin/mount` patch
- 108993-27 SunOS 5.8: LDAP2 client, `libc`, `libthread` and `libnsl` libraries patch
- 108528-24 SunOS 5.8: kernel update patch

Refer to the IBM WebSphere Application Server supported hardware, software, and APIs Web site for a complete list of required patches and to determine whether your operating system is supported when you receive a message from the `prereqChecker` program. The Web site lists all supported operating systems and the operating system fixes and patches that you must install to have a compliant operating system.

9. Verify the system `cp` command when using emacs or other freeware.

If you have emacs or other freeware installed on your operating system, verify that the system `cp` command is used.

- a. Type `which cp` at the command prompt before running the installation program for the WebSphere Application Server product.
- b. Remove the freeware directory from your `PATH` if the resulting directory output includes freeware. For example, assume that the output is similar to the following message: `.../freeware/bin/cp`. If so, remove the directory from the `PATH`.



- c. Install the WebSphere Application Server product.
- d. Add the freeware directory back to the PATH.

If you install with a **cp** command that is part of a freeware package, the installation might appear to complete successfully, but the Java 2 SDK that the product installs might have missing files in the *install\_root/java* directory.

Missing files can destroy required symbolic links. If you remove the freeware **cp** command from the PATH, you can install the Application Server product successfully.

Perform the following step to verify that the Java 2 SDK is working correctly.

10. Verify that the Java 2 SDK on your copy of the product CD is functioning correctly.

If you created your own product CD from an ISO image or by copying the actual CD, perform the following steps to verify that the Java 2 SDK on the product CD-ROM is working correctly.

- a. Change directories to the `/mnt/JDK/repository/prereq.jdk/java/bin` directory on the product CD-ROM. For example:

```
cd /mnt/JDK/repository/prereq.jdk/java/bin
```

- b. Verify the Java 2 SDK version. Type the following command:

```
./java -version
```

The command completes successfully with no errors when the Java 2 SDK is intact.

---

## Preparing Windows systems for installation

This topic describes how to prepare your Windows systems for the installation of IBM WebSphere Application Server, Version 6 products.

The installation uses a InstallShield for Multiplatforms (ISMP) wizard. You can use the graphical interface of the wizard. You can also use the wizard in silent mode.

Silent mode is a command line invocation with a parameter that identifies an options response file. Edit the options response file before installing.

If you encounter a problem such as needing more temporary space or missing prerequisite packages on your operating system, cancel the installation, make the required changes, and restart the installation.

Use the following procedure to prepare the operating system for the installation of WebSphere Application Server products.

1. Log on to a user ID that belongs to the administrator group.  
Log on as a member of the administrator group to successfully install the product. You cannot create Windows services from a user ID that does not belong to the administrator group. The creation of Windows services requires the user to have the advanced user rights *Act as part of the operating system* and *Log on as a service*.
2. **Optional:** Download the latest supported version of Internet Explorer so that you can use the launchpad on the product disc. Download Internet Explorer 6 from the following location:  
<http://www.microsoft.com/windows/ie/default.mspx>
3. Stop all WebSphere Application Server-related Java processes on the machine where you are installing the product.
4. Stop any Web server process such as the IBM HTTP Server.
5. Stop all instances of the `process_spawner.exe` program.
6. Provide adequate disk space.

### Attention:

The Network Deployment product requires the following disk space:

### **730 MB for the C:\Program Files\IBM\WebSphere\AppServer directory before creating profiles**

The installation root directory includes the core product files. This size does not include space for profiles or applications. Profiles require 40 MB of temp space in addition to the sizes shown. Profiles have the following space requirements:

#### **30 MB for the Deployment manager profile**

This size does not include space for Sample applications that you might install. The size also does not include space for applications that you might deploy.

#### **200 MB for an Application Server profile with the Sample applications**

This size does not include space for applications that you might develop and install.

#### **10 MB for an unfederated custom profile**

This size does not include space for applications that you might develop and install. The requirement does include space for the node agent. However, you must federate a custom profile to create an operational managed node.

After federating a custom profile, the resulting managed node contains a functional node agent only. Use the deployment manager to create server processes on the managed node.

#### **100 MB for the C:\temp directory**

The temporary directory is the working directory for the installation program.

#### **830 MB total requirement**

This amount is the total space requirement when installing the product from the CD and when not installing service. Installing profiles requires more space.

The following space is required for the IBM HTTP Server product:

#### **110 MB for the C:\Program Files\IBM HTTP Server directory**

The IBM HTTP Server product requires this space.

#### **25 MB for the C:\Program Files\ibm\gsk7 directory**

The Tivoli Global Security Kit requires this space.

The following space is the maximum amount that is required for the Web server plug-ins for WebSphere Application Server:

#### **200 MB for the C:\Program Files\IBM\WebSphere\Plugins directory**

The Web server plug-ins require this disk space.

#### **25 MB for the C:\Program Files\ibm\gsk7 directory**

The Tivoli Global Security Kit requires this space.

The following space is required for the Application Client for WebSphere Application Server:

#### **150 MB for the C:\Program Files\IBM\WebSphere\AppClient directory**

The amount of space required to install the application clients is actually less than 150 MB. The amount of space depends on the clients that you install as features.

#### **Attention:**

The following space is required for the Application Server Toolkit:

## 550 MB maximum for the C:\Program Files\IBM\WebSphere\AST directory

The Application Server Toolkit does not include an integrated test environment.

Install the WebSphere Application Server Network Deployment product and configure an application server to create a test environment.

The Installation wizard for each component displays required space on the confirmation panel before you install the product files and selected features. The Installation wizard also warns you if you do not have enough space to install the product.

If you plan to migrate applications and the configuration from a previous version, verify that the application objects have enough disk space. As a rough guideline, plan for space equal to 110 percent of the size of the application objects:

- **For Version 4.0.x:** The size of enterprise archive (EAR) files
- **For Version 5.0.x:** The size of EAR files

7. Verify that prerequisites and corequisites are at the required release levels.

Although the Installation wizard checks for prerequisite operating system patches with the prereqChecker application, review the prerequisites on the IBM WebSphere Application Server supported hardware, software, and APIs Web site if you have not already done so.

Refer to the documentation for non-IBM prerequisite and corequisite products to learn how to migrate to their supported versions.

8.  Download Windows Script 5.6 to create Start menu items correctly on Windows 2000.

Download Windows Script 5.6 from the following Microsoft Web page: <http://www.microsoft.com/downloads/details.aspx?FamilyId=C717D943-7E4B-4622-86EB-95A22B832CAA>.

9. Verify that the Java 2 SDK on your copy of the product CD is functioning correctly.

If you created your own product CD from an ISO image or by copying the actual CD, perform the following steps to verify that the Java 2 SDK on the product CD-ROM is working correctly.

- a. Change directories to the \win\WAS\jdk\java\bin directory on the product CD-ROM. Type the following command:

```
cd CD_Drive_D:\WAS\jdk\java\bin
```

- b. Verify the version of the Java 2 SDK. Type the following command:

```
.\java -version
```

The command completes successfully with no errors when the Java 2 SDK is intact.

This procedure results in preparing the operating system for installing the product.

After preparing the operating system for installation, you can install the WebSphere Application Server product.

Go to “Preparing to install Network Deployment on a Windows system” on page 103 for the next step in the overall procedure, which is selecting the type of installation to perform.

---


## Preparing to install Network Deployment on an AIX system

This topic describes how to get started installing the IBM WebSphere Application Server Network Deployment, Version 6 product on an AIX operating system.

Before using this topic to prepare for installing the Network Deployment product, decide the sort of topology you want to install and prepare your operating system. See the following topics for more information:

- “Planning to install Network Deployment” on page 54
- “Preparing AIX systems for installation” on page 81

Log on as root to successfully install the product. You cannot install the product correctly from a non-root user.

 Manually verify that the required space for creating a profile is available on AIX. A known problem in the underlying InstallShield for Multiplatforms (ISMP) code prevents proper space checking on AIX systems at the time that the product disc was created.

After verifying prerequisites, verifying the product CD, and setting your installation goals, you can start installing. Use the following procedure to link to the type of installation procedure that you require. Then install the Network Deployment product on your operating system.

- Perform a custom installation.  
The custom installation lets you select features to install. The features include the core product files (required), the Javadoc (optional), and the Sample applications (optional).  
See “Installing Network Deployment” on page 111.
- Perform a silent installation.  
A silent installation requires you to edit the example response file to contain all of your installation choices. After creating a valid response file, issue the **install** command with the silent parameter from a command window.  
See “Installing silently” on page 120.
- Install additional features to an existing product.  
After using a custom installation to omit one or both of the optional features, you can install the product again to add the features. Select a missing feature to install it.  
See “Installing additional features on an existing Network Deployment installation” on page 116.

The Installation wizard installs the core product files and selected features into the installation root directory.

However, you must create a profile that defines a deployment manager, a managed node, or a stand-alone application server to have an operational environment.

Create a deployment manager profile!

Use the Profile creation wizard to create profiles in the default *install\_root/profiles* directory or in a directory that you select.

See “Using the Profile creation wizard” on page 234.

Deploy an application to get started!

After creating an application server profile, you can deploy an application. See “Fast paths for WebSphere Application Server” on page 9 to get started deploying applications.

---

## Preparing to install Network Deployment on an HP-UX system

This topic describes how to install the IBM WebSphere Application Server Network Deployment, Version 6 product on an HP-UX operating system.

Before using this topic to prepare for installing the Network Deployment product, decide the sort of topology you want to install and prepare your operating system. See the following topics for more information:

- “Planning to install Network Deployment” on page 54
- “Preparing HP-UX systems for installation” on page 85

Log on as root to successfully install the product. You cannot install the product correctly from a non-root user.

After verifying prerequisites, verifying the product CD, and setting your installation goals, you can start installing. Use the following procedure to link to the type of installation procedure that you require. Then install the Network Deployment product on your operating system.

- Perform a custom installation.

The custom installation lets you select features to install. The features include the core product files (required), the Javadoc (optional), and the Sample applications (optional).

See “Installing Network Deployment” on page 111.

- Perform a silent installation.

A silent installation requires you to edit the example response file to contain all of your installation choices. After creating a valid response file, issue the **install** command with the silent parameter from a command window.

See “Installing silently” on page 120.

- Install additional features to an existing product.

After using a custom installation to omit one or both of the optional features, you can install the product again to add the features. Select a missing feature to install it.

See “Installing additional features on an existing Network Deployment installation” on page 116.

The Installation wizard installs the core product files and selected features into the installation root directory.

However, you must create a profile that defines a deployment manager, a managed node, or a stand-alone application server to have an operational environment.

Create a deployment manager profile!

Use the Profile creation wizard to create profiles in the default *install\_root/profiles* directory or in a directory that you select.

See “Using the Profile creation wizard” on page 234.

Deploy an application to get started!

After creating an application server profile, you can deploy an application. See “Fast paths for WebSphere Application Server” on page 9 to get started deploying applications.

---

## Preparing to install Network Deployment on a Linux system

This topic describes how to install IBM WebSphere Application Server Network Deployment, Version 6 on a Linux platform.

Before using this topic to prepare for installing the Network Deployment product, decide the sort of topology you want to install and prepare your operating system. See the following topics for more information:

- “Planning to install Network Deployment” on page 54
- “Preparing Linux systems for installation” on page 90

Log on as root to successfully install the product. You cannot install the product correctly from a non-root user.

After verifying prerequisites, verifying the product CD, and setting your installation goals, you can start installing. Use the following procedure to link to the type of installation procedure that you require. Then install the Network Deployment product on your operating system.

- Perform a custom installation.

The custom installation lets you select features to install. The features include the core product files (required), the Javadoc (optional), and the Sample applications (optional).

See “Installing Network Deployment” on page 111.

- Perform a silent installation.

A silent installation requires you to edit the example response file to contain all of your installation choices. After creating a valid response file, issue the **install** command with the silent parameter from a command window.

See “Installing silently” on page 120.

- Install additional features to an existing product.

After using a custom installation to omit one or both of the optional features, you can install the product again to add the features. Select a missing feature to install it.

See “Installing additional features on an existing Network Deployment installation” on page 116.

The Installation wizard installs the core product files and selected features into the installation root directory.

However, you must create a profile that defines a deployment manager, a managed node, or a stand-alone application server to have an operational environment.

Create a deployment manager profile!

Use the Profile creation wizard to create profiles in the default *install\_root/profiles* directory or in a directory that you select.

See “Using the Profile creation wizard” on page 234.

Deploy an application to get started!

After creating an application server profile, you can deploy an application. See “Fast paths for WebSphere Application Server” on page 9 to get started deploying applications.

---

## Preparing to install Network Deployment on a Solaris system

This topic describes how to install the IBM WebSphere Application Server Network Deployment, Version 6 product on a Solaris platform.

Before using this topic to prepare for installing the Network Deployment product, decide the sort of topology you want to install and prepare your operating system. See the following topics for more information:

- “Planning to install Network Deployment” on page 54
- “Preparing AIX systems for installation” on page 81

Log on as root to successfully install the product. You cannot install the product correctly from a non-root user.

After verifying prerequisites, verifying the product CD, and setting your installation goals, you can start installing. Use the following procedure to link to the type of installation procedure that you require. Then install the Network Deployment product on your operating system.

- Perform a custom installation.

The custom installation lets you select features to install. The features include the core product files (required), the Javadoc (optional), and the Sample applications (optional).

See “Installing Network Deployment” on page 111.

- Perform a silent installation.

A silent installation requires you to edit the example response file to contain all of your installation choices. After creating a valid response file, issue the **install** command with the silent parameter from a command window.

See “Installing silently” on page 120.

- Install additional features to an existing product.

After using a custom installation to omit one or both of the optional features, you can install the product again to add the features. Select a missing feature to install it.

See “Installing additional features on an existing Network Deployment installation” on page 116.

The Installation wizard installs the core product files and selected features into the installation root directory.

However, you must create a profile that defines a deployment manager, a managed node, or a stand-alone application server to have an operational environment.

Create a deployment manager profile!

Use the Profile creation wizard to create profiles in the default *install\_root/profiles* directory or in a directory that you select.

See “Using the Profile creation wizard” on page 234.

Deploy an application to get started!

After creating an application server profile, you can deploy an application. See “Fast paths for WebSphere Application Server” on page 9 to get started deploying applications.

---

## Preparing to install Network Deployment on a Windows system

This topic describes how to install the IBM WebSphere Application Server Network Deployment, Version 6 product on a Windows operating system.

Before using this topic to prepare for installing the Network Deployment product, decide the sort of topology you want to install and prepare your operating system. See the following topics for more information:

- “Planning to install Network Deployment” on page 54
- “Preparing Windows systems for installation” on page 97

Log on as a member of the administrator group to successfully install the product. Creating a Windows service for the server1 process requires you to have the following advanced user right: *Act as part of the operating system*.

During the procedure, you can assign another ID or the one you are using to install as the user who will log on the Windows service. That user requires the advanced user right: *Log on as a service*.

However, if you do not have this advanced user right or if the user ID that is to log on the Windows service does not have the advanced user right, the Installation wizard assigns the advanced right to the user.



After verifying prerequisites, verifying the product CD, and setting your installation goals, you can start installing. Use the following procedure to link to the type of installation procedure that you require. Then install the Network Deployment product on your operating system.

- Perform a custom installation.

The custom installation lets you select features to install. The features include the core product files (required), the Javadoc (optional), and the Sample applications (optional).

See “Installing Network Deployment” on page 111.

- Perform a silent installation.

A silent installation requires you to edit the example response file to contain all of your installation choices. After creating a valid response file, issue the **install** command with the silent parameter from a command window.

See “Installing silently” on page 120.

- Install additional features to an existing product.

After using a custom installation to omit one or both of the optional features, you can install the product again to add the features. Select a missing feature to install it.

See “Installing additional features on an existing Network Deployment installation” on page 116.

The Installation wizard installs the core product files and selected features into the installation root directory.

However, you must create a profile that defines a deployment manager, a managed node, or a stand-alone application server to have an operational environment.

Create a deployment manager profile!

Use the Profile creation wizard to create profiles in the default *install\_root/profiles* directory or in a directory that you select.

See “Using the Profile creation wizard” on page 234.

Deploy an application to get started!

After creating an application server profile, you can deploy an application. See “Fast paths for WebSphere Application Server” on page 9 to get started deploying applications.



---

## Part 3. Installing



---

## Chapter 9. Installing the product and additional software

This topic describes how to install WebSphere Application Server Network Deployment as the root user on a Linux platform or on a UNIX platform, or from a user ID that belongs to the administrator group on a Windows platform. This topic also describes installing IBM HTTP Server and the application clients.

Read through this topic and its related topics to prepare for installation and to make yourself familiar with installation options, before you start to use the installation tools. Review the hardware and software requirements on the IBM WebSphere Application Server supported hardware and software Web site to get started.

The launchpad is the preferred method of starting the installation. Use the Launchpad to access the product overview, the installation guide, and the readme file. Of course, the main purpose of the launchpad is to launch the installation procedure.

See “Using the launchpad to start the installation” on page 41 for more information.

If you encounter a problem such as needing more disk space or temporary space or missing prerequisite packages on your Linux or UNIX system, cancel the installation, make the required changes, and restart the installation.

**Windows** Log on as a member of the administrator group to successfully install the product. You cannot install the product correctly from a user ID that does not belong to the administrator group. Some steps of the installation procedure require the administrator group ID to have the following advanced user rights, *Act as part of the operating system* and *Log on as a service*.

This topic provides an overview of installation procedures. Use the information in this topic to learn about the types of installation available. This topic has links to more detailed installation topics.

The Installation wizard performs the following actions:

- Automatically checks prerequisites
- Looks for a previous WebSphere Application Server V6 installation, to determine whether to let you add features to the product binaries or to install a new set of product binaries
- Looks for a previous V6 installation, to determine whether to let you upgrade from a trial installation to the real product
- Prompts you to start the Profile creation wizard, where you can create a deployment manager profile, an application server profile, or a custom profile

Perform the following procedure to start your installation.

1. Plan your installation, as described in “Planning to install Network Deployment” on page 54.
2. Prepare your operating platform for installation, as described in Chapter 8, “Preparing the operating system for product installation,” on page 81.
3. Review the roadmap for installing the Network Deployment product, as described in “Roadmap for installing the Network Deployment product” on page 108.
4. Install the Network Deployment product.

Choose an installation scenario to continue the installation:

- Perform the installation with the wizard, as described in “Installing Network Deployment” on page 111.

The installation lets you select features to install.

- Perform a silent installation, as described in “Installing silently” on page 120.

A silent installation requires you to edit the response file that contains all of your installation choices. After creating a valid response file, you issue the **install** command with the silent parameter from a command window.

- Install additional features to an existing installation, as described in “Installing additional features on an existing Network Deployment installation” on page 116.

After installing the product, you can install it again to update the system files by installing previously uninstalled features.

The installer program does not support the ISMP -console installation method at this time.

5. **Optional:** Install IBM HTTP Server.

After installing a WebSphere Application Server product, you can use the application server to serve applications over the Internet. It is not necessary to install the IBM HTTP Server or another Web server. It is also necessary to install Web server plug-ins. You also do not need the application clients unless the application you are deploying was designed to run as a client application. You also do not have to install an integrated development environment. The development environment is for developers of Web applications.

The installation solution diagrams in “Planning to install Network Deployment” on page 54 show the components that are present in different types of environments.

See Installing IBM HTTP Server for more information about installing IBM HTTP Server.

6. **Optional:** Install Web server plug-ins for IBM HTTP Server or another supported Web server.

If you install the IBM HTTP Server or another Web server, you must install a binary module for the Web server to enable it to communicate with WebSphere Application Server products. Before you can install a Web server plug-in, you must install the Web server.

After installing a Web server plug-in, the Plug-ins installation wizard configures a new element in the application server configuration called a *Web server definition*. You can then manage applications for the Web server using the administrative console.

See Installing Web server plug-ins for more information.

7. **Optional:** Install the application clients.

See Installing Application Client for WebSphere Application Server for more information.

This procedure results in installing WebSphere Application Server Network Deployment, Version 6.

You can also install IBM HTTP Server, the Web server plug-ins for WebSphere Application Server, and the application clients for WebSphere Application Server.

---

## Roadmap for installing the Network Deployment product

This topic provides an overview of how to install IBM WebSphere Application Server Network Deployment, Version 6.

See Chapter 9, “Installing the product and additional software,” on page 107 for information about installing any of the products in the WebSphere Application Server family of products. This topic describes installing the Network Deployment product.

## Using the launchpad

The launchpad gives you important clues about creating typical scenarios. The installation solution diagrams identify some of the system topologies that you can create. The launchpad is in the root directory of the product CD.

Use the launchpad to launch the installation procedure for each of the installable components in the primary disc packet. See “Using the launchpad to start the installation” on page 41. The launchpad also provides access to the documentation for each of the installable components on the product CD.

Start the launchpad with the **launchpad** command:

-   ./launchpad.sh
-  launchpad.bat

The launchpad requires a supported browser. Supported browsers are:

- Internet Explorer, Version 5.5 with SP 2 or later
- Mozilla browser, Version 1.4 or later




## Roadmaps for installing the Network Deployment product

At a high level view, the steps for installing and configuring the Network Deployment product are:

1. Install the Network Deployment product
2. Use the Profile creation wizard to create one of the following profiles:
  - Deployment manager profile
  - Application Server profile
  - Custom profile

### Roadmap for installing a deployment manager

The steps for installing the Network Deployment product and creating a deployment manager are:

1. Insert the product disc.
2. Start the launchpad.
3. Launch the installation program for the WebSphere Application Server Network Deployment product.
4. Install the Network Deployment product
5. Use the Profile creation wizard to create a deployment manager profile.
6. Start the deployment manager with the **startManager** command.
  -   ./install\_root/profiles/profile\_name/bin/startManager.sh
  -  install\_root\profiles\profile\_name\bin\startManager.bat

### Roadmap for installing a stand-alone application server

The steps for installing the Network Deployment product and creating a stand-alone application server are:

1. Insert the product disc.
2. Start the launchpad.
3. Launch the installation program for the WebSphere Application Server Network Deployment product.
4. Install the Network Deployment product
5. Use the Profile creation wizard to create an application server profile.
6. Start the application server with the **startServer** command. If you have a default profile that is not the stand-alone application server profile, address the **startServer** command with the **-profileName** parameter or use the **startServer** command in the bin directory of the *install\_root/profile\_root/bin* directory.

See the description of the startServer command for more information.

### Roadmap for installing a managed node

The steps for installing the Network Deployment product and creating a managed node are:

1. Insert the product disc.
2. Start the launchpad.
3. Launch the installation program for the WebSphere Application Server Network Deployment product.
4. Install the Network Deployment product

5. Use the Profile creation tool to create a deployment manager profile.
6. Start the deployment manager with the **startManager** command. If you have a default profile that is not the deployment manager profile, you must direct the **startManager** command to address the correct profile with the `-profileName` parameter.  
See the description of the `startManager` command for more information.
7. Use the Profile creation wizard to create an application server profile.
8. Start the application server with the **startServer** command. If you have a default profile that is not the stand-alone application server profile, address the **startServer** command with the `-profileName` parameter or use the **startServer** command in the `bin` directory of the `install_root/profile_root/bin` directory.
9. Use the administrative console of the deployment manager to add the application server node to the deployment manager cell. Click **System administration > Nodes > Add node** to identify the application server node.  
You can also use the **addNode** command. Use the **addNode** command that is in the application server profile `bin` directory.  
See the description of the `addNode` command for more information.

## Roadmap for creating multiple application servers on a single machine

After installing the Network Deployment product, you can use the Profile creation wizard to create the following profiles:

- Deployment manager profile
- Application server profile
- Custom profile

You do not have to install the Network Deployment product more than once on the same machine to have multiple profiles. Install the product once and use the Profile creation wizard to create multiple profiles.

For example, the following procedure shows the basic steps for creating two stand-alone application server profiles on one machine:

1. Insert and mount the product disc.
2. Start the launchpad.
3. Launch the installation program for the Network Deployment product.
4. Launch the Profile creation wizard from the Installation wizard.
5. Create an application server profile.  
Give the profile a unique name, such as `Profile01`.
6. Launch the First steps console from the Profile creation wizard.
7. Select the Profile creation wizard option and create a new application server profile.  
Give the new profile a unique name, such as `Profile02`.
8. Change directories to the `install_root/profiles/Profile01/bin` directory:
  -   `./cd /opt/IBM/WebSphere/AppServer/profiles/Profile01/bin`
  -  `cd C:\IBM\WebSphere\AppServer\profiles\Profile01\bin`
9. Start the application server named `server1` with the **startServer** command:
  -   `./startServer.sh server1`
  -  `startServer.bat server1`
10. Change directories to the `install_root/profiles/Profile02/bin` directory:
  -   `./cd /opt/IBM/WebSphere/AppServer/profiles/Profile02/bin`

-  `cd C:\IBM\WebSphere\AppServer\profiles\Profile02\bin`
11. Start the application server named server1 with the **startServer** command:
-   `./startServer.sh server1`
  -  `startServer.bat server1`

## Understanding what you can do with the Network Deployment product

Select the *Installation solutions diagrams* link in the navigation area on the launchpad to view typical topologies for the Network Deployment product. The solutions are shadowed in “Planning to install Network Deployment” on page 54. The topic describes the installation steps for each solution are described in detail.

## WebSphere Application Server uses ISMP to install products

The installation program uses the InstallShield for Multiplatforms (ISMP) product to perform the installation. You can use the Installation wizard or perform the installation from a command line, using the silent installation method.

## What to do when an installation problem occurs

If you encounter a problem such as not having enough space or not having the right prerequisite updates on your system, cancel the installation, make the required changes, and restart the installation. The installation procedure has a description of more comprehensive troubleshooting procedures.

## Getting the latest information

This topic is available on the product CD-ROM and online in an information center. The information center always has the most current information. The information center displays in the language of your machine locale if possible.

The information center for Version 6 is at the <http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp> IBM Web site.

The *Getting Started* PDF is updated occasionally. Download a new *Getting Started* guide from [ftp://ftp.software.ibm.com/software/webserver/appserv/library/wasv60nd\\_gs.pdf](ftp://ftp.software.ibm.com/software/webserver/appserv/library/wasv60nd_gs.pdf).

The HTML version of the *Getting Started* guide is not available for downloading because you can download and install the entire information center for the base product. See *Using this documentation offline* for more information.

## What to do next

After installing the Network Deployment product, create a deployment manager cell. See “Planning to install Network Deployment” on page 54 for a scenario that describes creating a cell.

After creating the cell, use the administrative console of the deployment manager to deploy an existing application. See *Fast paths for WebSphere Application Server* for more information.

---

## Installing Network Deployment

This topic describes how to use the custom installation path to install IBM WebSphere Application Server Network Deployment, Version 6.

This path is one of the installation paths that is available for any operating system. Two other installation paths exist:

- “Installing silently” on page 120

- “Installing additional features on an existing Network Deployment installation” on page 116

Select the appropriate installation path before you begin the installation.

Before starting the installation, see “Using the launchpad to start the installation” on page 41 for information about the *Installation solutions diagrams*. The diagrams show typical topologies for the product. The solutions are also in “Planning to install Network Deployment” on page 54.

Use the following procedure to install the WebSphere Application Server Network Deployment product.

1. Log on as root on a Linux or UNIX operating system, or as a member of the administrator group on a Windows system.



In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```



Some steps of the installation on a Windows platform require the administrator group user to have the following advanced user rights:

- *Act as part of the operating system*
- *Log on as a service*

For example, on some Windows systems, click **Administrative Tools > Local Security Policy > User Rights Assignments** to see the advanced options. See your Windows documentation for more information.



The installation wizard grants your Windows user ID the advanced user rights, if the user ID belongs to the administrator group. The silent installation does not grant these rights. If you create a new user ID on a Windows platform to perform a silent installation, you must restart the system to activate the proper authorizations for the user ID before you can perform a successful silent installation.



When installing the WebSphere Application Server as a Windows service, do not use a user ID that contains spaces. A user ID with spaces cannot be validated. Such a user ID is not allowed to continue the installation. To work around this problem, install with a user ID that does not contain spaces, or do not choose to install Windows services.

2. Insert the product CD-ROM labeled **WebSphere Application Server Network Deployment** into the CD-ROM drive and mount the drive if necessary, as described in “Mounting CD-ROMS on Linux and UNIX operating systems” on page 133.
3. Start the installation with the **launchpad.sh** command on Linux and UNIX platforms or the **launchpad.bat** command on Windows platforms.

You can also start the installation using the **install** command:

-   /WAS/install
-  \WAS\install

If you have a problem starting the launchpad, use the troubleshooting information in “Using the launchpad to start the installation” on page 41 to correct the problem.

When using the launchpad, launch the Installation wizard for Network Deployment.

After launching the Installation wizard from the launchpad or from the command line, the InstallShield MultiPlatform (ISMP) wizard initializes and then displays the Welcome panel.

4. Click **Next** to continue.

The license agreement displays.



5. Click the radio button beside the **I accept the terms in the license agreement** message to agree to the license agreement and click **Next** to continue.

After you accept the licensing terms, the Installation wizard checks for a supported operating system and prerequisite patches.

If you encounter a problem such as not having the right prerequisite updates on your system, cancel the installation, make the required changes, and restart the installation.

Although the Installation wizard checks for prerequisite operating system patches with the prereqChecker application, review the prerequisites on the IBM WebSphere Application Server supported hardware, software, and APIs Web site if you have not already done so.

The Web site lists all supported operating systems and the operating system fixes and patches that you must install to have a compliant operating system.

Refer to the documentation for non-IBM prerequisite and corequisite products to learn how to migrate to their supported versions.

6. After confirming that your operating system is supported and that you have installed all necessary patches, click **Next**.

The Installation wizard checks for a previous installation at the same product level.








If the wizard detects a previous installation, it displays the Existing installation panel. This procedure assumes that you do not have an existing installation that you intend to update. See “Installing additional features on an existing Network Deployment installation” on page 116 for more information about that scenario.

The wizard displays a panel that prompts you for the location of the installation root directory for the core product files.

7. Specify the destination of the installation root directory and click **Next**.

Specify the location of the installation root directory for the product binaries, which are also known as the system files. The system files do not change unless you add features; install service such as refresh packs, fix packs, or interim fixes; or install a product that extends the Network Deployment product.

The following list shows default installation locations for supported platforms:

-  /usr/IBM/WebSphere/AppServer
-    /opt/IBM/WebSphere/AppServer
-  C:\Program Files\IBM\WebSphere\AppServer
-  


**Important:** Do not use symbolic links as the destination directory. Symbolic links are not supported.

  Spaces are not supported in the name of the installation directory.

Deleting the default target location and leaving an installation directory field empty prevents you from continuing.

Non-ASCII special characters are not supported in the name of the installation directory.

The installer program checks for required space at the beginning of the installation. If you do not have enough space, stop the installation program, free space by deleting unused files and emptying the recycle bin, and restart the installation.

 The installation directory path must be no longer than 60 characters to successfully create the Application Server profile on a Windows 2000 system.

The wizard displays the Features selection panel.

8. Select features to install and click **Next**.

Table 6. Features for the Network Deployment product

Feature	Description
Core product files	Installs the product binary code.
Application Server Samples	Installs the Sample applications.
Javadoc	Installs the API documentation of WebSphere Application Server classes.

**Performance considerations**

For better performance in either a development or production environment, do not install the Samples. By omitting the Samples, you can improve application server startup time by 60 percent and save 15 percent of disk space. You can save up to 30 percent of process footprint (based on a maximum heap size of 256 MB).

The Installation wizard displays a confirmation panel.

- Review the summary information and click **Next** to install the product code or **Back** to change your specifications.

The Installation wizard creates the uninstaller program and displays a progress panel that shows which components are being installed.

At the end of the installation, the wizard displays the Installation completion panel.

- Verify the success of the installer program by examining the completion panel and the *install\_root/logs/log.txt* file for installation status.

If problems occur, consult the following applicable logs:

**Log files for WebSphere Application Server products:** The following table shows the installation logs, content, and indicators of success and failure for WebSphere Application Server products for Version 6.

Table 7. Installation logs for WebSphere Application Server products

Log	Content	Indicators
<i>install_root / logs / log.txt</i>	Logs all installation events	<p><b>INSTCONFFAIL</b> Total installation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful installation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Installation errors occurred but the installation is still usable. Additional information identifies the errors.</p>
<i>install_root / logs / wasprofile / wasprofile_create_profile_name.log</i>	<ul style="list-style-type: none"> <li>Traces all events that occur during the creation of the named profile</li> <li>Created when using the Profile creation wizard or the <b>wasprofile</b> command</li> </ul>	<p><b>INSTCONFFAIL</b> Total profile creation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile creation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile creation errors occurred but the profile is still functional. Additional information identifies the errors.</p>

Table 7. Installation logs for WebSphere Application Server products (continued)


Log	Content	Indicators
<i>install_root</i> / logs / wasprofile / wasprofile _ delete _ profile_name.log	<ul style="list-style-type: none"> <li>Traces all events that occur during the deletion of the named profile</li> <li>Created when using the Profile creation wizard or the <b>wasprofile</b> command</li> </ul>	<p><b>INSTCONFFAIL</b> Total profile deletion failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile deletion.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile deletion errors occurred but the profile is still deleted. Additional information identifies the errors.</p>
<i>install_root</i> / profiles / profile_name / logs / pctLog.txt	Logs all profile creation events that occur when using the Profile creation wizard	<p><b>INSTCONFFAIL</b> Total profile creation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile creation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile creation errors occurred but the profile is still functional. Additional information identifies the errors.</p>



If the installation of the core product files fails, fix the error and reinstall.

If the creation of a profile fails, fix the error and run the Profile creation tool to recreate the profile.

Troubleshoot the installation if the *install\_root*/logs/log.txt file or the *install\_root*/logs/wasprofile/wasprofile\_create\_default.log file does not contain a record of any problems, but problems exist.

See Appendix B, “Troubleshooting installation,” on page 337 and “Installation component troubleshooting tips” on page 355 for more information.

 Examine the Start menu entry. The menu is an indication of a successful installation:

-  **Programs > IBM WebSphere > Application Server Network Deployment v6**
-  **All Programs > IBM WebSphere > Application Server Network Deployment v6**

If problems exist that cause you to reinstall the product, correct the errors, uninstall the product as described in Appendix A, “Uninstalling the product,” on page 317, reboot a Windows machine or log off and back on as root on a Linux or UNIX machine, and reinstall.

- Select the check box to launch the Profile creation wizard and click **Next** to close the Installation wizard and launch the Profile creation wizard.

See “Using the Profile creation wizard” on page 234 for information about creating an operational environment that includes a deployment manager, a managed node, or a stand-alone application server.

Network Deployment V6 includes the ability to create a deployment manager profile, an application server profile, or a custom profile. The second step of installation is to use the Profile creation wizard to create an operational environment that includes one of these profiles.

The Installation wizard installs the core product files and selected features into the installation root directory.

However, you must create a profile that defines a deployment manager, a managed node, or a stand-alone application server to have an operational environment.

Create a deployment manager profile!

Use the Profile creation wizard to create profiles in the default *install\_root/profiles* directory or in a directory that you select.

See “Using the Profile creation wizard” on page 234.

Deploy an application to get started!

After creating an application server profile, you can deploy an application. See “Fast paths for WebSphere Application Server” on page 9 to get started deploying applications.

---

## Installing additional features on an existing Network Deployment installation

This topic describes how to install additional features on an existing IBM WebSphere Application Server Network Deployment, Version 6 installation.

This path is one of the installation paths that is available for any operating system. Two other installation paths exist:

- “Installing Network Deployment” on page 111
- “Installing silently” on page 120

Select the appropriate installation path before you begin the installation.

Before starting the installation, see “Using the launchpad to start the installation” on page 41 for information about the *Installation solutions diagrams*. The diagrams show typical topologies for the product. The solutions are also in “Planning to install Network Deployment” on page 54.

Use the following procedure to add features to an existing WebSphere Application Server Network Deployment installation.

1. Log on as root on a Linux or UNIX operating system, or as a member of the administrator group on a Windows system.



In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```



Some steps of the installation on a Windows platform require the administrator group user to have the following advanced user rights:

- *Act as part of the operating system*
- *Log on as a service*

For example, on some Windows systems, click **Administrative Tools > Local Security Policy > User Rights Assignments** to see the advanced options. See your Windows documentation for more information.



The installation wizard grants your Windows user ID the advanced user rights, if the user ID belongs to the administrator group. The silent installation does not grant these rights. If you create a new user ID on a Windows platform to perform a silent installation, you must restart the system to activate the proper authorizations for the user ID before you can perform a successful silent installation.



When installing the WebSphere Application Server as a Windows service, do not use a user ID that contains spaces. A user ID with spaces cannot be validated. Such a user ID is not

allowed to continue the installation. To work around this problem, install with a user ID that does not contain spaces, or do not choose to install Windows services.

2. Insert the product CD-ROM labeled **WebSphere Application Server Network Deployment** into the CD-ROM drive and mount the drive if necessary, as described in “Mounting CD-ROMS on Linux and UNIX operating systems” on page 133.
3. Start the installation with the **launchpad.sh** command on Linux and UNIX platforms or the **launchpad.bat** command on Windows platforms.

You can also start the installation using the **install** command:

-   /WAS/install
-  \WAS\install

If you have a problem starting the launchpad, use the troubleshooting information in “Using the launchpad to start the installation” on page 41 to correct the problem.

When using the launchpad, launch the Installation wizard for Network Deployment.

After launching the Installation wizard from the launchpad or from the command line, the InstallShield MultiPlatform (ISMP) wizard initializes and then displays the Welcome panel.

4. Click **Next** to continue.

The license agreement displays.

5. Click the radio button beside the **I accept the terms in the license agreement** message to agree to the license agreement and click **Next** to continue.

After you accept the licensing terms, the Installation wizard checks for a supported operating system and prerequisite patches.

If you encounter a problem such as not having the right prerequisite updates on your system, cancel the installation, make the required changes, and restart the installation.

Although the Installation wizard checks for prerequisite operating system patches with the prereqChecker application, review the prerequisites on the IBM WebSphere Application Server supported hardware, software, and APIs Web site if you have not already done so.

The Web site lists all supported operating systems and the operating system fixes and patches that you must install to have a compliant operating system.

Refer to the documentation for non-IBM prerequisite and corequisite products to learn how to migrate to their supported versions.

6. After confirming that your operating system is supported and that you have installed all necessary patches, click **Next** to continue.

The Installation wizard checks for a previous installation at the same product level. This task describes the scenario where a previous installation exists and you are adding features with an incremental installation.

When the wizard detects a previous installation, it displays the Existing installation panel. The panel lets you add features to the existing installation, or perform a new installation to another directory.

7. Select the check box for installing additional features and click **Next**.

Do not change the installation directory unless another existing installation is in the directory that you specify. An incremental installation adds features to the shared binaries identified in the field.

You can add features to the core product files at any time by running the Installation wizard again. Installing additional features does not affect profiles that you might have already created.

When adding features during an incremental installation, the wizard displays the Features selection panel next.

8. Select features to install and click **Next**.

Table 8. Features for the Network Deployment product

Feature	Description
<b>Core product files</b>	Installs the product binary code.
<b>Application Server Samples</b>	Installs the Sample applications.
<b>Javadoc</b>	Installs the API documentation of WebSphere Application Server classes.

### Performance considerations

For better performance in either a development or production environment, do not install the Samples. By omitting the Samples, you can improve application server startup time by 60 percent and save 15 percent of disk space. You can save up to 30 percent of process footprint (based on a maximum heap size of 256 MB).

The Installation wizard displays a confirmation panel.

- Review the summary information and click **Next** to install the product code or **Back** to change your specifications.

The disk space shown on the pre-installation summary panel includes space for the Java Virtual Machine (JVM). Subtract the size of the JVM for your operating system to determine the installed size on disk for the features that you selected.

Table 9. JVM sizes

AIX	HP-UX	Linux	Solaris	Windows
59 MB	pa-risc: 109 MB	ia32: 83 MB	81 MB	ia32: 75 MB
	64 bit: 137 MB	64 bit: 133 MB		64 bit: 71 MB

Verify that you have the total amount of free space shown before installing. Space is required for a working copy of the JVM during the installation.

The Installation wizard creates the uninstaller program and then displays a progress panel that shows which components are being installed. At the end of the installation, the wizard displays the Installation completion panel.

At the end of the installation, the wizard displays the Installation completion panel.

- Verify the success of the installer program by examining the completion panel and the *install\_root/logs/log.txt* file for installation status.

If problems occur, consult the following applicable logs:

**Log files for WebSphere Application Server products:** The following table shows the installation logs, content, and indicators of success and failure for WebSphere Application Server products for Version 6.

Table 10. Installation logs for WebSphere Application Server products

Log	Content	Indicators
<i>install_root</i> / logs / log.txt	Logs all installation events	<p><b>INSTCONFFAIL</b> Total installation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful installation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Installation errors occurred but the installation is still usable. Additional information identifies the errors.</p>

Table 10. Installation logs for WebSphere Application Server products (continued)


Log	Content	Indicators
<i>install_root</i> / logs / wasprofile / wasprofile _ create _ profile_name.log	<ul style="list-style-type: none"> <li>Traces all events that occur during the creation of the named profile</li> <li>Created when using the Profile creation wizard or the <b>wasprofile</b> command</li> </ul>	<p><b>INSTCONFFAIL</b> Total profile creation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile creation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile creation errors occurred but the profile is still functional. Additional information identifies the errors.</p>
<i>install_root</i> / logs / wasprofile / wasprofile _ delete _ profile_name.log	<ul style="list-style-type: none"> <li>Traces all events that occur during the deletion of the named profile</li> <li>Created when using the Profile creation wizard or the <b>wasprofile</b> command</li> </ul>	<p><b>INSTCONFFAIL</b> Total profile deletion failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile deletion.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile deletion errors occurred but the profile is still deleted. Additional information identifies the errors.</p>
<i>install_root</i> / profiles / profile_name / logs / pctLog.txt	Logs all profile creation events that occur when using the Profile creation wizard	<p><b>INSTCONFFAIL</b> Total profile creation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile creation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile creation errors occurred but the profile is still functional. Additional information identifies the errors.</p>



If the installation of the core product files fails, fix the error and reinstall.

If the creation of a profile fails, fix the error and run the Profile creation tool to recreate the profile.

Troubleshoot the installation if the *install\_root*/logs/log.txt file or the *install\_root*/logs/wasprofile/wasprofile\_create\_default.log file does not contain a record of any problems, but problems exist.

See Appendix B, “Troubleshooting installation,” on page 337 and “Installation component troubleshooting tips” on page 355 for more information.

 Examine the Start menu entry. The menu is an indication of a successful installation:

-  **Programs > IBM WebSphere > Application Server Network Deployment v6**
-  **All Programs > IBM WebSphere > Application Server Network Deployment v6**

If problems exist that cause you to reinstall the product, correct the errors, uninstall the product as described in Appendix A, “Uninstalling the product,” on page 317, reboot a Windows machine or log off and back on as root on a Linux or UNIX machine, and reinstall.

- Select the check box to launch the Profile creation wizard and click **Next** to close the Installation wizard and launch the Profile creation wizard.

See “Using the Profile creation wizard” on page 234 for information about creating an operational environment that includes a deployment manager, a managed node, or a stand-alone application server.

Network Deployment V6 includes the ability to create a deployment manager profile, an application server profile, or a custom profile. The second step of installation is to use the Profile creation wizard to create an operational environment that includes one of these profiles.



The Installation wizard installs the new features in the existing installation root directory. The Installation wizard configures the WebSphere Application Server Network Deployment product for the new features.

However, you must create a profile that defines a deployment manager, a managed node, or a stand-alone application server to have an operational environment.

Create a deployment manager profile!

Use the Profile creation wizard to create profiles in the default `install_root/profiles` directory or in a directory that you select.

See “Using the Profile creation wizard” on page 234.

Deploy an application to get started!

After creating an application server profile, you can deploy an application. See “Fast paths for WebSphere Application Server” on page 9 to get started deploying applications.

---

## Installing silently

This topic describes how to perform a silent installation of the product.

This is one of the installation paths available for any platform. Other installation paths include:

- “Installing Network Deployment” on page 111
- “Installing additional features on an existing Network Deployment installation” on page 116

Select the appropriate installation path before you begin the installation.

Before starting the silent installation, see “Using the launchpad to start the installation” on page 41 for information about the *Installation solution diagrams*. The diagrams show typical topologies for the product.

The solutions are also in “Planning to install Network Deployment” on page 54.

Use this procedure to perform a silent installation of the WebSphere Application Server Network Deployment product.

A silent installation uses the Installation wizard to install the product in silent mode, without the graphical user interface. Instead of displaying a wizard interface, the silent installation causes the installation program to read all of your responses from a file that you provide.

1. Log on as root on a Linux or UNIX operating system, or as a member of the administrator group on a Windows system.



In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```



Some steps of the installation on a Windows platform require the administrator group user to have the following advanced user rights:

- *Act as part of the operating system*
- *Log on as a service*



For example, on some Windows systems, click **Administrative Tools > Local Security Policy > User Rights Assignments** to see the advanced options. See your Windows documentation for more information.

**Windows** The installation wizard grants your Windows user ID the advanced user rights, if the user ID belongs to the administrator group. The silent installation does not grant these rights. If you create a new user ID on a Windows platform to perform a silent installation, you must restart the system to activate the proper authorizations for the user ID before you can perform a successful silent installation.

**Windows** When installing the WebSphere Application Server as a Windows service, do not use a user ID that contains spaces. A user ID with spaces cannot be validated. Such a user ID is not allowed to continue the installation. To work around this problem, install with a user ID that does not contain spaces, or do not choose to install Windows services.

2. Copy the response file as `myoptionsfile` to your disk drive and customize it, as described in “Customizing the options response file for Network Deployment” on page 123.

The name of the original file is “`responsefile.nd.txt`” on page 128.

You must also create a profile for the Network Deployment product to create an operational environment. You can create a deployment manager profile, an application server profile, or a custom profile that becomes a managed node when you add the node into a deployment manager cell.

You can create a profile silently by customizing the profile options response file for the profile that you intend to create. The names of the original profile response files are:

- **Deployment manager profile:** “`responsefile.pct.NDdmgrProfile.txt`” on page 242
- **Application server profile:** “`responsefile.pct.NDstandAloneProfile.txt`” on page 267
- **Custom profile:** “`responsefile.pct.NDmanagedProfile.txt`” on page 254

Do not add an options line to any of the profile creation response files that includes the following parameter:

`-silent`

The `-silent` parameter is not required. If it exists in any of the files, the file cannot create a profile during a silent product installation. If you use the Profile creation wizard in silent mode, the additional parameter does not affect the creation of a profile. However, when the silent installation of the Network Deployment product attempts to call a silent profile response file, the parameter prevents the creation of the profile.

3. Issue the proper command to use your custom response file. For example, issue one of the following commands:

- **Linux** **UNIX** `mnt_cdrom/WAS/install -options /tmp/WAS/myoptionsfile.txt -silent`
- **Windows** `"CD_drive_D:\WAS\install" -options "C:\temp\WAS\myoptionsfile.txt" -silent`

You can find the sample options response file in the WAS directory on the product CD.

The Installation wizard and the Profile creation wizard record installation events in the following log files:

**Log files for WebSphere Application Server products:** The following table shows the installation logs, content, and indicators of success and failure for WebSphere Application Server products for Version 6.

Table 11. Installation logs for WebSphere Application Server products

Log	Content	Indicators
<i>install_root</i> / logs / log.txt	Logs all installation events	<p><b>INSTCONFFAIL</b> Total installation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful installation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Installation errors occurred but the installation is still usable. Additional information identifies the errors.</p>
<i>install_root</i> / logs / wasprofile / wasprofile _ create _ profile_name.log	<ul style="list-style-type: none"> <li>Traces all events that occur during the creation of the named profile</li> <li>Created when using the Profile creation wizard or the <b>wasprofile</b> command</li> </ul>	<p><b>INSTCONFFAIL</b> Total profile creation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile creation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile creation errors occurred but the profile is still functional. Additional information identifies the errors.</p>
<i>install_root</i> / logs / wasprofile / wasprofile _ delete _ profile_name.log	<ul style="list-style-type: none"> <li>Traces all events that occur during the deletion of the named profile</li> <li>Created when using the Profile creation wizard or the <b>wasprofile</b> command</li> </ul>	<p><b>INSTCONFFAIL</b> Total profile deletion failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile deletion.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile deletion errors occurred but the profile is still deleted. Additional information identifies the errors.</p>
<i>install_root</i> / profiles / profile_name / logs / pctLog.txt	Logs all profile creation events that occur when using the Profile creation wizard	<p><b>INSTCONFFAIL</b> Total profile creation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile creation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile creation errors occurred but the profile is still functional. Additional information identifies the errors.</p>

**Log more information when InstallShield for MultiPlatforms (ISMP) cannot start the Installation wizard.**

Verify or troubleshoot the installation if the *install\_root*/logs/log.txt file or the *profiles\_install\_root/profile\_name*/logs/pctLog.txt file does not contain a record of any problems, but problems exist. If the *profiles\_install\_root/profile\_name* directory exists, the pctLog.txt file is in the logs directory. If the error happens early in the installation, look for the logs in the system temporary area. The installation program copies the logs from the system temporary area to the logs directory at the end of the installation.

If the *profiles\_install\_root/profile\_name* directory does not exist, the pctLog.txt file is in the USER\_HOME directory and is named .\${pctLog.txt}. Issue the **env** command to display the USER\_HOME directory on Linux and UNIX systems. See the environment variable settings on Windows system to display the value of the variable.

Certain events can prevent ISMP from starting the Installation wizard. Such an event is not enough disk space to launch the Installation wizard, for example. If your installation fails and there is no information in the installation logs, use the `-log` parameter to record entries for events that cause the ISMP program to fail to start the installation wizard. The syntax of the **install** command for logging such events is:

```
install -options fully_qualified_options_response_file_name
        -silent
        -log # !fully_qualified_log_file_name @ALL
```

The following example is for AIX systems:

```
install -options "/usr/IBM/WebSphere/silentFiles/myresponsefile.txt"
        -silent -log # !/usr/IBM/WebSphere/myOptionFiles/log.txt @ALL
```

The following example is for Linux systems, HP-UX systems, and Solaris systems:

```
install -options "/opt/IBM/WebSphere/silentFiles/myresponsefile.txt"
        -silent -log # !/opt/IBM/WebSphere/myOptionFiles/log.txt @ALL
```

The following example is for Windows systems:

```
install.exe -options "C:\IBM\WebSphere\silentFiles\myresponsefile.txt"
            -silent -log # !C:\IBM\WebSphere\silentFiles\log.txt @ALL
```

### Attention:

The following examples show how to use the `-log` parameter when creating a deployment manager profile.

#### AIX

```
pctAIX.bin
  -options /usr/IBM/WebSphere/silentFiles/responsefile.pct.NDstandAloneProfile.txt
  -silent
  -log # !/usr/IBM/WebSphere/silentFiles/pctlog.txt @ALL
```

#### Linux HP-UX Solaris

This example shows the command for Linux on 32-bit platforms. Use the appropriate command for your Linux or UNIX operating system. The options and attributes are the same with only file path differences.

```
pctLinux.bin
  -options "/opt/IBM/WebSphere/silentFiles/responsefile.pct.NDstandAloneProfile.txt"
  -silent
  -log # !/opt/IBM/WebSphere/silentFiles/log.txt @ALL
```

#### Windows

The following command is for Windows platforms:

```
pctWindows.exe -options
"C:\IBM\WebSphere\silentFiles\responsefile.pct.NDstandAloneProfile.txt"
  -silent -log # !C:\IBM\WebSphere\silentFiles\log.txt
@ALL
```

## Customizing the options response file for Network Deployment

This topic describes how to customize the response file for installing the WebSphere Application Server Network Deployment product from a command line without using the graphical user interface of the Installation wizard.

Before using the `install -options myoptionsfile -silent` command on a Linux or UNIX platform, or the `install.exe -options myoptionsfile -silent` command on a Windows platform to invoke a silent installation, you must customize the response file to add your selections.

Use the response file to supply values to the Installation wizard as the wizard runs in silent mode. The wizard does not display interactive panels when it runs in silent mode, but reads values from the response file instead.

**Be precise when supplying values in the file:** Customize the options response file precisely to let the installation program read the option values that the file contains. Incorrect specifications affect the silent interface of the Installation wizard. For example, always enclose values in double quotation marks.

If you customize the response file incorrectly, the Installation wizard cannot install the product unless you remove the -options parameter. If the error is an invalid option value, ISMP displays a warning message that you must confirm and stops the installation.

Compare your options response file to the responsefile.nd.txt file that is shipped with the product to make the necessary corrections. After correcting the file, reinstall.

**Guidance:** The options response file has comments that describe each directive in the file. Read the explanations in the file to further understand the purpose of each directive.

See “responsefile.nd.txt” on page 128 for an example of the comments and directives in the response file.

Use the following description of options as a worksheet to plan your values. Refer to the individual descriptions of directives in the following list to help plan your responses:

```
-W silentInstallLicenseAcceptance.value="false"
# -W detectedexistingcopypanelInstallWizardBean.choice="addFeatures"
Platform specific: -P wasProductBean.installLocation="/usr/IBM/WebSphere/AppServer"
Platform specific: -P wasProductBean.installLocation="/opt/IBM/WebSphere/AppServer"
Platform specific: -P wasProductBean.installLocation="C:\Program Files\IBM\WebSphere\AppServer"
-P samplesProductFeatureBean.active="true"
-P javadocsProductFeatureBean.active="true"
-W ndsummarypanelInstallWizardBean.launchPCT="true"
-W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation=""
```

See “responsefile.nd.txt” on page 128 for an example of the response file. Refer to the following list of values to plan your responses to the options in the file:

**-W silentInstallLicenseAcceptance.value="false"**

Valid values are true or false. You must accept the license agreement before you are allowed to install the product.






Default value	Description	Your value
false	By changing the silentInstallLicenseAcceptance.value in this response file to "true", you agree that you have reviewed and agree to the terms of the IBM International Program License Agreement accompanying this program, which is located at CD_ROOT\WAS\lfiles. If you do not agree to these terms, do not change the value or otherwise download, install, copy, access, or use the program and promptly return the program and proof of entitlement to the party from whom you acquired it to obtain a refund of the amount you paid.  Change the value to true to accept the license agreement and install the product. You cannot install the product with the default value.	





**-W detectedexistingcopypanelInstallWizardBean.choice= "addFeatures"**

The only valid value for this directive is addFeatures. Leave the line commented if you are not adding features to a previous installation.

Default value	Description	Your value
Commented out	<p>Uncomment the line to add features to an existing installation in an incremental migration.</p> <p>Change the value of the following <code>-P wasProductBean.installLocation=</code> directive to the directory of the existing installation.</p> <p>To avoid problems, set the feature active properties for currently installed features to false. To install a new feature, set the feature active property to true.</p> <p>For example, suppose that you have an existing installation of the product with the Javadocs feature installed, but not the Samples feature. To install the Samples feature, set the following directives accordingly:</p> <ul style="list-style-type: none"> <li>• <code>-P samplesProductFeatureBean.active=true</code></li> <li>• <code>-P javadocsProductFeatureBean.active=false</code></li> </ul> <p>The feature selection options are described after the installation type option.</p>	

#### **-P wasProductBean.installLocation=**

-  `-P wasProductBean.installLocation="/usr/IBM/WebSphere/AppServer"`
-    `-P wasProductBean.installLocation="/opt/IBM/WebSphere/AppServer"`
-  `-P wasProductBean.installLocation="C:\Program Files\IBM\WebSphere\AppServer"`

Default value	Description	Your value
Varies per product disc	<p>The installation root directory for the product. Specify a valid directory.</p> <p>The default installation location is for a Windows system. Comment the Windows line to install on a Linux or UNIX operating system.</p> <p>Uncomment the line that describes your platform and optionally change the location. The installation path must be 80 characters or less.</p> <p> </p> <p><b>Important:</b> Do not use symbolic links as the destination directory. Symbolic links are not supported.</p> <p>  Spaces are also not supported in the name of the installation directory.</p> <p>On a Windows system, the directory can contain spaces. Enclose a directory name with spaces within double quotation marks as shown in the Windows example.</p>	

#### **-P samplesProductFeatureBean.active="true"**

Valid values are true or false.

Default value	Description	Your value
true	<p>This option installs the Samples feature.</p> <p>Change the value to false to skip the installation of the Samples feature.</p> <p>Change the value to false during an incremental installation if the Samples feature is already installed.</p> <p>Do not use a value of true if the Samples feature is already installed.</p>	

**-P javadocsProductFeatureBean.active="true"**

Valid values are true or false.

Default value	Description	Your value
true	<p>This option installs the Javadocs feature.</p> <p>Change the value to false to skip the installation of the Javadocs feature.</p> <p>Change the value to false during an incremental installation if the Javadocs feature is already installed.</p> <p>Do not use a value of true if the Javadocs feature is already installed.</p>	

**-W ndsummarypanellInstallWizardBean.launchPCT="true"**

Valid values are true or false.

Default value	Description	Your value
true	<p>A value of true launches the Profile creation wizard silently.</p> <p>This option controls launching the Profile creation wizard silently at the end of the installation of the core product files and features. The Profile creation wizard can create one of the three available profile types for the Network Deployment product.</p>	




**-W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation=""**

Valid values are the absolute path and file name of the customized options file for a profile. The original files are located in the WAS directory on the product disc or the downloaded installation image.

You must copy the appropriate file to a location on your machine where you can edit and save the file. The names of the original profile response files are:

- **Deployment manager profile:** "responsefile.pct.NDmgrProfile.txt" on page 242
- **Application Server profile:** "responsefile.pct.NDstandAloneProfile.txt" on page 267
- **Custom profile:** "responsefile.pct.NDmanagedProfile.txt" on page 254

Each file contains a planning area for marking your values.

Default value	Description	Your value
none	<p>This option must specify the absolute path to the customized options file for the appropriate profile.</p> <p>For example:</p> <p> </p> <p>-W pctresponse...fileLocation= "/tmp/WAS/responsefile.pct.NDmgrProfile.txt"</p> <p></p> <p>-W pctresponse...fileLocation= "C:\temp\WAS\responsefile.pct.NDmgrProfile.txt"</p>	

Perform the following procedure to use the options response file

1. Locate the sample options response file. The file is named `responsefile.nd.txt` in the WAS directory on the product CD-ROM or in a downloaded installation image.
2. Copy the file to preserve it in its original form. For example, copy it as *myoptionsfile* on your disk drive.
3. Edit the copy in your flat file editor of choice, on the target operating system. Read the directions within the response file to choose appropriate values.

**Important:** To prepare the file for a silent installation on AIX, use UNIX line-end characters (0x0D0A) to end each line of the options response file. The safest method of preparing the file is to edit the file on the target operating system.




4. Include custom option responses that reflect parameters for your system.  
Read the directions within the response file to choose appropriate values.
5. Save the file.
6. **Optional:** Edit the response file for the profile that you selected while editing the response file for the product installation. The directives for each file are described in the following topics:
  - “responsefile.pct.NDmgrProfile.txt” on page 242
  - “responsefile.pct.NDstandAloneProfile.txt” on page 267
  - “responsefile.pct.NDmanagedProfile.txt” on page 254
7. Save the response file for creating the profile.
8. Log on with the proper authorizations for installing the product.

Log on as root on a Linux platform or a UNIX platform, or as a user that belongs to the administrator group on a Windows platform. Installing as a non-root user is not supported on Linux or UNIX operating systems.

Some steps of the silent installation procedure on a Windows platform require that the administrator group user have the advanced user rights *Act as part of the operating system* and *Log on as a service*.

9. Issue the command to start the installation.

Use the following command to install WebSphere Application Server Network Deployment with a file that you customize with your own responses. The following command assumes that you copy the `responsefile.nd.txt` file as `myoptions.txt` before customizing the file.

-   `install -options /tmp/WAS/myoptions.txt -silent`
-  `install.exe -options "C:\temp\WAS\myoptions.txt" -silent`



This procedure results in creating a customized response file and using the file to start a silent installation.

Edit the version of the file that ships with the Network Deployment product. The examples in “responsefile.nd.txt” and the example profile response files are not guaranteed to be an accurate representation of what ships with the product. The profile creation file examples are:

- “responsefile.pct.NDdmgrProfile.txt” on page 242
- “responsefile.pct.NDstandAloneProfile.txt” on page 267
- “responsefile.pct.NDmanagedProfile.txt” on page 254

## responsefile.nd.txt

This topic describes the response file for silently installing the WebSphere Application Server Network Deployment product.

Install the product with an options response file after logging on as root on a Linux or UNIX platform, or a user that belongs to the administrator group on a Windows platform. Some steps of the installation procedure on a Windows platform require the user to belong to the administrator group and to have the advanced user rights *Act as part of the operating system* and *Log on as a service*.

**Location of the response file:** The sample options response file is named responsefile.nd.txt. The file is in the WAS directory on the product disc or in the downloaded installation image.

**Mode of use:** The installation wizard can read an existing options response file and run silently without displaying the graphical user interface.

**Installing silently:** The options file lets the Installation wizard run in silent mode. The wizard reads the options file to determine responses and does not display the graphical user interface. Use the following command to use a copy of the options file named myresponsefile.txt for a silent installation:

```
install -options "myresponsefile.txt" -silent
```

**Creating an operational environment:** Version 6 installation of the Network Deployment product is a two-step process:

1. Installing the core product files and feature files.
2. Creating a profile for the deployment manager, a managed node, or a stand-alone application server.

The sample options response file, responsefile.nd.txt, controls the first part of the installation and can also start the second part of the installation. To create a profile after installing the core product files, use the option in the responsefile.nd.txt file that identifies the response file for creating a profile. The profile response file lets you use the Profile creation wizard silently.

To edit and use the appropriate response file for creating a profile, perform the following procedure:

1. Copy the appropriate file from the WAS directory on the product disc to a place that you can easily identify on your machine. The example files are:

To create one:	Copy the following response file:
Deployment manager profile	responsefile.pct.NDdmgrProfile.txt
Application server profile	responsefile.pct.NDstandAloneProfile.txt
Custom profile	responsefile.pct.NDmanagedProfile.txt

2. Edit the file to customize the values for your installation.
3. Save the file.
4. Edit the responsefile.nd.txt file to identify the location and name of the profile response file. The installation path must be 80 characters or less. Change the value of the -W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation option to identify the file. For example:



Linux UNIX

```
-W pctresponse...fileLocation=
"/tmp/WAS/responsefile.pct.NDdmgrProfile.txt"
```

Windows

```
-W pctresponse...fileLocation=
"C:\temp\WAS\responsefile.pct.NDdmgrProfile.txt"
```

5. Start the installation. For example:

Linux UNIX

```
install -options /tmp/WAS/myresponsefile.txt -silent
```

Windows

```
install -options "C:\temp\WAS\myresponsefile.txt" -silent
```

6. After the installation, examine the logs for success.

### Logging:

The Installation wizard and the Profile creation wizard record installation events in the following log files.

**Log files for WebSphere Application Server products:** The following table shows the installation logs, content, and indicators of success and failure for WebSphere Application Server products for Version 6.

Table 12. Installation logs for WebSphere Application Server products

Log	Content	Indicators
<i>install_root</i> / logs / log.txt	Logs all installation events	<p><b>INSTCONFFAIL</b> Total installation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful installation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Installation errors occurred but the installation is still usable. Additional information identifies the errors.</p>
<i>install_root</i> / logs / wasprofile / wasprofile _ create _ profile_name.log	<ul style="list-style-type: none"> <li>Traces all events that occur during the creation of the named profile</li> <li>Created when using the Profile creation wizard or the <b>wasprofile</b> command</li> </ul>	<p><b>INSTCONFFAIL</b> Total profile creation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile creation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile creation errors occurred but the profile is still functional. Additional information identifies the errors.</p>
<i>install_root</i> / logs / wasprofile / wasprofile _ delete _ profile_name.log	<ul style="list-style-type: none"> <li>Traces all events that occur during the deletion of the named profile</li> <li>Created when using the Profile creation wizard or the <b>wasprofile</b> command</li> </ul>	<p><b>INSTCONFFAIL</b> Total profile deletion failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile deletion.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile deletion errors occurred but the profile is still deleted. Additional information identifies the errors.</p>

Table 12. Installation logs for WebSphere Application Server products (continued)

Log	Content	Indicators
<i>install_root</i> / <i>profiles</i> / <i>profile_name</i> / <i>logs</i> / <i>pctLog.txt</i>	Logs all profile creation events that occur when using the Profile creation wizard	<p><b>INSTCONFFAIL</b> Total profile creation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile creation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile creation errors occurred but the profile is still functional. Additional information identifies the errors.</p>

See Appendix B, “Troubleshooting installation,” on page 337 for more information.

**Usage notes:**

- The file is not a read-only file.
- Edit this file directly with your flat file editor of choice, such as WordPad on a Windows platform.
- The file is updated when you specify the `-options` parameter when using the Installation wizard.
- The file must exist to perform a silent installation. The installation program reads this file to determine installation option values when you install silently.
- Save the file in a location that you can identify when you specify the fully qualified path as part of the installation command.
- The file must identify the appropriate response file for creating a profile if you want to have an operational application server environment when you finish installing.

**Example responsefile.nd.txt file:** Edit the version of the file that ships with the WebSphere Application Server Network Deployment product. The following example is not guaranteed to be an accurate representation of the file that ships with the product.

```
#####
#
# InstallShield Options File
#
# Wizard name: Install
# Wizard source: setup.jar
#
# This file can be used to configure Install with the options specified below
# when the wizard is run with the "-options" command line option. Read each
# setting's documentation for information on how to change its value.
#
# A common use of an options file is to run the wizard in silent mode. This lets
# the options file author specify wizard settings without having to run the
# wizard in graphical or console mode. To use this options file for silent mode
# execution, use the following command line arguments when running the wizard:
#
# -options "D:\installImage\WAS\responsefile.nd.txt" -silent
#
#####

#####
#
# License Acceptance
#
# Valid Options : true  Accepts the license. Will install the product.
#                 false Declines the license. Install will not occur.
#
# If no install occurs, this will be logged to a temporary log file in the user's
# temporary directory.
#
# By changing the silentInstallLicenseAcceptance.value in this response file to "true",
# you agree that you
```

```

# have reviewed and agree to the terms of the IBM International Program License Agreement
# accompanying this
# program, which is located at CD_ROOT\WAS\lfiles. If you do not agree to these terms,
# do not change
# the value or otherwise download, install, copy, access, or use the program and promptly
# return the
# program and proof of entitlement to the party from whom you acquired it to obtain a
# refund of the amount you paid.
#
#

```

```
-W silentInstallLicenseAcceptance.value="false"
```

```

#####
#
# Incremental Install
#
# If you are installing additional features on top of an existing installation,
# (e.g. incremental install), uncomment the following line. This will notify
# the installer that you are doing an incremental install.
#
# -W NDdetectedexistingcopypanelInstallWizardBean.choice="addFeatures"
#
# Please make sure installLocation is set to your existing install location.
#
# For already installed features, you need to set the feature active property
# to false. For new features you want to install, you need to set the active
# properties to true.
#
# For example, you have installed core product feature, now you want to install
# samples feature, the active properties should be:
#
# -P samplesProductFeatureBean.active="true"
# -P javadocsProductFeatureBean.active="false"

```

```

#####
#
# IBM WebSphere Application Server Network Deployment, V6.0 Install
# Location
#
# The install location of the product. Specify a valid directory into which the
# product should be installed. If the directory contains spaces, enclose it in
# double-quotes as shown in the Windows example below. Note that spaces in the
# install location is only supported on Windows operating systems.
#
# Below is the list of default install locations for each supported operating
# system. By default, in this response file, the Windows install location is
# used. If you want to use the default install location for another operating
# system, uncomment the appropriate default install location entry (by removing '#') and
# then comment out (by adding '#') the Windows operating system entry below.
#
#
# AIX Default Install Location:
#
# -P wasProductBean.installLocation=/usr/IBM/WebSphere/AppServer
#
#
# HP-UX, Solaris or Linux Default Install Location:
#
# -P wasProductBean.installLocation=/opt/IBM/WebSphere/AppServer
#
#
# Windows Default Install Location:
#

```

```
-P wasProductBean.installLocation="C:\Program Files\IBM\WebSphere\AppServer"
```

```
#####  
#  
# "Core Product Files" feature  
#  
# This feature is required by Application Server and will be installed by default.  
# Selection is not required.  
#
```

```
#####  
#  
# "Application Server samples" feature  
#  
# The selection state of the "Application Server samples" feature. Legal values  
# are:  
#  
# true - Indicates that the feature is selected for installation  
# false - Indicates that the feature is not selected for installation  
#  
# For example, to select "Application Server samples" for installation, use  
#  
# -P samplesProductFeatureBean.active="true"  
#
```

```
-P samplesProductFeatureBean.active="true"
```

```
#####  
#  
# "Javadocs" feature  
#  
# The selection state of the "Javadocs" feature. Legal values are:  
#  
# true - Indicates that the feature is selected for installation  
# false - Indicates that the feature is not selected for installation  
#  
# For example, to select "Javadocs" for installation, use  
#  
# -P javadocsProductFeatureBean.active="true"  
#
```

```
-P javadocsProductFeatureBean.active="true"
```

```
#####  
#  
# WebSphere Application Server Profile Creation Tool  
#  
# To launch the Profile Creation Tool SILENTLY at the end  
# of the installation, please change the value to "true".  
# Otherwise, change the value to "".  
#
```

```
-W ndsummarypanelInstallWizardBean.launchPCT="true"
```

```
#####  
# If you choose to run Profile Creation Tool silently at the end of the  
# installation, you must  
# 1. get a copy of the PCT response file of Profile creator (from CD) and put to a temp dir  
# such as: /tmp/WAS/responsefile.pct.NDdmgrProfile.txt For DMGR profile creation
```

```
#           /tmp/WAS/responsefile.pct.NDmanagedProfile.txt    For Manage profile creation
#           /tmp/WAS/responsefile.pct.NDstandAloneProfile.txt For Stand alone profile creation
# 2. modify the response file by filling in the required options/values.
# Note: If the values are not completed properly, the configuration may be unsuccessful.
# 3. specify below the absolute path to your modified response file.
#
# For example, -W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation=
#           "/tmp/WAS/responsefile.pct.NDmgrProfile.txt"
```

```
-W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation=""
```

---

## Mounting CD-ROMS on Linux and UNIX operating systems

The following sections describe how to mount the CD-ROM on Linux and UNIX operating systems.

After inserting a CD-ROM into a drive, some Linux and UNIX operating systems require you to mount the drive.

Use these procedures to mount the product discs.

- Mounting the CD-ROM on AIX To mount the CD-ROM on AIX using the System Management Interface Tool (SMIT), perform the following steps:
  1. Log in as a user with root authority.
  2. Insert the CD-ROM in the drive.
  3. Create a CD-ROM mount point by entering the `mkdir -p /cdrom` command, where `cdrom` represents the CD-ROM mount point directory.
  4. Allocate a CD-ROM file system using SMIT by entering the **smit storage** command.
  5. After SMIT starts, click **File Systems > Add / Change / Show / Delete File Systems > CDROM File Systems > Add CDROM File System**.
  6. In the Add a File System window:
    - Enter a device name for your CD-ROM file system in the **DEVICE Name** field. Device names for CD-ROM file systems must be unique. If there is a duplicate device name, you may need to delete a previously-defined CD-ROM file system or use another name for your directory. The example uses `/dev/cd0` as the device name.
    - Enter the CD-ROM mount point directory in the **MOUNT POINT** window. In our example, the mount point directory is `/cdrom`.
    - In the **Mount AUTOMATICALLY at system restart** field, select yes to enable automatic mounting of the file system.
    - Click **OK** to close the window, then click **Cancel** three times to exit SMIT.
  7. Next, mount the CD-ROM file system by entering the **smit mountfs** command.
  8. In the Mount a File System window:
    - Enter the device name for this CD-ROM file system in the **FILE SYSTEM name** field. In our example, the device name is `/dev/cd0`.
    - Enter the CD-ROM mount point in the **Directory over which to mount** field. In our example, the mount point is `/cdrom`.
    - Enter `cdrfs` in the **Type of Filesystem** field. To view the other kinds of file systems you can mount, click List.
    - In the **Mount as READ-ONLY system** field, select yes.
    - Accept the remaining default values and click **OK** to close the window.

Your CD-ROM file system is now mounted. To view the contents of the CD-ROM, place the disk in the drive and enter the `cd /cdrom` command where `cdrom` is the CD-ROM mount point directory.

- Mounting the CD-ROM on HP-UX Because the product contains several files with long file names, the mount command can fail. The following steps successfully mount the product CD-ROM on the HP-UX platform:

1. Log in as a user with root authority.
2. In the /etc directory, add the following line to the pfs\_fstab file:  

```
/dev/dsk/c0t2d0 mount_point pfs-rrip ro,hard
```

where *mount\_point* represents the mount point of the CD-ROM.

3. Start the *pfs* daemon by entering the following commands (if they are not already running):

```
/usr/sbin/pfs_mountd &  
/usr/sbin/pfsd 4 &
```

4. Insert the CD-ROM in the drive and enter the following commands:

```
mkdir /cdrom  
/usr/sbin/pfs_mount /cdrom
```

The */cdrom* variable represents the mount point of the CD-ROM.

5. Log out.

- Mounting the CD-ROM on Linux To mount the CD-ROM on Linux:

1. Log in as a user with root authority.
2. Insert the CD-ROM in the drive and enter the following command:

```
mount -t iso9660 -o ro /dev/cdrom /cdrom
```

The */cdrom* variable represents the mount point of the CD-ROM.

3. Log out.

Some window managers can automatically mount your CD-ROM for you. Consult your system documentation for more information.

- Mounting the CD-ROM on Solaris To mount the CD-ROM on Solaris:

1. Log in as a user with root authority.
2. Insert the CD-ROM into the drive.
3. If the Volume Manager is not running on your system, enter the following commands to mount the CD-ROM:

```
mkdir -p /cdrom/unnamed_cdrom  
mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom/unnamed_cdrom
```

The */cdrom/unnamed\_cdrom* variable represents the CD-ROM mount directory and the */dev/dsk/c0t6d0s2* represents the CD-ROM drive device.

If you are mounting the CD-ROM drive from a remote system using NFS, the CD-ROM file system on the remote machine must be exported with root access. You must also mount that file system with root access on the local machine.

If the Volume Manager (vold) is running on your system, the CD-ROM is automatically mounted as:

```
/cdrom/unnamed_cdrom
```

4. Log out.

---

## Installing IBM HTTP Server

Install the IBM HTTP Server Version 6 product and its plug-in, or install a plug-in for another supported Web server to enable the Web server to work with WebSphere Application Server, Version 6.

Before using this topic to install the IBM HTTP Server, see the Information center for IBM HTTP Server.

To use a Web server other than IBM HTTP Server Version 6, install and configure the Web server before or after installing the WebSphere Application Server product, but before installing the Web server plug-ins for WebSphere Application Server.

The Plug-ins installation wizard configures supported Web servers. You can also manually configure supported Web servers for WebSphere Application Server, Version 6, as described in “Editing Web server configuration files” on page 203.

See the information center for IBM HTTP Server Version 6 for more information.

After installing the Web server and the Application Server, install the appropriate Web server plug-in for a supported, installed Web server. No further configuration is required for most Web servers.

This topic describes installing IBM HTTP Server.

1. Prepare your operating platform for installing IBM HTTP Server as you would for installing any of the installable components on the product disc. See Chapter 8, “Preparing the operating system for product installation,” on page 81.
2. Insert the product disc and mount the disc if necessary. See “Packaging” on page 30 for information about identifying the product disc. See “Mounting CD-ROMS on Linux and UNIX operating systems” on page 133 for information about mounting the product disc, if you are installing IBM HTTP Server on a Linux or UNIX platform.
3. Start the installation with the **launchpad.sh** command on Linux and UNIX platforms or the **launchpad.bat** on Windows platforms. You can also start the installation using the **/IHS/install** command, where IHS is the installable component directory on the product disc:



-   /IHS/install
-  \IHS\install

If you have a problem launching the launchpad, use the troubleshooting information in “Using the launchpad to start the installation” on page 41 to correct the problem.

When using the launchpad, launch the Installation wizard for IBM HTTP Server:

After launching the Installation wizard from the launchpad or from the command line, the ISMP wizard initializes and then displays the Welcome panel.

Separate installation procedures for the WebSphere Application Server product, the IBM HTTP Server product, and the Web server plug-ins let you install only what you need on a particular machine. Installation of the product code is simplified and faster than in Version 5. The installation program installs the system files (shared binary files), which you do not update until you install a service fix. Creation and configuration of application servers do not change the system files.

4. Click **Next** to display the License agreement panel.
5. Accept the license agreement and click **Next** to display the installation root directory panel.
6. Specify the root directory information and click **Next** to display the feature type selection panel.  
The panel lets you bypass features selection by accepting typical features. Selecting Custom lets you select features in the Features selection panel.
7. Click **Custom** to select features and click **Next** to display the Features selection panel.
8.  Select features to install and click **Next** to display the Windows service authorization panel, when installing IBM HTTP Server on a Windows platform.
9.  Specify your user ID and password information and click **Next** when installing IBM HTTP Server on a Windows platform.
10. Review the confirmation panel to verify your selections. Click **Back** to change any of your specifications. Click **Next** to begin installing IBM HTTP Server.

11. Review the confirmation panel to verify your selections. Click **Back** to change any of your specifications. Click **Next** to begin installing IBM HTTP Server.  
After displaying installation status, the wizard displays the Completion status panel that indicates a successful installation.
12. Click Next to display the Web server plug-ins prompt panel.
13. Click Next to launch the Plug-ins installation wizard. See “Installing Web server plug-ins” to continue the installation.  
If the plugin directory does not exist at the same level as the IHS directory, the prompt panel for selecting the plug-ins installer does not display and the installation is finished. In that case, launch the Plug-ins installation wizard using the launchpad.

You can install the IBM HTTP Server product.

You can get started easily with Secure Sockets Layer (SSL) connections, by making only a few configuration changes, as described in Configuring IBM HTTP Server for SSL mutual authentication. If you are using the IBM HTTP Server on a Windows platform, you can configure the Fast Response Cache Accelerator to boost performance. You can also make many other configuration changes with Apache directives. Refer to the Information center for IBM HTTP Server at <http://www-3.ibm.com/software/webservers/htpservers/library.html>, for a description of configuring the Web server for SSL, the Fast Response Cache Accelerator, or Apache directives.

Return to Chapter 9, “Installing the product and additional software,” on page 107 to continue.

---

## Installing Web server plug-ins

This topic describes installing a Web server plug-in for WebSphere Application Server. WebSphere Application Server products supply a unique binary plug-in module for each supported Web server. The plug-in configuration file that the WebSphere Application Server products supply is associated with the binary module. The Plug-ins installation wizard installs required files and configures the Web server and the application server to allow communication between the servers.

You must install a supported Web server before you can install a plug-in for the Web server. If the Web server is not already installed, you cannot install the plug-in for it. If the WebSphere Application Server product is not installed, you can still install the plug-in.

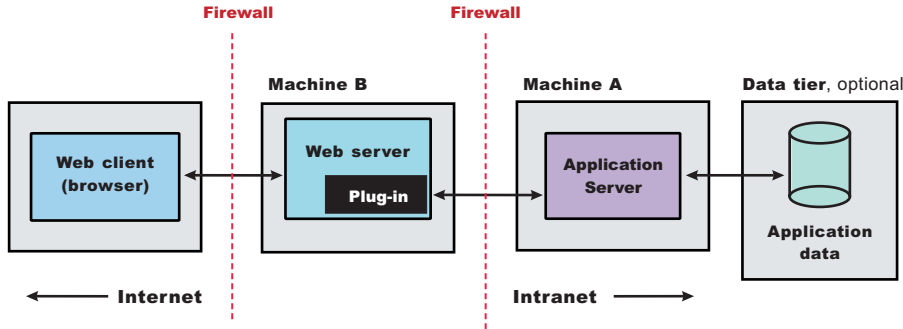
The Plug-ins installation wizard installs the plug-in module, configures the Web server for communicating with the application server, and creates a Web server configuration definition in the application server, if possible.

Some topologies, such as the Web server on one machine and the application server on another machine, prevent the Plug-ins installation wizard from creating the Web server definition in the application server configuration on the remote machine. In such a case, the Plug-ins installation wizard creates a script that you can copy to the application server machine. Run the script to create the Web server configuration definition within the application server configuration.

Select one of the following topology scenarios and follow the steps below the diagram to install the plug-in and configure both the Web server and the application server.

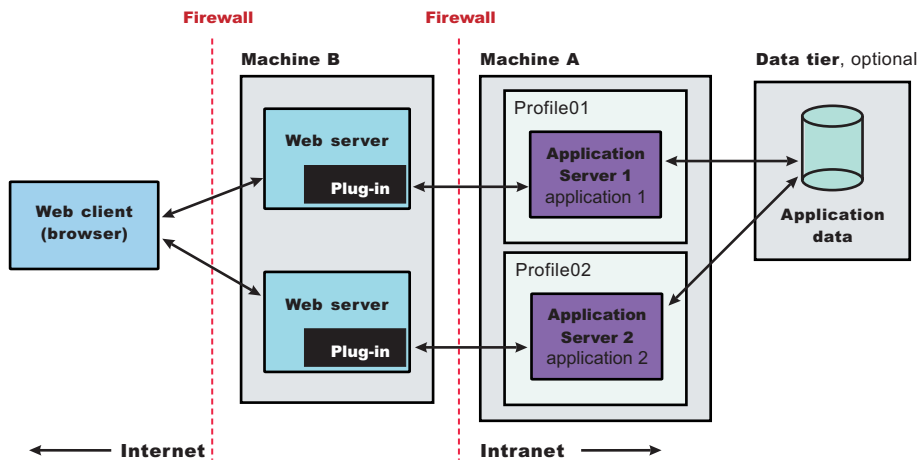
- **Scenario 1: Remote** Install the Web server plug-ins when the application server and the Web server are on separate machines.





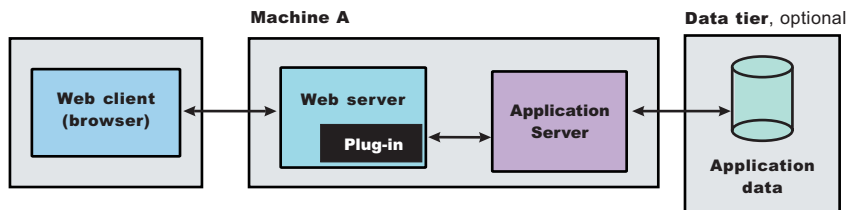
See “Configuring a Web server and an application server on separate machines (remote)” on page 157 for the procedure that explains how to create this Web server topology.

- **Scenario 2: Remote** Install the Web server plug-ins when multiple stand-alone application servers are on one machine, and each application server has a dedicated Web server on a separate machine.

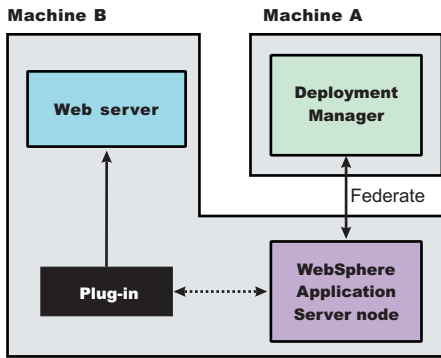


See “Configuring multiple Web servers and remote stand-alone Application Servers” on page 163 for the procedure that explains how to create this Web server topology.

- **Scenario 3: Local Application Server profile** Install the Web server plug-ins when the application server and the Web server are on a single machine.



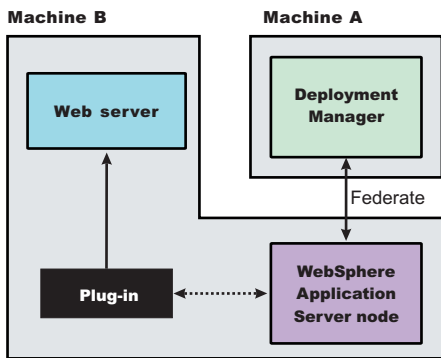
A local distributed installation includes the Web server plug-in, the Web server, and a *managed* application server on the same machine:



See “Configuring a Web server and an application server profile on the same machine” on page 170 for the procedure that explains how to create this Web server topology for an application server profile.

- **Scenario 4: Local custom profile** Install the Web server plug-ins when a managed node and the Web server are on the same machine.

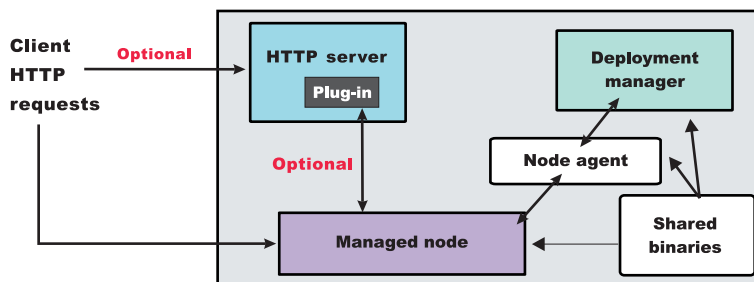
A local distributed installation includes the Web server plug-in, the Web server, and the managed custom node on the same machine:



See “Configuring a Web server and a custom profile on the same machine” on page 180 for the procedure that explains how to create this Web server topology for a federated custom profile.

- **Scenario 5: Local deployment manager profile** Install the Web server plug-ins when a deployment manager node and the Web server are on a single machine.

A local distributed installation includes the Web server plug-in, the Web server, and the application server on the same machine:



See “Configuring a Web server and a deployment manager profile on the same machine” on page 185 for the procedure that explains how to create this Web server topology for a deployment manager profile.

## Selecting a Web server topology diagram and roadmap

This topic helps you use the Plug-ins installation wizard to install Web server plug-ins for WebSphere Application Server. This topic is the information center version of the roadmap that is displayed when you use the Plug-ins installation wizard.

The primary production configuration is an application server on one machine and a Web server on a separate machine. This configuration is referred to as a *remote* configuration. Contrast the remote configuration to the local configuration, where the application server and the Web server are on the same machine.

The Plug-ins installation wizard has four main tasks:

- Installs the binary plug-in module on the Web server machine.
- Configures the Web server configuration file on the Web server machine to point to the binary plug-in module and to the XML configuration file for the binary module.
- Installs a temporary XML configuration file for the binary module (`plugin-cfg.xml`) on the Web server machine in remote scenarios.
- Creates the configuration for a Web server definition on the application server machine. The wizard processes the creation of the Web server definition differently depending on the scenario:

### **Web server plug-in installation for stand-alone application server environments**

- Recommended remote stand-alone application server installation:

Creates a configuration script that you run on the application server machine. Install the Web server and its plug-in on a different machine than the application server. This configuration is recommended for a production environment.

- Local stand-alone application server installation:

Detects the default profile on a local application server machine and creates the Web server definition for it directly. Install the Web server and its plug-in on the same machine with the application server. This configuration is for development and test environments.

### **Web server plug-in installation for distributed environments (cells)**

- Recommended remote distributed installation:

Creates a configuration script that you run on the application server machine. Install the Web server and its plug-in on a different machine than the deployment manager or managed node. This configuration is recommended for a production environment.

- Local distributed installation:

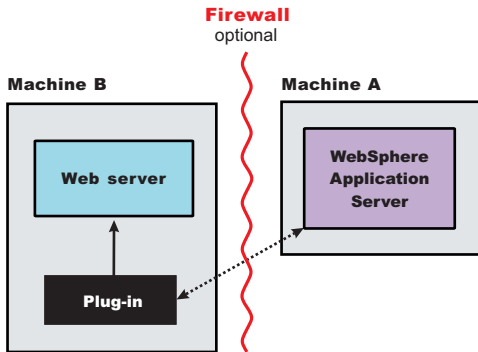
Creates a configuration script that you run when the deployment manager is running. Install the Web server and its plug-in on the same machine with the deployment manager or a managed node. This configuration is for development and test environments.

Select a link to go to the appropriate steps in the following procedure.

- **Set up a remote Web server installation.**



The remote Web server configuration is recommended for production environments.

The remote installation installs the Web server plug-in on the Web server machine when the application server is on a separate machine, such as shown in the following graphic:



### Remote installation scenario

Table 13. Installation and configuration

Step	Machine	Task
1	A	Install your WebSphere Application Server product. See Installing the product and additional software.
2	A	Create an application server profile. Using the Profile creation wizard.
3	B	Install IBM HTTP Server or another supported Web server. See Installing IBM HTTP Server.
4	B	Install the binary plug-in module using the Plug-ins installation wizard. See “Configuring a Web server and an application server on separate machines (remote)” on page 157.  The script for creating and configuring the Web server is created under the <i>plug-ins_install_root/ bin</i> directory.
5	B	Copy the <i>configureWeb_server_name</i> script to Machine A. If one machine is running under Linux or UNIX and the other machine is running under Windows, copy the script from the <i>plug-ins_install_root/ bin/ crossPlatformScripts</i> directory.
6	A	Paste the <i>configureWeb_server_name</i> script from Machine B to the <i>was_install_root/ bin</i> directory on Machine A.
7	A	Start the application server, then run the script from a command line.
8	A	Verify that the application server is running. Open the administrative console and save the changed configuration.
9	B	  Run the <i>plug-ins_install_root/setupPluginCfg.sh</i> script for a Domino Web Server before starting a Domino Web server. Otherwise, start the Web server.
10	B	Run the snoop servlet.  To verify with your own application, regenerate and propagate the <i>plugin-cfg.xml</i> file after installing the application.

### Regeneration of the plugin-cfg.xml file

During the installation of the plug-ins, the temporary *plugin-cfg.xml* file is installed on Machine B in the *plug-ins\_install\_root/ config/ web\_server\_name* directory.

The Web server plug-in configuration service regenerates the *plugin-cfg.xml* file automatically.

To use the real *plugin-cfg.xml* file from the application server, propagate the *plugin-cfg.xml* file as described in the next section.

### Propagation of the plugin-cfg.xml file

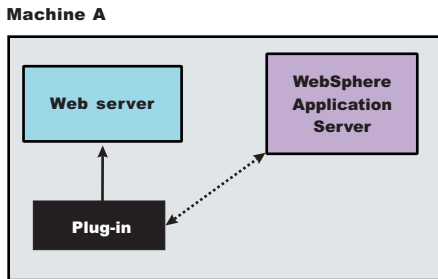
The Web server plug-in configuration service propagates the `plugin-cfg.xml` file automatically for IBM HTTP Server 6.0 only.

For all other Web servers, propagate the plug-in configuration file manually. Copy the `plugin-cfg.xml` file from the `profiles_install_root/config/cells/cell_name/nodes/Web_server_name_node/servers/web_server_name` directory on Machine A. Paste the file into the `plug-ins_install_root/config/web_server_name` directory on Machine B.

- **Set up a local Web server configuration.**

The local Web server configuration is recommended for a development or test environment.

A local installation includes the Web server plug-in, the Web server, and the Application Server on the same machine:



### Local installation scenario

Table 14. Installation and configuration

Step	Machine	Task
1	A	Install your WebSphere Application Server product. See Installing the product and additional software.
2	A	Create an application server profile. See Using the Profile creation wizard.
3	A	Install IBM HTTP Server or another supported Web server. See Installing IBM HTTP Server.
4	A	Install the binary plug-in module using the Plug-ins installation wizard. See "Configuring a Web server and an application server profile on the same machine" on page 170.  The Web server definition is automatically created and configured during the installation of the plug-ins.
5	A	Verify that the application server is running. Open the administrative console and save the changed configuration.
6	B	<div style="display: flex; align-items: center; gap: 10px;"> <span style="background-color: #f0ad4e; padding: 2px 5px;">Linux</span> <span style="background-color: #555; color: white; padding: 2px 5px;">UNIX</span> </div> Run the <code>plug-ins_install_root/setupPluginCfg.sh</code> script for a Domino Web Server before starting a Domino Web server.  Start the Web server.
7	B	Run the snoop servlet.  To verify with your own application, regenerate and propagate the <code>plugin-cfg.xml</code> file after installing the application.

### Regeneration of the `plugin-cfg.xml` file

The Web server plug-in configuration service regenerates the `plugin-cfg.xml` file automatically.

The `plugin-cfg.xml` file is generated in the `profiles_install_root/profile_name/config/cells/cell_name/nodes/Web_server_name_node/servers/web_server_name` directory. The generation occurs when the Web server definition is created.

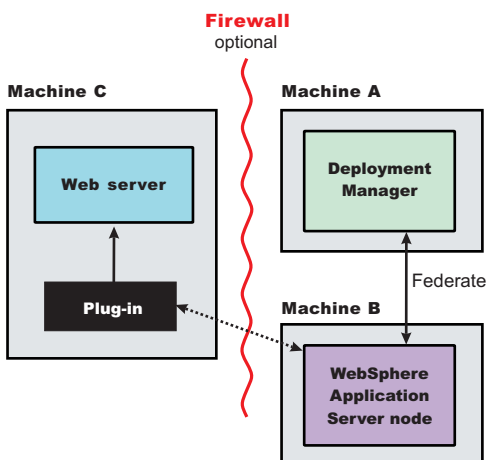
### Propagation of the `plugin-cfg.xml` file

The local file does not require propagation.

- **Set up a remote Web server installation in a cell.**

The remote Web server configuration is recommended for production environments.

The remote installation installs the Web server plug-in on the Web server machine when the application server is on a separate machine, such as shown in the following graphic:




### Remote installation scenario

Table 15. Installation and configuration

Step	Machine	Task
1	A	Install WebSphere Application Server Network Deployment. See Installing the product and additional software.
2	A	Create a deployment manager profile. See Using the Profile creation wizard.
3	A	Start the deployment manager with the <code>./profiles_install_root/profile_name/bin/startManager.sh</code> command or its Windows equivalent.
4	B	Install WebSphere Application Server Network Deployment. See Chapter 9, "Installing the product and additional software," on page 107.
5	B	Create an application server profile. See Using the Profile creation wizard.
6	B	Federate the node with the <code>./profiles_install_root/profile_name/bin/addNode.sh dmgrhost 8879 -includeapps</code> command or its Windows equivalent. Federating the node starts the nodeagent process, which is required to be running during this configuration.
7	C	Install IBM HTTP Server or another supported Web server. See Installing IBM HTTP Server.
8	C	Install the binary plug-in module using the Plug-ins installation wizard. See "Configuring a Web server and an application server on separate machines (remote)" on page 157.  The script for creating and configuring the Web server is created under the <code>plug-ins_install_root/bin</code> directory.
9	C	Copy the <code>configureWeb_server_name</code> script to Machine A.  If one machine is running under Linux or UNIX and the other machine is running under Windows, copy the script from the <code>plug-ins_install_root/bin/crossPlatformScripts</code> directory.
10	A	Paste the <code>configureWeb_server_name</code> script from Machine C to the <code>was_install_root/bin</code> directory on Machine A.

Table 15. Installation and configuration (continued)

Step	Machine	Task
11	A	Start the node agent and the deployment manager if they are not already running, then run the script from a command line.  If you have enabled security or changed the default JMX connector type, edit the script and include the appropriate parameters on the <b>wsadmin</b> command.
12	A/B	Use the administrative console of the deployment manager on Machine A to start the application server on Machine B. Wait for synchronization to occur and save the new configuration.
13	C	 Run the <code>plug-ins_install_root/setupPluginCfg.sh</code> script for a Domino Web Server before starting a Domino Web server. Otherwise, start the Web server.
14	C	Run the snoop servlet.  To verify with your own application, regenerate and propagate the <code>plugin-cfg.xml</code> file after installing the application.

### Regeneration of the plugin-cfg.xml file

During the installation of the plug-ins, the temporary `plugin-cfg.xml` file is installed on Machine C in the `plug-ins_install_root/config/web_server_name` directory.

The Web server plug-in configuration service regenerates the `plugin-cfg.xml` file automatically.

To use the real `plugin-cfg.xml` file from the application server, propagate the `plugin-cfg.xml` file as described in the next section.

### Propagation of the plugin-cfg.xml file

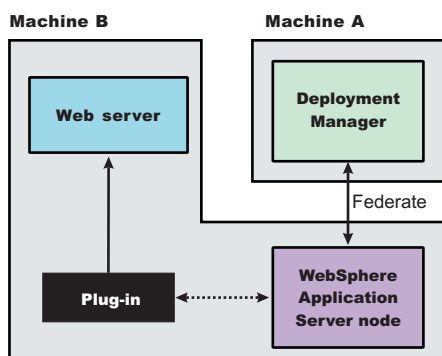
The Web server plug-in configuration service propagates the `plugin-cfg.xml` file automatically for IBM HTTP Server 6.0 only.

For all other Web servers, propagate the plug-in configuration file, by manually copying the `plugin-cfg.xml` file from the `profiles_install_root/profile_name/config/cells/cell_name/nodes/node_name/servers/web_server_name` directory on Machine A to the `plug-ins_install_root/config/web_server_name` directory on Machine C.

- **Set up a local distributed Web server configuration.**


The local Web server configuration is recommended for a development or test environment.

A local distributed installation includes the Web server plug-in, the Web server, and the managed application server on the same machine:



## Local distributed installation scenario

Table 16. Installation and configuration

Step	Machine	Task
1	A	Install WebSphere Application Server Network Deployment. See Installing the product and additional software.
2	A	Create a deployment manager profile. See Using the Profile creation wizard.
3	A	Start the deployment manager with the <code>profiles_install_root/profile_name/bin/startManager.sh</code> command or its Windows equivalent.
4	B	Install WebSphere Application Server Network Deployment. See Installing the product and additional software.
5	B	Create an application server profile. See Using the Profile creation wizard.
6	B	Federate the node with the <code>./profiles_install_root/profile_name/bin/addNode.sh dmgrhost 8879 -includeapps</code> command or its Windows equivalent. Federating the node starts the nodeagent process, which is required to be running during this configuration.
7	B	Install IBM HTTP Server or another supported Web server. See Installing IBM HTTP Server.
8	B	Install the binary plug-in module using the Plug-ins installation wizard. See “Configuring a Web server and an application server profile on the same machine” on page 170.  The script for creating and configuring the Web server is created in the <code>plug-ins_install_root/bin</code> directory.
11	B	After verifying that the deployment manager and the node agent are running on Machine A, run the <code>configureWeb_server_name</code> script from a command line in the <code>plug-ins_install_root/bin</code> directory on Machine B.  If you have enabled security or changed the default JMX connector type, edit the script and include the appropriate parameters.
12	A/B	Use the administrative console of the deployment manager on Machine A to start the application server on Machine B. Wait for synchronization to occur and save the new configuration.
13	B	 Run the <code>plug-ins_install_root/setupPluginCfg.sh</code> script for a Domino Web Server before starting a Domino Web server. Otherwise, start the Web server.
14	B	Run the Snoop servlet.

### Regeneration of the plugin-cfg.xml file

The Web server plug-in configuration service regenerates the `plugin-cfg.xml` file automatically.

The `plugin-cfg.xml` file is generated at the location `profiles_install_root/profile_name/config/cells/cell_name/nodes/node_name/servers/web_server_name` directory, when the Web server definition is created.

Regenerate the `plugin-cfg.xml` file in the Web server definition in the application server whenever the configuration changes. The Web server has immediate access to the file whenever it is regenerated.

When the Web server plug-in configuration service (an administration service) is enabled on Machine A, the `plugin-cfg.xml` file is automatically generated for all Web servers.

### Propagation of the plugin-cfg.xml file

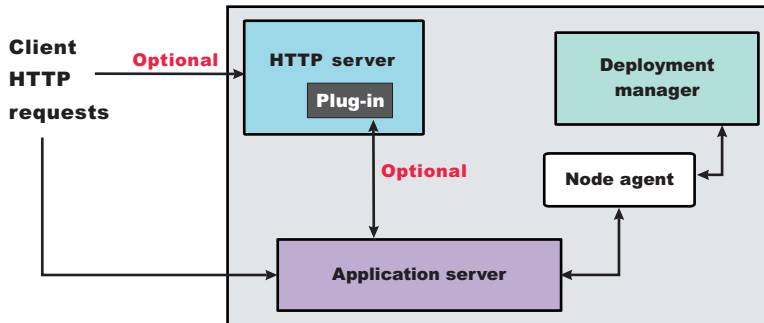
Node synchronization is used to propagate the `plugin-cfg.xml` file from Machine A to Machine B.

When the Web server plug-in configuration service (an administration service) is enabled on Machine A, the `plugin-cfg.xml` file is automatically propagated for all Web servers.

### Alternate configuration



This procedure describes installing the plug-ins on two machines. However, you can perform this procedure on a single machine as shown in the following graphic. A local distributed installation also includes the Web server plug-in, the Web server, the Application Server, and the deployment manager on the same machine:



## Configuration behavior of the Plug-ins installation wizard

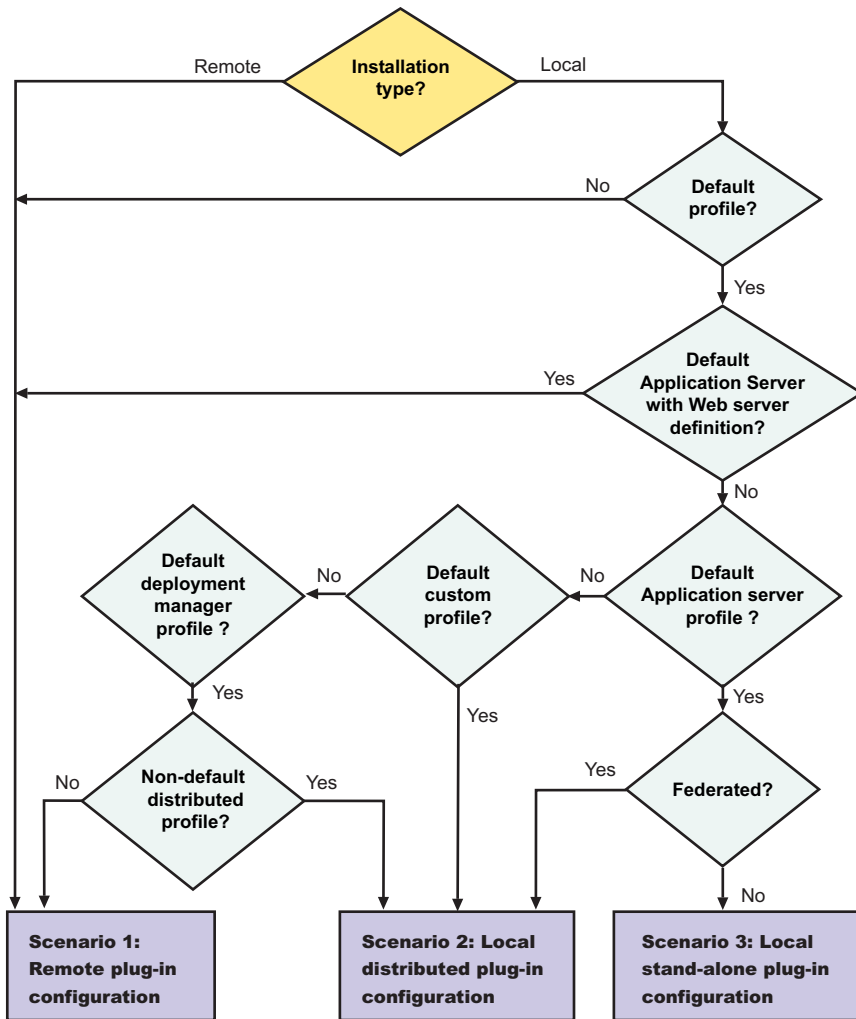
This topic describes installing the Web server plug-ins for WebSphere Application Server. The Plug-ins installation wizard installs a binary plug-in module and a plug-in configuration file for the Web server. The Plug-ins installation wizard configures the supported Web server for the Application Server. The Plug-ins installation wizard also creates a Web server definition in the configuration of the application server.

This topic describes the three ways that the Plug-ins installation wizard configures a Web server to locate the `plugin-cfg.xml` file, which is the plug-in configuration file.

### Configuration flows for the Network Deployment product

The Plug-ins installation wizard resolves all configurations of Web server and WebSphere Application Server to three scenarios: a remote application server, local distributed application Server, and local stand-alone Application Server. The logic implemented in determining which scenario applies to a configuration is shown in the following diagram.

## Web server plug-ins for WebSphere Application Server



### Legend:

#### Default application server with Web server definition?

If the default profile is an application server with an existing Web server definition, then the installation is considered a remote installation. You cannot have more than one Web server definition in a stand-alone application server.

Use the same name for the Web server to configure a new Web server to use the existing Web server definition.

Use a different Web server name to create a script that creates a new Web server definition in a federated application server profile.

#### Default profile?

If the product is installed but the Profile creation wizard has not yet created a profile, the scenario is considered to be a remote installation.

#### Default *profile\_type*?

The Plug-ins installation wizard can configure only one profile at a time. The wizard always works with the default profile. These three paths show how processing varies for different types of profiles.

## Federated?

If the application server node is federated, the Plug-ins installation wizard configures the Web server definition on the managed node. This has advantages. Suppose the Web server and the managed node are on a separate machine. The `plugin-cfg.xml` file is automatically propagated to the remote node during node synchronization because the Web server definition is part of the node configuration.

## Installation type?

The installation type is either remote or local.

## Non-default distributed profile?

If the deployment manager has a federated custom node (custom profile), the Plug-ins installation wizard configures the Web server definition on the managed node. This has advantages. Suppose the Web server and the managed node are on a separate machine. The `plugin-cfg.xml` file is automatically propagated to the remote node during node synchronization because the Web server definition is part of the node configuration.

If a federated custom profile is not found, the Plug-ins installation wizard looks for and configures the first federated application server node (application server profile) that it finds. So, the logic is:

1. Look for a federated managed (custom) profile and configure the first one found.
2. If no federated managed profile is found, look for a federated application server profile and configure the first one found.

## Scenario 1. Remote plug-in configuration

The Plug-ins installation wizard does not automatically create a Web server definition within the default distributed profile on a remote machine. The wizard creates the `configureWeb_server_name` script instead.

The Plug-ins installation wizard configures the Web server to use the `plugin-cfg.xml` file that will be maintained on the Web server machine in the `plugins_install_root/config/web_server_name` directory. This file requires periodic propagation. Propagation is copying the current `plugin-cfg.xml` file from the Application Server machine to replace the `plugins_install_root/config/web_server_name/plugin-cfg.xml` file.

After installing the binary plug-in for the local Web server, you do not have to run the script before you can start the application server and the Web server. However, you do not have the benefits of a Web server definition in the application server node until you run the script.

Four configurations qualify for the remote application server scenario:

Profile type	Federation status	Creation of Web server definition?	Web server already defined in Application Server configuration?
Any profile anywhere if you select a remote installation type in the Plug-ins installation wizard	N/A	By script	N/A
No default profile detected	N/A	By script	N/A
Default unfederated stand-alone application server profile with an existing Web server definition	Not federated	By script	Yes
Default deployment manager profile with no managed nodes	N/A	By script	N/A




**Testing the application server without a Web server definition:** The following overview shows the procedure for verifying the temporary `plugins_install_root/config/web_server_name/plugin-cfg.xml` file.

The Web server communicates with the remote Application Server using the temporary `plugin-cfg.xml` file.

If the application server has an HTTP Transport port assignment other than 9080, the test is not successful. Continue to the next section to create the Web server definition on the application server and complete your test of the configuration.




1. Start the Web server with the proper procedure for your Web server.

For example, start the IBM HTTP Server from a command line:

-   `./IHS_install_root/bin/apachectl start`
-  `IHS_install_root\bin\apache`

2. Start the application server on the remote machine.




Change directories to the `install_root/profiles/profile_name/bin` directory and run the **startServer** command:

-   `./WAS_install_root/profiles/default/bin/startServer.sh server1`
-  `WAS_install_root\profiles\default\bin\startServer server1`

3. Point your browser to `http://localhost:9080/snoop` to test the internal HTTP transport provided by the application server. Point your browser to `http://Host_name_of_Web_server_machine/snoop` to test the Web server plug-in.

4. Verify that both Web addresses display the Snoop Servlet - Request/Client Information page.

**Completing the installation by configuring a Web server definition:** The following overview shows the procedure for completing the configuration. The configuration is not complete until the Web server definition exists in the configuration of the Application Server node. The Web server definition is a central element in the regeneration of a valid plug-in configuration file, `plugin-cfg.xml`.

1. Start the deployment manager if you are configuring the deployment manager or a managed node.
2. Federate a remote application server node or custom node now if you are planning to federate the node at some point. If a Web server definition already exists when you federate a node, the definition is lost.
3. Create the Web server definition in the application server. You have two options for a managed node. Use the script option for a deployment manager node without managed nodes.
  - Use the administrative console of the deployment manager to create a Web server definition for a managed node. Click **Servers > Web servers > New** and use the Create new Web server entry wizard to create the Web server definition.
  - Run the script to manually create the Web server definition within the configuration of the Application Server node:
    - a. Copy the script from the `plugins_install_root/bin` directory to the remote `was_install_root/bin` directory.
    - b. Open a command window and run the script:
      -   `./configureWeb_server_name.sh`
      -  `configureWeb_server_name.bat`
4. Open the administrative console of the deployment manager if the node is federated. Wait for node synchronization to occur on the managed node and save the changed configuration that includes the new Web server definition. If the remote node is not federated, open the administrative console of the application server and save the changed configuration.
5. Copy the current plug-in configuration file, `plugin-cfg.xml`, in the `profiles_install_root/profile_name/config/cells/cell_name/nodes/Web_server_name_node/`

`servers/web_server_name/` directory. Paste the file on the Web server machine to replace the temporary `plugins_install_root/config/web_server_name/plugin-cfg.xml` file. The IBM HTTP Server supports automatic propagation. Other Web servers require manual propagation.

6. Start the Web server with the proper procedure for your Web server.
7. Point your browser to `http://localhost:9080/snoop` to test the internal HTTP transport provided by the Application Server. Point your browser to `http://Host_name_of_Web_server_machine/snoop` to test the Web server plug-in.
8. Verify that both Web addresses display the Snoop Servlet - Request/Client Information page.

## Scenario 2. Local distributed plug-in configuration

The Plug-ins installation wizard does not automatically create a Web server definition within a federated application server profile. The wizard creates the `configureWeb_server_name` script instead in the `plugins_install_root/bin` directory.

The Plug-ins installation wizard configures the Web server to use the `plugin-cfg.xml` file that will be created within the application server profile when you run the script. The deployment manager regenerates `plugin-cfg.xml` file in the `install_root/profiles/profile_name/config/cells/cell_name/nodes/node_name/servers/Web_server_name` directory. Regeneration occurs whenever a change occurs in the application server configuration that affects deployed applications on the managed node.

After installing the binary plug-in for the local Web server, you must run the script before you can start the Web server. The Web server has already been configured to use the `plugin-cfg.xml` file in the application server configuration. That file does not exist until you run the `configureWeb_server_name` script.

Four configurations qualify for the local distributed application server scenario:




Profile type	Federation status	Creation of Web server definition?	Web server already defined in Application Server configuration?
Default application server profile	Federated	By script	N/A
Default Custom profile	Not federated	By script	N/A
Default Custom profile	Federated	By script	N/A
Default deployment manager profile with a managed node (non-default distributed profile)	N/A	By script	N/A

The following overview shows the procedure for completing the configuration and verifying the Web server configuration:

1. Start the deployment manager.
2. If you are planning to add an application server node into a deployment manager cell but have not done so yet, federate the node before installing the plug-ins. If the Web server definition exists when you federate the node, the Web server definition is lost when you federate.
3. Create the Web server definition in the application server. You have two options:
  - Use the administrative console of the deployment manager to create a Web server definition for a managed node. Click **Servers > Web servers > New** and use the Create new Web server entry wizard to create the Web server definition.

- Run the script to manually create the Web server definition within the configuration of the deployment manager. Run the script from the *plugins\_install\_root/bin* directory. The script can address the deployment manager on the same machine.




Open a command window to run the appropriate script:

-   `./configureWeb_server_name.sh`
-  `configureWeb_server_name.bat`

If you have enabled security or changed the default JMX connector type, edit the script and include the appropriate parameters.




4. Start the Web server with the proper procedure for your Web server.

For example, start the IBM HTTP Server from a command line:

-   `./IHS_install_root/bin/apachectl start`
-  `IHS_install_root\bin\apache`

5. Start the application server.

Change directories to the *install\_root/profiles/profile\_name/bin* directory and run the **startServer** command:

-   `./WAS_install_root/profile/default/bin/startServer.sh server1`
-  `WAS_install_root\profile\default\bin\startServer server1`

6. Open the administrative console of the deployment manager. Wait for node synchronization to occur and save the changed configuration that includes the new Web server definition.
7. Point your browser to `http://localhost:9080/snoop` to test the internal HTTP transport provided by the application server. Point your browser to `http://Host_name_of_Web_server_machine/snoop` to test the Web server plug-in.
8. Verify that both Web addresses display the Snoop Servlet - Request/Client Information page.

### Scenario 3. Local stand-alone plug-in configuration

The Plug-ins installation wizard creates a Web server definition within the application server profile.

The Plug-ins installation wizard configures the Web server to use the `plugin-cfg.xml` file that is within the application server profile. The stand-alone application server regenerates the *install\_root/profiles/profile\_name/config/cells/cell\_name/nodes/Web\_server\_name\_node/servers/Web\_server\_name/plugin-cfg.xml* file whenever a change occurs in the application server configuration that affects deployed applications.

After installing the binary plug-in for the local Web server, you can start the Application Server and the Web server immediately upon completion of the installation.

Suppose that you create a Web server definition on a stand-alone application server and then federate the node. The Web server definition is not federated into the cell because the Web server definition is defined as a separate node in a stand-alone Application Server. You must recreate the Web server definition on the managed node. See Scenario 2.

Only one configuration qualifies for the local stand-alone application server scenario:

Profile type	Federation status	Automatic creation of Web server definition?	Web server already defined in Application Server configuration?
Application server	Not federated	Yes	No

## Redirection to Scenario 1

An unfederated default standalone application server that has an existing Web server definition is processed as a remote plug-in configuration.

An existing Web server definition on a stand-alone application server causes the Plug-ins installation wizard to follow the remote installation path. A stand-alone application server can have just one Web server definition. Specify the same nick name for the Web server if you want to configure a new Web server.

You can use the `plugin-cfg.xml` file that is within the Web server definition in the configuration of the Application Server. Simply click **Browse** on the appropriate panel in the Plug-ins installation wizard to select the file. This file must exist. Otherwise, the Plug-ins installation wizard displays a warning and prevents you from proceeding until you select an existing file. The Web server is configured to use this existing `plugin-cfg.xml` file.

See Scenario 1 for a description of this type of node.

## Redirection to Scenario 2




A federated default standalone Application Server is processed as a local distributed plug-in configuration. See Scenario 2 for a description of this type of node.

## Overview of the verification procedure

The following overview shows the procedure for verifying the Web server configuration after installing the binary plug-in module:




1. Start the Web server with the proper procedure for your Web server.

For example, start the IBM HTTP Server from a command line:

-   `./IHS_install_root/bin/apachectl start`
-  `IHS_install_root\bin\apache`

2. Start the application server.

Change directories to the `install_root/profiles/profile_name/bin` directory and run the **startServer** command:

-   `./WAS_install_root/profile/default/bin/startServer.sh server1`
-  `WAS_install_root\profile\default\bin\startServer server1`

Open the administrative console and save the changed configuration.

3. Point your browser to `http://localhost:9080/snoop` to test the internal HTTP transport provided by the application server. Point your browser to `http://Host_name_of_Web_server_machine/snoop` to test the Web server plug-in.
4. Verify that both Web addresses display the Snoop Servlet - Request/Client Information page.

## Summary

Three scenarios exist for Web server plug-ins for WebSphere Application Server. Each scenario revolves around a unique location for the plug-in configuration file, `plugin-cfg.xml`. The application server generates the plug-in configuration file. The purpose of the file is to publish the location of all of the application server elements that are relevant to a Web server. Such elements include applications, virtual hosts for serving applications, clusters, and cluster members, for example.



If the Web server cannot get to the file on the application server machine, you must take the file to the Web server. That process is called propagation. Propagation is reserved for the remote plug-in configuration scenario, which is **Scenario 1** in this topic.

In each of the local scenarios, the Web server can get to the `plugin-cfg.xml` file because it is on the same machine as the file. Two local scenarios exist because of two distinct locations for a local `plugin-cfg.xml` file.

The configuration scheme for Version 6 of WebSphere Application Server puts the plug-in configuration file in a Web server definition that is either within a Web server node or a managed node. The type of node is the difference between Scenario 2 and Scenario 3 in this topic. All **Scenario 2** configurations require the Web server definition to exist within a managed application server node. All **Scenario 3** configurations have the Web server definition within its own Web server node.

Limited management options do not let you create or delete the one Web server definition in the administrative console of a stand-alone application server. The inability of a stand-alone application server to create a Web server definition is the basis for the configuration scripts created by the Web server plug-ins for WebSphere Application Server. Without the scripts you could not easily create a Web server definition on a stand-alone application server node.

The location of the `plugin-cfg.xml` file for each configuration described in this topic is shown in the following table:

Table 17. Plug-in configuration file locations

Scenario	Profile type	Location of the <code>plugin-cfg.xml</code> file		
		Plugins_ install_ root	Profiles_ install_ root: within the managed node	Profiles_ install_ root: within the Web server node
1	Any profile anywhere if you select a remote installation type in the Plug-ins installation wizard	X		
	No default profile detected	X		
	Default unfederated (stand-alone) Application Server profile with an existing Web server definition	X		
	Default deployment manager profile with no managed nodes	X		
2	Default application server profile		X	
	Default custom profile		X	
	Default deployment manager profile with a managed node (non-default distributed profile)		X	
3	Default application server profile			X

**Legend:**

**Plugins\_install\_root**

`plugins_install_root/config/  
web_server_name/plugin-cfg.xml`

**Profiles\_install\_root: within the managed node**

`WAS_install_root/profiles/profile_name  
/config/cells/cell_name/nodes/  
node_name_of_AppServer/servers/  
Web_server_name/plugin-cfg.xml`



## Profiles\_install\_root: within the Web server node

```
WAS_install_root/profiles/profile_name  
/config/cells/cell_name/nodes/  
Web_server_name_node/servers/  
Web_server_name/plugin-cfg.xml
```

## Web server configuration

After installing a supported Web server, you must install a binary plug-in module for the Web server. The plug-in module lets the Web server communicate with the application server. The Plug-ins installation wizard installs the Web server plug-in. The wizard configures the Web server. The wizard also creates a Web server definition in the configuration of the application server.

The Plug-ins installation wizard uses five files to configure a plug-in for the Web server you select:

- The **Web server configuration file** on the Web server machine, such as the `httpd.conf` file for IBM HTTP Server.
- The **binary Web server plug-in file** that the Plug-ins installation Wizard installs on the Web server machine.
- The **plug-in configuration file, `plugin-cfg.xml`**, on the application server machine that you propagate (copy) to a Web server machine.
- The **default (temporary) plug-in configuration file, `plugin-cfg.xml`**, on the Web server machine.
- The **configure*Web\_server\_name* script** that you copy from the Web server machine to the application server machine.

See the following descriptions of each file.

### Web server configuration file

The Web server configuration file is installed as part of the Web server. The wizard must reconfigure the configuration file for a supported Web server. Reconfiguration consists of adding directives that identify file locations of two files:

- The binary plug-in file
- The `plugin-cfg.xml` configuration file

### The binary Web server plug-in file

See Web server plug-ins for a description of the binary plug-in module. An example of a binary plug-in module is the `mod_ibm_app_server_http.dll` file for IBM HTTP Server on the Windows platform.

The binary plug-in file does not change. However, the configuration file for the binary plug-in is an XML file. The application server changes the configuration file whenever you change the configuration of deployed applications.

The binary module reads the XML file to adjust settings and to locate deployed applications for the Web server.

### The plug-in configuration file, `plugin-cfg.xml`

The plug-in configuration file is an XML file with settings that you can tune in the administrative console. The file lists all of the applications installed on the Web server definition. The binary module reads the XML file to adjust settings and to locate applications for the Web server.

The stand-alone application server regenerates the `plugin-cfg.xml` file in the `install_root/ profiles/ profile_name/ config/ cells/ cell_name/ nodes/ Web_server_name_node/ servers/ Web_server_name` directory. Regeneration occurs whenever a change occurs in the application server configuration that affects deployed applications.

The deployment manager regenerates the `plugin-cfg.xml` file in the `install_root/profiles/profile_name/config/cells/cell_name/nodes/node_name/servers/Web_server_name` directory whenever a change occurs in application server configuration that affects deployed applications on the managed node.

After regeneration, you must propagate (copy) the file to the Web server machine. The binary plug-in then has access to the most current copy of its configuration file.

The Version 6 Web server plug-in configuration service automatically regenerates the `plugin-cfg.xml` file whenever the configuration changes. The configuration service automatically propagates the `plugin-cfg.xml` file to an IBM HTTP Server machine. You must manually copy the file on other Web servers.

See Web server plug-in configuration service properties settings for more information.

### Default plug-in configuration file, `plugin-cfg.xml`

The Plug-ins installation wizard creates the temporary `plugin-cfg.xml` file in the `plugins_install_root/Plugins/config/Web_server_name` directory. The wizard creates the file for every remote installation scenario. The wizard creates the file at the same time that it installs the binary plug-in module for the Web server.

The default file is a placeholder that you must replace with the `plugin-cfg.xml` file from the Web server definition on the application server. The default file is a replica of the file that the application server creates for a default stand-alone application server that has the samples installed.

Run the `configureWeb_server_name` script from the `was_install_root/bin` directory of the application server machine for a remote installation, or directly from the `plugins_install_root/bin` directory for a local installation. The script creates the Web server definition in the configuration files of the default profile. To configure a different profile than the default, edit the `configureWeb_server_name` script. Use the `-profileName` parameter to identify a different profile than the default.

After the Web server definition is created, the Web server plug-in configuration service within the application server creates the first `plugin-cfg.xml` file in the Web server definition on the application server machine. If you install an application, create a virtual host, or do anything that changes the configuration, you must propagate the updated `plugin-cfg.xml` file from the application server machine to the Web server machine to replace the default file.

### The `configureWeb_server_name` script

The Plug-ins installation wizard creates the `configureWeb_server_name` script on the Web server machine in the `plugins_install_root/bin` directory. If one machine in a remote scenario is running under Linux or UNIX and the other machine is running under Windows, use the script created in the `install_root/Plugins/bin/crossPlatformScripts` directory. The script is created for remote installation scenarios only.

Copy the script from the Web server machine to the `was_install_root/bin` directory on a remote application server machine. You do not have to copy the script on a local installation. Run the script to create a Web server definition in the configuration of the application server.

When using the IBM HTTP Server, configure the IBM HTTP Administration Server also. The IBM HTTP Administration Server works with the administrative console to manage Web server definitions. Also, use the administrative console to update your Web server definition with remote Web server management options. Click **Servers > Web servers > Web\_server** to see configuration options. For example, click **Remote Web server management** to change such properties as:

- Host name

- Administrative port
- User ID
- Password

**Important:** Always open a new command window before running this script. You can avoid a potential problem by using a new command window to run this script.

The problem is a potential conflict between a shell environment variable, the `WAS_USER_SCRIPT` environment variable, and the actual default profile. The script always works against the default profile. However, if the `WAS_USER_SCRIPT` environment variable is set, a conflict arises as the script attempts to work on the profile identified by the variable.

The variable is easy to set accidentally. Issue any command from the `profiles_install_root/profile_name/bin` directory of any profile and the variable is set to that profile.

If you have more than one profile on your system, the potential exists that the default profile and the profile identified by the variable are different profiles. If so, a conflict occurs and the script might not create the Web server definition in the correct profile, or might not create the Web server definition at all.

Reset the variable in either of two ways:

- Close the command window where the variable is set and open a new one.
- Change directories to the `profiles_install_root/profile_name/bin` directory of the default profile and source the `setupCmdLine.sh` script:

```

- > Linux source install_root/bin/setupCmdLine.sh
- > UNIX . install_root/bin/setupCmdLine.sh (Source the command with a space
  between the period and the installation root directory.)
- > Windows install_root\bin\setupCmdLine.bat

```

If a Web server definition already exists for a stand-alone application server, running the script does not add a new Web server definition. Each stand-alone application server can have only one Web server definition.

You cannot use the administrative console of a stand-alone application server to add or delete a Web server definition. However, you can do both tasks using the administrative scripting interface:

- Add a Web server definition through the `wsadmin` facility using the `configureWeb_server_name` script. The script uses a Java Command Language (Jacl) script named `configureWebserverDefintion.jacl` to create and configure the Web server definition.
- Delete a Web server definition using `wsadmin` commands. The Web server is named `webserver1` in the following example:

```

$AdminTask deleteServer { -serverName webserver1 -nodeName webserver1_node }
$AdminTask removeUnmanagedNode { -nodeName webserver1_node }
$AdminConfig save

```

A managed node, on the other hand, can have multiple Web server definitions. The script creates a new Web server definition unless the Web server name is the same.

### Replacing the default plug-in configuration file with the file from the Web server definition (propagation)

The default file uses the following fixed parameter values that might not match the parameter values in the actual file on the application server. The default file is a placeholder only.

The file cannot reflect changes that occur in the application server configuration. The file also cannot reflect non-default values that might be in effect on the application server.

The application server must have the following values in the actual `plugin-cfg.xml` file. If so, the default file can successfully configure the binary plug-in module. Then, the plug-in module can successfully communicate with the Web server and the application server.

Suppose that the application server does not have the following values in the actual `plugin-cfg.xml` file. In that case, the default file configures the binary plug-in module incorrectly. The plug-in module can always communicate with the Web server. But with an improper configuration file, the plug-in module cannot communicate successfully with the application server.

### **Fixed parameter values in the temporary plug-in configuration file**

#### **Virtual host name**

`default_host`

The `default_host` value is the default virtual host. This virtual host is configured to serve the `DefaultApplication` and the `Sample` applications. This value is probably the same as the value in the real `plugin-cfg.xml` file. However, suppose that you create another virtual host for serving applications and install the `DefaultApplication` on it. If so, the actual `plugin-cfg.xml` file is regenerated. The Web server cannot access the `DefaultApplication`. (The application includes the `snoop` servlet and the `hitcount` servlet.)

To access applications on the new virtual host, propagate the real `plugin-cfg.xml` file. Propagation is copying the updated file from the application server machine to the Web server machine.

#### **HTTP transport port**

`9080`

The `9080` value is the default value for the HTTP transport port for the `default_host` virtual host. This value is probably the same as the value in the updated file. However, this value changes for every profile on the application server machine. The HTTP transport port value must be unique for every application server.

To communicate over a different port, propagate the real `plugin-cfg.xml` file.

#### **Web server listening port**

`80`

The `80` value is the default value for the port that controls communication with the Web server. However, each application server profile must have a unique port value to communicate to a Web server. The actual port value might be `81` or another number.

To communicate over a different port, propagate the real `plugin-cfg.xml` file.

#### **HTTPS transport port**

`9443`

The `9443` value is the default value for the HTTPS (secure) transport port for the `default_host` virtual host. This value is probably the same as the value in the updated file. However, this value changes for every profile on the application server machine. The HTTPS transport port value must be unique for every application server.

To communicate over a different secure port, propagate the real `plugin-cfg.xml` file.

#### **Applications installed on the server1 application server**

All of the default servlets and applications are included in the default file, including the `WSsamples` application and the `SamplesGallery` application.

The default file lists all of the default applications and samples. The list can be inaccurate. If you performed a custom installation and did not install the samples, for example, the list is inaccurate.

To serve an application that you developed with the Web server, propagate the real `plugin-cfg.xml` file.

## Configuring a Web server and an application server on separate machines (remote)

This topic describes installing a Web server plug-in that WebSphere Application Server provides to communicate with a particular brand of Web server. This procedure describes installing the Web server and its Web server plug-in for WebSphere Application Server on one machine and the Application Server on another.

If the WebSphere Application Server product family supports a particular brand of Web server, such as IBM HTTP Server or Microsoft Internet Information Services (IIS), then your WebSphere Application Server product provides a binary plug-in for the Web server that you must install.

If the WebSphere Application Server product family does not provide a binary plug-in for a particular brand of Web server, then the Web server is not supported. The purpose of the binary plug-in is to provide the communication protocol between the Web server and the application server.

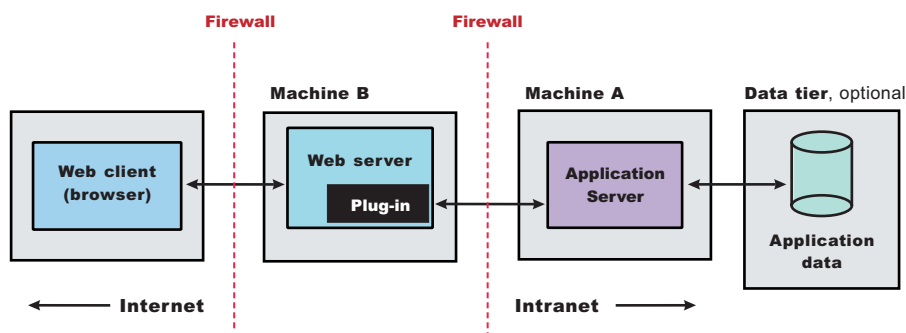
Suppose that you create a new profile. Suppose also that you want to use a Web server. You must install a new Web server for the new profile and use the Plug-ins installation wizard to install the binary plug-in module and to configure both the Web server and the application server.

If the Web server is not already installed, you can still install the plug-ins for future use. If the WebSphere Application Server product is not installed, you can still install the plug-ins. However, it is recommended that you install the Web server and the WebSphere Application Server product before installing the plug-ins for the supported Web server.

The Plug-ins installation wizard installs the plug-in module, configures the Web server for communicating with the application server, and creates a Web server configuration definition in the application server, if possible.

This procedure configures the application server profile that is the default profile on the machine. A one-to-one relationship exists between a Web server and the application server.

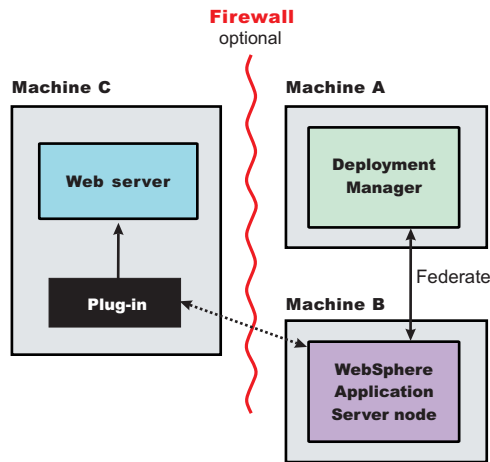
This topic describes how to create the following topology:



### Attention:

If you are planning to add the application server node into a deployment manager cell but have not done so yet, start the deployment manager and federate the node before installing the plug-in. You cannot add an application server with a Web server definition into the deployment manager cell.

The following topology is considered a remote topology because the Web server is on a separate machine. The diagram shows a typical remote topology for a distributed environment:



This topic describes the installation of a Web server on one machine and the application server on a separate machine. In this situation, the Plug-ins installation wizard on one machine cannot create the Web server definition in the application server configuration on the other machine.

In such a case, the Plug-ins installation wizard creates a script on the Web server machine that you can copy to the application server machine. Run the script on the application server machine to create the Web server configuration definition within the application server configuration.

Perform the following procedure to install the plug-in and configure both the Web server and the application server.

1. Log on as root on a Linux or UNIX operating system, or as a member of the administrator group on a Windows system.

Linux

UNIX

In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```

Windows

Some steps of the installation on a Windows platform require the administrator group user to have the following advanced user rights:

- *Act as part of the operating system*
- *Log on as a service*

For example, on some Windows systems, click **Administrative Tools > Local Security Policy > User Rights Assignments** to see the advanced options. See your Windows documentation for more information.






Windows

The installation wizard grants your Windows user ID the advanced user rights, if the user ID belongs to the administrator group. The silent installation does not grant these rights. If you create a new user ID on a Windows platform to perform a silent installation, you must restart the system to activate the proper authorizations for the user ID before you can perform a successful silent installation.

Windows

When installing the WebSphere Application Server as a Windows service, do not use a user ID that contains spaces. A user ID with spaces cannot be validated. Such a user ID is not allowed to continue the installation. To work around this problem, install with a user ID that does not contain spaces, or do not choose to install Windows services.



2. Install WebSphere Application Server Network Deployment on Machine A.  
See Chapter 9, “Installing the product and additional software,” on page 107.
3. Install the IBM HTTP Server or another supported Web server on Machine B.  
See “Installing IBM HTTP Server” on page 134 or refer to the product documentation for your Web server for more information.
4. Launch the Plug-ins installation wizard on the machine with the Web server.  
Select the Plug-ins installation wizard from the launchpad or change directories to the `plugin` directory on the product disc or in the downloaded installation image and issue the **install** command.  
The installation of the plug-ins tolerates installing as a non-root user. However, the automated installation of the required Global Services Kit (GSKit) fails if you install as non-root.  
If you decide to install the plug-in as a non-root user, log on as root after the entire procedure is complete. Reinstall the GSKit from the GSKit directory on the product disc.
5. Clear the check box for the roadmap or select the check box to view the roadmap, then click **Next**.  
If you are unsure of which installation scenario to follow, display the roadmap instead. Print and keep the roadmap as a handy overview of the installation steps.  
Press **Ctrl-P** to print the roadmap if the Web browser navigation controls and the menu bar are not present on the browser window that displays the Plug-ins roadmap. Press **Ctrl-W** to close the browser window if the navigation controls and the menu bar do not display. Or close the browser window with the window control in the title bar.
6. Read the license agreement and accept the agreement if you agree to its terms. Click **Next** when you are finished.
7. If your system does not pass the prerequisites check, stop the installation, correct any problems, and restart the installation. If your system passes the prerequisites check, click **Next**.  
Look for the appropriate log file for information about missing prerequisites:
  - If you stop the installation, see the `temporaryPluginInstallLog.txt` file in the temporary directory of the user who installed the plug-ins. For example, the `/tmp/temporaryPluginInstallLog.txt` file might exist if the root user installed the plug-ins on a Linux or UNIX system.
  - If you continue the installation in spite of warnings about missing prerequisites, see the `plugins_install_root/logs/install/log.txt` file after the installation is complete.
 See Appendix B, “Troubleshooting installation,” on page 337 for more information about log files.
8. Select the type of Web server that you are configuring and click **Next**.  
The Plug-ins installation wizard panel prompts you to identify the *Web servers* to configure. Actually you can select only one Web server each time you run the Plug-ins installation wizard.  
Stop any Web server while you are configuring it. A step later in the procedures directs you to start the Web server as you begin the snoop servlet test.  
If you select the Web server identification option labeled **None**, the Web server installs the binary plug-ins but does not configure the Web server. A known problem in the wizard panel causes the English word *None* to appear in translated versions of the wizard. However, the selectable option is functional in every locale in spite of the missing translation.
9. Select **Web server machine (remote)** and click **Next**.
10. Accept the default location for the installation root directory for the plug-ins. Click **Next**.  
You can type another new directory or click **Browse** to select an empty directory. The fully qualified path identifies the plug-ins installation root directory.  
The default location is:
  -  `/usr/IBM/WebSphere/Plugins`
  -    `/opt/IBM/WebSphere/Plugins`
  -  `C:\Program Files\IBM\WebSphere\Plugins`

A possibility exists that the Web server might run on a platform that WebSphere Application Server does not support.

11. Click **Browse** to select the configuration file for your Web server, verify that the Web server port is correct, and then click **Next** when you are finished.

Select the file and not just the directory of the file. Some Web servers have two configuration files and require you to browse for each file.

The following list shows configuration files for supported Web servers:

#### **Apache HTTP Server**

*apache\_install\_root/config/httpd.conf*

#### **Domino Web Server**

names.nsf and Notes.jar

The wizard prompts for the notes.jar file. The actual name is Notes.jar.

The Plug-ins installation wizard verifies that the files exist but the wizard does not validate either file.

#### **IBM HTTP Server**

*ihs\_install\_root/conf/httpd.conf*

#### **Microsoft Internet Information Services (IIS)**

The Plug-ins installation wizard can determine the correct files to edit.

#### **Sun ONE Web Server 6.0 or Sun Java System Web Server, V6.1**

obj.conf and magnus.conf

The wizard displays a naming panel for the nickname of the Web server definition.

12. Specify a nickname for the Web server. Click **Next** when you are finished.

The wizard uses the value to name configuration folders in the plug-ins installation root directory. The wizard also uses the name in the configuration script for the application server to name the Web server definition.

If the application server profile already has a Web server definition, specify a nickname that matches the name of the existing Web server definition. Using the same name configures the Web server to use the Web server definition on the application server.

13. Accept the default location for the plugin-cfg.xml file that the wizard creates on the Web server machine, then click **Next**.

You can type a change to the value or click **Browse** to select a file in another location. If you do accept the default location, the plugin-cfg.xml file must exist.

14. Identify the host name or IP address of Machine A, which is the application server machine, then click **Next**.

15. Examine the summary panel. Click **Next** when you are finished.

The panel notifies you that you have manual steps to perform to complete the installation and configuration. The type of Web server, the nickname of the Web server, and the location of the plugin-cfg.xml file displays on the panel.

The Plug-ins installation wizard creates the *configureWeb\_server\_name* script in the *plug-ins\_installation\_root/bin/* directory on Machine B.

The Plug-ins installation wizard also creates the plugin-cfg.xml file in the *plug-ins\_installation\_root/config/Web\_server\_name* directory.

The Web server reads the plugin-cfg.xml file to determine the applications that the application server on Machine A can serve to the Web server on Machine B. Whenever the configuration changes, the application server regenerates the file. When regeneration occurs, propagate, or copy the actual plugin-cfg.xml file from the application server machine to the Web server machine. You can automatically propagate the file to the IBM HTTP Server product.



16. Click **Next** on the pre-installation summary panel to begin the installation or click **Back** to change any characteristics of the installation.

The panel specifies the plug-ins installation root directory, the Web server plug-ins feature, and the disk size of the code that installs when you click **Next**.

17. After the wizard installs the code and creates the uninstaller program, examine the post-installation summary panel. Click **Next** when you are finished to display the Plug-ins installation roadmap.

The Plug-ins installation wizard installs the binary plug-in module. On a Linux system, for example, the installation creates the `/opt/IBM/WebSphere/Plugins` directory. The `/opt/IBM/WebSphere/Plugins/config/Web_server_name` directory contains the `plugin-cfg.xml` file. The wizard displays the name and location of the configuration script and the `plugin-cfg.xml` file. The wizard also displays the type of Web server that is configured and the nickname of the Web server.

If a problem occurs and the installation is unsuccessful, examine the logs in the `plug-ins_install_root/logs` directory. Correct any problems and reinstall.

18. Close the road map and click **Finish** to exit the wizard.

Log files from the installation are in the `plug-ins_install_root/logs/install` directory.

19. Copy the `configureWeb_server_name.sh` script on Linux and UNIX systems or the `configureWeb_server_name.bat` script on Windows systems from Machine B to the `was_install_root/bin` directory on Machine A.




For example, on a Linux system with an IBM HTTP Server named `webserver1` in the default location, copy `/opt/IBM/WebServer/Plugins/bin/configurewebserver1.sh` from Machine B to the `/opt/IBM/WebSphere/AppServer/bin` directory on Machine A.

If one platform is a Linux or UNIX system and the other is a Windows platform, copy the script from the `crossPlatformScripts` directory. For example:

-   `plug-ins_install_root/bin/configureWeb_server_name.sh`
-  `plug-ins_install_root/bin/crossPlatformScripts/configureWeb_server_name.bat`

20. Start the application server on Machine A.

Use the **startServer** command, for example:

-   `profiles_install_root/profile_name/bin/startServer.sh server1`
-  `profiles_install_root\profile_name\bin\startServer server1`



21. Run the script on Machine A to create the Web server definition in the configuration for the application server.

Open a command window to run the script that you copied to Machine A.


As soon as the Web server definition is created, the application server creates a `plugin-cfg.xml` file for the Web server. For example, the file on a Linux system might have this file path: `/opt / IBM / WebSphere / AppServer / profiles / default / config / cells / cellname / nodes / webserver1_node / servers / webserver1 / plugin-cfg.xml`.

22. From the administrative console of the deployment manager, click **System administration > Save Changes to Master Repository > Synchronize changes with Nodes > Save**.

23.   **Domino Web Server only:** Set the `WAS_PLUGIN_CONFIG_FILE` environment variable.

  To configure Domino, you must set the `WAS_PLUGIN_CONFIG_FILE` environment variable. On Linux and UNIX-based platforms, sourcing a script to the parent shell allows child processes to inherit the exported variables. On Windows systems, run the script as you would run any other command. Sourcing is automatic on Windows systems.

- a. Open a command window.
- b. Change directories to the plug-ins installation root directory.

- c. Issue the appropriate command for the `plug-ins_install_root/bin/setupPluginCfg.sh` script:
-  `. plug-ins_install_root/bin/setupPluginCfg.sh` (Notice the space between the period and the installation root directory.)

-  `source plug-ins_install_root/bin/setupPluginCfg.sh`

The script is also in the `lotus_install_root/notesdata` directory on Linux and UNIX systems.




Issue the appropriate command for the script before starting the Domino Web Server.

24. Start the Snoop servlet to verify the ability of the Web server to retrieve an application from the Application Server.

Test your environment by starting your Application Server, your Web server, and using the snoop servlet with an IP address.

- a. Start the Application Server. In a Network Deployment environment, the Snoop servlet is available in the cell only if you included the DefaultApplication when adding the Application Server to the cell. The `-includeapps` option for the `addNode` command migrates the DefaultApplication to the cell. If the application is not present, skip this step.

Change directories to the `install_root/profiles/profile_name/bin` directory and run the `startServer` command:

-   `./startServer.sh server1`
-  `startServer server1`

- b. Start the IBM HTTP Server or the Web server that you are using.

Use a command window to change the directory to the IBM HTTP Server installed image, or to the installed image of your Web server. Issue the appropriate command to start the Web server, such as these commands for IBM HTTP Server:

**To start the IBM HTTP Server from the command line:**

Access the `apache` and `apachectl` commands in the `IBMHttpServer/bin` directory.

-   `./apachectl start`
-  `apache`

- c. Point your browser to `http://localhost:9080/snoop` to test the internal HTTP transport provided by the Application Server. Point your browser to `http://Host_name_of_Web_server_machine/snoop` to test the Web server plug-in.

The HTTP Transport port is 9080 by default and must be unique for every profile. The port is associated with a virtual host named `default_host`, which is configured to host the installed DefaultApplication and any installed Samples. The snoop servlet is part of the DefaultApplication. Change the port to match your actual HTTP Transport port.

- d. Verify that snoop is running.

Either Web address should display the Snoop Servlet - Request/Client Information page.

- e. **Remote IBM HTTP Server only:** Verify that the automatic propagation function can work on a remote IBM HTTP Server by using the following steps. This procedure is not necessary for local Web servers.
- 1) Run the `htpasswd` command on the Web server machine to create the administrative user ID and password information.
  - 2) Use the administrative console of the deployment manager or the Application Server to enter the User ID and password information that you created for the administrative user of IBM HTTP Server. Go to **Servers > Web server > Web\_server\_definition > Remote Web server administration**.

25. Regenerate the `plugin-cfg.xml` file on Machine A using the administrative console. Click **Servers > Web server**. Select the Web server, then click **Generate Plug-in**.

During the installation of the plug-ins, the temporary `plugin-cfg.xml` file is installed on Machine B in the `plugins_install_root/config/web_server_name` directory. The Web server plug-in configuration service regenerates the `plugin-cfg.xml` file automatically. To use the real `plugin-cfg.xml` file from the application server, propagate the `plugin-cfg.xml` file as described in the next step.

This step shows you how to regenerate the `plugin-cfg.xml` file. WebSphere Application Server products are configured to automatically regenerate the file each time a significant event occurs. Such events include installing applications on the application server and the Web server, for example. Creating a new virtual host is another such event.

26. Propagate the `plugin-cfg.xml` file from the application server to the Web server using the administrative console. Click **Servers > Web server**. Select the Web server, then click **Propagate Plug-in**. Web servers other than IBM HTTP Server require manual propagation.

The Web server plug-in configuration service propagates the `plugin-cfg.xml` file automatically for IBM HTTP Server 6.0 only. For all other Web servers, propagate the plug-in configuration file by manually copying the `plugin-cfg.xml` file from the `profiles_install_root/config/cells/cell_name/nodes/node_name/servers/web_server_name` directory on Machine A to the `plugins_install_root/config/web_server_name` directory on Machine B.

This procedure results in the installation of the Web server plug-ins for WebSphere Application Server on a Web server machine. The Plug-ins installation wizard also configures the Web server to support an application server on a separate machine.

The installation of the binary plug-in modules results in the creation of the `Plugins` directory and several subdirectories. The following directories are among those created on a Linux system, for example:

- `/opt/IBM/WebSphere/Plugins/_uninstPlugin/_jvmForPlugin` contains the WebSphere Application Server SDK, Java technology edition used to uninstall the product
- `/opt/IBM/WebSphere/Plugins/_uninstPlugin` contains the uninstaller program
- `/opt/IBM/WebSphere/Plugins/bin` contains the binary plug-ins for all supported Web servers
- `/opt/IBM/WebSphere/Plugins/logs` contains log files
- `/opt/IBM/WebSphere/Plugins/properties` contains version information
- `/opt/IBM/WebSphere/Plugins/roadmap` contains the roadmap for the Plug-ins installation wizard

See “Selecting a Web server topology diagram and roadmap” on page 139 for an overview of the installation procedure.

See “Web server configuration” on page 153 for more information about the files involved in configuring a Web server.

See “Configuration behavior of the Plug-ins installation wizard” on page 145 for information about the location of the plug-in configuration file.

See “Editing Web server configuration files” on page 203 for information about how the Plug-ins installation wizard configures supported Web servers.

## Configuring multiple Web servers and remote stand-alone Application Servers

This topic describes installing a Web server plug-in that WebSphere Application Server provides to communicate with a particular brand of Web server. This procedure describes installing multiple Web servers and their Web server plug-ins for WebSphere Application Server on one machine and multiple application servers on another.

If the WebSphere Application Server product family supports a particular brand of Web server, such as IBM HTTP Server or Microsoft Internet Information Services (IIS), then your WebSphere Application Server product provides a binary plug-in for the Web server that you must install.

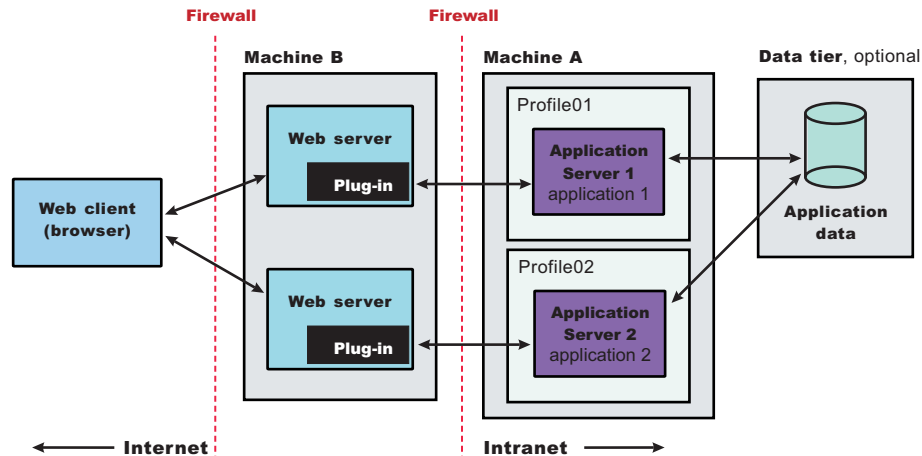
If the WebSphere Application Server product family does not provide a binary plug-in for a particular brand of Web server, then the Web server is not supported. The purpose of the binary plug-in is to provide the communication protocol between the Web server and the application server.

Suppose that you create a new profile. Suppose also that you want to use a Web server. You must install a new Web server for the new profile and use the Plug-ins installation wizard to install the binary plug-in module and to configure both the Web server and the application server.

If the Web server is not already installed, you can still install the plug-ins for future use. If the WebSphere Application Server product is not installed, you can still install the plug-ins. However, it is recommended that you install the Web server and the WebSphere Application Server product before installing the plug-ins for the supported Web server.

The Plug-ins installation wizard installs the plug-in module, configures the Web server for communicating with the application server, and creates a Web server configuration definition in the application server, if possible.

This topic describes how to create the following topology:



Perform the following procedure to install the plug-ins and configure both Web servers and both application servers.

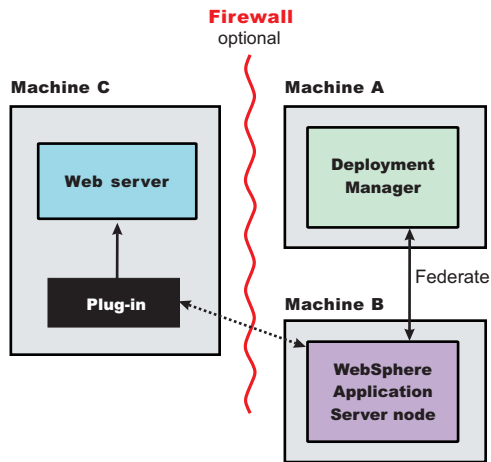
This topology lets each profile have unique applications, configuration settings, data, and log files, while sharing the same set of system files. Creating multiple profiles creates multiple application server environments that you can then dedicate to different purposes.

For example, each application server on a Web site can serve a different application. In another example, each application server can be a separate test environment that you assign to a programmer or a development team.

#### Attention:

If you are planning to add the application server node into a deployment manager cell but have not done so yet, start the deployment manager and federate the node before installing the plug-in. You cannot add an application server with a Web server definition into the deployment manager cell.

The following topology is considered a remote topology because the Web server is on a separate machine. The diagram shows a typical remote topology for a distributed environment:



A deployment manager by itself is also considered a remote scenario if the deployment manager has no managed nodes. Although multiple application servers are not shown in the preceding diagram, Machine B could have more than one application server profile.

1. Log on as root on a Linux or UNIX operating system, or as a member of the administrator group on a Windows system.

Linux

UNIX

In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```

Windows

Some steps of the installation on a Windows platform require the administrator group user to have the following advanced user rights:

- *Act as part of the operating system*
- *Log on as a service*

For example, on some Windows systems, click **Administrative Tools > Local Security Policy > User Rights Assignments** to see the advanced options. See your Windows documentation for more information.

Windows

The installation wizard grants your Windows user ID the advanced user rights, if the user ID belongs to the administrator group. The silent installation does not grant these rights. If you create a new user ID on a Windows platform to perform a silent installation, you must restart the system to activate the proper authorizations for the user ID before you can perform a successful silent installation.

Windows

When installing the WebSphere Application Server as a Windows service, do not use a user ID that contains spaces. A user ID with spaces cannot be validated. Such a user ID is not allowed to continue the installation. To work around this problem, install with a user ID that does not contain spaces, or do not choose to install Windows services.

2. Install WebSphere Application Server Network Deployment on Machine A.  
See Chapter 9, “Installing the product and additional software,” on page 107.
3. Create the first application server profile using the Profile creation wizard on Machine A.
4. Install the IBM HTTP Server or another supported Web server on Machine B.  
See “Installing IBM HTTP Server” on page 134 or refer to the product documentation for your Web server for more information.

5. Launch the Plug-ins installation wizard on the machine with the Web server.  
Select the Plug-ins installation wizard from the launchpad or change directories to the `plugin` directory on the product disc or in the downloaded installation image and issue the **install** command.

The installation of the plug-ins tolerates installing as a non-root user. However, the automated installation of the required Global Services Kit (GSKit) fails if you install as non-root.

If you decide to install the plug-in as a non-root user, log on as root after the entire procedure is complete. Reinstall the GSKit from the GSKit directory on the product disc.

6. Clear the check box for the roadmap or select the check box to view the roadmap, then click **Next**.  
If you are unsure of which installation scenario to follow, display the roadmap instead. Print and keep the roadmap as a handy overview of the installation steps.

Press **Ctrl-P** to print the roadmap if the Web browser navigation controls and the menu bar are not present on the browser window that displays the Plug-ins roadmap. Press **Ctrl-W** to close the browser window if the navigation controls and the menu bar do not display. Or close the browser window with the window control in the title bar.

7. Read the license agreement and accept the agreement if you agree to its terms. Click **Next** when you are finished.
8. If your system does not pass the prerequisites check, stop the installation, correct any problems, and restart the installation. If your system passes the prerequisites check, click **Next**.

Look for the appropriate log file for information about missing prerequisites:

- If you stop the installation, see the `temporaryPluginInstallLog.txt` file in the temporary directory of the user who installed the plug-ins. For example, the `/tmp/temporaryPluginInstallLog.txt` file might exist if the root user installed the plug-ins on a Linux or UNIX system.
- If you continue the installation in spite of warnings about missing prerequisites, see the `plugins_install_root/logs/install/log.txt` file after the installation is complete.

See Appendix B, “Troubleshooting installation,” on page 337 for more information about log files.

9. Select the type of Web server that you are configuring and click **Next**.

The Plug-ins installation wizard panel prompts you to identify the *Web servers* to configure. Actually you can select only one Web server each time you run the Plug-ins installation wizard.

Stop any Web server while you are configuring it. A step later in the procedures directs you to start the Web server as you begin the snoop servlet test.






If you select the Web server identification option labeled **None**, the Web server installs the binary plug-ins but does not configure the Web server. A known problem in the wizard panel causes the English word *None* to appear in translated versions of the wizard. However, the selectable option is functional in every locale in spite of the missing translation.

10. Select **Web server machine (remote)** and click **Next**.

11. Accept the default location for the installation root directory for the plug-ins. Click **Next**.

You can type another new directory or click **Browse** to select an empty directory. The fully qualified path identifies the plug-ins installation root directory.

The default location is:

-  `/usr/IBM/WebSphere/Plugins`
-    `/opt/IBM/WebSphere/Plugins`
-  `C:\Program Files\IBM\WebSphere\Plugins`

A possibility exists that the Web server might run on a platform that WebSphere Application Server does not support.

12. Click **Browse** to select the configuration file for your Web server, verify that the Web server port is correct, and then click **Next** when you are finished.

Select the file and not just the directory of the file. Some Web servers have two configuration files and require you to browse for each file.



The following list shows configuration files for supported Web servers:

### Apache HTTP Server

*apache\_install\_root/config/httpd.conf*

### Domino Web Server

*names.nsf* and *Notes.jar*

The wizard prompts for the *notes.jar* file. The actual name is *Notes.jar*.

The Plug-ins installation wizard verifies that the files exist but the wizard does not validate either file.

### IBM HTTP Server

*ihs\_install\_root/conf/httpd.conf*

### Microsoft Internet Information Services (IIS)

The Plug-ins installation wizard can determine the correct files to edit.

### Sun ONE Web Server 6.0 or Sun Java System Web Server, V6.1

*obj.conf* and *magnus.conf*

The wizard displays a naming panel for the nickname of the Web server definition.

13. Specify a nickname for the Web server. Click **Next** when you are finished.

The wizard uses the value to name configuration folders in the plug-ins installation root directory. The wizard also uses the name in the configuration script for the application server to name the Web server definition.

If the application server profile already has a Web server definition, specify a nickname that matches the name of the existing Web server definition. Using the same name configures the Web server to use the Web server definition on the application server.
14. Accept the default location for the *plugin-cfg.xml* file that the wizard creates on the Web server machine, then click **Next**.

You can type a change to the value or click **Browse** to select a file in another location. If you do accept the default location, the *plugin-cfg.xml* file must exist.
15. Identify the host name or IP address of Machine A, which is the application server machine, then click **Next**.
16. Examine the summary panel. Click **Next** when you are finished.

The panel notifies you that you have manual steps to perform to complete the installation and configuration. The type of Web server, the nickname of the Web server, and the location of the *plugin-cfg.xml* file displays on the panel.

The Plug-ins installation wizard creates the *configureWeb\_server\_name* script in the *plug-ins\_installation\_root/bin/* directory on Machine B.

The Plug-ins installation wizard also creates the *plugin-cfg.xml* file in the *plug-ins\_installation\_root/config/Web\_server\_name* directory.

The Web server reads the *plugin-cfg.xml* file to determine the applications that the application server on Machine A can serve to the Web server on Machine B. Whenever the configuration changes, the application server regenerates the file. When regeneration occurs, propagate, or copy the actual *plugin-cfg.xml* file from the application server machine to the Web server machine. You can automatically propagate the file to the IBM HTTP Server product.
17. Click **Next** on the pre-installation summary panel to begin the installation or click **Back** to change any characteristics of the installation.

The panel specifies the plug-ins installation root directory, the Web server plug-ins feature, and the disk size of the code that installs when you click **Next**.
18. After the wizard installs the code and creates the uninstaller program, examine the post-installation summary panel. Click **Next** when you are finished to display the Plug-ins installation roadmap.

The Plug-ins installation wizard installs the binary plug-in module. On a Linux system, for example, the installation creates the `/opt/IBM/WebSphere/Plugins` directory. The `/opt/IBM/WebSphere/Plugins/config/Web_server_name` directory contains the `plugin-cfg.xml` file. The wizard displays the name and location of the configuration script and the `plugin-cfg.xml` file. The wizard also displays the type of Web server that is configured and the nickname of the Web server.

If a problem occurs and the installation is unsuccessful, examine the logs in the `plug-ins_install_root/logs` directory. Correct any problems and reinstall.

19. Close the road map and click **Finish** to exit the wizard.

Log files from the installation are in the `plug-ins_install_root/logs/install` directory.

20. Copy the `configureWeb_server_name.sh` script on Linux and UNIX systems or the `configureWeb_server_name.bat` script on Windows systems from Machine B to the `was_install_root/bin` directory on Machine A.




For example, on a Linux system with an IBM HTTP Server named `webserver1` in the default location, copy `/opt/IBM/WebSphere/Plugins/bin/configurewebserver1.sh` from Machine B to the `/opt/IBM/WebSphere/AppServer/bin` directory on Machine A.

If one platform is a Linux or UNIX system and the other is a Windows platform, copy the script from the `crossPlatformScripts` directory. For example:

-   `plug-ins_install_root/bin/configureWeb_server_name.sh`
-  `plug-ins_install_root/bin/crossPlatformScripts/configureWeb_server_name.bat`

21. Start the application server on Machine A.

Use the **startServer** command, for example:



-   `profiles_install_root/profile_name/bin/startServer.sh server1`
-  `profiles_install_root\profile_name\bin\startServer server1`



22. Run the script on Machine A to create the Web server definition in the configuration for the application server.

Open a command window to run the script that you copied to Machine A.



As soon as the Web server definition is created, the application server creates a `plugin-cfg.xml` file for the Web server. For example, the file on a Linux system might have this file path: `/opt / IBM / WebSphere / AppServer / profiles / default / config / cells / cellname / nodes / webserver1_node / servers / webserver1 / plugin-cfg.xml`.

23. From the administrative console of the deployment manager, click **System administration > Save Changes to Master Repository > Synchronize changes with Nodes > Save**.

24.   **Domino Web Server only:** Set the `WAS_PLUGIN_CONFIG_FILE` environment variable.

  To configure Domino, you must set the `WAS_PLUGIN_CONFIG_FILE` environment variable. On Linux and UNIX-based platforms, sourcing a script to the parent shell allows child processes to inherit the exported variables. On Windows systems, run the script as you would run any other command. Sourcing is automatic on Windows systems.

- a. Open a command window.
- b. Change directories to the plug-ins installation root directory.
- c. Issue the appropriate command for the `plug-ins_install_root/bin/setupPluginCfg.sh` script:

-  `. plug-ins_install_root/bin/setupPluginCfg.sh` (Notice the space between the period and the installation root directory.)
-  `source plug-ins_install_root/bin/setupPluginCfg.sh`

The script is also in the `lotus_install_root/notesdata` directory on Linux and UNIX systems.






Issue the appropriate command for the script before starting the Domino Web Server.

25. Start the Snoop servlet to verify the ability of the Web server to retrieve an application from the Application Server.

Test your environment by starting your Application Server, your Web server, and using the snoop servlet with an IP address.

- a. Start the Application Server. In a Network Deployment environment, the Snoop servlet is available in the cell only if you included the DefaultApplication when adding the Application Server to the cell. The `-includeapps` option for the **addNode** command migrates the DefaultApplication to the cell. If the application is not present, skip this step.

Change directories to the `install_root/profiles/profile_name/bin` directory and run the **startServer** command:

-   `./startServer.sh server1`
-  `startServer server1`

- b. Start the IBM HTTP Server or the Web server that you are using.

Use a command window to change the directory to the IBM HTTP Server installed image, or to the installed image of your Web server. Issue the appropriate command to start the Web server, such as these commands for IBM HTTP Server:

**To start the IBM HTTP Server from the command line:**

Access the **apache** and **apachectl** commands in the `IBMHttpServer/bin` directory.

-   `./apachectl start`
-  `apache`

- c. Point your browser to `http://localhost:9080/snoop` to test the internal HTTP transport provided by the Application Server. Point your browser to `http://Host_name_of_Web_server_machine/snoop` to test the Web server plug-in.

The HTTP Transport port is 9080 by default and must be unique for every profile. The port is associated with a virtual host named `default_host`, which is configured to host the installed DefaultApplication and any installed Samples. The snoop servlet is part of the DefaultApplication. Change the port to match your actual HTTP Transport port.

- d. Verify that snoop is running.

Either Web address should display the Snoop Servlet - Request/Client Information page.

- e. **Remote IBM HTTP Server only:** Verify that the automatic propagation function can work on a remote IBM HTTP Server by using the following steps. This procedure is not necessary for local Web servers.
  - 1) Run the **htpasswd** command on the Web server machine to create the administrative user ID and password information.
  - 2) Use the administrative console of the deployment manager or the Application Server to enter the User ID and password information that you created for the administrative user of IBM HTTP Server. Go to **Servers > Web server > Web\_server\_definition > Remote Web server administration**.

26. Regenerate the `plugin-cfg.xml` file on Machine A using the administrative console. Click **Servers > Web server**. Select the Web server, then click **Generate Plug-in**.

During the installation of the plug-ins, the temporary `plugin-cfg.xml` file is installed on Machine B in the `plugins_install_root/config/web_server_name` directory. The Web server plug-in configuration service regenerates the `plugin-cfg.xml` file automatically. To use the real `plugin-cfg.xml` file from the application server, propagate the `plugin-cfg.xml` file as described in the next step.

This step shows you how to regenerate the `plugin-cfg.xml` file. WebSphere Application Server products are configured to automatically regenerate the file each time a significant event occurs. Such events include installing applications on the application server and the Web server, for example. Creating a new virtual host is another such event.

27. Propagate the `plugin-cfg.xml` file from the application server to the Web server using the administrative console. Click **Servers > Web server**. Select the Web server, then click **Propagate Plug-in**. Web servers other than IBM HTTP Server require manual propagation.  
The Web server plug-in configuration service propagates the `plugin-cfg.xml` file automatically for IBM HTTP Server 6.0 only. For all other Web servers, propagate the plug-in configuration file by manually copying the `plugin-cfg.xml` file from the `profiles_install_root/config/cells/cell_name/nodes/node_name/servers/web_server_name` directory on Machine A to the `plugins_install_root/config/web_server_name` directory on Machine B.
28. Create the second application server profile using the Profile creation wizard on Machine A. Make the profile the default profile during the profile creation by selecting the check box on the appropriate panel.  
The script that the Plug-ins installation wizard creates works on the default profile only. So, this script can create only a Web server definition on the profile that is the default profile at the time that the script runs.
29. Install a second IBM HTTP Server or another supported Web server on Machine B.
30. On Machine B, install the Web server plug-ins to configure the second Web server using the Plug-ins installation wizard. Both Web servers share a single installation of the plug-in binaries but must be configured individually.
31. The Plug-ins installation wizard creates a script named `configureWeb_server_name` for the second Web server. The script is in the `plugins_install_root/bin` directory on Machine B. Copy the script to the `install_root/bin` directory on Machine A.
32. Start the second application server.
33. Run the `configureWeb_server_name` script to create a Web server definition in the administrative console. You can then use the administrative console to manage the Web server.
34. Propagate the `plugin-cfg.xml` file from the second application server to the Web server using the administrative console. Click **Servers > Web server > Propagate Plug-in**. Web servers other than IBM HTTP Server require manual propagation.
35. Run the snoop servlet on the second Web server to verify that it is operational.

This procedure results in installing two or more application servers on one machine and installing dedicated Web servers on another machine. This procedure installs the Web server plug-ins for both Web servers and configures both Web servers and both application servers.

See “Selecting a Web server topology diagram and roadmap” on page 139 for an overview of the installation procedure.

See “Web server configuration” on page 153 for more information about the files involved in configuring a Web server.

See “Editing Web server configuration files” on page 203 for information about how the Plug-ins installation wizard configures supported Web servers.

## Configuring a Web server and an application server profile on the same machine

This topic describes installing a Web server plug-in that WebSphere Application Server provides to communicate with a particular brand of Web server. This procedure describes installing the Web server and its Web server plug-in for WebSphere Application Server and the Application Server on the same machine .

If the WebSphere Application Server product family supports a particular brand of Web server, such as IBM HTTP Server or Microsoft Internet Information Services (IIS), then your WebSphere Application Server product provides a binary plug-in for the Web server that you must install.

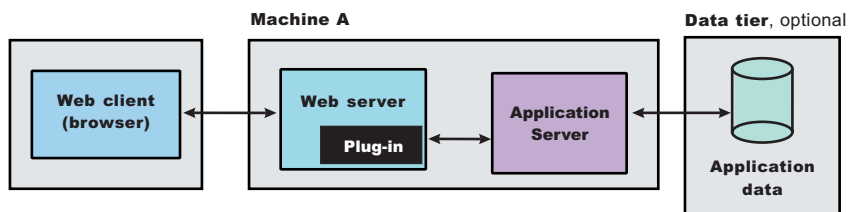
If the WebSphere Application Server product family does not provide a binary plug-in for a particular brand of Web server, then the Web server is not supported. The purpose of the binary plug-in is to provide the communication protocol between the Web server and the application server.

Suppose that you create a new profile. Suppose also that you want to use a Web server. You must install a new Web server for the new profile and use the Plug-ins installation wizard to install the binary plug-in module and to configure both the Web server and the application server.

If the Web server is not already installed, you can still install the plug-ins for future use. If the WebSphere Application Server product is not installed, you can still install the plug-ins. However, it is recommended that you install the Web server and the WebSphere Application Server product before installing the plug-ins for the supported Web server.

This procedure configures the application server profile that is the default profile on the machine. A one-to-one relationship exists between a Web server and the application server.

This topic describes how to create the following topology:



The set of steps leading up to the next diagram show how to configure a stand-alone application server. The set of steps after the next diagram show how to configure an application server that is federated into a deployment manager cell.

The wizard performs three steps to properly configure a Web server for Version 6. The wizard performs the steps in the following order:

1. The wizard installs the unique binary plug-in module for the supported Web server after collecting the following information:
  - The type of Web server
  - The location of the configuration file for the Web server that the wizard configures
  - The plug-ins installation root directory for the Web server plug-in modules that the wizard installs
  - The installation root directory of the WebSphere Application Server product, where the wizard creates a Web server definition

2. The wizard prompts you for the location of the configuration file or files for the Web server. You must browse for and select the correct file.

The wizard edits the configuration file or files for a Web server by creating directives that point to the location of the binary plug-in module and the plug-in configuration file.

The name of the binary plug-in module varies per Web server type. The plug-in configuration file is always the `plugin-cfg.xml` file.

3. The wizard creates a Web server definition in the configuration of the application server unless one already exists.

You can use the administrative console to manage the Web server configuration. For example, when you install an application on the application server, you can also choose to install it on the Web server

definition. If so, the updated `plugin-cfg.xml` file shows that the new application is available. When the Web server reads the updated plug-in configuration file, the Web server becomes aware of the new application that it can serve to Web clients.

If you choose not to install the new application on the Web server definition, the application is not added to the plug-in configuration file. The Web server is not aware of the application and cannot serve it to Web clients.

Use the following procedure to install the Web server plug-in, configure the Web server, and create a Web server definition in the default application server profile.

1. Log on as root on a Linux or UNIX operating system, or as a member of the administrator group on a Windows system.

**Linux**

**UNIX**

In addition, verify that the `umask` setting is 022. To verify the `umask` setting, issue the following command:

```
umask
```

To set the `umask` setting to 022, issue the following command:

```
umask 022
```

**Windows**

Some steps of the installation on a Windows platform require the administrator group user to have the following advanced user rights:

- *Act as part of the operating system*
- *Log on as a service*

For example, on some Windows systems, click **Administrative Tools > Local Security Policy > User Rights Assignments** to see the advanced options. See your Windows documentation for more information.









**Windows**






The installation wizard grants your Windows user ID the advanced user rights, if the user ID belongs to the administrator group. The silent installation does not grant these rights. If you create a new user ID on a Windows platform to perform a silent installation, you must restart the system to activate the proper authorizations for the user ID before you can perform a successful silent installation.

**Windows**

When installing the WebSphere Application Server as a Windows service, do not use a user ID that contains spaces. A user ID with spaces cannot be validated. Such a user ID is not allowed to continue the installation. To work around this problem, install with a user ID that does not contain spaces, or do not choose to install Windows services.

2. Install WebSphere Application Server Network Deployment on Machine A.  
See Chapter 9, “Installing the product and additional software,” on page 107.
3. Install the IBM HTTP Server or another supported Web server on Machine B.  
See “Installing IBM HTTP Server” on page 134 or refer to the product documentation for your Web server for more information.
4. Launch the Plug-ins installation wizard on the machine with the Web server.  
Select the Plug-ins installation wizard from the launchpad or change directories to the `plugin` directory on the product disc or in the downloaded installation image and issue the **install** command.  
The installation of the plug-ins tolerates installing as a non-root user. However, the automated installation of the required Global Services Kit (GSKit) fails if you install as non-root.  
If you decide to install the plug-in as a non-root user, log on as root after the entire procedure is complete. Reinstall the GSKit from the GSKit directory on the product disc.
5. Stop the stand-alone application server before installing the Web server plug-ins. For example, assuming that the profile name is `default`, use one of the following commands.
  - **AIX** `/usr/IBM/WebSphere/AppServer/profiles/default/bin/stopServer.sh server1`

-   /opt/ IBM/ WebSphere/ AppServer/ profiles/ *default*/ bin/ stopServer.sh server1
  -  C:\ Program Files\ IBM\ WebSphere\ AppServer\ profiles\ *default*\ bin\ stopServer.sh server1
6. Clear the check box for the roadmap or select the check box to view the roadmap, then click **Next**.  
If you are unsure of which installation scenario to follow, display the roadmap instead. Print and keep the roadmap as a handy overview of the installation steps.  
Press **Ctrl-P** to print the roadmap if the Web browser navigation controls and the menu bar are not present on the browser window that displays the Plug-ins roadmap. Press **Ctrl-W** to close the browser window if the navigation controls and the menu bar do not display. Or close the browser window with the window control in the title bar.
  7. Read the license agreement and accept the agreement if you agree to its terms. Click **Next** when you are finished.
  8. If your system does not pass the prerequisites check, stop the installation, correct any problems, and restart the installation. If your system passes the prerequisites check, click **Next**.  
Look for the appropriate log file for information about missing prerequisites:
    - If you stop the installation, see the temporaryPluginInstallLog.txt file in the temporary directory of the user who installed the plug-ins. For example, the /tmp/temporaryPluginInstallLog.txt file might exist if the root user installed the plug-ins on a Linux or UNIX system.
    - If you continue the installation in spite of warnings about missing prerequisites, see the *plugins\_install\_root/logs/install/log.txt* file after the installation is complete.
 See Appendix B, “Troubleshooting installation,” on page 337 for more information about log files.
  9. Select the type of Web server that you are configuring and click **Next**.  
The Plug-ins installation wizard panel prompts you to identify the *Web servers* to configure. Actually you can select only one Web server each time you run the Plug-ins installation wizard.  
Stop any Web server while you are configuring it. A step later in the procedures directs you to start the Web server as you begin the snoop servlet test.  
If you select the Web server identification option labeled **None**, the Web server installs the binary plug-ins but does not configure the Web server. A known problem in the wizard panel causes the English word *None* to appear in translated versions of the wizard. However, the selectable option is functional in every locale in spite of the missing translation.
  10. Select **Application Server machine (local)** and click **Next**.
  11. Accept the default location for the installation root directory for the plug-ins. Click **Next**.  
You can type another new directory or click **Browse** to select an empty directory. The fully qualified path identifies the plug-ins installation root directory.  
The default location is:
    -  /usr/IBM/WebSphere/Plugins
    -    /opt/IBM/WebSphere/Plugins
    -  C:\Program Files\IBM\WebSphere\Plugins
 A possibility exists that the Web server might run on a platform that WebSphere Application Server does not support.
  12. Click **Browse** on the Application Server installation location panel to browse for the location of the Application Server profile, if necessary. Click **Next** when the installation root directory is correct.  
The fully qualified path identifies the installation root directory for the WebSphere Application Server product.  
The default location is:

-  /usr/IBM/WebSphere/AppServer
-    /opt/IBM/WebSphere/AppServer
-  C:\Program Files\IBM\WebSphere\AppServer

13. Click **Browse** to select the configuration file for your Web server, verify that the Web server port is correct, and then click **Next** when you are finished.

Select the file and not just the directory of the file. Some Web servers have two configuration files and require you to browse for each file.

The following list shows configuration files for supported Web servers:

#### Apache HTTP Server

*apache\_install\_root/config/httpd.conf*

#### Domino Web Server

names.nsf and Notes.jar

The wizard prompts for the notes.jar file. The actual name is Notes.jar.

The Plug-ins installation wizard verifies that the files exist but the wizard does not validate either file.

#### IBM HTTP Server

*ihs\_install\_root/conf/httpd.conf*

#### Microsoft Internet Information Services (IIS)

The Plug-ins installation wizard can determine the correct files to edit.

#### Sun ONE Web Server 6.0 or Sun Java System Web Server, V6.1

obj.conf and magnus.conf

The wizard displays a naming panel for the nickname of the Web server definition.

14. Specify a nickname for the Web server. Click **Next** when you are finished.

The wizard uses the value to name configuration folders in the plug-ins installation root directory. The wizard also uses the name in the configuration script for the application server to name the Web server definition.

If the application server profile already has a Web server definition, specify a nickname that matches the name of the existing Web server definition. Using the same name configures the Web server to use the Web server definition on the application server.

15. Specify the location for the plugin-cfg.xml file and click **Next**.

This is a critical selection.

See “Configuration behavior of the Plug-ins installation wizard” on page 145 for a description of the logic that determines what path is configured by default. The following possibilities exist for the default location of the plug-in configuration file. The wizard determines the characteristics of the application server to determine the best path for the file:

- An application server that has an existing Web server definition has the following path:

*plug-ins\_install\_root/config/  
web\_server\_name/plugin-cfg.xml*

- A stand-alone application server that does not have a Web server definition has the following path:

*was\_install\_root/profiles/profile\_name  
/config/cells/cell\_name/nodes/  
Web\_server\_name\_node/servers/  
Web\_server\_name/plugin-cfg.xml*

You can accept the default value if the application server does not have a Web server definition.

#### Using an existing Web server definition



If the application server has a Web server definition, the wizard cannot create a new Web server definition within the application server configuration. However, the wizard can reconfigure the Web server. Click **Browse** and select the existing `plugin-cfg.xml` file in the application server configuration.

To find the plug-in configuration file in a stand-alone application server, follow this file path:

```
was_install_root/profiles/profile_name
/config/cells/cell_name/nodes/
Web_server_name_node/servers/
Web_server_name/plugin-cfg.xml
```

If the existing `Web_server_name` is different than the nickname that you gave the Web server in the wizard, click **Back** to return to the naming panel for the Web server and change the name to match the existing Web server definition name.

If you cannot find an existing `plugin-cfg.xml` file after all, you must install the temporary `plugin-cfg.xml` file. In such a case, type the path to the plug-ins installation root directory so that the wizard can install the temporary plug-in configuration file:

```
plug-ins_install_root/config/
web_server_name/plugin-cfg.xml
```

16. Click **Next** after verifying the characteristics of the plug-ins installation or click **Back** to make changes.

Once created, a Web server definition on a stand-alone application server node cannot be removed except through scripting. (See “Uninstalling the Web server plug-ins for WebSphere Application Server” on page 200 for the procedure.)

You can, however, reuse the same definition for a different type of Web server. Run the Plug-ins installation wizard to configure a new Web server in that situation. The Plug-ins installation wizard configures the new Web server to use the existing `plugin-cfg.xml` file.



17. Click **Next** on the pre-installation summary panel to begin the installation or click **Back** to change any characteristics of the installation. The wizard begins installing the plug-ins and configuring the Web server and the application server.



The wizard shows an installation status panel as it installs the plug-ins.

The wizard displays the Installation summary panel at the completion of the installation.



18. Verify the success of the installation on the Installation summary panel and click **Finish** to exit the wizard.

If a problem occurs and the installation is unsuccessful, examine the logs in the `plug-ins_install_root/logs` directory. Correct any problems and reinstall.

19.   **Domino Web Server only:** Set the `WAS_PLUGIN_CONFIG_FILE` environment variable.

  To configure Domino, you must set the `WAS_PLUGIN_CONFIG_FILE` environment variable. On Linux and UNIX-based platforms, sourcing a script to the parent shell allows child processes to inherit the exported variables. On Windows systems, run the script as you would run any other command. Sourcing is automatic on Windows systems.

- a. Open a command window.
- b. Change directories to the plug-ins installation root directory.
- c. Issue the appropriate command for the `plug-ins_install_root/bin/setupPluginCfg.sh` script:

-  `. plug-ins_install_root/bin/setupPluginCfg.sh` (Notice the space between the period and the installation root directory.)
-  `source plug-ins_install_root/bin/setupPluginCfg.sh`

The script is also in the `lotus_install_root/notesdata` directory on Linux and UNIX systems.



Issue the appropriate command for the script before starting the Domino Web Server.

20. Start the Snoop servlet to verify the ability of the Web server to retrieve an application from the Application Server.

Test your environment by starting your Application Server, your Web server, and using the snoop servlet with an IP address.

- a. Start the Application Server. In a Network Deployment environment, the Snoop servlet is available in the cell only if you included the DefaultApplication when adding the Application Server to the cell. The `-includeapps` option for the **addNode** command migrates the DefaultApplication to the cell. If the application is not present, skip this step.

Change directories to the `install_root/profiles/profile_name/bin` directory and run the **startServer** command:

-   `./startServer.sh server1`
-  `startServer server1`

- b. Start the IBM HTTP Server or the Web server that you are using.

Use a command window to change the directory to the IBM HTTP Server installed image, or to the installed image of your Web server. Issue the appropriate command to start the Web server, such as these commands for IBM HTTP Server:

**To start the IBM HTTP Server from the command line:**

Access the **apache** and **apachectl** commands in the `IBMHttpServer/bin` directory.

-   `./apachectl start`
-  `apache`

- c. Point your browser to `http://localhost:9080/snoop` to test the internal HTTP transport provided by the Application Server. Point your browser to `http://Host_name_of_Web_server_machine/snoop` to test the Web server plug-in.

The HTTP Transport port is 9080 by default and must be unique for every profile. The port is associated with a virtual host named `default_host`, which is configured to host the installed DefaultApplication and any installed Samples. The snoop servlet is part of the DefaultApplication. Change the port to match your actual HTTP Transport port.

- d. Verify that snoop is running.

Either Web address should display the Snoop Servlet - Request/Client Information page.

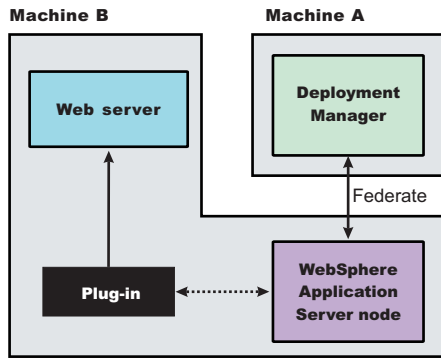
- e. **Remote IBM HTTP Server only:** Verify that the automatic propagation function can work on a remote IBM HTTP Server by using the following steps. This procedure is not necessary for local Web servers.
- 1) Run the **htpasswd** command on the Web server machine to create the administrative user ID and password information.
  - 2) Use the administrative console of the deployment manager or the Application Server to enter the User ID and password information that you created for the administrative user of IBM HTTP Server. Go to **Servers > Web server > Web\_server\_definition > Remote Web server administration**.

21. Configure a Web server and a distributed application server profile on the same machine.

The rest of these steps describe how to configure an application server that is federated into a deployment manager cell.

The following topology is considered a local distributed topology because it involves a cell:





This part of the procedure assumes that you have already installed the Network Deployment product on both machines. Also assumed is that you have already configured a deployment manager profile on Machine A and an application server profile on Machine B.

If you are planning to add the application server node into a deployment manager cell but have not done so yet, start the deployment manager and federate the node before installing the plug-in. You cannot add an application server with a Web server definition into the deployment manager cell.

A Web server definition on a federated application server is installed on the same managed node as the application server. There is one node, but with two server processes, the application server and the Web server definition.

If you are installing the plug-ins for use with a federated application server, start the deployment manager. Verify that the node agent process on the managed node is also running. Both the deployment manager and the node agent must be running to successfully configure a managed node.

22. Install IBM HTTP Server or another supported Web server on Machine B.  
See “Installing IBM HTTP Server” on page 134 or refer to the product documentation for your Web server for more information.
23. Launch the Plug-ins installation wizard on the machine with the Web server.
24. Clear the check box for the roadmap or select the check box to view the roadmap, then click **Next**.
25. Read the license agreement and accept the agreement if you agree to its terms, then click **Next**.
26. If your system does not pass the prerequisites check, stop the installation, correct any problems, and restart the installation. If your system passes the prerequisites check, click **Next**.
27. Select the type of Web server that you are configuring, then click **Next**.
28. Select **Application Server machine (local)** and click **Next**.
29. Accept the default location for the installation root directory for the plug-ins, then click **Next**.
30. Click **Browse** on the Application Server installation location panel to browse for the location of the Application Server profile, if necessary. Click **Next** when the installation root directory is correct.
31. Click **Browse** to select the configuration file for your Web server, verify that the Web server port is correct, and then click **Next**.
32. Specify a nickname for the Web server, then click **Next**.
33. Specify the location for the `plugin-cfg.xml` file and click **Next**.

This is a critical selection. A federated application server that does not have a Web server definition has the following path:

```

was_install_root/profiles/profile_name
  /config/cells/cell_name/nodes/
  node_name_of_AppServer/servers/
  Web_server_name/plugin-cfg.xml

```

An application server that has an existing Web server definition has the following path:

```

plug-ins_install_root/config/
  web_server_name/plugin-cfg.xml

```

See “Configuration behavior of the Plug-ins installation wizard” on page 145 for a description of the logic that determines what path is configured by default.

34. Click **Next** after verifying the characteristics of the plug-ins installation or click **Back** to make changes.

You can use the administrative console of the deployment manager to delete an existing Web server or to create new ones. Federated nodes can have more than one Web server definition.

35. Click **Next** on the pre-installation summary panel to begin the installation or click **Back** to change any characteristics of the installation.

The wizard begins installing the plug-ins and configuring the Web server and the application server.

The wizard shows an installation status panel as it installs the plug-ins.

The wizard displays the Installation summary panel at the completion of the installation.

36. Verify the success of the installation on the Installation summary panel and click **Finish** to exit the wizard.

37. Complete the installation by creating the Web server definition.

You can use the administrative console of the deployment manager to create the Web server definition on a federated node. Or, you can run the configuration script that the Plug-ins installation wizard created.

The script already contains all of the information that you must gather when using the administrative console option.

Select one of the following options:

- **Using the administrative console**

Click **Servers > Web servers > New** and use the Create new Web server entry wizard to create the Web server definition.

- **Running the configuration script**

Issue the appropriate command from a command window:

```
–   ./plug-ins_install_root/bin/configureWeb_server_name.sh
```

```
–  plug-ins_install_root\bin\configureWeb_server_name.bat
```

If you have enabled security or changed the default JMX connector type, edit the script and include the appropriate parameters on the **wsadmin** command.

38. From the administrative console of the deployment manager, click **System administration > Save Changes to Master Repository > Synchronize changes with Nodes > Save**.

39. Source the Domino Web server script if necessary.

40. Start the snoop servlet.

See the snoop procedure for the stand-alone application server for the full procedure.

The installation of the binary plug-in modules results in the creation of the `Plugins` directory and several subdirectories. The following directories are among those created on a Linux system, for example:

- `/opt/IBM/WebSphere/Plugins/_uninstPlugin/_jvmForPlugin` contains the WebSphere Application Server SDK, Java technology edition used to uninstall the product
- `/opt/IBM/WebSphere/Plugins/_uninstPlugin` contains the uninstaller program
- `/opt/IBM/WebSphere/Plugins/bin` contains the binary plug-ins for all supported Web servers
- `/opt/IBM/WebSphere/Plugins/logs` contains log files
- `/opt/IBM/WebSphere/Plugins/properties` contains version information
- `/opt/IBM/WebSphere/Plugins/roadmap` contains the roadmap for the Plug-ins installation wizard

The Plug-ins installation wizard creates a Web server definition within the application server profile unless one already exists.

The Plug-ins installation wizard configures the Web server to use the *profiles\_install\_root/profile\_name/plugin-cfg.xml* file.

The application server regenerates the Web server plug-in configuration file, *plugin-cfg.xml* whenever an event occurs that affects the file. Such events include the addition or removal of an application, server, or virtual host. The stand-alone application server regenerates the file in the following location:

```
was_install_root/profiles/profile_name  
/config/cells/cell_name/nodes/  
Web_server_name_node/servers/  
Web_server_name/plugin-cfg.xml
```

On a federated node, the creation or removal of clusters and cluster members also causes file regeneration. The deployment manager regenerates the file for a federated application server in the following location:

```
was_install_root/profiles/profile_name  
/config/cells/cell_name/nodes/  
node_name_of_AppServer/servers/  
Web_server_name/plugin-cfg.xml
```

You can start a stand-alone application server and the Web server immediately after installing of the binary plug-in for the local Web server. Open the administrative console of the application server after you start the server and save the changed configuration.

After installing the binary plug-in for the local Web server, you can start a federated application server and the Web server after running the configuration script that completes the installation. Open the administrative console of the deployment manager. Wait for node synchronization to occur. Save the changed configuration that includes the new Web server definition.

See “Selecting a Web server topology diagram and roadmap” on page 139 for an overview of the installation procedure.

See “Configuration behavior of the Plug-ins installation wizard” on page 145 for information about the location of the plug-in configuration file.

See “Web server configuration” on page 153 for information about the files involved in configuring a Web server.

See “Editing Web server configuration files” on page 203 for information about how the Plug-ins installation wizard configures supported Web servers.

See “Installing Web server plug-ins” on page 136 for information about other installation scenarios for installing Web server plug-ins.

#### **Related concepts**

“Configuration behavior of the Plug-ins installation wizard” on page 145

This topic describes installing the Web server plug-ins for WebSphere Application Server. The Plug-ins installation wizard installs a binary plug-in module and a plug-in configuration file for the Web server. The Plug-ins installation wizard configures the supported Web server for the Application Server. The Plug-ins installation wizard also creates a Web server definition in the configuration of the application server.

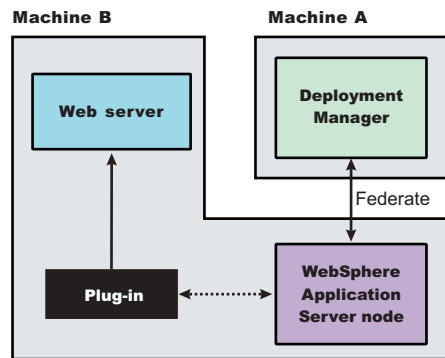
“Web server configuration” on page 153

After installing a supported Web server, you must install a binary plug-in module for the Web server. The plug-in module lets the Web server communicate with the application server. The Plug-ins installation wizard installs the Web server plug-in. The wizard configures the Web server. The wizard also creates a Web server definition in the configuration of the application server.

## Configuring a Web server and a custom profile on the same machine

This procedure describes installing a Web server and its plug-in on a machine where the default profile is a custom profile.

This procedure configures the custom profile that is the default profile on the machine. This procedure assumes that you already have installed a deployment manger on Machine A.



The WebSphere Application Server node on Machine B is the custom node that you create in this procedure. This procedure starts the deployment manager and federates the custom node before installing the Web server plug-ins.

Start the deployment manager. The deployment manager must be running to successfully federate and configure the custom node.

Use the following procedure to install the Web server plug-in, configure the Web server, and create a Web server definition in the default custom profile (custom node).

1. Log on as root on a Linux or UNIX operating system, or as a member of the administrator group on a Windows system.

**Linux** **UNIX** In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```

**Windows** Some steps of the installation on a Windows platform require the administrator group user to have the following advanced user rights:

- *Act as part of the operating system*
- *Log on as a service*

For example, on some Windows systems, click **Administrative Tools > Local Security Policy > User Rights Assignments** to see the advanced options. See your Windows documentation for more information.

**Windows** The installation wizard grants your Windows user ID the advanced user rights, if the user ID belongs to the administrator group. The silent installation does not grant these rights. If you create a new user ID on a Windows platform to perform a silent installation, you must restart the system to activate the proper authorizations for the user ID before you can perform a successful silent installation.

**Windows** When installing the WebSphere Application Server as a Windows service, do not use a user ID that contains spaces. A user ID with spaces cannot be validated. Such a user ID is not

- allowed to continue the installation. To work around this problem, install with a user ID that does not contain spaces, or do not choose to install Windows services.
2. Install the WebSphere Application Server Network Deployment product.  
See *Installing the product and additional software*.
  3. Create a custom profile as the first profile on the machine and federate the node as you create it.  
See *Using the Profile creation wizard to create a custom profile*.
  4. **Optional:** Use the administrative console of the deployment manager to create an Application Server on the custom node.  
Click **Servers > Applications servers > New** and follow the instructions to create a server. A server is not required for installing the plug-ins but it lets you verify the functionality of the Web server.
  5. **Optional:** Install the DefaultApplication on the new server while you are in the administrative console of the deployment manager.  
The DefaultApplication includes the snoop servlet. The verification step uses the snoop servlet.
  6. Stop the application server if it is running.  
Use the stopServer command or the administrative console of the deployment manager to stop the managed node.
  7. Install the IBM HTTP Server or another supported Web server on Machine B.  
See “Installing IBM HTTP Server” on page 134 or refer to the product documentation for your Web server for more information.
  8. Launch the Plug-ins installation wizard on the machine with the Web server.  
Select the Plug-ins installation wizard from the launchpad or change directories to the plugin directory on the product disc or in the downloaded installation image and issue the **install** command.  
The installation of the plug-ins tolerates installing as a non-root user. However, the automated installation of the required Global Services Kit (GSKit) fails if you install as non-root.  
If you decide to install the plug-in as a non-root user, log on as root after the entire procedure is complete. Reinstall the GSKit from the GSKit directory on the product disc.
  9. Clear the check box for the roadmap or select the check box to view the roadmap, then click **Next**.  
If you are unsure of which installation scenario to follow, display the roadmap instead. Print and keep the roadmap as a handy overview of the installation steps.  
Press **Ctrl-P** to print the roadmap if the Web browser navigation controls and the menu bar are not present on the browser window that displays the Plug-ins roadmap. Press **Ctrl-W** to close the browser window if the navigation controls and the menu bar do not display. Or close the browser window with the window control in the title bar.
  10. Read the license agreement and accept the agreement if you agree to its terms. Click **Next** when you are finished.
  11. If your system does not pass the prerequisites check, stop the installation, correct any problems, and restart the installation. If your system passes the prerequisites check, click **Next**.  
Look for the appropriate log file for information about missing prerequisites:
    - If you stop the installation, see the temporaryPluginInstallLog.txt file in the temporary directory of the user who installed the plug-ins. For example, the /tmp/temporaryPluginInstallLog.txt file might exist if the root user installed the plug-ins on a Linux or UNIX system.
    - If you continue the installation in spite of warnings about missing prerequisites, see the *plugins\_install\_root/logs/install/log.txt* file after the installation is complete.
 See Appendix B, “Troubleshooting installation,” on page 337 for more information about log files.
  12. Select the type of Web server that you are configuring and click **Next**.  
The Plug-ins installation wizard panel prompts you to identify the *Web servers* to configure. Actually you can select only one Web server each time you run the Plug-ins installation wizard.  
Stop any Web server while you are configuring it. A step later in the procedures directs you to start the Web server as you begin the snoop servlet test.






If you select the Web server identification option labeled **None**, the Web server installs the binary plug-ins but does not configure the Web server. A known problem in the wizard panel causes the English word *None* to appear in translated versions of the wizard. However, the selectable option is functional in every locale in spite of the missing translation.

13. Select **Application server machine (local)** and click **Next**.

14. Accept the default location for the installation root directory for the plug-ins. Click **Next**.

You can type another new directory or click **Browse** to select an empty directory. The fully qualified path identifies the plug-ins installation root directory.

The default location is:





-  /usr/IBM/WebSphere/Plugins
-    /opt/IBM/WebSphere/Plugins
-  C:\Program Files\IBM\WebSphere\Plugins

A possibility exists that the Web server might run on a platform that WebSphere Application Server does not support.

15. Click **Browse** on the Application Server installation location panel to browse for the location of the managed node, if necessary. Click **Next** when the installation root directory is correct.

The fully qualified path identifies the installation root directory for the Network Deployment product core files.

The default location is:

-  /usr/IBM/WebSphere/AppServer
-    /opt/IBM/WebSphere/AppServer
-  C:\Program Files\IBM\WebSphere\AppServer

16. Click **Browse** to select the configuration file for your Web server, verify that the Web server port is correct, and then click **Next** when you are finished.

Select the file and not just the directory of the file. Some Web servers have two configuration files and require you to browse for each file.

The following list shows configuration files for supported Web servers:

#### **Apache HTTP Server**

*apache\_install\_root/config/httpd.conf*

#### **Domino Web Server**

names.nsf and Notes.jar

The wizard prompts for the notes.jar file. The actual name is Notes.jar.

The Plug-ins installation wizard verifies that the files exist but the wizard does not validate either file.

#### **IBM HTTP Server**

*ihs\_install\_root/conf/httpd.conf*

#### **Microsoft Internet Information Services (IIS)**

The Plug-ins installation wizard can determine the correct files to edit.

#### **Sun ONE Web Server 6.0 or Sun Java System Web Server, V6.1**

obj.conf and magnus.conf

The wizard displays a naming panel for the nickname of the Web server definition.

17. Specify a nick name for the Web server and click **Next**.



The wizard uses the value to name configuration folders in the plug-ins installation root directory. The wizard also uses the name in the configuration script for the application server to name the Web server definition.

18. Accept the location for the `plugin-cfg.xml` file and click **Next**.

See “Configuration behavior of the Plug-ins installation wizard” on page 145 for a description of the logic that determines what path is configured by default. The wizard determines the characteristics of the managed application server node to determine the best path for the file:

A federated custom node has the following path:

```
WAS_install_root/profiles/profile_name  
/config/cells/cell_name/nodes/  
node_name_of_custom_profile/servers/  
Web_server_name/plugin-cfg.xml
```

Accept the default value.

19. Click **Next** after verifying the characteristics of the plug-ins installation or click **Back** to make changes.

You can use the administrative console of the deployment manager to delete an existing Web server or to create new ones. Federated nodes can have more than one Web server definition.

20. Click **Next** on the pre-installation summary panel to begin the installation or click **Back** to change any characteristics of the installation. The wizard begins installing the plug-ins and configuring the Web server and the managed custom node.

The wizard shows an installation status panel as it installs the plug-ins. The wizard displays the Installation summary panel at the completion of the installation.

21. After the wizard installs the code and creates the uninstaller program, examine the post-installation summary panel. Click **Next** when you are finished to display the Plug-ins installation roadmap.

The Plug-ins installation wizard installs the binary plug-in module. On a Linux system, for example, the installation creates the `/opt/IBM/WebSphere/Plugins` directory. The `/opt/IBM/WebSphere/Plugins/config/Web_server_name` directory contains the `plugin-cfg.xml` file. The wizard displays the name and location of the configuration script and the `plugin-cfg.xml` file. The wizard also displays the type of Web server that is configured and the nickname of the Web server.

If a problem occurs and the installation is unsuccessful, examine the logs in the `plug-ins_install_root/logs` directory. Correct any problems and reinstall.

22. Close the road map and click **Finish** to exit the wizard.

Log files from the installation are in the `plug-ins_install_root/logs/install` directory.

23. Complete the installation by creating the Web server definition.

You can use the administrative console of the deployment manager to create the Web server definition on a federated node. Or, you can run the configuration script that the Plug-ins installation wizard created.

The script already contains all of the information that you must gather when using the administrative console option.



Select one of the following options:

- **Using the administrative console**



Click **Servers > Web servers > New** and use the Create new Web server entry wizard to create the Web server definition.



- **Running the configuration script**



Issue the appropriate command from a command window:

-   `./plug-ins_install_root/bin/configureWeb_server_name.sh`
-  `plug-ins_install_root\bin\configureWeb_server_name.bat`

If you have enabled security or changed the default JMX connector type, edit the script and include the appropriate parameters on the **wsadmin** command.

24.   **Domino Web Server only:** Set the `WAS_PLUGIN_CONFIG_FILE` environment variable.

  To configure Domino, you must set the `WAS_PLUGIN_CONFIG_FILE` environment variable. On Linux and UNIX-based platforms, sourcing a script to the parent shell allows child processes to inherit the exported variables. On Windows systems, run the script as you would run any other command. Sourcing is automatic on Windows systems.

- a. Open a command window.
- b. Change directories to the plug-ins installation root directory.
- c. Issue the appropriate command for the `plug-ins_install_root/bin/setupPluginCfg.sh` script:
  -  `. plug-ins_install_root/bin/setupPluginCfg.sh` (Notice the space between the period and the installation root directory.)
  -  `source plug-ins_install_root/bin/setupPluginCfg.sh`

The script is also in the `lotus_install_root/notesdata` directory on Linux and UNIX systems.




Issue the appropriate command for the script before starting the Domino Web Server.

25. Start the Snoop servlet to verify the ability of the Web server to retrieve an application from the Application Server.

Test your environment by starting your Application Server, your Web server, and using the snoop servlet with an IP address.

- a. Start the Application Server. In a Network Deployment environment, the Snoop servlet is available in the cell only if you included the DefaultApplication when adding the Application Server to the cell. The `-includeapps` option for the **addNode** command migrates the DefaultApplication to the cell. If the application is not present, skip this step.

Change directories to the `install_root/profiles/profile_name/bin` directory and run the **startServer** command:

-   `./startServer.sh server1`
-  `startServer server1`

- b. Start the IBM HTTP Server or the Web server that you are using.

Use a command window to change the directory to the IBM HTTP Server installed image, or to the installed image of your Web server. Issue the appropriate command to start the Web server, such as these commands for IBM HTTP Server:

**To start the IBM HTTP Server from the command line:**

Access the **apache** and **apachectl** commands in the `IBMHttpServer/bin` directory.

-   `./apachectl start`
-  `apache`

- c. Point your browser to `http://localhost:9080/snoop` to test the internal HTTP transport provided by the Application Server. Point your browser to `http://Host_name_of_Web_server_machine/snoop` to test the Web server plug-in.

The HTTP Transport port is 9080 by default and must be unique for every profile. The port is associated with a virtual host named `default_host`, which is configured to host the installed DefaultApplication and any installed Samples. The snoop servlet is part of the DefaultApplication. Change the port to match your actual HTTP Transport port.

- d. Verify that snoop is running.

Either Web address should display the Snoop Servlet - Request/Client Information page.
- e. **Remote IBM HTTP Server only:** Verify that the automatic propagation function can work on a remote IBM HTTP Server by using the following steps. This procedure is not necessary for local Web servers.



- 1) Run the **htpasswd** command on the Web server machine to create the administrative user ID and password information.
  - 2) Use the administrative console of the deployment manager or the Application Server to enter the User ID and password information that you created for the administrative user of IBM HTTP Server. Go to **Servers > Web server > Web\_server\_definition > Remote Web server administration**.
26. If the deployment manager does not have the DefaultApplication installed, you can test the functionality of the Web server and the custom node using an application of your own. See Fast paths for WebSphere Application Server for help in deploying applications.
27. From the administrative console of the deployment manager, click **System administration > Save Changes to Master Repository > Synchronize changes with Nodes > Save**.
28. To create multiple Web server definitions for the managed node, use the Plug-ins installation wizard to configure each Web server.  
Identify the same managed node each time. Give each Web server a different nick name.

This procedure results in the installation of the Web server plug-ins for WebSphere Application Server on a Web server machine. The Plug-ins installation wizard creates a Web server definition within the managed node.

The Plug-ins installation wizard configures the Web server to use the `plugin-cfg.xml` file that is within the managed custom node.

The deployment manager regenerates the Web server plug-in configuration file, `plugin-cfg.xml` whenever an event occurs that affects the file. Such events include the addition or removal of an application, server, or virtual host.

The creation or removal of clusters and cluster members also causes file regeneration. Automatic propagation through node synchronization copies the file after each regeneration to the following location on the custom node machine:

```
was_install_root/profiles/profile_name
/config/cells/cell_name/nodes/
node_name_of_custom_node/servers/
web_server_name/plugin-cfg.xml
```

The installation of the binary plug-in modules results in the creation of the `Plugins` directory and several subdirectories. The following directories are among those created on a Linux system, for example:

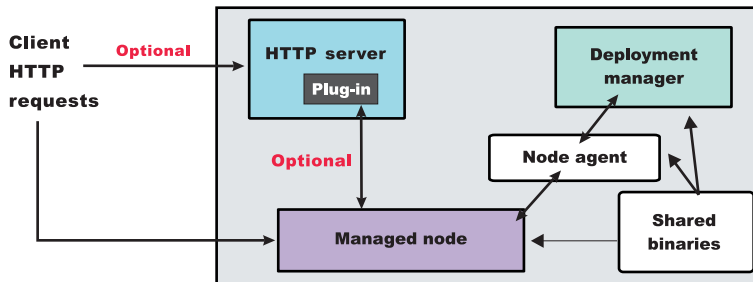
- `/opt/IBM/WebSphere/Plugins/_uninstPlugin/_jvmForPlugin` contains the WebSphere Application Server SDK, Java technology edition used to uninstall the product
- `/opt/IBM/WebSphere/Plugins/_uninstPlugin` contains the uninstaller program
- `/opt/IBM/WebSphere/Plugins/bin` contains the binary plug-ins for all supported Web servers
- `/opt/IBM/WebSphere/Plugins/logs` contains log files
- `/opt/IBM/WebSphere/Plugins/properties` contains version information
- `/opt/IBM/WebSphere/Plugins/roadmap` contains the roadmap for the Plug-ins installation wizard

After installing the binary plug-in for the local Web server, you can start the managed node and the Web server after running the configuration script that completes the installation.

## Configuring a Web server and a deployment manager profile on the same machine

This procedure describes installing a Web server and its plug-in on a machine where the default profile is a deployment manager.

This procedure configures the deployment manager profile that is the default profile on the machine. A managed node must exist to define a Web server definition, which is always on a managed node. If the deployment manager is the default profile, the Plug-ins installation wizard looks for a managed custom node in the deployment manager configuration. If the deployment manager does not have a managed custom node, the Plug-ins installation wizard looks for a managed application server node. If the deployment manager does not have a managed node, then the Plug-ins installation wizard classifies the installation as a remote installation.



Start the deployment manager and the node agent for the managed node. The deployment manager and the node must be running to successfully change its configuration.

Use the following procedure to install the Web server plug-in, configure the Web server, and create a Web server definition in the default profile.

1. Log on as root on a Linux or UNIX operating system, or as a member of the administrator group on a Windows system.



In addition, verify that the umask setting is 022. To verify the umask setting, issue the following command:

```
umask
```

To set the umask setting to 022, issue the following command:

```
umask 022
```



Some steps of the installation on a Windows platform require the administrator group user to have the following advanced user rights:

- *Act as part of the operating system*
- *Log on as a service*

For example, on some Windows systems, click **Administrative Tools > Local Security Policy > User Rights Assignments** to see the advanced options. See your Windows documentation for more information.








The installation wizard grants your Windows user ID the advanced user rights, if the user ID belongs to the administrator group. The silent installation does not grant these rights. If you create a new user ID on a Windows platform to perform a silent installation, you must restart the system to activate the proper authorizations for the user ID before you can perform a successful silent installation.



When installing the WebSphere Application Server as a Windows service, do not use a user ID that contains spaces. A user ID with spaces cannot be validated. Such a user ID is not allowed to continue the installation. To work around this problem, install with a user ID that does not contain spaces, or do not choose to install Windows services.

2. Install the WebSphere Application Server Network Deployment product.  
See *Installing the product and additional software*.
3. Create a deployment manager profile as the first profile on the machine. See *Using the Profile creation wizard to create a deployment manager*.

4. Install the IBM HTTP Server or another supported Web server.  
See Installing IBM HTTP Server or the product documentation for your Web server.
5. Launch the Plug-ins installation wizard on the machine with the Web server.  
Select the Plug-ins installation wizard from the launchpad or change directories to the `plugin` directory on the product disc or in the downloaded installation image and issue the **install** command.  
The installation of the plug-ins tolerates installing as a non-root user. However, the automated installation of the required Global Services Kit (GSKit) fails if you install as non-root.  
If you decide to install the plug-in as a non-root user, log on as root after the entire procedure is complete. Reinstall the GSKit from the GSKit directory on the product disc.
6. Clear the check box for the roadmap or select the check box to view the roadmap, then click **Next**.  
If you are unsure of which installation scenario to follow, display the roadmap instead. Print and keep the roadmap as a handy overview of the installation steps.  
Press **Ctrl-P** to print the roadmap if the Web browser navigation controls and the menu bar are not present on the browser window that displays the Plug-ins roadmap. Press **Ctrl-W** to close the browser window if the navigation controls and the menu bar do not display. Or close the browser window with the window control in the title bar.
7. Read the license agreement and accept the agreement if you agree to its terms. Click **Next** when you are finished.
8. If your system does not pass the prerequisites check, stop the installation, correct any problems, and restart the installation. If your system passes the prerequisites check, click **Next**.  
Look for the appropriate log file for information about missing prerequisites:
  - If you stop the installation, see the `temporaryPluginInstallLog.txt` file in the temporary directory of the user who installed the plug-ins. For example, the `/tmp/temporaryPluginInstallLog.txt` file might exist if the root user installed the plug-ins on a Linux or UNIX system.
  - If you continue the installation in spite of warnings about missing prerequisites, see the `plugins_install_root/logs/install/log.txt` file after the installation is complete.
 See Appendix B, “Troubleshooting installation,” on page 337 for more information about log files.
9. Select the type of Web server that you are configuring and click **Next**.  
The Plug-ins installation wizard panel prompts you to identify the *Web servers* to configure. Actually you can select only one Web server each time you run the Plug-ins installation wizard.  
Stop any Web server while you are configuring it. A step later in the procedures directs you to start the Web server as you begin the snoop servlet test.  
If you select the Web server identification option labeled **None**, the Web server installs the binary plug-ins but does not configure the Web server. A known problem in the wizard panel causes the English word *None* to appear in translated versions of the wizard. However, the selectable option is functional in every locale in spite of the missing translation.
10. Select **Application Server machine (local)** and click **Next**.
11. Accept the default location for the installation root directory for the plug-ins. Click **Next**.  
You can type another new directory or click **Browse** to select an empty directory. The fully qualified path identifies the plug-ins installation root directory.  
The default location is:
  -  `/usr/IBM/WebSphere/Plugins`
  -    `/opt/IBM/WebSphere/Plugins`
  -  `C:\Program Files\IBM\WebSphere\Plugins`
 A possibility exists that the Web server might run on a platform that WebSphere Application Server does not support.
12. Click **Browse** on the Application Server installation location panel to browse for the location of the deployment manager, if necessary. Click **Next** when the installation root directory is correct.

The fully qualified path identifies the installation root directory for the Network Deployment product core files.

The default location is:

-  /usr/IBM/WebSphere/AppServer
-    /opt/IBM/WebSphere/AppServer
-  C:\Program Files\IBM\WebSphere\AppServer

13. Click **Browse** to select the configuration file for your Web server, verify that the Web server port is correct, and then click **Next** when you are finished.

Select the file and not just the directory of the file. Some Web servers have two configuration files and require you to browse for each file.

The following list shows configuration files for supported Web servers:

#### Apache HTTP Server

*apache\_install\_root/config/httpd.conf*

#### Domino Web Server

names.nsf and Notes.jar

The wizard prompts for the notes.jar file. The actual name is Notes.jar.

The Plug-ins installation wizard verifies that the files exist but the wizard does not validate either file.

#### IBM HTTP Server

*ihs\_install\_root/conf/httpd.conf*

#### Microsoft Internet Information Services (IIS)

The Plug-ins installation wizard can determine the correct files to edit.

#### Sun ONE Web Server 6.0 or Sun Java System Web Server, V6.1

obj.conf and magnus.conf

The wizard displays a naming panel for the nickname of the Web server definition.

14. Specify a nickname for the Web server and click **Next**.

The wizard uses the value to name configuration folders in the plug-ins installation root directory. The wizard also uses the name within the deployment manager as the name of the Web server definition.

15. Accept the location for the `plugin-cfg.xml` file and click **Next**.

See “Configuration behavior of the Plug-ins installation wizard” on page 145 for a description of the logic that determines what path is configured by default. The wizard determines the characteristics of the deployment manager to determine the best path for the file.

When the deployment manager is the default profile, the path is:

*plugins\_install\_root/config/  
web\_server\_name/plugin-cfg.xml*

Accept the default value.

**Important:** If there is a managed custom node on the deployment manager machine, the Plug-ins installation wizard uses the following file path:

*WAS\_install\_root/profiles/profile\_name  
/config/cells/cell\_name/nodes/  
node\_name\_of\_custom\_profile/servers/  
Web\_server\_name/plugin-cfg.xml*

In this case, accept the path and resume the procedure at this point in “Configuring a Web server and a custom profile on the same machine” on page 180.

16. Click **Next** after verifying the characteristics of the plug-ins installation or click **Back** to make changes.

You can use the administrative console of the deployment manager to delete an existing Web server or to create new ones. Federated nodes can have more than one Web server definition.

17. Click **Next** on the pre-installation summary panel to begin the installation or click **Back** to change any characteristics of the installation.

The wizard begins installing the plug-ins and configuring the Web server and the deployment manager.

The wizard shows an installation status panel as it installs the plug-ins.

The wizard displays the Installation summary panel at the completion of the installation.

18. After the wizard installs the code and creates the uninstaller program, examine the post-installation summary panel. Click **Next** when you are finished to display the Plug-ins installation roadmap.

The Plug-ins installation wizard installs the binary plug-in module. On a Linux system, for example, the installation creates the `/opt/IBM/WebSphere/Plugins` directory. The

`/opt/IBM/WebSphere/Plugins/config/Web_server_name` directory contains the `plugin-cfg.xml` file.

The wizard displays the name and location of the configuration script and the `plugin-cfg.xml` file.

The wizard also displays the type of Web server that is configured and the nickname of the Web server.

If a problem occurs and the installation is unsuccessful, examine the logs in the `plugin_install_root/logs` directory. Correct any problems and reinstall.

19. Close the road map and click **Finish** to exit the wizard.

Log files from the installation are in the `plugin_install_root/logs/install` directory.

20. Complete the installation by creating the Web server definition.

You must create an application server profile or a custom profile and federate the node before you can use the administrative console of the deployment manager to create a Web server definition. The same is true for running the configuration script that the Plug-ins installation wizard created. You must assign the Web server to a managed node when you create it.

The managed node must exist before running the Plug-ins installation wizard. Otherwise, the installation is considered a remote installation.

If you install the plug-in, save the script to run after you create a managed node. Otherwise an error occurs. Wait until the script runs successfully and creates the Web server definition on the managed node and node synchronization occurs before starting the Web server.

Adding the node starts the nodeagent process. If the node agent is not running for some reason, start the node.

See `startNode` command for more information.

**Tip:** If you want the Web server to handle requests for an application for multiple managed nodes, install the application on each managed node and on the Web server definition.

The script already contains all of the information that you must gather when using the administrative console option.

Select one of the following options:

- **Using the administrative console**

Click **Servers > Web servers > New** and use the Create new Web server entry wizard to create the Web server definition.

- **Running the configuration script**

If the node has only a deployment manager profile, then the plug-ins installer reverts to a remote plug-in configuration. You must manually copy the `plugin_install_root/bin/configureWeb_server_name.sh` script or the `plugin_install_root\bin\configureWeb_server_name.bat` script to the `was_install_root/bin` directory of the deployment manager to run the script.

Issue the appropriate command from a command window:

```
- > Linux > UNIX ./was_install_root/bin/configureWeb_server_name.sh
```

– **Windows** `was_install_root\bin\configureWeb_server_name.bat`

If you have enabled security or changed the default JMX connector type, edit the script and include the appropriate parameters on the **wsadmin** command.

21. **Linux** **UNIX** **Domino Web Server only:** Set the `WAS_PLUGIN_CONFIG_FILE` environment variable.

**Linux** **UNIX** To configure Domino, you must set the `WAS_PLUGIN_CONFIG_FILE` environment variable. On Linux and UNIX-based platforms, sourcing a script to the parent shell allows child processes to inherit the exported variables. On Windows systems, run the script as you would run any other command. Sourcing is automatic on Windows systems.

- Open a command window.
- Change directories to the plug-ins installation root directory.
- Issue the appropriate command for the `plug-ins_install_root/bin/setupPluginCfg.sh` script:
  - UNIX** `. plug-ins_install_root/bin/setupPluginCfg.sh` (Notice the space between the period and the installation root directory.)
  - Linux** `source plug-ins_install_root/bin/setupPluginCfg.sh`

The script is also in the `lotus_install_root/notesdata` directory on Linux and UNIX systems.

Issue the appropriate command for the script before starting the Domino Web Server.

22. From the administrative console of the deployment manager, click **System administration > Save Changes to Master Repository > Synchronize changes with Nodes > Save**.

23. Start the Snoop servlet to verify the ability of the Web server to retrieve an application from the Application Server.

Test your environment by starting your Application Server, your Web server, and using the snoop servlet with an IP address.

- Start the Application Server. In a Network Deployment environment, the Snoop servlet is available in the cell only if you included the `DefaultApplication` when adding the Application Server to the cell. The `-includeapps` option for the **addNode** command migrates the `DefaultApplication` to the cell. If the application is not present, skip this step.

Change directories to the `install_root/profiles/profile_name/bin` directory and run the **startServer** command:

- Linux** **UNIX** `./startServer.sh server1`
- Windows** `startServer server1`

- Start the IBM HTTP Server or the Web server that you are using.

Use a command window to change the directory to the IBM HTTP Server installed image, or to the installed image of your Web server. Issue the appropriate command to start the Web server, such as these commands for IBM HTTP Server:

**To start the IBM HTTP Server from the command line:**

Access the **apache** and **apachectl** commands in the `IBMHttpServer/bin` directory.

- Linux** **UNIX** `./apachectl start`
- Windows** `apache`

- Point your browser to `http://localhost:9080/snoop` to test the internal HTTP transport provided by the Application Server. Point your browser to `http://Host_name_of_Web_server_machine/snoop` to test the Web server plug-in.

The HTTP Transport port is 9080 by default and must be unique for every profile. The port is associated with a virtual host named `default_host`, which is configured to host the installed `DefaultApplication` and any installed `Samples`. The snoop servlet is part of the `DefaultApplication`. Change the port to match your actual HTTP Transport port.



- d. Verify that snoop is running.  
Either Web address should display the Snoop Servlet - Request/Client Information page.
- e. **Remote IBM HTTP Server only:** Verify that the automatic propagation function can work on a remote IBM HTTP Server by using the following steps. This procedure is not necessary for local Web servers.
  - 1) Run the **htpasswd** command on the Web server machine to create the administrative user ID and password information.
  - 2) Use the administrative console of the deployment manager or the Application Server to enter the User ID and password information that you created for the administrative user of IBM HTTP Server. Go to **Servers > Web server > Web\_server\_definition > Remote Web server administration**.

The installation of the binary plug-in modules results in the creation of the Plugins directory and several subdirectories. The following directories are among those created on a Linux system, for example:

- /opt/IBM/WebSphere/Plugins/\_uninstPlugin/\_jvmForPlugin contains the WebSphere Application Server SDK, Java technology edition used to uninstall the product
- /opt/IBM/WebSphere/Plugins/\_uninstPlugin contains the uninstaller program
- /opt/IBM/WebSphere/Plugins/bin contains the binary plug-ins for all supported Web servers
- /opt/IBM/WebSphere/Plugins/logs contains log files
- /opt/IBM/WebSphere/Plugins/properties contains version information
- /opt/IBM/WebSphere/Plugins/roadmap contains the roadmap for the Plug-ins installation wizard

The Plug-ins installation wizard configures the Web server to use the *plug-ins\_install\_root/plugin-cfg.xml* file.

After installing the binary plug-in for the local Web server, you must create a managed node before you can successfully run the configuration script and use the Web server.

## responsefile.txt

This topic describes the response file for performing a silent installation of the Web server plug-ins for WebSphere Application Server.

Install the product silently using an options response file.

The responsefile.txt file has directives that set installation options. Comments in the file describe how to set the string value for each directive.

Use the options file to run the Plug-ins installation wizard in silent mode, which is referred to as installing silently. The wizard reads the options file to determine responses and does not display the graphical user interface. Use the following command to use a copy of the options file named myresponsefile.txt for a silent installation:

```
install -options "myresponsefile.txt" -silent
```

## Location of the response file

The sample options response file is named responsefile.txt. The file is in the plugin directory on the product disc or in the downloaded installation image.

## Mode of use

The Plug-ins installation wizard can read an existing options response file and run silently without displaying the graphical user interface.

**Installing silently:** The options file supplies the values to the Plug-ins installation wizard when installing silently. The wizard reads the options file to determine responses and does not display the graphical user interface. Use the following command to use a copy of the options file named `myresponsefile.txt` for a silent installation:

```
install -options "myresponsefile.txt" -silent
```

**Creating an operational environment:** The installation of the plug-ins is a three-step process:

1. Installing the binary plug-in modules for supported Web servers
2. Configuring the Web servers to use the binary module to communicate with the application server
3. Creating a Web server definition in the application server

As you install an application, you can install it on the Web server definition in addition to the application server. All applications on the Web server definition are listed in its plug-in configuration file. After propagation, the real Web server can access the applications.

The granular assignment of applications to a Web server is new in Version 6.

The sample options response file, `responsefile.txt`, controls installing the binary plug-ins, configuring the Web server, and creates a script for creating the Web server definition on a remote application server machine. The script is customized according to values supplied in the `responsefile.txt` file. The script is generated to run on the application server machine to create the Web server definition.

If the Web server is on the same machine as a stand-alone application server, the `responsefile.txt` file can create the Web server definition directly without creating a script.

To edit and use the response file for installing the plug-ins and configuring the Web server and application server, perform the following procedure:

1. Copy the `responsefile.txt` file from the `plugins` directory on the product disc to a place that you can easily identify on your machine.
2. Edit the file to customize the values for your installation.
3. Save the file.
4. Start the installation. For example:



```
install -options /tmp/plugins/myresponsefile.txt -silent
```



On non-Windows operating systems, silent installations run in a synchronous process. The process does not return until the silent installation finishes. For a silent installation to run synchronously on Windows, issue the first of the following commands:

- **Synchronous processing:** `START /WAIT install.exe -options "C:\temp\myresponsefile.txt" -silent`
- **Asynchronous processing:** `install -options "C:\temp\plugins\myresponsefile.txt" -silent`

5. After the installation, examine the logs for success.

## Logging

If no installation logs exist, refer to temporary log file, `temporaryPluginInstallLog.txt` in your temporary directory. You can also cause ISMP to record status about a problem that is preventing the installation from occurring, as described in the following section.

For example, if you start the silent installation without accepting the license in the `-W silentInstallLicenseAcceptance.value="true"` directive, the installation does not occur. The fact that the license entry was not accepted is recorded in `temporaryPluginInstallLog.txt` in the user temporary directory.



For example, the root user on a Linux system would normally find the file in the `/tmp/temporaryPluginInstallLog.txt` directory.

If any response file validation results in a failure, the failure is listed in the `temporaryPluginInstallLog.txt` file, then the installation fails.

If all validations pass, the installation occurs. Then, the Plug-ins installation wizard records installation events in the following log files. The log files are in the `plug-ins_install_root/logs/install` directory:

#### **log.txt**

Records all of the ISMP events that occur during the installation. The log also describes whether the installation was local or remote. Messages at the end of the file indicate whether manual configuration steps are required to complete the installation.

Key elements to look for in the installation record are:

#### **Manual steps warning**

When the wizard requires you to run a script to create the Web server definition, the wizard refers to the fact that manual steps are required.

If manual steps are required, the name and location of the script that you must run are written in the log file at the end of the installation record.

#### **Web server type**

The log has a record of the Web server type, such as IHS for the IBM HTTP Server, for example.

#### **Location of the plug-in configuration file**

The log has a record of the `plugin-cfg.xml` file location currently in the Web server configuration.


#### **masterConfigurationLog.txt**

Lists all of the configuration events that occur during the installation.

#### **installGSKit.log**

Lists events that occur during the installation of the GSKit code.

The command line for the installation is listed when the installation occurs. The GSKit 7 installation record is written after the `GSKIT 7 :` entry in the log.

 The GSKit 4 installation record is written after the `GSKIT 4:` entry in the log.

#### **installWeb\_server\_typePlugin.log**

Records events that occur during the installation of a Web server plug-in. The name of the file varies to reflect the Web server:

- `installAPACHEPlugin.log`
- `installIHSPPlugin.log`
- `installIISPPlugin.log`
- `installSunOnePlugin.log`
- `installDomino5Plugin.log`
- `installDomino6Plugin.log`

Each log lists the following critical information:

- The plug-in binary module that is currently installed
- The current location of the plug-in configuration file that is configured for the Web server

### **configure\_Web\_server\_type\_webserver.log**

Lists events that occur during the configuration of a Web server plug-in. The name of the file varies to reflect the Web server:

- configure\_APACHE\_webserver.log
- configure\_IHS\_webserver.log
- configure\_IIS\_webserver.log
- configure\_SUNJAVASYSTEM\_webserver.log
- configure\_DOMINO\_webserver.log

The `configure_Web_server_type_webserver.log` file reports the actions that the Plug-ins installation wizard performs as it updates the Web server configuration file.

In a remote scenario, this log is not present because you must run the script to create the Web server definition manually.

In a federated scenario, the script is created and this log is not present.

### **Information that ISMP can log when it cannot start the Plug-ins installation wizard**

Verify or troubleshoot the installation if the `install_root/product/logs/log.txt` file or the `install_root/product/logs/pctLog.txt` file does not contain a record of any problems, but problems exist.

If the error happens early in the installation, look for the logs in the system temporary directory. The installation program copies the logs from the system temporary directory to the logs directory at the end of the installation.

Certain events can prevent ISMP from starting the Installation wizard. Such an event is not enough disk space to launch the Installation wizard, for example. If your installation fails and there is no information in the installation logs, use the `-log` parameter to record entries for events that cause the ISMP program to fail to start the installation wizard. The syntax of the **install** command for logging such events is:

```
install -options fully_qualified_options_response_file_name
        -silent
        -log # !fully_qualified_log_file_name @ALL
```

The following example is for AIX systems:

```
install -options "/usr/IBM/WebSphere/silentFiles/myresponsefile.txt"
        -silent -log # !/usr/IBM/WebSphere/myOptionFiles/log.txt @ALL
```

The following example is for Linux systems, HP-UX systems, and Solaris systems:

```
install -options "/opt/IBM/WebSphere/silentFiles/myresponsefile.txt"
        -silent -log # !/opt/IBM/WebSphere/myOptionFiles/log.txt @ALL
```

The following example is for Windows systems:

```
install.exe -options "C:\IBM\WebSphere\silentFiles\myresponsefile.txt"
            -silent -log # !C:\IBM\WebSphere\silentFiles\log.txt @ALL
```

See Appendix B, "Troubleshooting installation," on page 337 and "Installation component troubleshooting tips" on page 355 for more information.

### **Response file user entry validation**

In a silent install, responsefile validation has been coded into the install. If response file validation does not pass, the failure is recorded in the temporary `PluginInstallLog.txt` file.

#### **Accepted license agreement**

##### **Default directive setting**

```
-W silentInstallLicenseAcceptance.value="false"
```

**Valid setting**

You must set this directive to true to accept the license and install the plug-ins.

**Error identifier in temporaryPluginInstallLog.txt**

INSTCONFFAILED : LICENSE \_ NOT \_ ACCEPTED.

**Valid install type****Default directive setting**

-P pluginSettings.installType="local"

**Valid setting**

You must set this directive to remote or local. Any other value fails validation.

**Error identifier in temporaryPluginInstallLog.txt**

INSTCONFFAILED : INVALID \_ PLUGIN \_ INSTALL \_ TYPE \_ SCENARIO \_ SELECTED.

**Valid WAS V6.0 install location****Default directive setting**

-W websphereLocationWizardBean.wasExistingLocation= "C: \ Program Files \ IBM \ WebSphere \ AppServer"

The setting varies per operating system.

**Valid setting**

If you set the -P pluginSettings.installType="" directive above to local, this validation checks that the path is a valid WebSphere Application Server V6 directory. Any other path fails validation for a local installation type.

**Error identifier in temporaryPluginInstallLog.txt**

INSTCONFFAILED : NON \_ WAS6 \_ DIRECTORY

**Valid Web server configuration file 1****Default directive setting**

-P pluginSettings.webServerConfigFile1= "C:\ Program Files \ IBM HTTP Server \ conf \ httpd.conf"

The setting varies per operating system. Valid file names for each Web server are:

- IBM HTTP Server: httpd.conf
- Apache: httpd.conf
- Domino 5: notes.jar
- Domino 6: notes.jar
- Sun Java System Web Server: obj.conf

**Valid setting**

Validation checks that the file exists. A known problem in ISMP validates a directory specification. However, you must identify the file to establish a working configuration.

**Error identifier in temporaryPluginInstallLog.txt**

INSTCONFFAILED : NON \_ EXISTENT \_ WS \_ CONFIG \_ FILE.

**Valid Web server configuration file 2****Default directive setting**

-P pluginSettings.webServerConfigFile2=""

Valid file names for affected Web servers are:

- Domino 5: names.nsf
- Domino 6: names.nsf
- Sun Java System Web Server: magnus.conf

**Valid setting**

Validation checks that the file exists. A known problem in ISMP validates a directory specification. However, you must identify the file to establish a working configuration.

**Error identifier in temporaryPluginInstallLog.txt**

INSTCONFFAILED : NON \_ EXISTENT \_ WS \_ CONFIG \_ FILE2.

**Valid Web server definition name****Default directive setting**

-P pluginSettings.webServerDefinition="webserver1"

**Valid setting**

Validation verifies that spaces do not exist in the Web server definition name. If spaces exist, validation fails.

**Error identifier in temporaryPluginInstallLog.txt**

INSTCONFFAILED : INVALID \_ WEB \_ SERVER \_ DEFINITION \_ NAME.

**Verify that application mapping is true or false****Default directive setting**

-P pluginSettings.mapWebserverToApplications="true"

**Valid setting**

If you set the directive to a value other than true or false, no error is generated or recorded in the temporaryPluginInstallLog.txt file. The Plug-ins installation wizard defaults the value to true and continues the installation.

**Error identifier in temporaryPluginInstallLog.txt**






None

**Valid Web server****Default directive setting**

-P pluginSettings.webServerSelected="ihs"

**Valid setting**

You must specify the correct Web server for your operating system. Valid values include:

-  none, ihs, apache, domino6, sunone
-  none, ihs, apache, domino6, sunone
-  none, ihs, apache, domino5, sunone
-  none, ihs, apache, domino6, sunone
-  none, ihs, apache, domino6, iis5, iis6, sunone

**Error identifier in temporaryPluginInstallLog.txt**

INSTCONFFAILED : INVALID \_ WEBSERVER \_ SELECTED.

**Usage notes**

- The file is not a read-only file.
- Edit this file directly with your flat file editor of choice, such as WordPad on a Windows platform.
- The file is updated when you specify the -options parameter when using the Plug-ins installation wizard.
- The file must exist to perform a silent installation. The installation program reads this file to determine installation option values when you install silently. Provide the fully qualified file path.
- Save the file in a location that you can identify when you specify the fully qualified path as part of the installation command.

## Example responsefile.txt file

Edit the version of the file that ships with your WebSphere Application Server product. The following example is not guaranteed to be an accurate representation of the file that ships with the product.

```
#####
#
# A common use of an options file is to run the wizard in silent mode. This lets
# the options file author specify wizard settings without having to run the
# wizard in graphical or console mode. To use this options file for silent mode
# execution, use the following command line arguments when running the wizard:
#
#   install -options "responsefile.txt" -silent
#
#####
# Note: Synchronous Silent Installs
# -----
# On non-Windows operating systems, silent installs will execute in a synchronous process.
# The process will not return until the silent install has completed.
#
# For the silent install to run synchronously on Windows, you must run the
# executable in the following way:
#
#   START /WAIT install.exe -options "response_local.txt" -silent
#
#####

#####
#
# TROUBLE SHOOTING TIP
#
# If no signs of an install are visible, reference the temporary log file
# (temporaryPluginInstallLog.txt)
# in the user's temporary directory for signs of cause.
#

#####
# Note: Default Web server Plugins install locations
# -----
#
# Below are the default install locations for each supported operating system.
#
# AIX:  "/usr/IBM/WebSphere/Plugins"
#
# HP-UX, Linux, and Solaris: "/opt/IBM/WebSphere/Plugins"
#
# Windows: "C:\Program Files\IBM\WebSphere\Plugins"
#
# Note: On unix operating systems, spaces in the install location is not supported.
#####

#####
#
# License Acceptance
#
# Valid Options : true  Accepts the license. Will install the product.
#                 false Declines the license. Install will not occur.
#
# If no install occurs, this will be logged to a temporary log file
# (temporaryPluginInstallLog.txt)
# in the user's temporary directory.
#
# By changing the silentInstallLicenseAcceptance.value in this response file to "true",
# you agree that you
# have reviewed and agree to the terms of the IBM International Program License Agreement
# accompanying this
# program, which is located at CD_ROOT\plugins\lafiles. If you do not agree to these terms,
```

```
# do not change
# the value or otherwise download, install, copy, access, or use the program and promptly
# return the
# program and proof of entitlement to the party from whom you acquired it to obtain a refund
# of the amount you paid.
#
#
```

```
-W silentInstallLicenseAcceptance.value="false"
```

```
#####
#
# install Type
#
# Valid Options : local  WebSphere Application Server and Web server on same machine
#                 remote WebSphere Appliication Server and Web server on seperate machines.
#
```

```
-P pluginSettings.installType="local"
```

```
#####
#
# pluginDestination
#
# Web server Plugins for IBM WebSphere Application Server V6 install location
#
```

```
-P pluginProductBean.installLocation="C:\Program Files\IBM\WebSphere\Plugins"
```

```
#####
#
# WAS V6 Existing Location
#
# Valid Options : Existing WebSphere Application Server Version 6.0 install home directory.
#
# Note : This option is valid for local install type.
#        The install will use the directory entered below.
#
```

```
-W websphereLocationWizardBean.wasExistingLocation="C:\Program Files\IBM\WebSphere\AppServer"
```

```
#####
#
# Web server to configure
#
# Valid options
# : none  Install binaries only. No Web server configuration.
# : ihs   IBM HTTP Server V6
# : apache Apache Web Server V2
# : domino5 Lotus Domino Web Server V5 (supported on HP-UX)
# : domino6 Lotus Domino Web Server V6 or V6.5 (not supported on HP-UX)
# : sunone Sun ONE Web Server 6.0 or Sun Java System Web Server V6.1
# : iis5 Microsoft Internet Information Services V5 (supported on Windows)
# : iis6 Microsoft Internet Information Services V6 (supported on Windows)
#
# Note : Specify only one Web server to configure.
#
```

```
-P pluginSettings.webServerSelected="ihs"
```

```
#####
#
# Web server Configuration File 1
#
# Valid options for Web server configuration file 1
#
# ihs : httpd.conf
```

```

# apache : httpd.conf
# domino5 : Notes.jar
# domino6 : Notes.jar
# sunone : obj.conf
#
# Note : File must exist
#

-P pluginSettings.webServerConfigFile1="C:\Program Files\IBM HTTP Server\conf\httpd.conf"

#####
#
# Web server Configuration File 2
#
# Valid options for Web server configuration file 2
#
# domino5 : names.nsf
# domino6 : names.nsf
# sunone : magnus.conf
#
# Note : File must exist
#

-P pluginSettings.webServerConfigFile2=""

#####
#
# Web server port number
#
# Specify the Web server port for the Web server selected to be configured.
#

-P pluginSettings.portNumber="80"

#####
#
# Domino 6 User ID
#
# Specify the Domino 6 User ID. (only specify value when configuring Domino 6 on
# Linux, Solaris, and AIX platforms)
#
# Note : Domino 6 is supported on Windows, but this value is not required on Windows.
#

-W domino6UserIDPanel.userID="notes"

#####
#
# Web server Definition Name
#
# A web server definition allows for Web server administration through the WebSphere
# admin console.
#
# Note : No spaces are allowed in the Web server definition name.
#

-P pluginSettings.webServerDefinition="webserv1"

#####
#
# plugin-cfg.xml File Location
#
# This file will be generated by the plugin installer.
#
# Valid options:

```

```

# ""      : leaving the string empty will result in installer generating the plugin-cfg.xml
#      file location at install time and configuring Web server to use this location.
#      This is the recommended option.
#
# "<file_location>" : User may enter an existing file location. Web server will be
#      configured to use this existing plugin-cfg.xml file location.
#      If file is specified, it must exist, otherwise install will not proceed.
#

-P pluginSettings.pluginCfgXmlLocation=""

#####
#
# WebSphere Application Server Machine HostName
#
# remote install type : enter the hostname of the WebSphere Application Server machine.
# local install type : hostname of machine being installed to will be used.
#

-P pluginSettings.wasMachineHostName=""

#####
#
# Advanced User Options available in silent installs only
#

# Map all the existing deployed applications to the Web server.
#
# Valid Options
# true : Web server Definition is mapped as a target to
#       the existing deployed applications such as snoop and hitcount
#       (Recommended)
# false : No applications are mapped to the Web server definition.
#
# Note : If not set to a valid option of true or false, the installer will set to
#       true and continue install.
#

-P pluginSettings.mapWebserverToApplications="true"

# Web server Hostname
#
# In advanced scenarios where a user has multiple Web server hostnames on a machine,
# set the entry
# below to the Web server hostname used to configure.
#
# Valid Options :
# "" : Install will resolve to hostname detected on machine (Recommended)
# "<HOSTNAME>" : Where <HOSTNAME> is a Web server hostname on the machine.
#

-P pluginSettings.webServerHostName=""

```

## Uninstalling the Web server plug-ins for WebSphere Application Server

You can uninstall Web server plug-ins for WebSphere Application Server without uninstalling the supported Web server or the Application Server.

This topic describes uninstalling the Web server plug-ins for WebSphere Application Server.

If you used the IBM Key Management wizard to create SSL key files in the Web server Plug-ins home directory, back up the files to a directory outside of the Web server Plugins directory. After the uninstall procedure is complete, you can delete the SSL key files if they are no longer required.



After uninstalling the Web server plug-ins for WebSphere Application Server, you can reinstall them using any of the installation procedures described in “Installing Web server plug-ins” on page 136

1. Open a command window.
2. Change directories to the `_uninstPlugin` directory in the installation root directory of the Web server plug-ins for WebSphere Application Server.  
This procedure occasionally refers to the installation root directory as the `plugins_install_root` directory.
3. Issue the **uninstall** command. The Uninstall wizard displays a welcome panel.
4. Click **Next** on the Welcome panel. The Uninstaller wizard displays a confirmation panel for you to confirm what is to be uninstalled. Each feature that displays in the panel is a separate Web server installation. Each Web server installation is configured to use one of the binary plug-in modules that is being deleted.
5. Click **Next** on the confirmation panel to begin uninstalling the plug-ins. The Uninstaller wizard displays a summary panel that provides status.
6. Click **Finish** to close the Uninstaller wizard.


Uninstalling the Web server plug-ins for WebSphere Application Server removes many files in the installation root directory, but leaves the following logs in the `plugins_install_root/log/uninstall` directory:

- ISMP uninstall log: `log.txt`
- Configuration uninstall log: `masterConfigurationLog.txt`
- Web server deconfiguration log: `uninstallWeb_server_namePlugin.log`, such as the `uninstallApachePlugin.log`.

7.   Uninstall the IBM Global Services Kit (GSKit) on Linux and UNIX systems.

The Plug-ins installation wizard does not uninstall GSKit 7 because there is no registry describing products that are enrolled to use the GSKit. Without this critical information, uninstalling the GSKit product might affect other products that use GSKit.

You can uninstall GSKit once it is no longer in use on your systems.

 On Solaris systems, you must also uninstall GSKit 4 in addition to GSKit 7.

8.  Uninstall GSKit on Windows systems.

The Plug-ins uninstaller program unregisters the GSKit. The registry key is **HKEY\_LOCAL\_MACHINE > SOFTWARE > IBM > GSK7 > REGAPPS > WASPlugins60\_unique\_key**.

If the registry key `WASPlugins60_unique_key` is the last remaining key in the GSKit registry entry, the wizard also uninstalls the GSK7 product. If another product, such as IBM HTTP Server, is registered to use GSKit, the wizard does not uninstall the GSKit. The wizard always unregisters the registry key for the Web server plug-ins for WebSphere Application Server, which is `WASPlugins60_unique_key`.

The GSKit uninstall log is the `plugins_install_root/log/uninstall/uninstallGSKit.log` file.

9. Delete files from the temporary directory.

Delete the following files:

- Install temporary log : `temporaryPluginInstallLog.txt`
- Uninstall temporary log : `temporaryPluginUnInstallLog.txt`

10. Delete the Web server definition in a stand-alone application server.

The uninstaller program for the Web server plug-ins for WebSphere Application Server does not delete Web server definitions. However, you can delete a Web server definition using the following `wsadmin` commands:

```
$AdminTask deleteServer { -serverName webserver1 -nodeName webserver1_node }
$AdminTask removeUnmanagedNode { -nodeName webserver1_node }
$AdminConfig save
```






After you exit from the *plugins\_install\_root/\_uninstPlugin* directory, the directory is removed. The only remaining directory is the *plugins\_install\_root/logs* directory. The logs directory contains the install process log, the uninstall process log, and the Web server creation logs.

**Important:** SSL key files might exist if you used the IBM Key Management Wizard. Move these files to a safe location before using the manual uninstalling procedure, if necessary.

To reinstall the Web server plug-ins for WebSphere Application Server, launch the installation procedure again.

To reinstall the Web server plug-ins for WebSphere Application Server into the original directory, delete the existing installation root directory for the plug-ins before reinstalling.

The default location is:

-  /usr/IBM/WebSphere/Plugins
-    /opt/IBM/WebSphere/Plugins
-  C:\Program Files\IBM\WebSphere\Plugins

## Manually uninstalling Web server plug-ins for WebSphere Application Server

You can use this manual procedure for uninstalling Web server plug-ins for WebSphere Application Server when the uninstaller program is not available for some reason.

This topic describes uninstalling the Web server plug-ins for WebSphere Application Server.

After uninstalling the Web server plug-ins for WebSphere Application Server, you can reinstall them using any of the installation procedures that are described in “Installing Web server plug-ins” on page 136

1. Delete the installation root directory for Web server plug-ins for WebSphere Application Server.
2. Delete the Web server definition in the application server configuration.

You can delete a Web server definition using the following wsadmin commands:

```
$AdminTask deleteServer { -serverName webserver1 -nodeName webserver1_node }
$AdminTask removeUnmanagedNode { -nodeName webserver1_node }
$AdminConfig save
```

3. Reconfigure any Web servers that were configured to use the binary plug-ins that you deleted.  
See “Editing Web server configuration files” on page 203.
4. Delete the following registry keys using the platform-specific steps described in “Uninstalling manually” on page 325.

Operating system	Registry keys
AIX, Linux and Windows	WSPAA60, WSPAA60DefineglobalconstantsComponent, WSPAA60DefinelocalvariablesComponent, WSPAA60LicensingComponent, WSPAA60Webserverplugins, WSPAA60WebserverpluginsComponent, WSPAA60AddBytes, WSPAA60gskit, WSPAA60gskitComponent
HP-UX	WSPAA60, WSPAA60DGCC, WSPAA60DLVC, WSPAA60LC, WSPAA60WSPC, WSPAA60AddBytesHS, WSPAA60gskitHP, WSPAA60gskitHPC, WSPAA60jdkHP
Solaris	WSPAA60, WSPAA60AC, WSPAA60BC, WSPAA60CC, WSPAA60DC, WSPAA60FC, WSPAA60FB, WSPAA60GC, WSPAA60HC

Deleting the installation root directory for the Web server plug-ins for WebSphere Application Server removes the binary plug-ins. Any Web servers that are configured to use the deleted binary modules do not work. Reinstall the Web server plug-ins for WebSphere Application Server and reconfigure the Web servers to restore their functionality.

## Editing Web server configuration files

The Plug-ins installation wizard automatically configures supported Web servers during the installation of the binary plug-in modules. Use this topic to understand how the wizard configures the Web server configuration files.

If you must change a configuration for some reason, you can run the Plug-ins installation wizard again to reconfigure the Web server, or you can edit the files.

The recommended approach is to always use the Plug-ins installation wizard to configure the Web server configuration file. However, sometimes you must edit the files. An example is when you are installing a Web server definition for a non-default profile.

To support a non-default profile, edit the configuration to point the Web server to the correct location of the plug-in configuration file (`plugin-cfg.xml`). In this case, you would change the profile path from the default profile to the secondary profile.

This task shows you how to edit the Web server configuration files.

Select one of the following links that identifies the Web server that you must reconfigure.

- Configure Apache HTTP Server 2.0. See “Configuring Apache HTTP Server V2.0.”
- Configure Domino Web Server Version 5 and Version 6.x. See “Configuring Lotus Domino” on page 205.
- Configure IBM HTTP Server Version 6.0. See “Configuring IBM HTTP Server Version 6.0” on page 210.
- Configure IBM HTTP Server Version 2.0. See “Configuring IBM HTTP Server powered by Apache 2.0” on page 208.
- Configure Microsoft Internet Information Services (IIS). See “Configuring Microsoft Internet Information Services (IIS)” on page 212.
- Configure Sun Java System Web Server (formerly Sun ONE and iPlanet). See “Configuring the Sun Java System Web Server” on page 215.

## Configuring Apache HTTP Server V2.0

This topic describes how to change configuration settings for Apache HTTP Server Version 2.0.

When you install the Web server plug-ins for WebSphere Application Server, as described in “Installing Web server plug-ins” on page 136, the Plug-ins installation wizard configures the Web server. This topic describes how to configure the Apache HTTP Server Version 2.0 Web server. Other procedures in “Editing Web server configuration files” describe configuring other supported Web servers.

Testing verified that the WebSphere Application Server plug-in for Apache HTTP Server, Version 2.0 works with Apache HTTP Server Version 2.0.48, the latest version of Apache 2.0. The plug-in was tested with the threaded worker multi-processing module (MPM) on all platforms except Windows. The plug-in was tested with the default threaded MPM on Windows.

The plug-in works with the Apache 2 prefork MPM but works best with the worker MPM. The plug-in maintains connection pools to backend WebSphere Application Servers and uses in-memory caching. These plug-in functions perform most efficiently when Apache 2.0 is configured to use a single child process with the `ThreadsPerChild` value equal to the `MaxClients` value. The plug-in can be used with the prefork MPM or the worker MPM that is configured with multiple child processes, but at reduced efficiency.

**Compatibility Statement** The plug-in works with versions of the Apache HTTP Server that claim full binary compatibility with Apache 2.0.42 and later, which are built with compilers and compiler options that are compatible with those used to build the plug-in.

Perform the step that configures Apache 2.0 for your operating system.

Examples and messages are shown on more than one line for clarity. Place each directive in a Web server configuration file on one line.

Local file path means a file path to the `plugin-cfg.xml` file on an Application Server that is on the same machine as the Web server. Remote file path means the file path to the `plugin-cfg.xml` file when the Application Server is on a remote machine. The Plug-ins installation wizard installs a dummy `plugin-cfg.xml` file during installation, but this file requires periodic propagation from the real file on the Application Server machine.

The `node_name` in the following Application Server local file paths is `Web_server_name_node` for a stand-alone Application Server or `managed_node_name` for a managed node. In each case, the remote file path is also shown.

The name of the Web server definition is `webserver1`.

- ▶ **AIX** Configure entries in the `httpd.conf` file.

Use the following examples of the `LoadModule` and the `WebSpherePluginConfig` directives as models for configuring your file:

```
LoadModule
  was_ap20_module /usr/IBM/WebSphere/Plugins/bin/mod_was_ap20_http.so
```

**Local distributed example:**

```
WebSpherePluginConfig
  /usr/IBM/WebSphere/AppServer/profiles/custom01/config/cells/
  dmgrcell/nodes/managednode/servers/webserver1/plugin-cfg.xml
```

**Local stand-alone example:**

```
WebSpherePluginConfig
  /usr/IBM/WebSphere/AppServer/profiles/default/config/cells/
  sa_cell/nodes/webserver1_node/servers/webserver1/plugin-cfg.xml
```

**Remote example:**

```
WebSpherePluginConfig
  /usr/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
```

- ▶ **Linux** ▶ **Solaris** Configure entries in the `httpd.conf` file.

Use the following examples of the `LoadModule` and the `WebSpherePluginConfig` directives as models for configuring your file:

```
LoadModule
  was_ap20_module /opt/IBM/WebSphere/Plugins/bin/mod_was_ap20_http.so
```

**Local distributed example:**

```
WebSpherePluginConfig
  /opt/IBM/WebSphere/AppServer/profiles/custom01/config/cells/
  dmgrcell/nodes/managednode/servers/webserver1/plugin-cfg.xml
```

**Local stand-alone example:**

```
WebSpherePluginConfig
  /opt/IBM/WebSphere/AppServer/profiles/default/config/cells/
  sa_cell/nodes/webserver1_node/servers/webserver1/plugin-cfg.xml
```

**Remote example:**

```
WebSpherePluginConfig
  /opt/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
```

- **HP-UX** Configure entries in the `httpd.conf` file.  
Use the following examples of the `LoadModule` and the `WebSpherePluginConfig` directives as models for configuring your file:

```
LoadModule
    was_ap20_module /opt/IBM/WebSphere/Plugins/bin/mod_was_ap20_http.s1
```

**Local distributed example:**

```
WebSpherePluginConfig
    /opt/IBM/WebSphere/AppServer/profiles/custom01/config/cells/
    dmgrcell/nodes/managednode/servers/webserver1/plugin-cfg.xml
```

**Local stand-alone example:**

```
WebSpherePluginConfig
    /opt/IBM/WebSphere/AppServer/profiles/default/config/cells/
    sa_cell/nodes/webserver1_node/servers/webserver1/plugin-cfg.xml
```

**Remote example:**

```
WebSpherePluginConfig
    /opt/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
```

- **Windows** Configure entries in the `httpd.conf` file.

Use the following examples of the `LoadModule` and the `WebSpherePluginConfig` directives as models for configuring your file:

```
LoadModule was_ap20_module
    drive:\IBM\WebSphere\Plugins\bin\mod_was_ap20_http.dll
```

**Local distributed example:**

```
WebSpherePluginConfig
    C:\IBM\WebSphere\AppServer\profiles\default\config\cells\
    dmgrcell\nodes\managednode\servers\webserver1\plugin-cfg.xml
```

**Local stand-alone example:**

```
WebSpherePluginConfig
    C:\IBM\WebSphere\AppServer\profiles\default\config\cells\
    sa_cell\nodes\webserver1_node\servers\webserver1\plugin-cfg.xml
```

**Remote example:**

```
WebSpherePluginConfig
    C:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml
```

This procedure results in reconfiguring the Apache 2.0 Web server.

## Configuring Lotus Domino

This task describes how to change configuration settings for Lotus Domino.

When you install the Web server plug-ins for WebSphere Application Server, as described in “Installing Web server plug-ins” on page 136, the Plug-ins installation wizard configures the Web server.

This topic describes how to reconfigure the Domino V6 Web server. Other procedures in “Editing Web server configuration files” on page 203 describe reconfiguring other supported Web servers.

Use the following procedure to enable the Web server plug-in to work with Lotus Domino.

1. Start the Domino server.
2. Access the `names.nsf` file using your Web browser (for example, `http://tarheels2.raleigh.ibm.com/names.nsf`). The browser prompts you for a password.
3. Give the administrator name and password.
4. Click **Configuration** on the left side of the page.
5. Click **Servers** on the left side of the page.
6. Click **All Server Documents** on the left side of the page.

7. Click the server that you intend to have work with WebSphere Application Server, Version 6.
8. Click **Edit Server** on the top-left of the center window.
9. Click **Internet Protocols** in the middle of the page.
10. Add the path to the Domino plug-in under the DSAPI Section in the middle-right of the page. The plug-in is installed in the bin directory of the Web server plug-ins for WebSphere Application Server. If specifications for filter files for the Domino Server Application Programming Interface (DSAPI) already exist, use a space to delimit the Web server plug-in for WebSphere Application Server.
11. Click **Save and Close** in the upper-left of the center window.
12. Define the location of the plugin-cfg.xml configuration file.

The location varies depending on how you have configured your system. If the Web server and the Application Server are on separate machines, you have a remote installation.

If the two servers are on the same machine, you have a local installation.

If the two servers are on the same machine and the Application Server node or the custom node is federated, you have a local distributed installation.

In the following examples, webserver1 is the Web server definition name. **Setting the path to the plug-in configuration file on a Linux or UNIX system**

Set the WAS\_PLUGIN\_CONFIG\_FILE environment variable to the location of the plug-in configuration file using one of the following paths:

Linux UNIX

If the type of installation is:	Then use this command to set the environment variable:
Remote	WAS_PLUGIN_CONFIG_FILE=/opt/ IBM/ WebSphere/ Plugins/ config/ webserver1/ plugin-cfg.xml
Local stand-alone	WAS_PLUGIN_CONFIG_FILE=/opt/ IBM/ WebSphere/ AppServer/ profiles/ default/ config/ cells/ sa_cell/ nodes/ <b>webserver1_node</b> / servers/webserver1/ plugin-cfg.xml
Local distributed	WAS_PLUGIN_CONFIG_FILE= /opt/ IBM/ WebSphere/ AppServer/ profiles/ custom01/ config/ cells/ dmgrcell/ nodes/ <b>managednode</b> / servers/ webserver1/ plugin-cfg.xml

Linux UNIX

During the installation process, the Plug-ins installation wizard creates the setupPluginCfg.sh file in two places:

- The *plug-ins\_install\_root/bin* directory
- The *lotus\_install\_root/notesdata* directory

Linux UNIX

You can run the script from either location to set the WAS\_PLUGIN\_CONFIG\_FILE environment variable. However, if you are reconfiguring the Web server, you might want to set the path yourself by setting the value of the environment variable with a path from the preceding table.

Linux UNIX

The setupPluginCfg.sh script sets the file path value to the file path that the wizard configured originally. If you are reconfiguring the Web server to change the original file path, do not use this script. **Setting the path to the plug-in configuration file on a Windows system**

Windows

Add the appropriate statement to your *lotus\_domino\_install\_root\notes.ini* file:



## Windows

If the type of installation is:	Then use this command to set the WebSpherePluginCfg variable:
Remote	WebSpherePluginCfg="C:\ Program Files\ IBM\ WebSphere\ Plugins\ config\ webserver1\ plugin-cfg.xml"
Local stand-alone	WebSpherePluginCfg="C:\ IBM\ WebSphere\ AppServer\ profiles\ default\ config\ cells\ sa_cell\ nodes\ <b>webserver1_node</b> \ servers\ webserver1\ plugin-cfg.xml"
Local distributed	WebSpherePluginCfg="C:\ IBM\ WebSphere\ AppServer\ profiles\ default\ config\ cells\ dmgrcell\ nodes\ <b>managednode</b> \ servers\ webserver1\ plugin-cfg.xml"

13. Restart the Domino server. When the server starts, information similar to the following example is displayed:

```
01/21/2005 01:21:51 PM JVM: Java virtual machine initialized
WebSphere Application Server DSAPI filter loaded
01/21/2005 01:21:52 PM HTTP Web Server started
```

This procedure results in reconfiguring Version 6.x of Lotus Domino.

After configuring a Web server, you can install applications on it. See the Applications section of the information center for more information.

For more information on configuring Lotus Domino to work with WebSphere Application Server, search the Lotus Support Services Web site. Enter the search term WebSphere in the keyword search field.

Related technotes:




- Enabling single sign-on for Domino and WebSphere Application Server
- Extended Search installation: Using horizontal clusters in WebSphere Application Server
- Prompted to authentication again when using single sign-on (SSO) for WebSphere Application Server and Domino
- Some CGI Variables do not work with DSAPI filter, Domino, and WebSphere Application Server
- WebSphere Application Server plug-in can hang when Domino encounters HTTP Error 500
- Must I install WebSphere Application Server Network Deployment before Lotus Workplace Messaging?

### **Lotus Domino tips:**

This topic describes configuration file locations, a Solaris configuration tip, and how to avoid a DSAPI filter-loading error when starting the Lotus Domino Web server as a non-root user on Linux and UNIX operating systems.

### **Lotus Domino Server file locations**

Lotus Domino server file locations are:

-  On AIX systems:
  - /usr/lotus/notes/50091/ibmpow/Notes.jar
  - /usr/notesdata/names.nsf
-  On the Solaris Operating Environment:
  - /opt/lotus/notes/5080/sunspa/Notes.jar
  - /opt/notesdata/names.nsf
-  On Windows systems:
  - c:\Program Files\lotus\notes\Notes.jar
  - c:\Program Files\lotus\notes\data\names.nsf

## The Domino Server plug-in might fail to configure on a Solaris platform

If this occurs during installation, a `dsapi_stderr.txt` file is created in the logs directory and you get the following error messages:

```
lotus.notes.NotesException: Could not load dll for system name SUNOS
    at lotus.notes.NotesThread.load(NotesThread.java:210)
    at lotus.notes.NotesThread.<clinit>(NotesThread.java:24)
java.lang.UnsatisfiedLinkError: NnotesInitThread
    at lotus.notes.NotesThread.NnotesInitThread(Native Method)
    at lotus.notes.NotesThread.initThread(NotesThread.java:99)
    at lotus.notes.NotesThread.run(NotesThread.java:133)
```

You can configure the WebSphere Application Server or Domino Server plug-in manually using the Domino Server Web administration tool. Use the following procedures:

1. Start the Domino Server.
2. Enter the URL for the Domino Server Web Administration site using a browser. For example, `http://host_name/names.nsf`. Enter the administrator user name and password.
3. Double-click **Server-Servers**.
4. Double-click **WebServer** to configure.
5. Double-click **Edit Server**.
6. Double-click **Internet Protocol**.
7. Add the WebSphere Application Server DSAPI plug-in to the **DSAPI** field. For example, `/opt/WebSphere/AppServer/bin/libdomino5_http.so` Note: If there are already DSAPI filter files specified use a space to delimit the WebSphere plug-in file.
8. Double-click **Save and Close**.
9. Restart the Domino Server.

## Avoiding a DSAPI filter-loading error when the Lotus Domino Server starts

On a Linux or UNIX operating system, if the Lotus Domino Web server starts using a non-root user, you are likely to generate a DSAPI filter-loading error when the Lotus Domino Server starts:

```
Error loading DSAPI filter.
```

```
Filter not loaded: /usr/WebSphere/AppServer/bin/libdomino6_http.a
```

Manually change the WebSphere Application Server `bin` directory permissions from 750 to 755 to run Lotus Domino Server as a non-root user and not generate the error.

You must also change permissions on the WebSphere Application Server `logs` directory to 777 to allow Lotus Domino Server to write to the log. However, this change poses a security risk.

If the Lotus Domino Server is started as root, the problem does not occur.

## Configuring IBM HTTP Server powered by Apache 2.0

This topic describes how to change configuration settings for IBM HTTP Server powered by Apache 2.0.

When you install the Web server plug-ins for WebSphere Application Server, as described in “Installing Web server plug-ins” on page 136, the Plug-ins installation wizard configures the Web server. This topic describes how to configure the IBM HTTP Server, V2.x. Other procedures in “Editing Web server configuration files” on page 203 describe configuring other supported Web servers.

Perform the step that configures IBM HTTP Server 6.0 version for your operating system.

Examples and messages are shown on more than one line for clarity. Place each directive in a Web server configuration file on one line.



Local file path means a file path to the `plugin-cfg.xml` file on an Application Server that is on the same machine as the Web server. Remote file path means the file path to the `plugin-cfg.xml` file when the Application Server is on a remote machine. The Plug-ins installation wizard installs a dummy `plugin-cfg.xml` file during installation, but this file requires periodic propagation from the real file on the Application Server machine.

The Web server definition name in the following examples is `webserver1`.

The node name `node_name` in the following Application Server local file paths is `Web_server_name_node` for a stand-alone Application Server or `managed_node_name` for a managed node. In each case, the remote file path is also shown.

-  Configure entries in the `httpd.conf` file.

Use the following examples of the `LoadModule` and the `WebSpherePluginConfig` directives as models for configuring your file:

```
LoadModule
    was_ap20_module /usr/IBM/WebSphere/Plugins/bin/mod_was_ap20_http.so
```

**Local distributed example:**

```
WebSpherePluginConfig
    /usr/IBM/WebSphere/AppServer/profiles/custom01/config/cells/
    dmgrcell/nodes/managednode/servers/webserver1/plugin-cfg.xml
```

**Local stand-alone example:**

```
WebSpherePluginConfig
    /usr/IBM/WebSphere/AppServer/profiles/default/config/cells/
    sa_cell/nodes/webserver1_node/servers/webserver1/plugin-cfg.xml
```

**Remote example:**

```
WebSpherePluginConfig
    /usr/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
```

-   Configure entries in the `httpd.conf` file.

Use the following examples of the `LoadModule` and the `WebSpherePluginConfig` directives as models for configuring your file:

```
LoadModule
    was_ap20_module /opt/IBM/WebSphere/Plugins/bin/mod_was_ap20_http.so
```

**Local distributed example:**

```
WebSpherePluginConfig
    /opt/IBM/WebSphere/AppServer/profiles/custom01/config/cells/
    dmgrcell/nodes/managednode/servers/webserver1/plugin-cfg.xml
```

**Local stand-alone example:**

```
WebSpherePluginConfig
    /opt/IBM/WebSphere/AppServer/profiles/default/config/cells/
    sa_cell/nodes/webserver1_node/servers/webserver1/plugin-cfg.xml
```

**Remote example:**

```
WebSpherePluginConfig
    /opt/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
```

-  Configure entries in the `httpd.conf` file.

Use the following examples of the `LoadModule` and the `WebSpherePluginConfig` directives as models for configuring your file:

```
LoadModule
    was_ap20_module /opt/IBM/WebSphere/Plugins/bin/mod_was_ap20_http.sl
```

**Local distributed example:**


```
WebSpherePluginConfig
    /opt/IBM/WebSphere/AppServer/profiles/custom01/config/cells/
    dmgrcell/nodes/managednode/servers/webserver1/plugin-cfg.xml
```

### Local stand-alone example:

```
WebSpherePluginConfig
/opt/IBM/WebSphere/AppServer/profiles/default/config/cells/
sa_cell/nodes/webserver1_node/servers/webserver1/plugin-cfg.xml
```

### Remote example:

```
WebSpherePluginConfig
/opt/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
```

-  Configure entries in the httpd.conf file.

Use the following examples of the LoadModule and the WebSpherePluginConfig directives as models for configuring your file:

```
LoadModule was_ap20_module
drive:\IBM\WebSphere\Plugins\bin\mod_was_ap20_http.dll
```

### Local distributed example:

```
WebSpherePluginConfig
C:\IBM\WebSphere\AppServer\profiles\default\config\cells\
dmgrcell\nodes\managednode\servers\webserver1\plugin-cfg.xml
```

### Local stand-alone example:

```
WebSpherePluginConfig
C:\IBM\WebSphere\AppServer\profiles\default\config\cells\
sa_cell\nodes\webserver1_node\servers\webserver1\plugin-cfg.xml
```

### Remote example:

```
WebSpherePluginConfig
C:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml
```

This procedure results in editing and reconfiguring IBM HTTP Server powered by Apache 2.0.

If the IBM HTTP Server 1.3.2x directive, LoadModule ibm\_app\_server\_http\_module, is present in an IBM HTTP Server 2.0 httpd.conf file, the IBM HTTP Server cannot start. You must comment or delete the directive to start the Version 2 server.

The mod\_was\_ap20\_http plug-in module requires the GSKIT Secure Sockets Layer (SSL) encryption library if the plug-in is configured to support encrypted connections to back-end WebSphere Application Servers.

The Plug-ins installation wizard installs the GSKIT SSL encryption library at the required level if it is not installed. If you manually copy the plug-in to a new machine, you might not have the required GSKIT libraries for encrypting back-end connections. See the *Security Guide* PDF for more information on configuring GSKIT.

After configuring a Web server, you can install applications on it. See the Applications section of the information center for more information.

## Configuring IBM HTTP Server Version 6.0

This topic describes how to change configuration settings for IBM HTTP Server 6.0.

When you install the Web server plug-ins for WebSphere Application Server, as described in “Installing Web server plug-ins” on page 136, the Plug-ins installation wizard configures the Web server. This topic describes how to configure the IBM HTTP Server, V6. Other procedures in “Editing Web server configuration files” on page 203 describe configuring other supported Web servers.

Perform the step that configures IBM HTTP Server 6.0 for your operating system.

Examples and messages are shown on more than one line for clarity. Place each directive in a Web server configuration file on one line.

Local file path means a file path to the `plugin-cfg.xml` file on an Application Server that is on the same machine as the Web server. Remote file path means the file path to the `plugin-cfg.xml` file when the Application Server is on a remote machine. The Plug-ins installation wizard installs a dummy `plugin-cfg.xml` file during installation, but this file requires periodic propagation from the real file on the Application Server machine.

The `node_name` in the following Application Server local file paths is `Web_server_name_node` for a stand-alone Application Server or `managed_node_name` for a managed node. In each case, the remote file path is also shown.

- ▶ **AIX** Configure entries in the `httpd.conf` file.

Use the following examples of the `LoadModule` and the `WebSpherePluginConfig` directives as models for configuring your file:

```
LoadModule
  was_ap20_module /usr/IBM/WebSphere/Plugins/bin/mod_was_ap20_http.so
```

**Local distributed example:**

```
WebSpherePluginConfig
  /usr/IBM/WebSphere/AppServer/profiles/custom01/config/cells/
  dmgrcell/nodes/managednode/servers/webserver1/plugin-cfg.xml
```

**Local stand-alone example:**

```
WebSpherePluginConfig
  /usr/IBM/WebSphere/AppServer/profiles/default/config/cells/
  sa_cell/nodes/webserver1_node/servers/webserver1/plugin-cfg.xml
```

**Remote example:**

```
WebSpherePluginConfig
  /usr/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
```

- ▶ **Linux** ▶ **Solaris** Configure entries in the `httpd.conf` file.

Use the following examples of the `LoadModule` and the `WebSpherePluginConfig` directives as models for configuring your file:

```
LoadModule
  was_ap20_module /opt/IBM/WebSphere/Plugins/bin/mod_was_ap20_http.so
```

**Local distributed example:**

```
WebSpherePluginConfig
  /opt/IBM/WebSphere/AppServer/profiles/custom01/config/cells/
  dmgrcell/nodes/managednode/servers/webserver1/plugin-cfg.xml
```

**Local stand-alone example:**

```
WebSpherePluginConfig
  /opt/IBM/WebSphere/AppServer/profiles/default/config/cells/
  sa_cell/nodes/webserver1_node/servers/webserver1/plugin-cfg.xml
```

**Remote example:**

```
WebSpherePluginConfig
  /opt/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
```

- ▶ **HP-UX** Configure entries in the `httpd.conf` file.

Use the following examples of the `LoadModule` and the `WebSpherePluginConfig` directives as models for configuring your file:

```
LoadModule
  was_ap20_module /opt/IBM/WebSphere/Plugins/bin/mod_was_ap20_http.sl
```

**Local distributed example:**

```
WebSpherePluginConfig
  /opt/IBM/WebSphere/AppServer/profiles/custom01/config/cells/
  dmgrcell/nodes/managednode/servers/webserver1/plugin-cfg.xml
```

**Local stand-alone example:**

```
WebSpherePluginConfig
/opt/IBM/WebSphere/AppServer/profiles/default/config/cells/
sa_cell/nodes/webserver1_node/servers/webserver1/plugin-cfg.xml
```

**Remote example:**

```
WebSpherePluginConfig
/opt/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml
```

-  Configure entries in the httpd.conf file.

Use the following examples of the LoadModule and the WebSpherePluginConfig directives as models for configuring your file:

```
LoadModule was_ap20_module
drive:\IBM\WebSphere\Plugins\bin\mod_was_ap20_http.dll
```

**Local distributed example:**

```
WebSpherePluginConfig
C:\IBM\WebSphere\AppServer\profiles\default\config\cells\
dmgrcell\nodes\managednode\servers\webserver1\plugin-cfg.xml
```

**Local stand-alone example:**

```
WebSpherePluginConfig
C:\IBM\WebSphere\AppServer\profiles\default\config\cells\
sa_cell\nodes\webserver1_node\servers\webserver1\plugin-cfg.xml
```

**Remote example:**

```
WebSpherePluginConfig
C:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml
```

This procedure results in editing and reconfiguring the IBM HTTP Server 6.0.

If the IBM HTTP Server 1.3.x directive, LoadModule ibm\_app\_server\_http\_module, is present in an IBM HTTP Server 6.0 httpd.conf file, the IBM HTTP Server cannot start. You must comment or delete the directive to start the Version 6 server.

The mod\_was\_ap20\_http plug-in module requires the GSKIT Secure Sockets Layer (SSL) encryption library if the plug-in is configured to support encrypted connections to back-end WebSphere Application Servers.

The Plug-ins installation wizard installs the GSKIT SSL encryption library at the required level if it is not installed. If you manually copy the plug-in to a new machine, you might not have the required GSKIT libraries for encrypting back-end connections. See the *Security Guide* PDF for more information on configuring GSKIT.

After configuring a Web server, you can install applications on it. See the Applications section of the information center for more information.

## Configuring Microsoft Internet Information Services (IIS)

This topic describes manual configuration settings for Internet Information Services (IIS).

When you install the Web server plug-ins for WebSphere Application Server, as described in “Installing Web server plug-ins” on page 136, the Plug-ins installation wizard configures the Web server. This topic describes how to configure the Internet Information Services (IIS) Web server. Other procedures in “Editing Web server configuration files” on page 203 describe configuring other supported Web servers.

You must have read/write access to the *plugins\_install\_root* directory to perform this task.

Use the following procedure to manually reproduce how the Installation wizard configures the Microsoft Internet Information Services Web server.

- Configure IIS V5.0.

1. Start the IIS application and create a new virtual directory for the Web site instance that you intend to work with WebSphere Application Server. These instructions assume that you are using the Default Web Site.
  2. Expand the tree on the left until you see Default Web Site. Right-click **Default Web Site**, then click **New > Virtual Directory** to create the directory with a default installation.
  3. Type `sePlugins` in the Alias to be used to Access Virtual Directory field.
  4. Browse to the WebSphere Application Server `install_root\bin` directory in the Enter the physical path of the directory containing the content you want to publish field.
  5. Select the **Allow Execute Access** check box in the What access permissions do you want to set for this directory field.
  6. Click **Finish** to add the `sePlugins` virtual directory to your default Web site.
  7. Add the Internet Services Application Programming Interface (ISAPI) filter into the IIS configuration.
  8. Right-click the host name in the tree on the left and click **Properties**.  
In the Properties dialog, perform the following steps:
    - a. Click the Internet Information Services tab.
    - b. Click **WWW Service** in the Master properties window.
    - c. Click **Edit** to open the WWW Service master properties window.
    - d. Click **ISAPI Filters > Add** to open the Filter properties window.
    - e. Type `iisWASPlugin` in the Filter Name field.
    - f. Click **Browse** in the Executable field.
    - g. Browse to the `Plugins_install_root\bin` directory.
    - h. Click the `iisWASPlugin_http.dll` file.
    - i. Click **OK** until all the open windows close.
- Configure IIS V6.0.
    1. Start the IIS application and create a new virtual directory for the Web site instance that you intend to work with WebSphere Application Server. These instructions assume that you are using the Default Web Site.
    2. Expand the tree on the left until you see Default Web Site. Right-click **Default Web Site**, then click **New > Virtual Directory** to create the directory with a default installation.
    3. Type `sePlugins` in the Alias to be used to Access Virtual Directory field.
    4. Browse to the WebSphere Application Server `install_root\bin` directory in the Enter the physical path of the directory containing the content you want to publish field.
    5. Select the **Allow Execute Access** check box in the What access permissions do you want to set for this directory field.
    6. Click **Finish** to add the `sePlugins` virtual directory to your default Web site.
    7. Add the Internet Services Application Programming Interface (ISAPI) filter into the IIS configuration.
    8. Right-click **Default Web Site** in the tree on the left and click **Properties**. In the Properties panel, perform the following steps:
      - a. Click the Internet Information Services tab.
      - b. Click the ISAPI Filters tab.
      - c. Click **Add** to open the Filter properties window.
      - d. Type `iisWASPlugin` in the Filter Name field.
      - e. Click **Browse** in the Executable field.
      - f. Browse to the `plugins_install_root\bin` directory.
      - g. Click the `iisWASPlugin_http.dll` file.
      - h. Click **OK** until all the open windows close.

9. Copy the plug-in binaries to the directory C:\program files\WebSphere\Plugins\bin\iis\_webserverName The plugin-cfg.loc file resides in this directory. The first line of the plugin-cfg.loc file is the location of the plugin-cfg.xml file.

10. Set the value of the Plugin Config variable to the location of the configuration file, which is the *plugins\_install\_root\config\webserver\_name\plugin-cfg.xml* directory.

The location varies, depending on how you have configured your system. If the Web server and the Application Server are on separate machines, you have a remote installation.

If the two servers are on the same machine, you have a local installation.

If the two servers are on the same machine and the Application Server is a managed node, you have a local distributed installation.

**Local distributed example:**

```
"C:\IBM\WebSphere\AppServer\profiles\custom01\config\cells\
dmgrcell\nodes\managed_node\servers\webserver1\plugin-cfg.xml"
```

**Local stand-alone example:**

```
"C:\IBM\WebSphere\AppServer\profiles\default\config\cells\
sa_cell\nodes\webserver1_node\servers\webserver1\plugin-cfg.xml"
```

**Remote example:**

```
"C:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml"
```

11. Configure the Web server to run WebSphere Application Server extensions:

- a. Expand the tree on the left until you see **Web service extensions**.
- b. Click **Web service extensions**.
- c. Click **All Unknown ISAPI extensions** on the right side of the page.
- d. Click **Allow**.

• **Optional:** Configure multiple Web sites. Given:

- There are two Web sites defined: website1, website2.
  - The DLL files are already created as bin/website1/iisWASPlugin\_http.dll and bin/website2/iisWebsite2/iisWASPlugin\_http.dll.
  - The plugin-cfg.loc files are created in the same folder as the DLL file.
1. Run IIS in worker process isolation mode (default). To enable worker process in isolation mode:
    - a. Open the IIS management console and expand the local computer by clicking the plus sign.
    - b. Right-click the Web Sites folder, click Properties, and then click the Service tab.
    - c. Under Isolation mode, clear the Run Web service in IIS 5.0 isolation mode check box to enable worker process isolation mode.
  2. Define two application pools; one for website1 and the other for website2. Do not use the pre-defined application pool DefaultAppPool.
  3. Define the two Websites, including the filter setting, virtual host setting, and extension settings.
  4. Assign an application pool for each website.
    - a. Under each Web site folder, right click on the Web site name.
    - b. Click **Property**, and select the **Home Directory** tab. 2.
    - c. At the bottom of the application settings, select the application pool you defined for website1 from the drop-down list of application pools.
    - d. Click **OK**.
    - e. Repeat the previous steps for the second Web site and select the application pool you defined for website2.
  5. Start the IIS service and start each Web site.

This procedure results in reconfiguring the Internet Information Services (IIS) Web server.

## Configuring the Sun Java System Web Server

This topic describes how to change configuration settings for the Sun Java System Web Server (formerly Sun ONE Web Server and iPlanet Web Server), Version 6.0 and later.

When you install the Web server plug-ins for WebSphere Application Server, as described in “Installing Web server plug-ins” on page 136, the Plug-ins installation wizard configures the Web server. This topic describes how to configure the Sun Java System Web Server if you must change something in the existing configuration. Other procedures in “Editing Web server configuration files” on page 203 describe configuring other supported Web servers.

Configure the Sun ONE Web Server 6.0 or Sun Java System Web Server, V6.1 and later, on a Linux, UNIX, or Windows platform.

Examples and messages are sometimes shown on more than one line for ease of presentation. Verify that each directive in a Web server configuration file is on one line.

1. Configure entries in the `obj.conf` configuration file and in the `magnus.conf` configuration file for Version 6.0 and later of Sun Java System Web Server.

- a. Add two directives to the `obj.conf` file after the `<Object name=default>` tag:

```
Service fn="as_handler"  
AddLog fn="as_term"
```

- b. Add two directives at the end of the `magnus.conf` file:

The location for the `bootstrap.properties` directive varies, depending on how you have configured your system. If the Web server and the application server are on separate machines, you have a remote installation.

If the two servers are on the same machine, you have a local installation.

If the two servers are on the same machine and the application server is a managed node, you have a local distributed installation.

### Local distributed example:

```
Init fn="load-modules"  
  funcs="as_init,as_handler,as_term"  
  shlib="/opt/IBM/WebSphere/Plugins/bin/libns41_http.so"  
Init fn="as_init"  
  bootstrap.properties="/opt/IBM/WebSphere/AppServer/profiles/custom01/config/  
  cells/dmgrcell/nodes/managed_node/servers/webserver1/plugin-cfg.xml"
```

### Local stand-alone example:

```
Init fn="load-modules"  
  funcs="as_init,as_handler,as_term"  
  shlib="/opt/IBM/WebSphere/Plugins/bin/libns41_http.so"  
Init fn="as_init"  
  bootstrap.properties="/opt/IBM/WebSphere/AppServer/profiles/default/config/  
  cells/sa_cell/nodes/webserver1_node/servers/webserver1/plugin-cfg.xml"
```

### Remote example:

```
Init fn="load-modules"  
  funcs="as_init,as_handler,as_term"  
  shlib="/opt/IBM/WebSphere/Plugins/bin/libns41_http.so"  
Init fn="as_init"  
  bootstrap.properties="/opt/IBM/WebSphere/Plugins/config/webserver1/plugin-cfg.xml"
```

### Local distributed example:

```
Init fn="load-modules"  
  funcs="as_init,as_handler,as_term"  
  shlib="C:\IBM\WebSphere\Plugins\bin\ns41_http.dll"  
Init fn="as_init"  
  bootstrap.properties=  
  "C:\IBM\WebSphere\AppServer\profiles\custom01\config\cells\  
  dmgrcell\nodes\managed_node\servers\webserver1\plugin-cfg.xml"
```

### Local stand-alone example:



```

Init fn="load-modules"
    funcs="as_init,as_handler,as_term"
    shlib="C:\IBM\WebSphere\Plugins\bin\ns41_http.dll"
Init fn="as_init"
    bootstrap.properties=
        "C:\IBM\WebSphere\AppServer\profiles\default\config\cells\
        sa_cell1\nodes\webserver1_node\servers\webserver1\plugin-cfg.xml"

```

#### Remote example:

```

Init fn="load-modules"
    funcs="as_init,as_handler,as_term"
    shlib="C:\IBM\WebSphere\Plugins\bin\ns41_http.dll"
Init fn="as_init"
    bootstrap.properties="C:\IBM\WebSphere\Plugins\config\webserver1\plugin-cfg.xml"

```

2. Set the shared library path on HP-UX machines. On some installations of Sun Java System Web Server on an HP-UX machine, it is necessary to manually set the SHLIB\_PATH variable to /usr/lib before starting Sun Java System Web Server with a plug-in that is configured for Secured Sockets Layer (SSL). For example, in the korn shell, issue the following command before invoking the command to start the Sun Java System Web Server:

```
export SHLIB_PATH=/usr/lib:$SHLIB_PATH
```

3. Disable the feature of Sun Java System Web Server V6.1 that supports servlets and JavaServer Pages files by default. Disable this feature so that the WebSphere Application Server plug-in can handle the requests.

Perform the following steps to disable the feature:

- a. Remove or comment out the following two lines from the obj.conf configuration file:

```
NameTrans fn="ntrans-j2ee" name="j2ee"
Error fn="error-j2ee"
```

- b. Remove or comment out the following line from the magnus.conf configuration file:

```

Init fn="load-modules"
    shlib="C:/Sun/WebServer6.1/bin/https/bin/j2eeplugin.so"
    shlib_flags="(global|now)"

Init fn="load-modules"
    shlib="C:\Sun\WebServer6.1\bin\https\bin\j2eeplugin.dll"
    shlib_flags="(global|now)"

```

This procedure results in editing and reconfiguring the Sun Java System Web Server.

After configuring a Web server, you can install applications on it. See the Applications section of the information center for more information.

## Allowing Web servers to access the administrative console

This topic describes how to add the virtual host that servers the administrative console to the plug-in configuration file so that you can access the administrative console through a Web server.

Install your Version 6 WebSphere Application Server product, a Web server, and the Web server plug-ins for WebSphere Application Server. The Plug-ins installation wizard creates a Web server definition on the Application Server system, either directly when they are on the same machine, or by a script for remote scenarios. After creating the Web server definition, the plug-in configuration file exists within the Web server definition.

This task gives you the option of configuring the admin\_host so that Web servers can access the administrative console. When the Web server plug-in configuration file is generated, it does not include admin\_host on the list of virtual hosts.

1. Use the administrative console to change the admin\_host virtual host group to include the Web server port (80 by default).



- a. Click **Environment > Virtual Host > admin\_host > Host Aliases > New**. The default port that displays is 80, unless you specify a different port during profile creation.
  - b. Specify the IP address, or the name of the machine that is hosting the administrative console application. For example, if you installed a WebSphere Application Server product on a machine that is named `wastricia.rtp.ibm.com`, specify the name in this field.
2. Click **Apply > Save**.
  3. Stop and restart the application server. For example, to access the administrative console of a stand-alone application server, stop and restart the `server1` process. To stop `server1`, open a command window and navigate to the `install_root/profiles/default/bin` directory. Then issue the following command:

```
./stopServer.sh server1
```

After receiving the following message, you can restart the application server:

```
Server server1 stop completed.
```

To start the application server, issue the following command:

```
./startServer.sh server1
```

When you receive a message that is similar to the following message, the `server1` process is running:

```
Server server1 open for e-business; process id is 1719
```

4. Stop and restart a deployment manager. For example, to access the administrative console of a deployment manager, stop and restart the deployment manager. To stop the deployment manager, open a command window and navigate to the `install_root/profiles/profile_name/bin` directory. Then issue this command:

```
./stopManager.sh
```

After receiving the following message, you can restart the deployment manager:

```
Server dmgr stop completed.
```

To start the deployment manager, issue the following command:

```
./startManager.sh
```

When you receive a message that is similar to the following message, the deployment manager is running:

```
Server dmgr open for e-business; process id is 1720
```

5. Edit the `plugin-cfg.xml` file to include the following entries:

```
<VirtualHostGroup Name="admin_host">
  <VirtualHost Name="*:9060"/>
  <VirtualHost Name="*:80"/>
  <VirtualHost Name="*:9043"/>
</VirtualHostGroup>
...
...
...
<ServerCluster Name="server1_SERVER1HOSTserver1_Cluster">
  <Server LoadBalanceWeight="1" Name="SERVER1HOSTserver1_dmgr">
    <Transport Hostname="SERVER1HOST" Port="9060" Protocol="http"/>
  </Server>

  <PrimaryServers>
    <Server Name="SERVER1HOSTserver1_dmgr"/>
  </PrimaryServers>
</ServerCluster>
...
...
...
<UriGroup Name="admin_host_server1_SERVER1HOSTserver1_Cluster_URIs">
  <Uri AffinityCookie="JSESSIONID"
    AffinityURLIdentifier="jsessionid" Name="/ibm/console/*"/>
</UriGroup>
<Route ServerCluster="server1_SERVER1HOSTserver1_Cluster"
  UriGroup="admin_host_server1_SERVER1HOSTserver1_Cluster_URIs" VirtualHostGroup="admin_host"/>
```

Replace *SERVER1HOST* with the host name of your deployment manager. For example, if the host name is *wastricia*:

```
<ServerCluster Name="server1_wastriciaserver1_Cluster">
```

You can configure your supported Web servers to access the administrative console application of a deployment manager or a stand-alone application server.

---

## Installing Application Client for WebSphere Application Server

Running J2EE and Thin application clients that communicate with WebSphere Application Server requires that elements of the Application Server are installed on the machine on which the client runs. However, if the system does not have the Application Server installed, you can install Application Clients, which provide a stand-alone client run-time environment for your client applications. See the Supported Prerequisites page for more information on supported product platforms.

This article describes how to install the Application Client for WebSphere Application Server using the installation image on the product CD-ROM. The steps that follow provide enough detail to guide you through preparing for, choosing, and installing the variety of options and features provided. To prepare for installation and to make yourself familiar with installation options, complete the steps in this article and read the related topics, before you start to use the installation tools. Specifically, read these topics before installing the product:

- Installing silently
- Best practices for installing

As a general rule, if you launch an installation and there is a problem such as not having enough temporary space or not having the right packages on your Linux or UNIX systems, then cancel the installation, and make the required changes. Restart the installation to initiate your changes.

Although it is not supported or recommended, you can install this product as a non-root user on a UNIX operating system. However, you must use a user ID that is part of the administrator group on Windows platforms.

In Version 6, the Application Server for WebSphere Application Server is installable on a machine with a previous version of Application Clients. However, you cannot install a Version 6 Application Client on top of a previous version of the Application Client. For example, if a Version 5 Application Clients install under the `C:\WebSphere\AppClient` directory, you can not choose the same install location for your V6 Application Clients installation.

**Note:** For Application Client coexisting, there is a limitation on Applet client and ActiveX client on Windows that can not be coexisted with V5.0.x and V4.x of the clients. For example, the Applet client feature in Version 6 cannot coexist with the Applet client feature in any previous release. This coexistence is not available because the installation of Applet client feature in Version 6 sets the browser default Java Virtual Machine (JVM) using the Java Runtime Environment (JRE) from the Version 6 installation, which is Java Runtime Environment Version 1.4.2. Similarly, the ActiveX to EJB Bridge feature in Version 6 sets the Windows system path to use the JRE from the Version 6 installation.

1. Install Application Client for WebSphere Application Server using the launchpad. The launchpad program is available on the root directory of the product CD in the program, `1launchpad.sh`, on Linux and UNIX platforms and `1launchpad.bat` on Windows platforms.

**Note:** The free download Application Client installation is not packaged with the launchPad program.

- a. Click **Launch the installation wizard for Application Client for WebSphere Application Server** from the launchpad tool to launch the InstallShield for MultiPlatforms installation wizard. This action launches the installation wizard.

The Readme documentation to which the launchpad links is the readme.html file in the CD root directory. The readme directory off the root of the CD has more detailed Readme files. The Installation Guide is in the /docs directory of the CD root directory.

**Note:** Readme file names are based on product offerings.

When you install application clients, the current working directory must be the directory where the installer binary program is located. This placement is important because the resolution of the class files location is done in the current working directory. For example:

```
cd /install_root/AppClient

./install

or

cd <CD mount point>/AppClient

    ./install
```

Failing to use the correct working directory can cause ISMP errors that abort the installation.

The installation wizard does not upgrade or remove previous Application Clients installation automatically. However, you must change the directory of any previously installed versions of the Application Client because you cannot install Version 6 on top of previously existing Application Client.

- b. As indicated in the previous example, you can start the installation wizard from the product CD-ROM, using the command line.

On other Linux platforms and UNIX-based platforms, run the `./install` command.

On Windows platforms, run the `Install.exe` command.

- c. You can also perform a silent installation using the `-options responsefile` parameter, which causes the installation wizard to read your responses from the options response file, instead of from the interactive graphical user interface. Customize the response file before installing silently. After customizing the file, issue the command to silently install. Silent installation is particularly useful if you install the product often.

The rest of this procedure assumes that you are using the installation wizard. There are corresponding entries in the response file for every prompt that is described as part of the wizard. Review the description of the response file for more information. Comments in the file describe how to customize its options.

2. Click **Next** to continue when the Welcome panel is displayed.
  - a. Click the radio button beside the **I accept the terms in the license agreement** message if you agree to the license agreement, and click **Next** to continue.
3. Specify a destination directory. Click **Next** to continue.
  - a. Ensure that there is adequate space available in the target directory.
  - b. Specify a target directory for the Application Client product.
  - c. Enter the required target directory to proceed to the next panel. Deleting the default target location and leaving an installation directory field empty prevents you from continuing the installation process.
4. Choose a type of installation, and click **Next**.

If you use the GUI, you can choose a Typical installation type, which installs J2EE and Thin application client, Samples and IBM Developer Kit, Java 2 Technology Edition or a Custom installation type.

The Custom installation type lets you select which features to install. However you can not install the J2EE and Java Thin application client feature and the Pluggable application client feature together.

(Windows Only) If you select the **ActiveX to EJB Bridge** feature, then the following is displayed in a dialog box: Do you want to add Java runtime to the system path and make it the default JRE? If you answer **Yes**, then the Java run time is added to the beginning of the system path. If you answer **No**, then the ActiveX to EJB Bridge does not function from the Active Server Pages (ASP), unless you add the Java run time to the path. To add the Java run time later, see the topic ActiveX application

clients or reinstall the Application Client.(Windows Only) If you select the **Applet client** feature, then the following message might be displayed: An existing JDK or JRE has been detected on your computer. You chose to install the Applet Client, which will overwrite the registry entries for this JDK or JRE. Do you want to continue and install the Applet Client? If you select **Yes**, the installation overrides the registry on your machine. If you select **No**, the Applet client feature is not installed, and you are directed back feature dialog box.

5. Install the Software Developer Kit, if you need to use any of the utilities that it provides, such as the javacompiler, the jarutility or the jarsinger utility. The Java 2 Development Kit that IBM provides has two components, Java Runtime Environment (JRE) and a complete Software Developer Kit (SDK). The JRE sub feature is selected by default when the J2EE or Thin application client feature is selected. The SDK component is optional; however, you must install the SDK component to compile the sample.
6. Enter the host name of the WebSphere Application Server machine. Click **Next** to continue. The default port number for product Version 6 server is 2809.
7. Review the summary information, and click **Next** to install the product code or you might also click **Back** to change your specifications.
8. Click **Finish** to exit the wizard, after the Application Client installs.
9. Verify the success of the installer program by examining the Completion panel and the `<install_root>/logs/WAS.Client.install.log` file for installation status. The installer program records the following indicators of success in the logs:
  - INSTCONFSUCCESS indicates that the installation is successful and that no further log analysis is required.
  - INSTCONFFAILED indicates an installation failure that you cannot retry or recover from without reinstalling.

You successfully installed the Application Client for WebSphere Application Server and the features you selected.

If the installation is not successful, fix the error as indicated in the installation error message. For example, if you do not have enough disk space, add more space, and reinstall the Application Client.

## Best practices for installing application clients

The following table offers tips for installing application clients on multiple platforms.

Operating environment	Tip
Linux and UNIX systems	Spaces are not supported in the name of the installation directory on Linux and UNIX platforms.
UNIX systems	When the application client installations are successful, the return code 1 is issued from the UNIX shell where you issued the <code>/install</code> command. Any other return code indicates an unsuccessful installation.
Solaris systems	Double-byte character set (DBCS) characters are not supported in the name of the installation directory on Solaris systems.
All platforms	Reserve at least 4 to 5MB free space in the target platform temporary directory.

All platforms	<p>When specifying a different temporary directory while installing application clients, the following message is displayed if the target platform default temporary directory does not have enough free space to install application clients:</p> <pre>Error writing file = There may not be enough temporary disk space. Try using -is:tempdir to use a temporary directory on a partition with more disk space.</pre> <p>Use the <code>-is:tempdir</code> installation option to specify a different temporary directory. For example, the following command uses <code>/swap</code> as a temporary directory during installation:</p> <pre>./install -is:tempdir /swap</pre>
All platforms	<p>After the installation, when changing the installation settings for the WebSphere Application Server host name and the port number, edit the <code>setupClient.bat</code> for Windows or <code>setupClient.sh</code> for UNIX. Change the <code>DEFAULTSERVERNAME</code> and <code>SERVERPORTNUMBER</code> to the new WebSphere Application Server host name and port number, respectively. If the <code>SERVERPORTNUMBER</code> is not set, then the default is 2809. Review the following example:</p> <pre>set DEFAULTSERVERNAME=NDServerName set SERVERPORTNUMBER=9810</pre> <p>The <code>setupClient.bat</code> file or <code>setupClient.sh</code> file is located in the <code>bin</code> sub-directory under the application clients installation destination.</p>

## Installing application clients silently

Use these steps to perform a silent installation, which uses the installation wizard to install the product. Instead of displaying a user interface, the silent installation provides interaction between you and the wizard by reading all of your responses from a file that you must customize.

1. Verify that the user ID that you are using to run the silent installation has sufficient authority to perform the task.
2. Customize the option response file.
  - a. Locate the sample options response file. The file name is `setup.response` in the operating system platform directory on the product CD-ROM.
  - b. Make a copy to preserve the original response file. For example, copy the file as `myoptionsfile`.
  - c. Edit the copy in your flat file editor of choice, on the target operating system. Read the directions within the response file to choose appropriate values.

**Note:** To prepare the file for a silent installation on AIX, use UNIX line-end characters (0x0D0A) to terminate each line of the options response file.

- d. Make a non-commented option to have a silent install.
  - e. Include custom option responses that reflect parameters for your system.
  - f. Follow the instructions in the response file to choose appropriate values.
  - g. Save the file.
3. Issue a command to use your custom response file: `Install.exe -options myoptionsfile` for Windows platforms and `install -options ./myoptionsfile` for Linux and UNIX platforms. The sample options response file is located in the operating-system platform directory on the product CD-ROM.
    - a. Issue the following command from a command prompt to update your response file: `-w silentInstallLicenseAcceptance.value='true'`.

Issuing this command changes the value for `silentInstallLicenseAcceptance.value` from `false` to `true`, which is necessary for installing application clients.

4. **Optional:** Restart your machine in response to the prompt that appears on Windows platforms when the installation is complete.

You installed application clients silently by using the response file.

To verify the silent install, look for the string `RC=INSTCONFSUCCESS` in the log file for successful installation and `RC=INSTCONFFAILED` for a failed install installation.

For UNIX platforms, the install command returns a return code of **1** to indicate a successful installation. Any other return code means that the installation failed.

## Uninstalling Application Client for WebSphere Application Server

This task describes using the uninstaller program to uninstall the Application Client for WebSphere Application Server.

Before uninstalling, verify that you have no open Web browsers that are accessing the administrative console.



If you want to uninstall Application Clients manually, see the topic, “Manually uninstalling on a Windows system” on page 334.

1. Stop any browsers and any Java processes related to WebSphere Application Server products as described in Appendix A, “Uninstalling the product,” on page 317.
2. Change directories to the `_uninst` directory before issuing the “uninstall command” on page 318. The command file is located in the `install_root/product/_uninst` directory on a Linux or UNIX platform, and in the `install_root\product\_uninst` directory on a Windows system.

For example, to change directories before uninstalling the product from a Linux platform, issue this command if your installation root is `/opt/IBM/WebSphere/AppServer`:

```
cd /opt/IBM/WebSphere/AppServer/product/_uninst
```


3. Issue the **uninstaller** command. The command file is named `uninstaller.bin` for Linux and UNIX platforms and `uninstaller.exe` on Windows platforms.

  On Linux and UNIX platforms, issue the **uninstaller.bin** command from the `install_root/product/_uninst` directory:

```
./uninstaller.bin
```

 On Windows platforms, call the **uninstaller.exe** command:

```
install_root\product\_uninst\uninstaller.exe
```

 Call the program directly from the `install_root\product\_uninst` directory. For example, if the installation root is `C:\IBM\WebSphere\AppServer`, issue the following command:

```
C:\IBM\WebSphere\AppServer\product\_uninst> uninstaller.exe
```

The Uninstaller wizard begins and displays the Welcome panel.

4. Click **Next** to begin uninstalling the product. The Uninstaller wizard displays a Confirmation panel that lists the product and features that you are uninstalling.
5. Click **Next** to continue uninstalling the product. The Uninstaller wizard deletes existing profiles first. After deleting profiles, the Uninstaller wizard deletes core product files by component.
6. Click **Finish** to close the wizard after the wizard removes the product.

Application Client for WebSphere Application Server is uninstalled.

Verify the uninstall procedure by viewing the *install\_root*/product/logs/uninstlog.txt log file for errors. Look for the INSTCONFSUCCESS, indicating a successful uninstall in the log file:

```
Uninstall, com.ibm.ws.install.ni.ismp.actions.ISMPLogSuccessMessageAction, msg1,  
INSTCONFSUCCESS
```





---

## Part 4. Configuring



---

## Chapter 10. Configuring the product after installation

This topic summarizes how to configure the application serving environment.

Use the First steps console to configure and test the WebSphere Application Server environment after installation.

This procedure starts the second and final step of Network Deployment installation. This second step creates and configures server processes by creating profiles.

1. At the end of product installation, select the **Launch the Profile creation wizard** check box to create a deployment manager profile.  
This step creates the deployment manager and the cell. Later steps in this procedure create an Application Server profile and optionally, a custom profile.  
See “Using the Profile creation wizard to create a deployment manager” on page 238.
2. Start the First steps console by selecting the check box on the last panel of the wizard.  
The First steps console for the deployment manager profile can start automatically at the end of profile creation. Select the check box on the last panel of the Profile creation wizard.  
The First steps console is an easy way to start using the product. The console provides one-stop access to the administrative console, Samples Gallery, Profile creation wizard, installation verification test, Migration wizard, and other activities.  
See the description of the “firststeps command” on page 228 for more information.
3. Click **Installation verification** on the First steps console.  
The installation verification test starts the deployment manager process named dmgr and runs several tests to verify that the dmgr process can start without error.  
See “Using the installation verification test” on page 290 for more information.
4. Click **Profile creation wizard** on the First steps console to create an Application Server profile.  
See “Using the Profile creation wizard to create an application server” on page 263.
5. Start the First steps console by selecting the check box on the last panel of the Profile creation wizard.  
This First steps console belongs to the Application Server profile that you just created. Each profile has its own First steps console.
6. Click **Installation verification** on the First steps console.  
The installation verification test starts the new Application Server process named server1 and runs several tests to verify that the server1 process can start without error.
7. Federate the Application Server into the deployment manager cell.  
If both server processes are running, use the administrative console of the deployment manager to add the Application Server node into the cell.  
Point your browser at <http://localhost:9060/ibm/console> for example, to start the administrative console. Or start it from the First steps console of the deployment manager profile.  
Log in and click **System administration > Nodes > Add Node** and follow the wizard to add the node into the cell. You can use localhost for the Host field if both processes are on the same machine. The SOAP port for the Application Server node is 8880 unless you changed the port during profile creation.  
If the deployment manager is running, you can use the **addNode** command instead.
8. **Optional:** Click **Profile creation wizard** on the First steps console to create a custom profile.  
Verify that the deployment manager is running. The Profile creation wizard can federate the custom node for you if the deployment manager is running.  
Supply the host name and the SOAP port for the deployment manager while creating the custom profile.  
Choose to federate the custom node into the deployment manager cell. A custom profile must be part of a cell.

Use the deployment manager to customize the node at your leisure. Add servers, add clusters, and install applications on the node, for example.

See “Using the Profile creation wizard to create a custom profile” on page 249.

This procedure results in configuring and testing the Application Server environment.

---

## firststeps command

The **firststeps** command starts the First steps console.

### The First steps console

The First steps console is a post-installation ease-of-use tool for directing WebSphere Application Server elements from one place. Options display dynamically on the First steps console, depending on features you install. With all of the options present, you can use the First steps console to start or stop the application server, verify the installation, access the information center, access the administrative console, launch the Migration wizard, or access the Samples gallery.

The First steps console for the Network Deployment product has several forms. A First steps console exists for the product itself before the creation of any profiles. This version lets you start the Profile creation wizard to get started defining a deployment manager and application servers for the cell. You can also define stand-alone application servers. Each stand-alone application server has its own First steps console. Each deployment manager has its own First steps console.

A prompt to launch the First steps console displays on the last panel of the Profile creation wizard.

You can also start the First steps console from the command line as described later.

The First steps console for the Network Deployment product has several forms. A console exists for the product itself before the creation of any profiles. Options that display on the First steps console in the core product files of the Network Deployment product include the following entries:

#### Profile creation wizard

This option starts the Profile creation wizard. The wizard lets you create a deployment manager profile, an application server profile, or a custom profile. A *profile* consists of files that define the run-time environment for the deployment manager or the application server. Each environment has its own administrative interface. A custom profile is an exception. A custom profile is an empty node that you can federate into a deployment manager cell and customize. No default server processes or applications are created for a custom profile.

Each deployment manager or application server profile has its own First steps console. The location of the command to launch the First steps console is within the set of files in the profile. A prompt to launch the First steps console that is associated with a profile displays on the last panel of the Profile creation wizard.

#### Information center for WebSphere Application Server

This option links you to the online information center at the <http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp> IBM Web address.

#### Migration wizard

This option starts the Migration wizard. The Migration wizard is a new graphical interface to the migration tools.

See WASPreUpgrade command and WASPostUpgrade command for more information.

See Using the Migration wizard for more information about the Migration wizard.

**Exit** This option closes the First steps console.

In addition to the generic First steps console for the Network Deployment product, other First steps consoles exist for the deployment manager profile and the application server profile that the Network Deployment product can create. Options that display on the First steps console for a deployment manager include the following entries:

### **Installation verification**

This option starts the installation verification test. The test consists of starting and monitoring the deployment manager during its start up.

If this is the first time that you have used the First steps console since creating a deployment manager profile, click **Installation verification** to verify that all is well with your installation. The verification process starts the deployment manager.

If you select the **Installation verification** option, the **Start the deployment manager** option is grayed out while the IVT is running.

The IVT provides the following useful information about the deployment manager:

- The deployment manager server name: dmgr
- The name of the profile
- The profile file path
- The type of profile: dmgr
- The cell name
- The node name
- The current encoding
- The port number for the administrative console
- Various informational messages that include the location of the `SystemOut.log` file and how many errors are listed within the file
- A completion message

### **Start the deployment manager**

This option displays when you use the Profile creation wizard to create a deployment manager. This option toggles to **Stop the deployment manager** when the deployment manager is running.

After selecting the **Start the deployment manager** option, an output screen displays with status messages. The success message informs you that the deployment manager is open for e-business. Then the menu item changes to **Stop the deployment manager**.

If you select the **Start the deployment manager** option, the **Installation verification** option is grayed out while the deployment manager is running.

### **Administrative console**

This option is grayed out until the application server is running.

The administrative console is a configuration editor that runs in a Web browser. The administrative console lets you work with XML configuration files for the deployment manager and all of the application servers that are in the cell. To launch the administrative console, click **Administrative console**. You can also point your browser to `http://localhost:9060/ibm/console` to start the administrative console. Substitute your own host name in the address if the localhost variable does not resolve correctly. As the administrative console opens, it prompts you for a login name. This is not a security item, but merely a tag to identify configuration changes that you make during the session. Secure signon is also available.

### **Profile creation wizard**

This option starts the Profile creation wizard. The wizard lets you create a deployment manager profile, an application server profile, or a custom profile. A *profile* consists of files that define the run-time environment for the deployment manager or the application server. Each environment has its own administrative interface. A custom profile is an exception. A custom profile is an empty

node that you can federate into a deployment manager cell and customize. No default server processes or applications are created for a custom profile.

Each deployment manager profile has its own First steps console. The location of the command to launch the First steps console is within the set of files in the profile. A prompt to launch the First steps console that is associated with a profile displays on the last panel of the Profile creation wizard.

### Information center for WebSphere Application Server

This option links you to the online information center at the <http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp> IBM Web address.

### Migration wizard

This option starts the Migration wizard. The Migration wizard is a new graphical interface to the migration tools.

See WASPreUpgrade command and WASPostUpgrade command for more information.

See Using the Migration wizard for more information about the Migration wizard.

**Exit** This option closes the First steps console.

In addition to the generic First steps console and the First steps console for the deployment manager, another First steps console exists for stand-alone application servers that you create with the Profile creation wizard. Options that display on the First steps console for an application server profile include the following entries:

### Installation verification

This option starts the installation verification test. The test consists of starting and monitoring the application server during its start up.

If this is the first time that you have used the First steps console since creating an application server profile, click **Installation verification** to verify that all is well with your installation. The verification process starts the application server.

If you select the **Installation verification** option, the **Start the server** option is grayed out while the IVT is running.

The IVT provides the following useful information about the application server:

- The server name: server1
- The name of the profile
- The profile file path
- The type of profile: default
- The cell name
- The node name
- The current encoding
- The port number for the administrative console
- Various informational messages that include the location of the SystemOut.log file and how many errors are listed within the file
- A completion message

### Start the server

This option toggles to **Stop the server** when the application server is running.

After selecting the **Start the server** option, an output screen displays with status messages. The success message informs you that the server is open for e-business. Then the menu item changes to **Stop the server**.

If you select the **Start the server** option, the **Installation verification** option is grayed out while the application server is running.

### Administrative console

This option is grayed out until the application server is running.

The administrative console is a configuration editor that runs in a Web browser. The administrative console lets you work with XML configuration files for the deployment manager and all of the application servers that are in the cell. To launch the administrative console, click **Administrative console**. You can also point your browser to <http://localhost:9060/ibm/console> to start the administrative console. Substitute your own host name in the address if the localhost variable does not resolve correctly. As the administrative console opens, it prompts you for a login name. This is not a security item, but merely a tag to identify configuration changes that you make during the session. Secure signon is also available.

### Profile creation wizard

This option starts the Profile creation wizard. The wizard lets you create a deployment manager profile, an application server profile, or a custom profile. A *profile* consists of files that define the run-time environment for the deployment manager or the application server. Each environment has its own administrative interface. A custom profile is an exception. A custom profile is an empty node that you can federate into a deployment manager cell and customize. No default server processes or applications are created for a custom profile.

Each application server profile has its own First steps console. The location of the command to launch the First steps console is within the set of files in the profile. A prompt to launch the First steps console that is associated with a profile displays on the last panel of the Profile creation wizard.

### Samples gallery

This option starts the Samples gallery. The option is grayed out until you start the application server. The option displays when you have installed the Samples during installation.

From the First steps console, click **Samples gallery** to explore the application Samples. Alternatively you can point your browser directly to <http://localhost:9080/WSsamples>. Substitute your own host name in the address if the localhost variable does not resolve correctly. The Web address is case sensitive. Substitute your own host name in the address.

### Information center for WebSphere Application Server

This option links you to the online information center at the <http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp> IBM Web address.

### Migration wizard

This option starts the Migration wizard. The Migration wizard is a new graphical interface to the migration tools.

See WASPreUpgrade command and WASPostUpgrade command for more information.




See Using the Migration wizard for more information about the Migration wizard.

**Exit** This option closes the First steps console.




The First steps console for a managed node has the same options as the generic First steps console for the Network Deployment product.

### Location of the command file

The location of the `firststeps.sh` or `firststeps.bat` script for any profile is:

-   `install_root/profiles/profile_name/firststeps/firststeps.sh`
-  `install_root\profiles\profile_name\firststeps\firststeps.bat`

The location of the **firststeps** command for the generic First steps console for the Network Deployment product is:

-   `install_root/firststeps/firststeps.sh`
-  `install_root\firststeps\firststeps.bat`

## Parameters

No parameters are associated with this command.






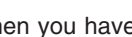
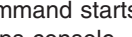
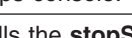
## Syntax for the firststeps command

Use the following syntax for the command:
















-   `./firststeps.sh`
-  `firststeps.bat`




## Usage tips

The following links exist on one of the First steps consoles for the Network Deployment product. Not all links display on each First steps console. For example, the First steps console for the deployment manager does not have the Samples Gallery option or the Start the server option.

Option	Link
<b>Installation verification</b>	<p>Calls the <b>ivt</b> command.</p> <p>The location of the installation verification test varies per platform:</p> <ul style="list-style-type: none"> <li>•  <code>install_root/profiles/profile_name/bin/ivt.sh</code></li> <li>•  <code>install_root/profiles/profile_name/bin/ivt.sh</code></li> <li>•  <code>install_root\profiles\profile_name\bin\ivt.bat</code></li> </ul>
<b>Start the server</b>	<p>Calls the <b>startServer</b> command.</p> <p>The location of the <b>startServer</b> command varies per platform:</p> <ul style="list-style-type: none"> <li>•  <code>install_root/profiles/profile_name/bin/startServer.sh</code> server1</li> <li>•  <code>install_root/profiles/profile_name/bin/startServer.sh</code> server1</li> <li>•  <code>install_root\profiles\profile_name\bin\startServer.bat</code> server1</li> </ul> <p>When you have more than one application server on the same machine, the command starts the same application server that is associated with the First steps console.</p>
<b>Stop the server</b>	<p>Calls the <b>stopServer</b> command.</p> <p>The location of the <b>stopServer</b> command varies per platform:</p> <ul style="list-style-type: none"> <li>•  <code>install_root/profiles/profile_name/bin/stopServer.sh</code> server1</li> <li>•  <code>install_root/profiles/profile_name/bin/stopServer.sh</code> server1</li> <li>•  <code>install_root\profiles\profile_name\bin\stopServer.bat</code> server1</li> </ul>



Option	Link
<b>Start the deployment manager</b>	<p data-bbox="636 220 1008 247">Calls the <b>startManager</b> command.</p> <p data-bbox="636 275 1321 302">The location of the <b>startManager</b> command varies per platform:</p> <ul data-bbox="636 317 1430 468" style="list-style-type: none"> <li data-bbox="636 317 1419 359">•  <code>install_root/profiles/profile_name/bin/startManager.sh</code></li> <li data-bbox="636 373 1419 415">•  <code>install_root/profiles/profile_name/bin/startManager.sh</code></li> <li data-bbox="636 430 1430 468">•  <code>install_root\profiles\profile_name\bin\startManager.bat</code></li> </ul> <p data-bbox="636 495 1446 575">When you have more than one deployment manager on the same machine, the command starts the same deployment manager that is associated with the First steps console.</p>
<b>Stop the deployment manager</b>	<p data-bbox="636 592 1008 619">Calls the <b>stopManager</b> command.</p> <p data-bbox="636 646 1321 674">The location of the <b>stopManager</b> command varies per platform:</p> <ul data-bbox="636 688 1419 840" style="list-style-type: none"> <li data-bbox="636 688 1408 730">•  <code>install_root/profiles/profile_name/bin/stopManager.sh</code></li> <li data-bbox="636 745 1408 787">•  <code>install_root/profiles/profile_name/bin/stopManager.sh</code></li> <li data-bbox="636 802 1419 840">•  <code>install_root\profiles\profile_name\bin\stopManager.bat</code></li> </ul>
<b>Administrative console</b>	<p data-bbox="636 854 1398 911">Opens the default browser to the <code>http://localhost:9060/ibm/console</code> Web address.</p> <p data-bbox="636 938 1446 1018">When you have more than one application server on the same machine, the port varies. The First steps console starts the administrative console that is associated with the First steps console.</p>
<b>Profile creation wizard</b>	<p data-bbox="636 1039 987 1066">Calls the <b>pctplatform</b> command.</p> <p data-bbox="636 1094 1446 1150">The command is in the <code>install_root/bin/ProfileCreator</code> directory. The name of the command varies per platform:</p> <ul data-bbox="636 1165 1446 1640" style="list-style-type: none"> <li data-bbox="636 1165 932 1207">•  <code>pctAIX.bin</code></li> <li data-bbox="636 1222 943 1264">•  <code>pctHPUX.bin</code></li> <li data-bbox="636 1278 1170 1320">•  64-bit platforms: <code>pctHPUXIA64.bin</code></li> <li data-bbox="636 1335 954 1377">•  <code>pctLinux.bin</code></li> <li data-bbox="636 1392 1446 1434">•  64-bit platforms: <code>pct.bin</code> S/390 platforms: <code>pctLinux390.bin</code></li> <li data-bbox="636 1449 1179 1491">•  Power platforms: <code>pctLinuxPPC.bin</code></li> <li data-bbox="636 1505 979 1547">•  <code>pctSolaris.bin</code></li> <li data-bbox="636 1562 979 1604">•  <code>pctWindows.exe</code></li> <li data-bbox="636 1619 1206 1661">•  64-bit platforms: <code>pctWindowsIA64.exe</code></li> </ul>

Option	Link
<b>Samples Gallery</b>	<p>Opens the default browser to the <a href="http://localhost:9080/WSsamples">http://localhost:9080/WSsamples</a> Web address.</p> <p>If you do not install Samples, the option does not appear on the First steps console. If you do not install the Samples during the initial installation of the product, the option does not display on the First steps console. You can perform an incremental installation to add the Samples feature. After adding the Samples, the options displays on the First steps console.</p> <p>When you have more than one profile on the same machine, the port varies. The First steps console starts the Samples gallery that is associated with the First steps console.</p>
<b>Information center for WebSphere Application Server products</b>	<p>Opens the default browser to the online information center at the <a href="http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp">http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp</a> Web address.</p>
<b>Migration wizard</b>	<p>Calls the <b>migration</b> command.</p> <p>The location of the <b>migration</b> command is:</p> <ul style="list-style-type: none"> <li>•   <code>install_root/bin/migration.sh</code></li> <li>•  <code>install_root\bin\migration.bat</code></li> </ul> <p>The migration tools are also in the <code>/migration</code> folder on the product disc.</p>

---

## Using the Profile creation wizard

This topic describes how to create run-time environments for WebSphere Application Server. Each run-time environment is created within a *profile*. A profile is the set of files that define the run-time environment. The Profile creation wizard creates the profile for each run-time environment.

Before using the Profile creation wizard, install the core product files.

The Profile creation wizard is the wizard interface to the profile creation tool, `wasprofile`. See the description of the “`wasprofile` command” on page 275 for more information.


An error can occur when you have not provided enough system temporary space to create a profile. Verify that you have a minimum of 40 MB of temp space available before creating a profile.

You must have 200 MB of available disk space in the directory where you create an Application Server profile.

### Important:

You must have 30 MB of available disk space in the directory where you create a deployment manager profile.

You must have 10 MB of available disk space in the directory where you create a custom profile.

 Manually verify that the required space for creating a profile is available on AIX. A known problem in the underlying InstallShield for Multiplatforms (ISMP) code prevents proper space checking on AIX systems at the time that the product disc was created.

**Important:** Concurrent profile creation is not supported at this time for one set of core product files. Concurrent attempts to create profiles result in a warning about a profile creation already in progress.

The installation procedure for the Network Deployment product does not create a run-time environment by default because three possibilities exist. After installing the core product files for the Network Deployment product, use the Profile creation wizard to create any combination of the following three profiles to have an operational run-time environment:

- A deployment manager profile.

See “Using the Profile creation wizard to create a deployment manager” on page 238. Create this run-time environment first, to create the administrative node for a multinode, multimachine group of application server nodes that you create later. This logical group of application server processes is known as a *cell*.

- An application server profile.

See “Using the Profile creation wizard to create an application server” on page 263.

When you create the application server profile, a default server1 process is created. The process has all of the Sample applications and its own administrative console. You can federate the server1 node into the deployment manager cell with the **addNode** command or from the administrative console of the deployment manager. The server1 process must be running to begin the federation from the deployment manager.

If you include all of the applications from the application server, the act of federation installs the applications on the deployment manager where they can be redeployed.

Optionally, you can create stand-alone application servers by creating an application server profile and not federating the node. If you remove a federated application server node from a deployment manager, the application server returns to its original configuration, which is a stand-alone application server.

- A custom profile.

See “Using the Profile creation wizard to create a custom profile” on page 249. A custom profile is an empty node that you can customize through the deployment manager to include application servers, clusters, or other Java processes, such as a messaging server. Create a custom profile on a distributed machine and add the node into the deployment manager cell to get started customizing the node.


Each use of the Profile creation wizard or the **wasprofile** command line tool creates one profile.



1. Install the product to create the core product files.
2. Start the Profile creation wizard to create a new run-time environment.

Several ways exist to start the wizard. The initial way to start the wizard is at the end of installation by selecting the check box to launch the Profile creation wizard.

One way to start the wizard is to issue the command directly from a command line.

The command is in the *install\_root/bin/ProfileCreator* directory. The name of the command varies per platform:



-  `pctAIX.bin`
-  `pctHPUX.bin`
-  64-bit platforms: `pctHPUXIA64.bin`
-  `pctLinux.bin`
-  64-bit platforms: `pct.bin` S/390 platforms: `pctLinux390.bin`
-  Power platforms: `pctLinuxPPC.bin`
-  `pctSolaris.bin`

-  `pctWindows.exe`
-  64-bit platforms: `pctWindowsIA64.exe`

Another way to start the Profile creation wizard is to select the wizard from the First steps console.

- Open a command window.
- Change directories to the `firststeps` directory in the installation root directory:

The installation root varies by platform:

-  `/usr/IBM/WebSphere/AppServer/firststeps`
  -    `/opt/IBM/WebSphere/AppServer/firststeps`
  -  `C:\Program Files\IBM\WebSphere\AppServer\firststeps`
- Issue the **firststeps** command to start the console:

-   `./firststeps.sh`
-  `firststeps.bat`

- Select the Profile creation wizard option on the console.

The Profile creation wizard is an InstallShield for Multiplatforms application. The wizard loads the Java 2 SDK and then displays its Welcome panel.

See the description of the “firststeps command” on page 228 for more information.

### 3. Create a profile.

You can create profiles in any order. However, to create a functioning cell in the shortest possible time, create a deployment manager profile first. Then create an application server profile and add it to the deployment manager cell. You now have a functioning cell with a managed node that you can manage from the administrative console of the deployment manager.

A custom profile requires a greater amount of customization. When you create a custom profile, you must use the **addNode** command to federate it into the deployment manager cell. In contrast to an application server profile, a custom profile does not have a default application server on its node. The `server1` application server does not exist by default on the custom node. Nor are there any default applications on the custom node. Use the administrative console of the deployment manager to customize the empty node for production or other uses. You can create application servers or clusters on the node, for example.

Create any of the following profiles:

- Create a deployment manager.

See “Using the Profile creation wizard to create a deployment manager” on page 238.

Create a deployment manager to establish a cell. Although you can create an application server profile and use it as a stand-alone application server, you must have a deployment manager to use a custom profile. So there is no point in creating a custom profile until you have created a deployment manager.

- Create an application server profile, as described in “Using the Profile creation wizard to create an application server” on page 263.

Federate the application server into the deployment manager cell to create a federated `server1` application server. A stand-alone application server has default applications. You can include the applications as you federate the application server to install the default applications on the deployment manager.

Two methods exist for federating application servers into a deployment manager cell:

- Start the deployment manager and the application server and use the administrative console of the deployment manager to federate the node. Click **System administration > Nodes > Add node > Managed node > Next** and identify the host name and the SOAP port of the machine where you created the application server.

- Start the deployment manager. Go to the *install\_root/profiles/profile\_name/bin* directory of the application server and issue the **addNode** command.
- Create a custom profile, as described in “Using the Profile creation wizard to create a custom profile” on page 249.

The first part of the process is to install the Network Deployment product to create the core product files. Then you can use the Profile creation wizard to create a managed profile.

The next part of the process is to federate the managed profile into the deployment manager cell. This changes the custom profile into a managed node.

After federation, a custom profile has a nodeagent process but does not have an application server process. Contrast this situation to an application server profile that has a server1 process, but does not have a nodeagent process until you federate the node.

Start the nodeagent process to allow the administrative console of the deployment manager to create server processes on the managed node.

Two methods exist for federating a custom node into a deployment manager cell:

- Federate the custom node during custom profile creation, either with the wizard or as the wizard runs in silent mode.

The deployment manager must be running and accessible at the host address you supply. The deployment manager must also use the default JMX connector type, which is SOAP. The deployment manager must not have security enabled. If any of these conditions are not met, do not federate the custom profile as you create it, but federate it later with the **addNode** command. Otherwise, you create a faulty profile that you must move or delete from the profiles repository directory before creating another profile.

- Use the **addNode** command to federate the custom node after you create the custom profile.
  - a. Start the deployment manager.
  - b. Go to the *install\_root/profiles/profile\_name/bin* directory of the custom profile and issue the **addNode** command.
  - c. Within the same directory, issue the **startNode** command.

After federation, go to the administrative console of the deployment manager to customize the empty node.

- Create a deployment manager and a managed node on the same machine:
  - a. Create a deployment manager, as described in “Using the Profile creation wizard to create a deployment manager” on page 238.
  - b. Start the deployment manager with the **startManager** command.
  - c. Create an application server profile, as described in “Using the Profile creation wizard to create an application server” on page 263.
  - d. Start the application server with the **startServer** command.
  - e. Use the administrative console of the deployment manager to add the application server node into the deployment manager cell. Click **System administration > Nodes > Add node > Managed node > Next** and identify the host name of the machine and the SOAP port of the application server.

The SOAP port is identified in the *install\_root/profiles/profile\_name/config/cells/cell\_name/nodes/node\_name/serverindex.xml* file. You can also use the administrative console of the application server to view its SOAP port setting. Click **Servers > Application servers > server1 > Ports**. The SOAP port is usually the second port in the list.

Select the check box to include applications that are installed on the application server. The default application has the snoop servlet and the hitcount servlet, which are useful for testing.

Adding the stand-alone application server node to the deployment manager node changes the server1 process into a managed node. Use the administrative console of the deployment manager to configure the server1 process.

You can also create a custom profile and federate the node during profile creation, or use the **addNode** command to federate the empty node into the cell after the custom profile exists. A managed node created from a custom profile requires customization to create an application server and install applications. Use the administrative console of the deployment manager to configure the custom node.

## Using the Profile creation wizard to create a deployment manager

This topic describes creating a run-time environment for a deployment manager.

Before using the Profile creation wizard, install the core product files.


The Profile creation wizard is the wizard interface to the profile creation tool, `wasprofile`. See the description of the “`wasprofile` command” on page 275 for more information.

An error can occur when you have not provided enough system temporary space to create a profile. Verify that you have a minimum of 40 MB of temp space available before creating a profile.

You must have 200 MB of available disk space in the directory where you create an Application Server profile.

You must have 30 MB of available disk space in the directory where you create a deployment manager profile.

You must have 10 MB of available disk space in the directory where you create a custom profile.

 Manually verify that the required space for creating a profile is available on AIX. A known problem in the underlying InstallShield for Multiplatforms (ISMP) code prevents proper space checking on AIX systems at the time that the product disc was created.

After installing the core product files, you must create one of the following profiles to have an operational run-time environment:

- An application server that is described in this topic.  
An application server profile has a default server (which is `server1`), the default application that includes the snoop servlet and the hitcount servlet, and application Samples. You can federate the application server or use it as a stand-alone application server.
- A deployment manager.  
See “Using the Profile creation wizard to create a deployment manager.” The deployment manager provides a single administrative interface to a logical group of application servers on one or more machines.
- A custom profile that you must federate to make operational.  
See “Using the Profile creation wizard to create a custom profile” on page 249. A custom profile is an empty node that you can customize to include application servers, clusters, or other Java processes, such as a messaging server.

This procedure describes creating a deployment manager profile. The procedure uses the graphical user interface provided by the Profile creation wizard. You can use the Profile creation wizard in silent mode with a response file, without the graphical user interface. See “`responsefile.pct.NDdmgrProfile.txt`” on page 242 for examples of using the Profile creation wizard in silent mode.










You can also use the **wasprofile** command to create a deployment manager. See the description of the “`wasprofile` command” on page 275 for more information.

1. Start the Profile creation wizard to create a new run-time environment.

Several ways exist to start the wizard. The initial way to start the wizard is at the end of installation by selecting the check box to launch the Profile creation wizard.

One way to start the wizard is to issue the command directly from a command line.

The command is in the *install\_root/bin/ProfileCreator* directory. The name of the command varies per platform:






-  `pctAIX.bin`
-  `pctHPUX.bin`
-  64-bit platforms: `pctHPUXIA64.bin`
-  `pctLinux.bin`
-  64-bit platforms: `pct.bin` S/390 platforms: `pctLinux390.bin`
-  Power platforms: `pctLinuxPPC.bin`
-  `pctSolaris.bin`
-  `pctWindows.exe`
-  64-bit platforms: `pctWindowsIA64.exe`

Another way to start the Profile creation wizard is to select the wizard from the First steps console.

a. Open a command window.

b. Change directories to the `firststeps` directory in the installation root directory:

The installation root varies by platform:

-  `/usr/IBM/WebSphere/AppServer/firststeps`
-    `/opt/IBM/WebSphere/AppServer/firststeps`
-  `C:\Program Files\IBM\WebSphere\AppServer\firststeps`

c. Issue the **firststeps** command to start the console:

-   `./firststeps.sh`
-  `firststeps.bat`

d. Select the Profile creation wizard option on the console.

The Profile creation wizard is an InstallShield for Multiplatforms application. The wizard loads the Java 2 SDK and then displays its Welcome panel.

See the description of the “firststeps command” on page 228 for more information.

2. Click **Next** on the Welcome panel.

The wizard displays the Profile type selection panel.

3. Select the type of profile to create and click **Next**. In this example, select the option for creating a deployment manager and click **Next**.

The wizard displays the Profile name panel.

4. Specify a name for the profile, then click **Next**.

**Profile naming guidelines:** 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
- Illegal special characters that are not allowed within the name of a directory on your operating system, such as `*&?`
- Slashes (`/`) or (`\`)



Double-byte characters are allowed.

### 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 issued from the `bin` directory in the product installation root. When only one profile exists on a machine, every command works on the only server process in the configuration.

### Addressing a profile in a multi-profile environment

When two or more profiles exist on a machine, certain commands require that you specify the profile to which the command applies. These commands use the `-profileName` parameter to identify which profile to address. You might find it easier to use the commands that in the `bin` directory of each profile.

A command in the `profiles/profile_name/bin` directory has two lines. The first line sets the `WAS_USER_SCRIPT` environment variable for the command window. The variable sets up the command environment to address the profile. The second line calls the actual command in the `install_root/bin` directory.

The actual command queries the command shell to determine the calling profile and to autonomically address the command to the calling profile.

The wizard then displays the Profile directory panel.

5. Accept the default directory, specify a non-default location, or click **Browse** to select a different location. Click **Next**.

If you click **Back** and change the name of the profile, you must manually change the name on this panel when it displays again.

The wizard displays the Node, host, and cell name panel.

6. Specify a unique node name, the actual host name of the machine, and a unique cell name. Click **Next**.

The deployment manager node has the following characteristics.

Field name	Default value	Constraints	Description
Node name	The name of your machine, or a unique derivation of the machine name.	Use a unique name for the deployment manager.  See the following note.	The name is used for administration within the deployment manager cell.
Host name	The DNS name of your machine.	The host name must be addressable through your network.  See the following note.	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	The arbitrary name of the deployment manager cell. The cell is a logical grouping of managed nodes, under the control of the deployment manager.	Use a unique name for the deployment manager cell. If you plan to migrate a V5 deployment manager cell to this V6 deployment manager, use the same cell name as the V5 deployment manager.  See the following note.	All federated nodes become members of the deployment manager cell, which you name in this panel.


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

- cells
- nodes
- servers
- clusters



- applications
- deployments

### Node and cell name considerations

 The profiles directory path must be no longer than 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 to communicate with this node. Selecting a host name that other machines can reach within your network is extremely important. Do not use the generic localhost identifier for this value.

If you define coexisting nodes on the same computer with unique IP addresses, 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 servers (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 advantage of being totally unambiguous and also 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 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 format disadvantage is being dependent 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 ability of being redefined in the local hosts file so that the system can run the Application Server even when disconnected from the network. Define the short name to `127.0.0.1` (local loopback) in the hosts file to run disconnected. A format disadvantage is being dependent 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 you name with a numeric IP address without DNS being available. A format disadvantage 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 format disadvantage is that you cannot use the node if the host is disconnected from the network.

After specifying deployment manager characteristics, the wizard displays the Port value assignment panel.

7. Verify that the ports specified for the deployment manager are unique and click **Next**.

 The wizard displays the Windows service definition panel.

8.  Choose whether to run the `dmgr` process as a Windows service on a Windows platform and click **Next**.

Version 6 attempts to start Windows services for `dmgr` processes started by a **startManager** command. For example, if you configure a deployment manager as a Windows service and issue the **startManager** command, the **wasservice** command attempts to start the defined service.

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

To perform this installation task, the user ID must not have spaces in its name. The ID must also belong to the administrator group and must have the advanced user rights *Act as part of the operating system* and *Log on as a service*. The Installation wizard grants the user ID the advanced user rights if it does not already have them, 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. See the *Administration: Setting up* PDF for more information.

The wizard displays the Profile summary panel.

9. Click **Next** to create the deployment manager or click **Back** to change the characteristics of the deployment manager.

The wizard displays a Status panel during the creation of the profile. When the installation is complete, the wizard displays the Profile creation is complete panel.

10. Click **Finish** to exit and then click **Profile creation wizard** on the First steps console to start the wizard again, if you intend to create an application server profile.

You can create a deployment manager profile. The node within the profile has a deployment manager named dmgr.

Refer to the description of the “wasprofile command” on page 275 to learn about creating this type of profile using a command instead of a wizard.

### **responsefile.pct.NDdmgrProfile.txt**

This topic describes the response file for creating a deployment manager profile.

Create a deployment manager profile with an options response file after logging on as root on a Linux or UNIX platform, or as a user that belongs to the administrator group on a Windows platform. Some steps of the installation procedure on a Windows platform require the user to belong to the administrator group and to have the advanced user rights *Act as part of the operating system* and *Log on as a service*.

The response file is shipped with default values.

A common use for an options file is to run the Installation wizard in silent mode, which is referred to as installing silently. The wizard reads the options file to determine responses and does not display the graphical user interface. Issue the following command to use a copy of the options file named myresponsefile.txt for a silent installation:

```
pctWindows.exe -options "myresponsefile.txt" -silent
```

### **Avoiding the use of the -silent option within the options response file**

A problem occurs when the -silent option exists in the file. The file works with the option during a direct call to the profile creation wizard, but fails when called from a silent product installation. See “Customizing the options response file for Network Deployment” on page 123 for information about creating a profile silently during a silent product installation.

The option is unnecessary. Avoid using the option to avoid problems.

### **Response file locations**

The example options response files are in two locations.

### **Example files:**

- responsefile.pct.NDmgrProfile.txt
- responsefile.pct.NDmanagedProfile.txt
- responsefile.pct.NDstandAloneProfile.txt

### Location:

Table 18. Option response file locations

Product disc location	Installed location
/WAS directory	install_root/bin/ProfileCreator directory

Use the file on the product disc to install the Network Deployment product silently and create a profile.

After installing the Network Deployment product, you can use the installed response file with the -options parameter on the Profile creation wizard command.

### Required disk space

Profile	Required disk space	Required temp space
Deployment manager profile	30 MB	40 MB
Custom profile	10 MB	40 MB
Application server profile	200 MB	40 MB

### Creating an operational environment during product installation

Version 6 installation of the Network Deployment product is a two-step process:

1. Installing the core product files and feature files.
2. Creating a deployment manager profile, a custom profile, or an application server profile.

The sample options response file, “responsefile.nd.txt” on page 128, controls the first part of the installation and can also start the second part of the installation. To create a profile as part of installing the core product files, use the option in the responsefile.nd.txt file that identifies the response file for creating a profile. The profile response file lets you use the Profile creation wizard silently.

To edit and use the appropriate response file for creating a profile, perform the following procedure:

1. Copy the appropriate file from the WAS directory on the product disc to a place that you can easily identify on your machine. The example files are:

To create a:	Copy the following response file:
Deployment manager profile	responsefile.pct.NDmgrProfile.txt
Custom profile	responsefile.pct.NDmanagedProfile.txt
Application server profile	responsefile.pct.NDstandAloneProfile.txt

2. Edit the file to customize the values for your installation.
3. Verify that no -silent option exists in the response file for the Profile creation wizard. If the option exists, the profile is not created.
4. Save the file.
5. Edit the responsefile.nd.txt file to identify the location and name of the profile response file. Change the value of the -W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation option to identify the file. For example:

```
-W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation=
"/opt/IBM/WebSphere/MyOptionFiles/customProfile.txt"
```

6. Start the installation. For example:  

```
install -options "myresponsefile.txt" -silent
```
7. After the installation, examine the logs for success.

### Creating a profile after installation

Version 6 installation of the Network Deployment product is a two-step process:

1. Installing the core product files and feature files
2. Creating a deployment manager profile, a custom profile, or an application server profile

When the core product files exist, create a profile at any time using the Profile creation wizard. Start the wizard from the First steps console or directly using the Profile creation wizard command.

You can also use one of the following sample options response files for profiles to create a profile silently using the Profile creation wizard in silent mode. To edit and use the appropriate response file for creating a profile, perform the following procedure:











1. Copy the appropriate file from the *install\_root/bin/ProfileCreator* directory to a place that you can easily identify on your machine. The example files are:

To create a profile for a:	Copy the following response file:
Deployment manager	responsefile.pct.NDmgrProfile.txt
Managed node	responsefile.pct.NDmanagedProfile.txt
Stand-alone application server	responsefile.pct.NDstandAloneProfile.txt

For example, copy the file as *my\_options\_file.txt*

2. Edit the file to customize the values for your installation.
3. Save the file.
4. Start the installation.

For example:

-  `./pctAIX.bin -options my_options_file.txt -silent`
-  `./pctHPUX.bin -options my_options_file.txt -silent`
-  64-bit platforms: `./pctHPUXIA64.bin -options my_options_file.txt -silent`
-  `./pctLinux.bin -options my_options_file.txt -silent`
-  64-bit platforms: `./pctLinuxIA64.bin -options my_options_file.txt -silent`
-  Power platforms: `./pctLinuxPPC.bin -options my_options_file.txt -silent`
-  S/390 platforms: `./pctLinux390.bin -options my_options_file.txt -silent`
-  `./pctSolaris.bin -options my_options_file.txt -silent`
-  `pctWindows.exe -options my_options_file.txt -silent`
-  64-bit platforms: `pctWindowsIA64.exe -options my_options_file.txt -silent`

5. After using the Profile creation wizard, examine the logs for success.

## Logging

The following log files record information about profile creation:

- The `install_root/logs/log.txt` file records installation status.
- The `install_root/profiles/profile_name/logs/pctLog.txt` file records installation events that occur when creating profiles with the Profile creation wizard.
- The `install_root/logs/wasprofile/wasprofile_create_profile_name.log` file records installation events that occur when creating profiles.
- The `install_root/logs/wasprofile/wasprofile_delete_profile_name.log` file records installation events that occur when deleting profiles.

See Appendix B, “Troubleshooting installation,” on page 337 for more information.

## Usage notes

- The file is not a read-only file.
- Edit this file directly with your flat file editor of choice, such as WordPad on a Windows platform.
- The file is updated when you specify the `-options` parameter when using the Profile creation wizard and the file does not yet exist.
- The file must exist to perform a silent installation. The installation program reads this file to determine installation option values when you install silently.
- Save the file in a location that you can identify when you specify the fully qualified path as part of the profile creation command.

## Naming considerations

Consider the following recommendations when supplying names for the profile and other objects.

### Naming the profile

Use the following guidelines when supplying a value for the profile name directive:

```
-W profilenamepanelInstallWizardBean.profileName
```

**Profile naming guidelines:** 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
- Illegal special characters that are not allowed within the name of a directory on your operating system, such as `*&?`
- Slashes (`/`) or (`\`)

Double-byte characters are allowed.

### Avoiding reserved names

Avoid the following reserved folder names as values for the directives in the `responsefile.pct.NDstandAloneProfile.txt` file and in the `responsefile.pct.NDmanagedProfile.txt` file:

```
-W nodehostnamepanelInstallWizardBean.nodeName=""  
-W nodehostnamepanelInstallWizardBean.hostName=""  
-W setnondmgrcellnameinglobalconstantsInstallWizardBean.value=""
```


Avoid the following reserved folder names as values for the directives in the `responsefile.pct.NDmgrProfile.txt` file:

```
-W nodehostandcellnamepanelInstallWizardBean.nodeName=""  
-W nodehostandcellnamepanelInstallWizardBean.hostName=""  
-W nodehostandcellnamepanelInstallWizardBean.cellName=
```

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

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

### Node and cell name considerations

 The profiles directory path must be no longer than 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 to communicate with this node. Selecting a host name that other machines can reach within your network is extremely important. Do not use the generic localhost identifier for this value.

If you define coexisting nodes on the same computer with unique IP addresses, 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 servers (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 advantage of being totally unambiguous and also 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 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 format disadvantage is being dependent 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 ability of being redefined in the local hosts file so that the system can run the Application Server even when disconnected from the network. Define the short name to `127.0.0.1` (local loopback) in the hosts file to run disconnected. A format disadvantage is being dependent 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 you name with a numeric IP address without DNS being available. A format disadvantage 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 format disadvantage is that you cannot use the node if the host is disconnected from the network.

## Example responsefile.pct.NDmgrProfile.txt file

```
#####
#
# Response file for Websphere Application Server 6.0 dmgr profile creation
#
# This options file is located in the CD_ROOT\WAS\ directory and in the
# install_root\bin\ProfileCreator directory.
#
# To use the options file under CD_ROOT\WAS\ directory, follow the instructions
# in CD_ROOT\WAS\responsefile.nd.txt. The WebSphere Application Server
# Network Deployment installer locates this file during silent installation
# and automatically runs the silent profile creation at the end of installation.
#
# To use the options file under install_root\bin\ProfileCreator for silent profile
# creation, you must change various values in the file and use the following
# command line arguments:
#
# -options "responsefile.pct.NDmgrProfile.txt" -silent
#
#####

#####
#
# Profile name
#
# Set the profile name for installing a deployment manager profile. The profile
# name must be unique for this WebSphere Application Server installation.
#
-W profilenamepanelInstallWizardBean.profileName="profileDmgr"

#####
# If you want to set this profile to be your default profile, set to "true".
# Otherwise set to "false". If this is the first profile being created, the profile
# automatically is the default.
#
-W profilenamepanelInstallWizardBean.isDefault="false"

#####
#
# Profile location
#
# Specify a directory to contain the files that define the run-time environment,
# such as commands, configuration files, and log files. If the directory contains
# spaces, enclose it in double-quotes as shown in the Windows example below.
#
# Note that spaces in the install location is only supported on Windows
# operating systems.
#
# Default Install Location:
#
# -P installLocation="<WAS_HOME>\profiles\<PROFILE_NAME>"
#
-P installLocation="C:\Program Files\IBM\WebSphere\AppServer\profiles\profileDmgr"

#####
#
# Node name
#
# Please select the node name for the Application Server. Node name under one cell
# has to be unique.
#
# Replace YOUR_NODE_NAME with the actual node name.
#
-W nodehostandcellnamepanelInstallWizardBean.nodeName="YOUR_NODE_NAME"
```



```

#####
#
# Host name
#
# Specify the host name for the Application Server. The host name is the domain
# name system (DNS) name (short or long) or the IP address of this computer.
#
# Replace YOUR_HOST_NAME with the actual host name. Comment the line to use
# the default value.
#
-W nodehostandcellnamepanelInstallWizardBean.hostName="YOUR_HOST_NAME"

#####
#
# Cell name
#
# Specify the cell name for the Application Server.
#
# If you plan to migrate a V5 deployment manager cell to this V6 deployment
# manager, specify the same cell name as the V5 cell.
#
# Replace YOUR_CELL_NAME with the actual cell name.
#
-W nodehostandcellnamepanelInstallWizardBean.cellName="YOUR_CELL_NAME"

#####
#
# Port value assignment
#
# The following entries are used to reset port numbers used in the configuration
#
# They are currently set to the defaults.
# Please check to make sure there are no Port Conflicts.
# Port numbrers for each profile can be found in:
# <profile>/config/cells/<cell name>/nodes/<node name>/serverindex.xml
#
-W pctdmgrprofileportspanelInstallWizardBean.WC_adminhost="9060"
-W pctdmgrprofileportspanelInstallWizardBean.WC_adminhost_secure="9043"
-W pctdmgrprofileportspanelInstallWizardBean.BOOTSTRAP_ADDRESS="9809"
-W pctdmgrprofileportspanelInstallWizardBean.SOAP_CONNECTOR_ADDRESS="8879"
-W pctdmgrprofileportspanelInstallWizardBean.SAS_SSL_SERVERAUTH_LISTENER_ADDRESS="9404"
-W pctdmgrprofileportspanelInstallWizardBean.CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS="9406"
-W pctdmgrprofileportspanelInstallWizardBean.CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS="9405"
-W pctdmgrprofileportspanelInstallWizardBean.ORB_LISTENER_ADDRESS="9101"
-W pctdmgrprofileportspanelInstallWizardBean.CELL_DISCOVERY_ADDRESS="7277"
-W pctdmgrprofileportspanelInstallWizardBean.DCS_UNICAST_ADDRESS="9352"

#####
#
# Windows service
#
# The following directives are to install services for WebSphere Application Server
# on Windows.
# Using Services, you can start and stop services,
# and configure startup and recovery actions.
# Set winServiceQuery="false" will turn off the function on windows system.
# You can ignore these or comment them out for other Operating Systems.
#
-W winservicepanelInstallWizardBean.winServiceQuery="true"

#####
# Specify account type of the service. Legal values are:
#
# localsystem - Indicates that you choose to use Local System account.

```



```

#    specifieduser - Indicates that you choose to use specified user account.
#
-W winservicepanelInstallWizardBean.accountType="localsystem"

#####
# If you chose to install a service above with the accountType="localsystem",
# the userName and password below can be ignored. If the accountType was set to:
# accountType="specifieduser", then you must specify the User Name and Password
# which are required to install the Services. The current user must be admin or must
# have admin authority to install a Service. Also the username
# which is given here must have "Log On as a Service " authority
# for the service to run properly.
#
# Replace YOUR_USER_NAME with your username.
#
-W winservicepanelInstallWizardBean.userName="YOUR_USER_NAME"

#####
# Replace YOUR_PASSWORD with your valid password.
#
-W winservicepanelInstallWizardBean.password="YOUR_PASSWORD"

#####
# Set the startup type of the WebSphere Application Server on Windows.
# Valid values are "automatic", "manual", and "disabled".
#
-W winservicepanelInstallWizardBean.startupType="manual"

#####
# Profile type
#
# Must be set to "dmgr" for installing a deployment manager Profile.
# Do not change this!
#
-W profiletypepanelInstallWizardBean.selection="dmgr"

```

## Using the Profile creation wizard to create a custom profile

This topic describes creating a run-time environment for a custom profile.

Before using the Profile creation wizard, install the core product files.


The Profile creation wizard is the wizard interface to the profile creation tool, wasprofile. See the description of the “wasprofile command” on page 275 for more information.

An error can occur when you have not provided enough system temporary space to create a profile. Verify that you have a minimum of 40 MB of temp space available before creating a profile.

You must have 30 MB of available disk space in the directory where you create a deployment manager profile.

You must have 200 MB of available disk space in the directory where you create an Application Server profile.

You must have 10 MB of available disk space in the directory where you create a custom profile.

 Manually verify that the required space for creating a profile is available on AIX. A known problem in the underlying InstallShield for Multiplatforms (ISMP) code prevents proper space checking on AIX systems at the time that the product disc was created.

After installing the core product files for the Network Deployment product, you must create one of the following profiles to have an operational run-time environment:

- A custom profile.

This topic describes creating a custom profile. A custom profile is an empty node that you can customize to include application servers, clusters, or other Java processes, such as a messaging server.

You must have 10 MB of available disk space in the directory where you create a custom profile.

- A deployment manager.

See “Using the Profile creation wizard to create a deployment manager” on page 238. The deployment manager provides a single administrative interface to a logical group of application servers on one or more machines.

- An application server profile.

See “Using the Profile creation wizard to create an application server” on page 263. An Application Server profile has a default server (which is server1), the default application that includes the snoop servlet and the hitcount servlet, and application Samples. You can federate the Application Server or use it as a stand-alone Application Server.

You must have 200 MB of available disk space in the directory where you create an Application Server profile.

This topic describes creating a custom profile using the Profile creation wizard. You can use the Profile creation wizard in silent mode with a response file instead of the graphical user interface. See “responsefile.pct.NDmanagedProfile.txt” on page 254 for examples of using the Profile creation wizard in silent mode.

You can also use the **wasprofile** command to create a custom profile. See the description of the “wasprofile command” on page 275 for more information.

After creating a custom profile, you must have access to a running deployment manager to federate the node. Federating the custom profile makes the node operational. If the custom profile is on a machine that does not have a deployment manager, the deployment manager must be accessible over the network to allow the federation of the node.






You can federate the custom node as you create the custom profile, either with the Profile creation wizard or when using the **wasprofile** command. If you choose to federate, but the deployment manager is not running, the custom node is not usable. You must then either delete the profile directory or move the directory out of the profiles repository (profiles installation root directory) before creating another profile with the same name.





1. Install the product to create the core product files.
2. Start the Profile creation wizard to create a new run-time environment.

Several ways exist to start the wizard. The initial way to start the wizard is at the end of installation by selecting the check box to launch the Profile creation wizard.

One way to start the wizard is to issue the command directly from a command line.

The command is in the *install\_root/bin/ProfileCreator* directory. The name of the command varies per platform:






-  `pctAIX.bin`
-  `pctHPUX.bin`
-  64-bit platforms: `pctHPUXIA64.bin`
-  `pctLinux.bin`
-  64-bit platforms: `pct.bin` S/390 platforms: `pctLinux390.bin`

-  Power platforms: pctLinuxPPC.bin
-  pctSolaris.bin
-  pctWindows.exe
-  64-bit platforms: pctWindowsIA64.exe

Another way to start the Profile creation wizard is to select the wizard from the First steps console.

- Open a command window.
- Change directories to the `firststeps` directory in the installation root directory:

The installation root varies by platform:

-  `/usr/IBM/WebSphere/AppServer/firststeps`
-    `/opt/IBM/WebSphere/AppServer/firststeps`
-  `C:\Program Files\IBM\WebSphere\AppServer\firststeps`

- Issue the **firststeps** command to start the console:

-   `./firststeps.sh`
-  `firststeps.bat`

- Select the Profile creation wizard option on the console.

The Profile creation wizard is an InstallShield for Multiplatforms application. The wizard loads the Java 2 SDK and then displays its Welcome panel.

See the description of the “firststeps command” on page 228 for more information.

- Click **Next** on the Welcome panel.

The wizard displays the Profile type selection panel.

- Select **Create a custom profile** and click **Next**.

The wizard displays the Custom-profile federation panel.

- Specify the host name and SOAP port of the deployment manager and click **Next**.

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

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

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

### Should you federate the node?

Federate the custom node at this time if the deployment manager is running. Select the check box to federate the node at a later time if the deployment manager is not running.

If you are unsure whether the deployment manager is running, do not federate now. Federate the node later.

If security is enabled on the deployment manager node, you must federate later using the **addNode** command to enter a user ID and password on the command.

A possibility exists that the deployment manager is reconfigured to use the non-default remote method invocation (RMI) as the preferred Java Management Extensions (JMX) connector. (Click

**System Administration > Deployment manager > Administrative services** in the administrative console of the deployment manager to verify the preferred connector type.)

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

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

#### **Federating when the deployment manager is not available**

If you federate a custom node when the deployment manager is not running or is not available because of security being enabled or for other reasons, the installation indicator in the logs is INSTCONFFAIL to indicate a complete failure. The resulting custom profile is unusable. You must move the custom profile directory out of the profile repository (the profiles installation root directory) before creating another custom profile with the same profile name.

Click **Next** to display the Profile name panel.

6. Specify a name for the profile, then click **Next**.

**Profile naming guidelines:** 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
- Illegal special characters that are not allowed within the name of a directory on your operating system, such as \*&?
- Slashes (/) or (\)

Double-byte characters are allowed.

#### **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 issued from the `bin` directory in the product installation root. When only one profile exists on a machine, every command works on the only server process in the configuration.

#### **Addressing a profile in a multi-profile environment**

When two or more profiles exist on a machine, certain commands require that you specify the profile to which the command applies. These commands use the `-profileName` parameter to identify which profile to address. You might find it easier to use the commands that in the `bin` directory of each profile.

A command in the `profiles/profile_name/bin` directory has two lines. The first line sets the `WAS_USER_SCRIPT` environment variable for the command window. The variable sets up the command environment to address the profile. The second line calls the actual command in the `install_root/bin` directory.

The actual command queries the command shell to determine the calling profile and to autonomically address the command to the calling profile.

The wizard then displays the Profile directory panel.

7. Specify a location for the profile and click **Next**.

If you click **Back** and change the name of the profile, you must manually change the name on this panel when it displays again.

The wizard displays the Node and host names panel.

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

#### **Migration considerations**

If you plan to migrate an installation of V5.x Network Deployment to V6, use the same cell name for the V6 deployment manager as you used for the V5.x cell.

After migrating the cell, the V5 managed nodes are now managed by the V6 deployment manager in compatibility mode. You can migrate individual V5 managed nodes in the cell to V6. To do so, you must create a V6 profile with the same node name as the V5 managed node.


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

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

The custom profile has the following characteristics:

Field name	Default value	Constraints	Description
Node name	The name of your machine, or a unique derivation of the machine name.	Avoid using the reserved words.  Use a unique name within the deployment manager cell.  If you plan to migrate a V5 managed node, use the same node name for this V6 custom profile.	The name is used for administration within the deployment manager cell to which the custom profile is added. Use a unique name within the deployment manager cell.  After migrating a V5 deployment manager cell to a V6 deployment manager, you can migrate the V5 custom profiles that are running in compatibility mode in the V6 deployment manager.
Host name	The DNS name of your machine.	The host name must be addressable through your network.	Use the actual DNS name or IP address of your machine to enable communication with your machine. See additional information about the host name that follows this table.

### Node name considerations

 The profiles directory path must be no longer than 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 to communicate with this node. Selecting a host name that other machines can reach within your network is extremely important. Do not use the generic localhost identifier for this value.

If you define coexisting nodes on the same computer with unique IP addresses, 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 servers (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 advantage of being totally unambiguous and also 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 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 format disadvantage is being dependent 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 ability of being redefined in the local hosts file so that the system can run the Application Server even when disconnected from the network. Define the short name to 127.0.0.1 (local loopback) in the hosts file to run disconnected. A format disadvantage is being dependent 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 you name with a numeric IP address without DNS being available. A format disadvantage 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 format disadvantage is that you cannot use the node if the host is disconnected from the network.

After specifying custom profile characteristics, the wizard displays the Port value assignment panel.

9. Specify port assignments that do not conflict for the custom profile and click **Next**.

When federating a custom profile, the **addNode** command uses non-conflicting ports. This means that you can take the default port assignments as you create the profile, and let the **addNode** command specify non-conflicting ports as you federate the node. Port assignments must be unique on a machine. application server processes on different machines can use the same port assignments without conflict.

After specifying non-conflicting port assignments, the wizard displays the Profile summary panel.

10. Click **Next** to create the custom profile or click **Back** to change the characteristics of the custom profile. The wizard displays a Status panel as the wizard creates the custom profile.

At the end of the installation, the wizard displays the Profile creation is complete panel.

11. Click **Finish** to exit the Profile creation wizard.

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

Refer to the description of the “wasprofile command” on page 275 to learn about creating this type of profile using a command instead of a wizard.

Federate the node into the deployment manager cell if you have not already done so as you created the node. Then use the deployment manager to create an application server on the node. Then you are ready to deploy an application.

Deploy an application to get started!

See “Fast paths for WebSphere Application Server” on page 9 to get started deploying applications.

### **responsefile.pct.NDmanagedProfile.txt**

This topic describes the response file for creating a custom profile. Federate the custom node into a running deployment manager cell to make the node operational.

Create a custom node using the with an options response file after logging on as root on a Linux or UNIX platform, or as a user that belongs to the administrator group on a Windows platform. Some steps of the installation procedure on a Windows platform require the user to belong to the administrator group and to have the advanced user rights *Act as part of the operating system* and *Log on as a service*.

The response file is shipped with default values.

A common use for an options file is to run the Installation wizard in silent mode, which is referred to as installing silently. The wizard reads the options file to determine responses and does not display the graphical user interface. Issue the following command to use a copy of the options file named `myresponsefile.txt` for a silent installation:

```
pctWindows.exe -options "myresponsefile.txt" -silent
```

### Federating the custom profile

Several directives in the file provide options for how the custom node is federated into the deployment manager cell:

- **-W pctfederationpanelInstallWizardBean.federateLater**  
Set this value to `true` if the deployment manager is not running or is not accessible for any of the reasons in the following description of federation.
- **-W pctfederationpanelInstallWizardBean.hostname**  
Specify a value that resolves to the system where the deployment manager is running. See the following description of host name considerations for more information.
- **-W pctfederationpanelInstallWizardBean.port**  
Specify the value of the deployment manager SOAP port. You must specify the correct value. An incorrect value prevents node federation and results in a total failure with an `INSTCONFFAILED` indicator. The default SOAP port for the deployment manager is 8879.

### Should you federate the node?

Federate the custom node if you can. However, if any of the parameters that you supply are faulty, you not only do not federate the node, you create a faulty custom profile. Federate the node at the time that you perform the silent creation of the node if, and only if, the deployment manager is running and is accessible at the host name that you specify and over the SOAP port that you specify.

If you are unsure whether the deployment manager is running, do not federate now. Federate the node later.

If security is enabled on the deployment manager node, you must federate later using the **addNode** command to enter a user ID and password on the command.

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

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

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

### Federating when the deployment manager is not available

If you federate a custom node when the deployment manager is not running or is not available because of security being enabled or for other reasons, the installation indicator in the logs is `INSTCONFFAIL` to indicate a complete failure. The resulting custom profile is unusable. You must move the custom profile directory out of the profile repository (the profiles installation root directory) before creating another custom profile with the same profile name.



## Avoiding the use of the -silent option within the options response file

A problem occurs when the -silent option exists in the file. The file works with the option during a direct call to the profile creation wizard, but fails when called from a silent product installation. See “Customizing the options response file for Network Deployment” on page 123 for information about creating a profile silently during a silent product installation.

The option is unnecessary. Avoid using the option to avoid problems.

## Response file locations

The example options response files are in two locations.

### Example files:

- responsefile.pct.NDdmgrProfile.txt
- responsefile.pct.NDmanagedProfile.txt
- responsefile.pct.NDstandAloneProfile.txt

### Location:

Table 19. Option response file locations

Product disc location	Installed location
/WAS directory	install_root/bin/ProfileCreator directory

Use the file on the product disc to install the Network Deployment product silently and create a profile.

After installing the Network Deployment product, you can use the installed response file with the -options parameter on the Profile creation wizard command.

## Required disk space

Profile	Required disk space	Required temp space
Deployment manager profile	30 MB	40 MB
Custom profile	10 MB	40 MB
Application server profile	200 MB	40 MB

## Creating an operational environment during product installation

Version 6 installation of the Network Deployment product is a two-step process:

1. Installing the core product files and feature files.
2. Creating a deployment manager profile, a custom profile, or an application server profile.

The sample options response file, “responsefile.nd.txt” on page 128, controls the first part of the installation and can also start the second part of the installation. To create a profile as part of installing the core product files, use the option in the responsefile.nd.txt file that identifies the response file for creating a profile. The profile response file lets you use the Profile creation wizard silently.

To edit and use the appropriate response file for creating a profile, perform the following procedure:

1. Copy the appropriate file from the WAS directory on the product disc to a place that you can easily identify on your machine. The example files are:



To create a:	Copy the following response file:
Deployment manager profile	responsefile.pct.NDdmgrProfile.txt
Custom profile	responsefile.pct.NDmanagedProfile.txt
Application server profile	responsefile.pct.NDstandAloneProfile.txt

- Edit the file to customize the values for your installation.
- Verify that no `-silent` option exists in the response file for the Profile creation wizard. If the option exists, the profile is not created.
- Save the file.
- Edit the `responsefile.nd.txt` file to identify the location and name of the profile response file. Change the value of the `-W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation` option to identify the file. For example:

```
-W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation=
"/opt/IBM/WebSphere/MyOptionFiles/customProfile.txt"
```
- Start the installation. For example:

```
install -options "myresponsefile.txt" -silent
```
- After the installation, examine the logs for success.

### Creating a profile after installation

Version 6 installation of the Network Deployment product is a two-step process:

- Installing the core product files and feature files
- Creating a deployment manager profile, a custom profile, or an application server profile

When the core product files exist, create a profile at any time using the Profile creation wizard. Start the wizard from the First steps console or directly using the Profile creation wizard command.

You can also use one of the following sample options response files for profiles to create a profile silently using the Profile creation wizard in silent mode. To edit and use the appropriate response file for creating a profile, perform the following procedure:





- Copy the appropriate file from the `install_root/bin/ProfileCreator` directory to a place that you can easily identify on your machine. The example files are:







To create a profile for a:	Copy the following response file:
Deployment manager	responsefile.pct.NDdmgrProfile.txt
Managed node	responsefile.pct.NDmanagedProfile.txt
Stand-alone application server	responsefile.pct.NDstandAloneProfile.txt

For example, copy the file as `my_options_file.txt`

- Edit the file to customize the values for your installation.
- Save the file.
- Start the installation.

For example:

-  `./pctAIX.bin -options my_options_file.txt -silent`
-  `./pctHPUX.bin -options my_options_file.txt -silent`
-  64-bit platforms: `./pctHPUXIA64.bin -options my_options_file.txt -silent`
-  `./pctLinux.bin -options my_options_file.txt -silent`

-  64-bit platforms: `./pct.bin -options my_options_file.txt -silent`
-  Power platforms: `./pctLinuxPPC.bin -options my_options_file.txt -silent`
-  S/390 platforms: `./pctLinux390.bin -options my_options_file.txt -silent`
-  `./pctSolaris.bin -options my_options_file.txt -silent`
-  `pctWindows.exe -options my_options_file.txt -silent`
-  64-bit platforms: `pctWindowsIA64.exe -options my_options_file.txt -silent`

5. After using the Profile creation wizard, examine the logs for success.

## Logging

The following log files record information about profile creation:

- The `install_root/logs/log.txt` file records installation status.
- The `install_root/profiles/profile_name/logs/pctLog.txt` file records installation events that occur when creating profiles with the Profile creation wizard.
- The `install_root/logs/wasprofile/wasprofile_create_profile_name.log` file records installation events that occur when creating profiles.
- The `install_root/logs/wasprofile/wasprofile_delete_profile_name.log` file records installation events that occur when deleting profiles.

See Appendix B, “Troubleshooting installation,” on page 337 for more information.

## Usage notes

- The file is not a read-only file.
- Edit this file directly with your flat file editor of choice, such as WordPad on a Windows platform.
- The file is updated when you specify the `-options` parameter when using the Profile creation wizard and the file does not yet exist.
- The file must exist to perform a silent installation. The installation program reads this file to determine installation option values when you install silently.
- Save the file in a location that you can identify when you specify the fully qualified path as part of the profile creation command.

## Naming considerations

Consider the following recommendations when supplying names for the profile and other objects.

### Naming the profile

Use the following guidelines when supplying a value for the profile name directive:

```
-W profilenamepanelInstallWizardBean.profileName
```

**Profile naming guidelines:** 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
- Illegal special characters that are not allowed within the name of a directory on your operating system, such as `*&?`
- Slashes (`/`) or (`\`)

Double-byte characters are allowed.

## Avoiding reserved names

Avoid the following reserved folder names as values for the directives in the responsefile.pct.NDstandAloneProfile.txt file and in the responsefile.pct.NDmanagedProfile.txt file:

```
-W nodehostnamepanelInstallWizardBean.nodeName=""  
-W nodehostnamepanelInstallWizardBean.hostName=""  
-W setnondmgrcellnameinglobalconstantsInstallWizardBean.value=""
```


Avoid the following reserved folder names as values for the directives in the responsefile.pct.NDdmgrProfile.txt file:

```
-W nodehostandcellnamepanelInstallWizardBean.nodeName=""  
-W nodehostandcellnamepanelInstallWizardBean.hostName=""  
-W nodehostandcellnamepanelInstallWizardBean.cellName=
```

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

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

### Node and cell name considerations

 The profiles directory path must be no longer than 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 to communicate with this node. Selecting a host name that other machines can reach within your network is extremely important. Do not use the generic localhost identifier for this value.

If you define coexisting nodes on the same computer with unique IP addresses, 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 servers (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 advantage of being totally unambiguous and also 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 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 format disadvantage is being dependent 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 ability of being redefined in the local hosts file so that the system can run the Application Server even when disconnected

from the network. Define the short name to 127.0.0.1 (local loopback) in the hosts file to run disconnected. A format disadvantage is being dependent 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 you name with a numeric IP address without DNS being available. A format disadvantage 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 format disadvantage is that you cannot use the node if the host is disconnected from the network.

### Example responsefile.pct.NDmanagedProfile.txt file

**Tip:** A custom profile must be added into a deployment manager cell to become operational. Because of this strong dependency on being a managed node, the profile is often referred to as a *managed profile* or as a managed node.

Of course, until you federate the node into a cell, the node is not managed. Another thing to keep in mind is that any federated node is a managed node, including federated nodes within Application Server profiles.

The following response file refers to the term *managed* instead of the term *custom* in many directive names. Even so, all of the directives in this response file help to define a custom profile.

```
#####
#
# Response file for WebSphere Application Server v6.0 custom profile creation
#
# This options file is located in the CD_ROOT\WAS\ directory and in the
# install_root\bin\ProfileCreator directory.
#
# To use the options file under CD_ROOT\WAS\ directory, follow the instructions
# in CD_ROOT\WAS\responsefile.nd.txt. The WebSphere Application Server
# Network Deployment installer locates this file during silent installation
# and automatically runs the silent profile creation at the end of installation.
#
# To use the options file under install_root\bin\ProfileCreator for silent profile
# creation, you must change various values in the file and use the following command
# line arguments:
#
# -options "responsefile.pct.NDmanagedProfile.txt" -silent
#
#####

#####
#
# Profile name
#
# Set the name for this custom profile. The profile name must be unique for this
# WebSphere Application Server installation.
#
#
-W profilenamepanelInstallWizardBean.profileName="profileManaged"

#####
# If you want to set this profile to be your default profile, set to "true".
# Otherwise set to "false". If this is the first profile being created, the profile
# automatically is the default.
#
-W profilenamepanelInstallWizardBean.isDefault="false"
```

```

#####
#
# Profile location
#
# Specify a directory to contain the files that define the run-time environment,
# such as commands, configuration files, and log files. If the directory contains
# spaces, enclose it in double-quotes as shown in the Windows example below.
#
# Note that spaces in the install location is only supported on Windows
# operating systems.
#
# Default Install Location:
#
#   -P installLocation="<WAS_HOME>\profiles\<PROFILE_NAME>"
#
-P installLocation="C:\Program Files\IBM\WebSphere\AppServer\profiles\profileManaged"

#####
#
# Node name
#
# Please select the node name for the Application Server. Node name under one cell
# has to be unique.
#
# If you plan to migrate a V5 deployment manager cell, the V5 managed nodes are also
# migrated to the V6 cell. To incrementally migrate an individual V5 managed node
# to V6, you must use the same node name for the V6 Application Server profile.
#
# Replace YOUR_NODE_NAME with the actual node name.
#
-W nodehostnamepanelInstallWizardBean.nodeName="YOUR_NODE_NAME"

#####
#
# Host name
#
# Specify the host name for the Application Server. The host name is the domain
# name system (DNS) name (short or long) or the IP address of this computer.
#
# Replace YOUR_HOST_NAME with the actual host name. Comment the line to use
# the default value.
#
-W nodehostnamepanelInstallWizardBean.hostName="YOUR_HOST_NAME"

#####
#
# Cell name
#
# You should not Modify this, unless absolutely necessary
#
# The Wizard would set this to short local host name + "Node##Cell" by default.
#
# If you would like to override the resolved cell name value, uncomment the line and
# replace YOUR_CELL_NAME with <YOUR_OWN_VALUE>
#
-W setnondmgrcellnameinglobalconstantsInstallWizardBean.value="YOUR_CELL_NAME"

#####
#
# Ports value assignment
#
# The following entries are used to reset port numbers used in the configuration
#

```

```

# They are currently set to the defaults.
# Please check to make sure there are no port conflicts.
# Port numbrers for each profile can be find in:
# <profile>/config/cells/<cell name>/nodes/<node name>/serverindex.xml
#
# If you specify true for the value of
# the -W pctfederationpanelInstallWizardBean.federateLater
# directive, port numbers are assigned automatically when you federate the
# node with the addNode command. The following port numbers do not apply.
#
-W pctmanagedprofileportspanelInstallWizardBean.BOOTSTRAP_ADDRESS="2809"
-W pctmanagedprofileportspanelInstallWizardBean.SOAP_CONNECTOR_ADDRESS="8878"
-W pctmanagedprofileportspanelInstallWizardBean.SAS_SSL_SERVERAUTH_LISTENER_ADDRESS="9901"
-W pctmanagedprofileportspanelInstallWizardBean.CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS="9201"
-W pctmanagedprofileportspanelInstallWizardBean.CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS="9202"
-W pctmanagedprofileportspanelInstallWizardBean.ORB_LISTENER_ADDRESS="9100"
-W pctmanagedprofileportspanelInstallWizardBean.NODE_DISCOVERY_ADDRESS="7272"
-W pctmanagedprofileportspanelInstallWizardBean.NODE_MULTICAST_DISCOVERY_ADDRESS="5000"
-W pctmanagedprofileportspanelInstallWizardBean.NODE_IPV6_MULTICAST_DISCOVERY_ADDRESS="5001"
-W pctmanagedprofileportspanelInstallWizardBean.DCS_UNICAST_ADDRESS="9353"

```

```
#####
```

```

#
# Federation
#
# A custom profile contains an empty node that must be federated to a deployment
# manager to become a functional managed node. Identify a running deployment
# manager that will administer the node or choose to federate the node later
# using the addNode command.
#
# Set to "true" if you want to federate this custom node later using the addNode
# command. You must federate this node later if the deployment manager :
#     - is not running.
#     - has security enabled.
#     - has the SOAP connector disabled
#
# If you want to federate it now, set to "" and fill in the entries for the host
# and port of the deployment manager.
#
-W pctfederationpanelInstallWizardBean.federateLater=""

```

```
#####
```

```

# Specify the host name of the deployment manager for federation.
#
-W pctfederationpanelInstallWizardBean.hostname="YOUR_DEPLOYMENT_MANAGER_HOST_NAME"

```

```
#####
```

```

# Specify the port number where the deployment manager (DMGR) is reachable on the
# above host. The default port value is "8879".
#
-W pctfederationpanelInstallWizardBean.port="YOUR_DEPLOYMENT_MANAGER_PORT_NUMBER"

```

```
#####
```

```

#
# Profile type
#
# Must be set to "managed" for installing a custom profile. Do not change this!
#
-W profiletypepanelInstallWizardBean.selection="managed"

```

## Using the Profile creation wizard to create an application server

The Profile creation wizard can create an application server profile on any machine where the core product files exist.

Before using the Profile creation wizard, install the core product files.

The Profile creation wizard is the wizard interface to the profile creation tool, `wasprofile`. See the description of the “`wasprofile` command” on page 275 for more information.


An error can occur when you have not provided enough system temporary space to create a profile. Verify that you have a minimum of 40 MB of temp space available before creating a profile.

You must have 200 MB of available disk space in the directory where you create an Application Server profile.

### Important:

You must have 30 MB of available disk space in the directory where you create a deployment manager profile.

You must have 10 MB of available disk space in the directory where you create a custom profile.

 Manually verify that the required space for creating a profile is available on AIX. A known problem in the underlying InstallShield for Multiplatforms (ISMP) code prevents proper space checking on AIX systems at the time that the product disc was created.

### Important:

After installing the core product files, you must create one of the following profiles to have an operational run-time environment:

- An application server that is described in this topic.  
An application server profile has a default server (which is `server1`), the default application that includes the snoop servlet and the hitcount servlet, and application Samples. You can federate the application server or use it as a stand-alone application server.
- A deployment manager.  
See “Using the Profile creation wizard to create a deployment manager” on page 238. The deployment manager provides a single administrative interface to a logical group of application servers on one or more machines.
- A custom profile that you must federate to make operational.  
See “Using the Profile creation wizard to create a custom profile” on page 249. A custom profile is an empty node that you can customize to include application servers, clusters, or other Java processes, such as a messaging server.

This procedure describes creating an application server profile using the graphical user interface provided by the Profile creation wizard.

You can use the Profile creation wizard in silent mode with a response file instead of a graphical user interface. See “`responsefile.pct.NDstandAloneProfile.txt`” on page 267 for examples of using the Profile creation wizard in silent mode.










You can also use the `wasprofile` command to create an application server profile. See the description of the “`wasprofile` command” on page 275 for more information.

1. Start the Profile creation wizard to create a new run-time environment.

Several ways exist to start the wizard. The initial way to start the wizard is at the end of installation by selecting the check box to launch the Profile creation wizard.

One way to start the wizard is to issue the command directly from a command line.



The command is in the *install\_root/bin/ProfileCreator* directory. The name of the command varies per platform:

-  `pctAIX.bin`
-  `pctHPUX.bin`
-  64-bit platforms: `pctHPUXIA64.bin`
-  `pctLinux.bin`
-  64-bit platforms: `pct.bin` S/390 platforms: `pctLinux390.bin`
-  Power platforms: `pctLinuxPPC.bin`
-  `pctSolaris.bin`
-  `pctWindows.exe`
-  64-bit platforms: `pctWindowsIA64.exe`

Another way to start the Profile creation wizard is to select the wizard from the First steps console.

- a. Open a command window.
- b. Change directories to the `firststeps` directory in the installation root directory:

The installation root varies by platform:

-  `/usr/IBM/WebSphere/AppServer/firststeps`
  -    `/opt/IBM/WebSphere/AppServer/firststeps`
  -  `C:\Program Files\IBM\WebSphere\AppServer\firststeps`
- c. Issue the **firststeps** command to start the console:

-   `./firststeps.sh`
-  `firststeps.bat`

- d. Select the Profile creation wizard option on the console.

The Profile creation wizard is an InstallShield for Multiplatforms application. The wizard loads the Java 2 SDK and then displays its Welcome panel.

See the description of the “firststeps command” on page 228 for more information.

2. Click **Next** on the Welcome panel.  
The wizard displays the Profile type selection panel.
3. Select the application server profile, then click **Next**.  
The wizard displays the Profile directory panel.
4. Specify a name for the profile, then click **Next**.

**Profile naming guidelines:** 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
- Illegal special characters that are not allowed within the name of a directory on your operating system, such as `*&?`
- Slashes (`/`) or (`\`)



Double-byte characters are allowed.

### 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 issued from the `bin` directory in the product installation root. When only one profile exists on a machine, every command works on the only server process in the configuration.

### Addressing a profile in a multi-profile environment

When two or more profiles exist on a machine, certain commands require that you specify the profile to which the command applies. These commands use the `-profileName` parameter to identify which profile to address. You might find it easier to use the commands that in the `bin` directory of each profile.

A command in the `profiles/profile_name/bin` directory has two lines. The first line sets the `WAS_USER_SCRIPT` environment variable for the command window. The variable sets up the command environment to address the profile. The second line calls the actual command in the `install_root/bin` directory.

The actual command queries the command shell to determine the calling profile and to autonomically address the command to the calling profile.

The wizard then displays the Profile directory panel.

5. Accept the default directory or specify a non-default location, then click **Next**. Or click **Browse** to select a different location.

If you click **Back** and change the name of the profile, you must manually change the name on this panel when it displays again.

The wizard displays the Node and host name panel.

6. Specify the characteristics for the application server, then 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 words are reserved:

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

Field name	Default value	Constraints	Description
Node name	Name of your machine	Avoid using the reserved words.	Pick 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.
Host name	DNS name of your machine	Addressable through your network.	Use the actual DNS name or IP address of your machine to enable communication with your machine. See additional information about the host name following this table.

**Node name considerations:** If you plan to migrate an installation of V5.x Network Deployment to V6 and migrate one of the managed nodes in the cell, use the same node name for the V6 application server as you used for the V5.x managed node.



The installation directory path must be no longer than 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 to communicate with this node. Selecting a host name that other machines can reach within your network is extremely important. Do not use the generic localhost identifier for this value.

If you define coexisting nodes on the same computer with unique IP addresses, 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 servers (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 advantage of being totally unambiguous and also 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 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 format disadvantage is being dependent 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 ability of being redefined in the local hosts file so that the system can run the Application Server even when disconnected from the network. Define the short name to `127.0.0.1` (local loopback) in the hosts file to run disconnected. A format disadvantage is being dependent 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 you name with a numeric IP address without DNS being available. A format disadvantage 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 format disadvantage is that you cannot use the node if the host is disconnected from the network.

After specifying application server characteristics, the wizard displays the Port value assignment panel.

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

 After specifying port assignments, the wizard displays the Windows service definition panel, if you are installing on a Windows platform.

8.  Choose whether to run the application server as a Windows service on a Windows platform and click **Next**.

Version 6 attempts to start Windows services for application server processes started by a **startServer** command. For example, if you configure an application server as a Windows service and issue the **startServer** command, the **wasservice** command attempts to start the defined service.

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

To perform this installation task, the user ID must not have spaces in its name. The ID must also belong to the administrator group and must have the advanced user rights *Act as part of the operating system* and *Log on as a service*. The Installation wizard grants the user ID the advanced user rights if it does not already have them, 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. See the *Administration: Setting up* PDF for more information.

The installation wizard shows which components are selected for installation in a pre-installation summary panel.

9. Click **Next** to create the application server or click **Back** to change the characteristics of the application server.




The wizard displays the Installation status panel that shows which components are installing.

When the installation is complete, the wizard displays the Profile creation is complete panel.

10. Click **Finish** to exit, then click **Profile creation wizard** on the First steps console to start the wizard again to create other application servers.

11. **Optional:** Add the node into a deployment manager cell.

Use the **addNode** command or start the deployment manager and the application server and use the administrative console of the deployment manager to add the application server node into the cell.

-   `application_server_install_root/ profiles/ profile_name/ bin/ addNode.sh dmgr_host 8879 -includeapps`
-  `application_server_install_root\ profiles\ profile_name\ bin\ addNode.sh dmgr_host 8879 -includeapps`

You can create an application server profile. The node within the profile has an application server named server1.

Refer to the description of the “wasprofile command” on page 275 to learn about creating this type of profile using a command instead of a wizard.

### **responsefile.pct.NDstandAloneProfile.txt**

This topic describes the response file for creating a stand-alone application server profile.

Create a stand-alone application server profile with an options response file after logging on as root on a Linux or UNIX platform, or a user that belongs to the administrator group on a Windows platform. Some steps of the installation procedure on a Windows platform require the user to belong to the administrator group and to have the advanced user rights *Act as part of the operating system* and *Log on as a service*.

The response file is shipped with default values.

A common use for an options file is to run the Installation wizard in silent mode, which is referred to as installing silently. The wizard reads the options file to determine responses and does not display the graphical user interface. Issue the following command to use a copy of the options file named myresponsefile.txt for a silent installation:

```
pctWindows.exe -options "myresponsefile.txt" -silent
```

### **Avoiding the use of the -silent option within the options response file**

A problem occurs when the -silent option exists in the file. The file works with the option during a direct call to the profile creation wizard, but fails when called from a silent product installation. See “Customizing the options response file for Network Deployment” on page 123 for information about creating a profile silently during a silent product installation.

The option is unnecessary. Avoid using the option to avoid problems.

## Response file locations

The example options response files are in two locations.

### Example files:

- responsefile.pct.NDdmgrProfile.txt
- responsefile.pct.NDmanagedProfile.txt
- responsefile.pct.NDstandAloneProfile.txt

### Location:

Table 20. Option response file locations

Product disc location	Installed location
/WAS directory	install_root/bin/ProfileCreator directory

Use the file on the product disc to install the Network Deployment product silently and create a profile.

After installing the Network Deployment product, you can use the installed response file with the -options parameter on the Profile creation wizard command.

### Required disk space

Profile	Required disk space	Required temp space
Deployment manager profile	30 MB	40 MB
Custom profile	10 MB	40 MB
Application server profile	200 MB	40 MB

## Creating an operational environment during product installation

Version 6 installation of the Network Deployment product is a two-step process:

1. Installing the core product files and feature files.
2. Creating a deployment manager profile, a custom profile, or an application server profile.

The sample options response file, “responsefile.nd.txt” on page 128, controls the first part of the installation and can also start the second part of the installation. To create a profile as part of installing the core product files, use the option in the responsefile.nd.txt file that identifies the response file for creating a profile. The profile response file lets you use the Profile creation wizard silently.

To edit and use the appropriate response file for creating a profile, perform the following procedure:

1. Copy the appropriate file from the WAS directory on the product disc to a place that you can easily identify on your machine. The example files are:

To create a:	Copy the following response file:
Deployment manager profile	responsefile.pct.NDdmgrProfile.txt
Custom profile	responsefile.pct.NDmanagedProfile.txt
Application server profile	responsefile.pct.NDstandAloneProfile.txt

2. Edit the file to customize the values for your installation.
3. Verify that no -silent option exists in the response file for the Profile creation wizard. If the option exists, the profile is not created.

4. Save the file.
5. Edit the `responsefile.nd.txt` file to identify the location and name of the profile response file. Change the value of the `-W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation` option to identify the file. For example:

```
-W pctresponsefilelocationqueryactionInstallWizardBean.fileLocation=
"/opt/IBM/WebSphere/MyOptionFiles/customProfile.txt"
```

6. Start the installation. For example:
 

```
install -options "myresponsefile.txt" -silent
```
7. After the installation, examine the logs for success.

### Creating a profile after installation

Version 6 installation of the Network Deployment product is a two-step process:

1. Installing the core product files and feature files
2. Creating a deployment manager profile, a custom profile, or an application server profile

When the core product files exist, create a profile at any time using the Profile creation wizard. Start the wizard from the First steps console or directly using the Profile creation wizard command.

You can also use one of the following sample options response files for profiles to create a profile silently using the Profile creation wizard in silent mode. To edit and use the appropriate response file for creating a profile, perform the following procedure:











1. Copy the appropriate file from the `install_root/bin/ProfileCreator` directory to a place that you can easily identify on your machine. The example files are:

To create a profile for a:	Copy the following response file:
Deployment manager	<code>responsefile.pct.NDmgrProfile.txt</code>
Managed node	<code>responsefile.pct.NDmanagedProfile.txt</code>
Stand-alone application server	<code>responsefile.pct.NDstandAloneProfile.txt</code>

For example, copy the file as `my_options_file.txt`

2. Edit the file to customize the values for your installation.
3. Save the file.
4. Start the installation.

For example:

-  `./pctAIX.bin -options my_options_file.txt -silent`
-  `./pctHPUX.bin -options my_options_file.txt -silent`
-  64-bit platforms: `./pctHPUXIA64.bin -options my_options_file.txt -silent`
-  `./pctLinux.bin -options my_options_file.txt -silent`
-  64-bit platforms: `./pct.bin -options my_options_file.txt -silent`
-  Power platforms: `./pctLinuxPPC.bin -options my_options_file.txt -silent`
-  S/390 platforms: `./pctLinux390.bin -options my_options_file.txt -silent`
-  `./pctSolaris.bin -options my_options_file.txt -silent`
-  `pctWindows.exe -options my_options_file.txt -silent`
-  64-bit platforms: `pctWindowsIA64.exe -options my_options_file.txt -silent`

5. After using the Profile creation wizard, examine the logs for success.

## Logging

The following log files record information about profile creation:

- The `install_root/logs/log.txt` file records installation status.
- The `install_root/profiles/profile_name/logs/pctLog.txt` file records installation events that occur when creating profiles with the Profile creation wizard.
- The `install_root/logs/wasprofile/wasprofile_create_profile_name.log` file records installation events that occur when creating profiles.
- The `install_root/logs/wasprofile/wasprofile_delete_profile_name.log` file records installation events that occur when deleting profiles.

See Appendix B, “Troubleshooting installation,” on page 337 for more information.

## Usage notes

- The file is not a read-only file.
- Edit this file directly with your flat file editor of choice, such as WordPad on a Windows platform.
- The file is updated when you specify the `-options` parameter when using the Profile creation wizard and the file does not yet exist.
- The file must exist to perform a silent installation. The installation program reads this file to determine installation option values when you install silently.
- Save the file in a location that you can identify when you specify the fully qualified path as part of the profile creation command.

## Naming considerations

Consider the following recommendations when supplying names for the profile and other objects.

### Naming the profile

Use the following guidelines when supplying a value for the profile name directive:

```
-W profilenamepanelInstallWizardBean.profileName
```

**Profile naming guidelines:** 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
- Illegal special characters that are not allowed within the name of a directory on your operating system, such as `*&?`
- Slashes (`/`) or (`\`)

Double-byte characters are allowed.

### Avoiding reserved names

Avoid the following reserved folder names as values for the directives in the `responsefile.pct.NDstandAloneProfile.txt` file and in the `responsefile.pct.NDmanagedProfile.txt` file:

```
-W nodehostnamepanelInstallWizardBean.nodeName=""  
-W nodehostnamepanelInstallWizardBean.hostName=""  
-W setnondmgrcellnameinglobalconstantsInstallWizardBean.value=""
```


Avoid the following reserved folder names as values for the directives in the `responsefile.pct.NDmgrProfile.txt` file:

```
-W nodehostandcellnamepanelInstallWizardBean.nodeName=""
-W nodehostandcellnamepanelInstallWizardBean.hostName=""
-W nodehostandcellnamepanelInstallWizardBean.cellName=
```

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

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

### Node and cell name considerations

 The profiles directory path must be no longer than 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 to communicate with this node. Selecting a host name that other machines can reach within your network is extremely important. Do not use the generic localhost identifier for this value.

If you define coexisting nodes on the same computer with unique IP addresses, 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 servers (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 advantage of being totally unambiguous and also 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 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 format disadvantage is being dependent 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 ability of being redefined in the local hosts file so that the system can run the Application Server even when disconnected from the network. Define the short name to `127.0.0.1` (local loopback) in the hosts file to run disconnected. A format disadvantage is being dependent 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 you name with a numeric IP address without DNS being available. A format disadvantage 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 format



disadvantage is that you cannot use the node if the host is disconnected from the network.

### Example responsefile.pct.NDstandAloneProfile.txt file

```
#####  
#  
# Response file for WebSphere Application Server v6.0 stand alone profile  
# creation  
#  
# This options file is located in the CD_ROOT\WAS\ directory and in the  
# install_root\bin\ProfileCreator directory.  
#  
# To use the options file under CD_ROOT\WAS\ directory, follow the instructions  
# in CD_ROOT\WAS\responsefile.nd.txt. The WebSphere Application Server  
# Network Deployment installer locates this file during silent installation  
# and automatically runs the silent profile creation at the end of installation.  
#  
# To use the options file under install_root\bin\ProfileCreator for silent profile  
# creation, you must change various values in the file and use the following  
# command line arguments:  
#  
# -options "responsefile.pct.NDstandAloneProfile.txt" -silent  
#  
#####  
  
#####  
#  
# Profile name  
#  
# Set the profile name for installing a stand alone profile. The profile  
# name must be unique for this WebSphere Application Server installation.  
#  
-W profilenamepanelInstallWizardBean.profileName="profileStandAlone"  
  
#####  
# If you want to set this profile to be your default profile, set to "true".  
# Otherwise set to "false". If this is the first profile being created, the profile  
# automatically is the default.  
#  
-W profilenamepanelInstallWizardBean.isDefault="false"  
  
#####  
#  
# Profile location  
#  
# Specify a directory to contain the files that define the run-time environment,  
# such as commands,configuration files, and log files. If the directory contains  
# spaces, enclose it in double-quotes as shown in the Windows example below.  
#  
# Note that spaces in the install location is only supported on Windows  
# operating systems.  
#  
# Default Install Location:  
#  
# -P installLocation="<WAS_HOME>\profiles\<PROFILE_NAME>"  
#  
-P installLocation="C:\Program Files\IBM\WebSphere\AppServer\profiles\profileStandAlone"  
  
#####  
#  
# Node name  
#  
# Please select the node name for the Application Server. Node name under one cell  
# has to be unique.
```



```

#
# If you plan to migrate a V5 deployment manager cell, the V5 managed nodes are also
# migrated to the V6 cell. To incrementally migrate an individual V5 managed node
# to V6, you must use the same node name for the V6 Application Server profile.
#
# Replace YOUR_NODE_NAME with the actual node name.
#
-W nodehostnamepanelInstallWizardBean.nodeName="YOUR_NODE_NAME"

#####
#
# Host name
#
# Specify the host name for the Application Server. The host name is the domain
# name system (DNS) name (short or long) or the IP address of this computer.
#
# Replace YOUR_HOST_NAME with the actual host name. Comment the line to use
# the default value.
#
-W nodehostnamepanelInstallWizardBean.hostName="YOUR_HOST_NAME"

#####
#
# Cell name
#
# You should not Modify this, unless absolutely necessary.
#
# The Wizard would set this to short local host name + "Node##Cell" by default.
#
# If you would like to override the resolved cell name value, uncomment the line and
# replace YOUR_CELL_NAME with <YOUR_OWN_VALUE>.
#
-W setnondmgrcellnameinglobalconstantsInstallWizardBean.value="YOUR_CELL_NAME"

#####
#
# Port value assignment
#
# The following entries are used to reset port numbers used in the configuration
#
# They are currently set to the defaults.
# Please check to make sure there are no Port Conflicts.
# Port numbers for each profile can be find in:
# <profile>/config/cells/<cell name>/nodes/<node name>/serverindex.xml
#
-W pctdefaultprofileportspanelInstallWizardBean.WC_defaulthost="9080"
-W pctdefaultprofileportspanelInstallWizardBean.WC_adminhost="9060"
-W pctdefaultprofileportspanelInstallWizardBean.WC_defaulthost_secure="9443"
-W pctdefaultprofileportspanelInstallWizardBean.WC_adminhost_secure="9043"
-W pctdefaultprofileportspanelInstallWizardBean.BOOTSTRAP_ADDRESS="2809"
-W pctdefaultprofileportspanelInstallWizardBean.SOAP_CONNECTOR_ADDRESS="8880"
-W pctdefaultprofileportspanelInstallWizardBean.SAS_SSL_SERVERAUTH_LISTENER_ADDRESS="9401"
-W pctdefaultprofileportspanelInstallWizardBean.CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS="9403"
-W pctdefaultprofileportspanelInstallWizardBean.CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS="9402"
-W pctdefaultprofileportspanelInstallWizardBean.ORB_LISTENER_ADDRESS="9100"
-W pctdefaultprofileportspanelInstallWizardBean.DCS_UNICAST_ADDRESS="9353"
-W pctdefaultprofileportspanelInstallWizardBean.SIB_ENDPOINT_ADDRESS="7276"
-W pctdefaultprofileportspanelInstallWizardBean.SIB_ENDPOINT_SECURE_ADDRESS="7286"
-W pctdefaultprofileportspanelInstallWizardBean.SIB_MQ_ENDPOINT_ADDRESS="5558"
-W pctdefaultprofileportspanelInstallWizardBean.SIB_MQ_ENDPOINT_SECURE_ADDRESS="5578"

#####
#
# Windows service

```

```

#
# The following directives are to install services for
# WebSphere Application Server on Windows.
# Using Services, you can start and stop services,
# and configure startup and recovery actions.
# Set winServiceQuery="false" will turn off the function on windows system.
# You can ignore these or comment them out for other Operating Systems.
#
-W winservicepanelInstallWizardBean.winServiceQuery="true"

#####
# Specify account type of the service. Legal values are:
#
#   localsystem   - Indicates that you choose to use Local System account.
#   specifieduser - Indicates that you choose to use specified user account.
#
-W winservicepanelInstallWizardBean.accountType="localsystem"

#####
# If you chose to install a service above with the accountType="localsystem",
# the userName and password below can be ignored. If the accountType was set to:
# accountType="specifieduser", then you must specify the User Name and Password
# which are required to install the Services. The current user must be admin or must
# have admin authority to install a Service. Also the username
# which is given here must have "Log On as a Service " authority
# for the service to run properly.
#
# Replace YOUR_USER_NAME with your username.
#
-W winservicepanelInstallWizardBean.userName="YOUR_USER_NAME"

#####
# Replace YOUR_PASSWORD with your valid password.
#
-W winservicepanelInstallWizardBean.password="YOUR_PASSWORD"

#####
# Set the startup type of the WebSphere Application Server on Windows.
# Valid values are "automatic", "manual", and "disabled".
#
-W winservicepanelInstallWizardBean.startupType="manual"

#####
# Profile type
#
# This must be set to "default" for installing a stand alone profile
# Do not change this!
#
-W profilenamepanelInstallWizardBean.selection="default"

```

## Deleting a profile

This topic describes how to manually delete a profile.

Before using the manual procedure to remove a profile, try the **wasprofile** command with the **-delete** option. For example, issue one of the following commands:



```

./wasprofile.sh -delete
                -profileName profile_name | -profilePath profile_path

```



```
wasprofile.bat -delete
               -profileName profile_name | -profilePath profile_path
```

See “wasprofile command.”

If the command does not work, use this procedure to delete the profile.




This procedure describes how to manually delete a profile when the **wasprofile -delete** command results in the following message:

```
INSTCONFFAILED: Cannot delete profile
```

1. Delete the *profiles\_install\_root/profile\_name* directory.
2. If the *install\_root/properties/profileRegistry.xml* file exists, edit the file in a flat-file editor to delete the entry for the profile, if the entry is present.

The entry resembles the following example:

```
<profile isDefault="true"
        name="BadProfile"
        path="E:\IBM\WebSphere\AppServer\profiles\BadProfile"
        template="E:\IBM\WebSphere\AppServer\profileTemplates\default"/>
```

3.   Compare the two batch files, *install\_root/properties/fsdb/\_was\_profile\_default/default.sh* and *install\_root/properties/fsdb/bad\_profile\_name.sh*. If the files are identical, delete the *install\_root/properties/fsdb/\_was\_profile\_default* directory and the *install\_root/properties/fsdb/bad\_profile\_name.sh* file. If the files are not identical, delete only the *install\_root/properties/fsdb/bad\_profile\_name.sh* file.
4.  Compare the two batch files, *install\_root\properties\fsdb\\_was\_profile\_default\default.bat* and *install\_root\properties\fsdb\bad\_profile\_name.bat*. If the files are identical, delete the *install\_root\properties\fsdb\\_was\_profile\_default* directory and the *install\_root\properties\fsdb\bad\_profile\_name.bat* file. If the files are not identical, delete only the *install\_root\properties\fsdb\bad\_profile\_name.bat* file.

See the description of the “wasprofile command” to learn more about the command-line method of working with profiles.

See “Using the Profile creation wizard” on page 234 for more information about creating profiles with the Profile creation wizard.

---

## wasprofile command

The **wasprofile** command line tool creates all Application Server run-time environments in Version 6. The command creates a profile, which is the set of files that define the run-time environment for a deployment manager, a custom profile, or a stand-alone Application Server.

The **wasprofile** command is also referred to as the *profile creation tool*.

## Introduction to terms that describe Version 6 profiles

The **wasprofile** command creates the run-time environment for a WebSphere Application Server process in a set of files called a *profile*. The profile defines the run-time environment and includes all of the files that the server processes in the run-time environment can change. The *profile creation tool* and its graphical user interface, the *Profile creation wizard*, are the only ways to create run-time environments in V6.

The *Profile creation wizard* is an InstallShield for Multiplatforms (ISMP) application. You can use the wizard to enter most of the parameters that are described in this topic. Some parameters, however, require you to

use the **wasprofile** command. You must use the **wasprofile** command to delete a profile, for instance, because the Profile creation wizard does not provide a deletion function.

However, the Profile creation wizard also performs tasks that the **wasprofile** command does not. For instance, the wizard can create a Windows service for each profile that it creates. It can also assign non-conflicting ports based on previous Version 6 port assignments.

## Core product files

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

The directory structure for V6 has 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 `profiles` directory is the default directory for creating profiles. 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.

All 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 provided there is enough available disk space.

If you put a profile in another existing folder in the installation root directory, 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.

## WebSphere Application Server profile

The **wasprofile** command line tool defines each Application Server instance of a Version 6 product.

You must run the wizard or the command line tool each time that you want to create a stand-alone Application Server. A need for more than one stand-alone Application Server on a machine is common.

Administration is greatly enhanced when using V6 profiles instead of multiple product installs. Not only is disk space saved, but updating the product is simplified when you only maintain a single set of product core files. Also, creating new profiles is faster and less prone to error than full product installs, allowing a developer to create new disposable profiles of the product for development and testing.

You can run the Profile creation wizard or the profile creation tool to create a new Application Server environment on the same machine as an existing one. Simply define unique characteristics (such as profile name and node name) for the new profile. Each profile has its own administrative console and administrative scripting interface. Each Application Server process shares all run-time scripts, libraries, the Software Development Kit, and other core product files.

## Profile types

Templates for each profile are located in the `was_home/profileTemplates` directory.

Within this directory are the `default` folder, the `dmgr` folder, and the `managed` folder. These are the paths that you indicate while using the **wasprofile** command with the `-templatePath` option.

For example: `-templatePath /usr/WebSphere/AppServer/profileTemplates/default`

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

### Deployment manager profile

The basic function of the deployment manager is to deploy applications to a cell of Application Servers, which it manages. Each Application Server that belongs to the cell is referred to as a *managed node*.

### Application Server profile

The basic function of the Application Server is to serve applications to the Internet or to an intranet.

An important product feature for the Network Deployment product 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 has 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.

There is no node agent process for a stand-alone Application Server unless you decide to add the Application Server node to a deployment manager cell. Adding the Application Server to a cell is known as *federation*. Federation changes the stand-alone Application Server into a managed node. At that point, you use the administrative console of the deployment manager to manage the node. If you remove the node from the deployment manager cell, use the administrative console and the scripting interface of the stand-alone Application Server to manage the process.

### Custom profile

The basic function of this profile that belongs to a deployment manager cell is to serve applications to the Internet or to an intranet under the management of the deployment manager.

The deployment manager changes a custom profile to a managed node by adding the node into the cell. This is also true when you add an Application Server into a cell. When either node is added to a cell, the node becomes a managed node. The *nodeagent* process is then instantiated on the managed node. The node agent acts on behalf of the deployment manager to control Application Server processes on the managed node. The node agent can start or stop Application Servers, for example.

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

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

A custom profile does not include default applications or a default server as the Application Server profile does. A custom profile is an empty node. Add the node to the deployment manager cell. Then you can use the administrative interface of the deployment manager to customize the managed node by creating clusters and Application Servers.

### 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. But you can change the location on the Profile creation wizard 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/devhost1Profile01  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile02
```

Suppose that you specify a different directory, such as `/opt/profiles`, for the profile directory field in the wizard. The profile directories resemble the following example:

```
/opt/profiles/devhost1Profile01  
/opt/profiles/devhost1Profile02
```

The following directories exist within a profile. This example assumes that a profile named `devhost1Profile01` exists:

```
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/bin  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/config  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/etc  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/firststeps  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/installableApps  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/installedApps  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/installedConnectors  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/installedFilters  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/logs  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/properties  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/samples  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/temp  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/tranlog  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01/wstemp
```

## The profile repository

The profile repository is the default location of profile-related metadata. The repository is the default location for new profiles, which is often referred to as the profiles installation root directory.

However, you can decide where to install a profile. The default location of the profile repository is the `install_root/profiles` directory. In the earlier example, creating two profiles on a Linux platform with host name `devhost1` results in the following example directories in the profile repository:

```
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile01  
/opt/IBM/WebSphere/AppServer/profiles/devhost1Profile02
```

When you specify a directory, such as `/opt/profiles`, the profiles are no longer in the default repository, which is not a problem. For example, the following locations are valid:

```
/opt/profiles/devhost1Profile01  
/opt/profiles/devhost1Profile02
```

## Location of the command file

The command file is located in the `install_root/bin` directory. The command file is a script named `wasprofile.sh` for Linux and UNIX platforms or `wasprofile.bat` for Windows platforms.

The Profile creation wizard is the graphical user interface to the command line tool. The file name of the command that calls the Profile creation wizard varies per operating system platform. See “Using the Profile creation wizard” on page 234 for more information.

## Logging

The **wasprofile** command creates a log for every profile that it creates. The logs are in the `install_root/logs/wasprofile` directory. The files are named in this pattern: `wasprofile_create_profile_name.log`.

The command also creates a log for every profile that it deletes. The logs are in the `install_root/logs/wasprofile` directory. The files are named in this pattern: `wasprofile_delete_profile_name.log`.

## Required disk space

 Manually verify that the required space for creating a profile is available on AIX. A known problem in the underlying InstallShield for Multiplatforms (ISMP) code prevents proper space checking on AIX systems at the time that the product disc was created.

An error can occur when you have not provided enough system temporary space to create a profile. Verify that you have a minimum of 40 MB of temp space available before creating a profile.

You must have 200 MB of available disk space in the directory where you create an Application Server profile.

## Network Deployment

### Tip:

You must have 30 MB of available disk space in the directory where you create a deployment manager profile.

You must have 10 MB of available disk space in the directory where you create a custom profile.

After installing the core product files, you must create one of the following profiles to have an operational run-time environment:

- An application server that is described in this topic.

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

- A deployment manager.

See “Using the Profile creation wizard to create a deployment manager” on page 238. The deployment manager provides a single administrative interface to a logical group of application servers on one or more machines.

- A custom profile that you must federate to make operational.

See “Using the Profile creation wizard to create a custom profile” on page 249. A custom profile is an empty node that you can customize to include application servers, clusters, or other Java processes, such as a messaging server.

## Concurrent profile creation

**Important:** Concurrent profile creation is not supported at this time for one set of core product files. Concurrent attempts to create profiles result in a warning about a profile creation already in progress.

## Entering lengthy commands on more than one line

The length of the `wasprofile` command can exceed the normal shell window limit for one line of 256 characters. If your command is longer than the limit, issue the command on multiple lines by ending a line with a backward slash, pressing **Enter**, and continuing the command on the next line.

For example, on a Solaris system, the following command requires input on multiple lines:



```

./wasprofile.sh \
-create -profileName bladetcb6profile \
-profilePath /usr/IBM/WebSphere/AppServer/profiles/bladetcb6profile \
-templatePath /usr/WebSphere/AppServer/profileTemplates/default \
-nodeName bladetcb6node \
-cellName bladetcb6Cell \
-hostName bladetcb6.rtp.raleigh.ibm.com

```

Omit the line continuation character from the last line to signal the end of the command to the operating system.

## wasprofile.sh command syntax

Get help for the command:

```

# ./wasprofile.sh -help

# ./wasprofile.sh -create -help

# ./wasprofile.sh -delete -help

# ./wasprofile.sh -getName -help

# ./wasprofile.sh -getPath -help

# ./wasprofile.sh -validateRegistry -help

# ./wasprofile.sh -validateAndUpdateRegistry -help

```

List existing profiles:

```

# ./wasprofile.sh -listProfiles
    [-debug]

```

Delete profiles:

```

# ./wasprofile.sh -delete
    -profileName profile_name | -profilePath profile_path
    [-debug]

```

Create new profiles:

```

wasprofile.sh -create
    -profileName profile_name
    -profilePath fully_qualified_profile_path
    -templatePath template_path
    -nodeName node_name
    -cellName cell_name
    -hostName host_name
    -server iSeries_server_name
    [-dmgrHost host_name]
    [-dmgrPort SOAP_port]
    [-startingPort starting_port | -portsFile filepath]
    -winserviceCheck true | false
    -winserviceAccountType specifieduser | localsystem
    -winserviceUserName yourusername
    -winservicePassword yourpassword
    -winserviceStartupType manual | automatic | disabled
    [-debug]

```

Get name of existing profile from path:

```

# ./wasprofile.sh -getName
    -profilePath profile_path
    [-debug]

```

Get path of existing profile from name:



```
# ./wasprofile.sh -getPath
    -profileName profile_name
    [-debug]
```

Check the integrity of the profile registry:

```
# ./wasprofile.sh -validateRegistry
    [-debug]
```

Check the integrity of the profile registry, removing profiles that are not found:

```
# ./wasprofile.sh -validateAndUpdateRegistry
    [-backup file_name]
    [-debug]
```

## wasprofile.bat command syntax

Get help for the command:

```
# ./wasprofile.bat -help

# ./wasprofile.bat -create -help

# ./wasprofile.bat -delete -help

# ./wasprofile.bat -getName -help

# ./wasprofile.bat -getPath -help

# ./wasprofile.bat -validateRegistry -help

# ./wasprofile.bat -validateAndUpdateRegistry -help
```

List existing profiles:

```
wasprofile.bat -listProfiles
    [-debug]
```

Delete profiles:

```
wasprofile.bat -delete
    -profileName profile_name | -profilePath profile_path
    [-debug]
```

Create new profiles:

```
wasprofile.bat -create
    -profileName profile_name
    -profilePath fully_qualified_profile_path
    -templatePath template_path
    -nodeName node_name
    -cellName cell_name
    -hostName host_name
    -serverName server_name
    [-dmgrHost host_name]
    [-dmgrPort SOAP port]
    [-startingPort starting_port | -portsFile filepath]
    -winserviceCheck true | false
    -winserviceAccountType specifieduser | localsystem
    -winserviceUserName yourusername
    -winservicePassword yourpassword
    -winserviceStartupType manual | automatic | disabled
    [-debug]
```

When the `-startingPort` parameter is not used, the profile creation tool uses the default port settings specified in the `serverindex.xml` file.

Get name of existing profile from path:

```
wasprofile.bat -getName  
               -profilePath fully_qualified_profile_path  
               [-debug]
```

Get path of existing profile from name:

```
wasprofile.bat -getPath  
               -profileName profile_name  
               [-debug]
```

Check integrity of profile registry:

```
wasprofile.bat -validateRegistry  
               [-debug]
```

Check integrity of profile registry, removing unfound profiles:

```
wasprofile.bat -validateAndUpdateRegistry  
               [-backup file_name]  
               [-debug]
```

## Parameters

Supported arguments include:

### **-augment**

Refreshes or augments the given profile using the template in the `templatePath` parameter.

### **-backup** *file\_name*

Backs up the profile registry file to a file with the file name specified.

### **-cellname** *file\_name*

Specifies the cell name of the profile.

### **-create**

Creates the profile.

### **-debug**

Turns on the debug function of the Ant utility, which the **wasprofile** command uses.

### **-delete**

Deletes the profile.

### **-dmgrHost** *host\_name*

Identifies the machine where the deployment manager is running. Specify this parameter and the `dmgrPort` parameter to federate a custom profile as it is created.

The host name can be the long or short DNS name or the IP address of the deployment manager machine.

Specifying this optional parameter directs the **wasprofile** command to attempt to federate the custom node into the deployment manager cell as it creates the custom profile. This parameter is ignored when creating a deployment manager profile or an Application Server profile.

### **Federating when the deployment manager is not available**

If you federate a custom node when the deployment manager is not running or is not available because of security being enabled or for other reasons, the installation indicator in the logs is `INSTCONFFAIL` to indicate a complete failure. The resulting custom profile is unusable. You must move the custom profile directory out of the profile repository (the profiles installation root directory) before creating another custom profile with the same profile name.

### **-dmgrPort** *port\_number*

Identifies the SOAP port of the deployment manager. Specify this parameter and the `dmgrHost` parameter to federate a custom profile as it is created. The deployment manager must be running and accessible.

If you have enabled security or changed the default JMX connector type, you cannot federate with the **wasprofile** command. Use the **addNode** command instead.

**-getName**

Gets the name for a profile registered at a given file system path. Requires the `-profilePath` parameter.

**-getPath**

Gets the file system location for a profile of a given name. Requires the `-profileName` parameter.

**-help**

Displays command syntax.

**-hostName** *host\_name*

Specifies the host name where you are creating the profile. This should match the host name that you specified during installation of the initial product.

**-listProfiles**

Lists all defined profiles.

**-nodeName** *node\_name*

Specifies the node name for the node that is created with the new profile. Use a unique value within the cell on the machine. Each profile that shares the same set of product binaries must have a unique node name.

**-portsFile** *file\_path*

An optional parameter that specifies the path to a file that defines port settings for the new profile. When omitted, the **wasprofile** tool looks for the `install_root/profileTemplates/profile_type/actions/portsUpdate/bin/portdef.props` file.


Do not use this parameter when using the `startingPort` parameter.

**-profileName** *profile\_name*


Specifies the name of the profile. Use a unique value when creating a profile. Each profile that shares the same set of product binaries must have a unique name.

**-profilePath** *profile\_path*

Specifies the fully qualified path to the profile.

 Specify the full path to avoid an ANT scripting limitation that can cause a failure when federating the profile into a cell. For example:

```
-profilePath C:\IBM\WebSphere\AppServer\profiles\profile01
```

 If the fully qualified path contains spaces, enclose the value in quotation marks.

Specifies the name of the application server. If not specified, the default application server name is `server1`. This parameter is supported only on the OS/400 operating system.

**-startingPort** *startingPort*

Specifies the starting port number for generating all ports for the profile. If not specified, the **wasprofile** command uses default ports specified in the `serverindex.xml` file.

**-templatePath** *template\_path*

Specifies the path to the templates in the shared binaries.

**-validateAndUpdateRegistry** *registry\_file backup\_file*

Checks all of the profiles that are listed in the profile registry to see if the profiles are present on the file system. Removes any missing profiles from the registry. Returns a list of the missing profiles that were deleted from the profile.

### **-validateRegistry** *registry\_file*

Checks all of the profiles that are listed in the profile registry to see if the profiles are present on the file system. Returns a list of missing profiles.

### **Windows** **-winserviceAccountType** *type\_of\_owner\_account*

The type of the owner account of the Windows service created for the profile can be either `specifieduser` or `localsystem`. The Windows service can run under the local account of the user who is creating the profile.

### **Windows** **winserviceCheck** *value*

The value can be either `true` or `false`. Specify `true` to create a Windows service for the server process that is created within the profile. Specify `false` to not create the Windows service.

### **Windows** **-winservicePassword** *yourpassword*

Specify the password for the specified user or the local account that is to own the Windows service.

### **Windows** **-winserviceStartupType** *startup\_type*

Possible `startup_type` values are:

- `manual`
- `automatic`
- `disabled`

See the *Administration: Setting up* PDF for more information.

### **Windows** **-winserviceUserName** *user\_ID*

Specify your user ID so that Windows can verify you as an ID that is capable of creating a Windows service. Your user ID must belong to the administrator group and have the following advanced user rights, *Act as part of the operating system* and *Log on as a service*

## Use case scenarios

Use cases are a description of common tasks for which the tool is used.

### **Scenario: Creating a deployment manager profile on a Linux or UNIX platform**

To create a deployment manager profile for user `shasti`:

```
wasprofile.sh -create
  -profileName shasti
  -profilePath ~/shasti/WebSphere
  -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/dmgr
  -cellName shaix1
  -hostName planetaix
  -nodeName dmgr1
```

On an Express product or a base product installation, the command does not create anything because the deployment manager template is not present.

On a Network Deployment product installation, the command creates a deployment manager profile named `shasti` in a cell named `shaix1` in location `~/shasti/WebSphere` with a node name of `dmgr1`.

### **Scenario: Creating a deployment manager profile on a Windows platform**

To create a server process for user `shasti`:

```
wasprofile.sh -create
  -profileName shasti
  -profilePath G:\shasti\WebSphere
  -templatePath C:\IBM\WebSphere\AppServer\profileTemplates\dmgr
  -cellName shwindows1
  -hostName planetnt
  -nodeName dmgr1
```

On an Express installation or on a base WebSphere Application Server product installation, the command does not create a deployment manager profile because the dmgr template does not exist in either product.

On a Network Deployment product installation, the command creates a deployment manager profile named shasti in a cell named shwindows1 in location G:\shasti\WebSphere with a node name of dmgr1.

### Scenario: Creating a deployment manager profile in a multiuser environment on a Linux or UNIX platform

Follow these steps to create a server process for user shasti in a multiuser environment:

1. Create the server process:

```
wasprofile.sh -create
              -profileName shasti
              -profilePath ~/shasti/WebSphere
              -templatePath /opt/IBM/WebSphere/AppServer/profileTemplates/dmgr
              -cellName shaix1
              -hostName myhost
              -nodeName dmgr1
              -startingPort 12000
```

2. Change the owner of the folder:

```
chown -R shasti ~/shasti2/WebSphere
```

3. As a convenience, add a call to script ~/shasti/WebSphere/bin/setupCmdLine.sh in the profile of user shasti to set the environment when user shasti logs in.

4. Give these folder permissions to user shasti:

```
install_root/bin          --- rx (read and execute)
install_root/java         --- rx
install_root/properties  ----r (read)
install_root/deploytool  ----r
install_root/config       ----r
install_root/lib          ----r
install_root/classes      ----r
install_root/null         ----r
install_root/samples      ----r
install_root/Web          ----r
```

### Scenario: Deleting a profile

The following command is on more than one line for clarity. Enter the command on one line to delete the profile named shasti:

```
wasprofile.sh -delete
              -profileName shasti
```

### Scenario: Using predefined port numbers

When you use the wasprofile tool without the -startingPort parameter, the tool uses the /profileTemplates/profile\_type/actions/portsUpdate/bin/portdef.props file to set the initial ports.

### Example of using the -portsFile parameter

Copy the file, edit the port settings, and use your copy by using the -portsFile parameter as shown in the following example:

```
wasprofile.bat
  -create
  -profileName Wow_Profile
  -profilePath
    C:\ExpressV6\IBM\WebSphere\AppServer\profiles\Wow_Profile
  -templatePath
    C:\ExpressV6\IBM\WebSphere\AppServer\profileTemplates\default
```

```

-nodeName Wow_node
-cellName Wow_cell
-hostName IoyAllen
-portsFile C:\temp\ports\portdef.props

```

Suppose that the portdef.props file has the following values:

```

WC_defaulthost=39080
WC_adminhost=39060
WC_defaulthost_secure=39443
WC_adminhost_secure=39043
BOOTSTRAP_ADDRESS=32809
SOAP_CONNECTOR_ADDRESS=38880
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=39401
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=39403
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=39402
ORB_LISTENER_ADDRESS=39100
DCS_UNICAST_ADDRESS=39353
SIB_ENDPOINT_ADDRESS=37276
SIB_ENDPOINT_SECURE_ADDRESS=37286
SIB_MQ_ENDPOINT_ADDRESS=35558
SIB_MQ_ENDPOINT_SECURE_ADDRESS=35578

```

As you run the command, messages similar to the following appear in the output stream:

```

replaceRegExpAllInstancesOfGivenTokenWithGivenValueForTheGivenFile:
[echo] File C:\ExpressV6\IBM\WebSphere\AppServer\profiles\
Wow_Profile/config/templates/default/serverentry-template.xml:
setting CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS to 39403

```

```

...
replaceRegExpAllInstancesOfGivenTokenWithGivenValueForTheGivenFile:
[echo] File C:\ExpressV6\IBM\WebSphere\AppServer\profiles\
Wow_Profile/config/templates/default/serverentry-template.xml:
setting CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS to 39402
...

```

The resulting serverindex.xml file looks similar to the following example:

```

<?xml version="1.0" encoding="UTF-8"?>
<serverindex:ServerIndex xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
...
  <specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="BOOTSTRAP_ADDRESS">
    <endPoint xmi:id="EndPoint_..." host="IBMmachine" port="32809"/>
  </specialEndpoints>
  <specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="SOAP_CONNECTOR_ADDRESS">
    <endPoint xmi:id="EndPoint_..." host="IBMmachine" port="38880"/>
  </specialEndpoints>
  <specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="SAS_SSL_SERVERAUTH_LISTENER_ADDRESS">
    <endPoint xmi:id="EndPoint_..." host="IBMmachine" port="39401"/>
  </specialEndpoints>
  <specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS">
    <endPoint xmi:id="EndPoint_..." host="IBMmachine" port="39403"/>
  </specialEndpoints>
  <specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS">
    <endPoint xmi:id="EndPoint_..." host="IBMmachine" port="39402"/>
  </specialEndpoints>
  <specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="WC_adminhost">
    <endPoint xmi:id="EndPoint_..." host="*" port="39060"/>
  </specialEndpoints>
  <specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="WC_defaulthost">

```

```

    <endPoint xmi:id="EndPoint_..." host="*" port="39080"/>
</specialEndpoints>
<specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="DCS_UNICAST_ADDRESS">
    <endPoint xmi:id="EndPoint_..." host="IBMmachine" port="39353"/>
</specialEndpoints>
<specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="WC_adminhost_secure">
    <endPoint xmi:id="EndPoint_..." host="*" port="39043"/>
</specialEndpoints>
<specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="WC_defaulthost_secure">
    <endPoint xmi:id="EndPoint_..." host="*" port="39443"/>
</specialEndpoints>
<specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="SIB_ENDPOINT_ADDRESS">
    <endPoint xmi:id="EndPoint_..." host="*" port="37276"/>
</specialEndpoints>
<specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="SIB_ENDPOINT_SECURE_ADDRESS">
    <endPoint xmi:id="EndPoint_..." host="*" port="37286"/>
</specialEndpoints>
<specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="SIB_MQ_ENDPOINT_ADDRESS">
    <endPoint xmi:id="EndPoint_..." host="*" port="35558"/>
</specialEndpoints>
<specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="SIB_MQ_ENDPOINT_SECURE_ADDRESS">
    <endPoint xmi:id="EndPoint_..." host="*" port="35578"/>
</specialEndpoints>
<specialEndpoints xmi:id="NamedEndPoint_..."
    endPointName="ORB_LISTENER_ADDRESS">
    <endPoint xmi:id="EndPoint_..." host="IBMmachine" port="39100"/>
</specialEndpoints>
</serverEntries>
</serverindex:ServerIndex>

```

The **wasprofile** command creates a copy of the current portdefs.props file in the *install\_root\profiles\profile\_name\logs* directory.

Do not use the portsFile parameter when using the startingPort parameter. The two parameters are mutually exclusive.

### Scenario: Incrementing default port numbers from a starting point

The **wasprofile** command can assign port numbers based on a starting port value that you give on the command line, using the `-startingPort` parameter. The tool assigns port numbers sequentially from the starting port number value.

The order of port assignments is arbitrary. Predicting assignments is not possible.

For example, ports created with `-startingPort 20002` would appear similar to the following example:

#### Assigned ports for an Application Server profile

```

WC_defaulthost=20002
WC_adminhost=20003
WC_defaulthost_secure=20004
WC_adminhost_secure=20005
BOOTSTRAP_ADDRESS=20006
SOAP_CONNECTOR_ADDRESS=20007
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=20008
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=20009
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=20010
ORB_LISTENER_ADDRESS=20011

```

```
DCS_UNICAST_ADDRESS=20012
SIB_ENDPOINT_ADDRESS=20013
SIB_ENDPOINT_SECURE_ADDRESS=20014
SIB_MQ_ENDPOINT_ADDRESS=20015
SIB_MQ_ENDPOINT_SECURE_ADDRESS=20016
```

### Assigned ports for a custom profile

```
WC_defaulthost=20002
WC_adminhost=20003
WC_defaulthost_secure=20004
WC_adminhost_secure=20005
BOOTSTRAP_ADDRESS=20006
SOAP_CONNECTOR_ADDRESS=20007
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=20008
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=20009
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=20010
ORB_LISTENER_ADDRESS=20011
CELL_DISCOVERY_ADDRESS=20012
DCS_UNICAST_ADDRESS=20013
```

### Assigned ports for a deployment manager profile

```
CELL_DISCOVERY_ADDRESS=20010
BOOTSTRAP_ADDRESS=20004
DRS_CLIENT_ADDRESS=7989
SOAP_CONNECTOR_ADDRESS=20005
ORB_LISTENER_ADDRESS=20009
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS=20006
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS=20008
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS=20007
WC_adminhost=20002
DCS_UNICAST_ADDRESS=20011
WC_adminhost_secure=20003
```

### Example of startingPort parameter use

The following example of using the **wasprofile** command creates ports from an initial value of 20002, with the content shown in the previous example:

```
wasprofile.bat -create
               -profileName shasti
               -profilePath G:\shasti\WebSphere
               -templatePath template_path
               -nodeName W2K03
               -cellName W2K03_Cell01
               -hostName plane.tnt
               -startingPort 20002
```

### Scenario: Setting up and using the profile environment

Most tasks that you perform in a profile are done using the `-profileName` attribute on the command line tools that you use. Such a scenario might be:

1. Create the server process using the `install_root/bin/wasprofile.sh` (or `wasprofile.bat`) script from the original installation. Assume that you create the Profile02 profile.
2. In that command window or in another, change directories to the `/bin` directory of the new server process.
3. Establish a temporary override for the normal WebSphere Application Server environment by using the `-profileName` attribute on a command you issue. In the same window, start `server1` by changing directories to the `install_root/bin` directory of the original installation and issuing the command. There is no such command in the `/bin` directory of the server process:

```
startServer.sh server1 -profileName Profile02
```



4. Notice the changes in the environment. Display all of the ports for the machine to see the ports that you set for the server process. For example, in a Linux bash shell or in the command window on a Windows platform, type:

```
netstat -a
```

5. Open a browser window and point it at the port defined for the HTTP\_TRANSPORT\_ADMIN port of the new process. For example, suppose the setting is HTTP\_TRANSPORT\_ADMIN=20003. Open the administrative console for server1 by pointing your browser at:

```
http://hostname_orIP_address:20003/ibm/console/
```

### Scenario: Profile creation for a non-root user

Two methods exist for a non-root user to create a profile:

- The root user creates the profile and assigns ownership to the non-root user.
- A non-root user creates a profile after getting write permission to the appropriate directories.

**Remember:** An ease-of-use limitation exists for non-root users who create profiles. Mechanisms within the Profile creation wizard 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 creation wizard for the profile name, node name, cell name, and port assignments. Consider assigning non-root users a range of values for each of the fields. You can assign responsibility to the non-root profilers for adhering to their proper value ranges and for maintaining the integrity of their own definitions.

**Root creates the profile and assigns ownership to a non-root user:** The root user can create a profile and assigns ownership of the profile directory to a non-root user.

1. The root user creates the profile with the following command:

```
./wasprofile.sh -create -profileName profile01 -profilePath install_root/profiles/profile01
```

2. The root user changes ownership of the directory for the profile to the non-root user with the following command:

```
chown -R user1 install_root/profiles/profile01
```

**A non-root user creates the profile (advanced option):** The root user can grant write permission to the appropriate files and directories to a non-root user. The non-root user can then create the profile. You can create a group for users who are authorized to create profiles. Or you can give everyone the ability to create profiles. The following example shows how to create a group that is authorized to create profiles.

1. Log on to the Application Server system as root.
2. Create the profilers group that you can use to create profiles.
3. Create a user named user1 to create profiles.
4. Add users root and user1 to the profilers group.
5. Log off and back on as root to pick up the new group.
6. As root, use operating system tools to change file permissions.

The following example assumes that the installation root directory is /opt/IBM/WebSphere/AppServer:

```
mkdir /opt/IBM/WebSphere/AppServer/logs/wasprofile
chgrp profilers /opt/IBM/WebSphere/AppServer/logs/wasprofile
chmod g+wr /opt/IBM/WebSphere/AppServer/logs/wasprofile
chgrp profilers /opt/IBM/WebSphere/AppServer/properties
chmod g+wr /opt/IBM/WebSphere/AppServer/properties
chmod g+wr /opt/IBM/WebSphere/AppServer/properties/profileRegistry.xml
```

You might have to change the permissions on additional /opt/IBM/WebSphere/AppServer directories if you encounter permission problems.

7. The non-root user who belongs to the profilers group can then create a profile in any directory to which the non-root user has write permission.

If the non-root user does not have write access to any directories, it is up to the root user to change that situation. If the non-root user does not have write access to the /tmp directory, it is up to the root user to change that as well.

The commands listed in step 6 give users assigned to the profilers group the ability to write to the /opt/IBM/WebSphere/AppServer/logs/wasprofile directory and to the /opt/IBM/WebSphere/AppServer/properties directory. It is not necessary to write to any other directories in the installation root of your WebSphere Application Server product.

Have non-root users create a profiles directory in their own area, not in the installation root directory of the product.

---

## Using the installation verification test

This topic describes how to use the installation verification test (IVT). The IVT verifies that the installation of the application server or deployment manager profile was successful. A *profile* consists of files that define the run-time environment for a deployment manager or an application server. Each profile has its own IVT command.

After installing the Network Deployment product and creating a deployment manager or application server profile, you are ready to use the installation verification test (IVT).

The IVT program scans product log files for errors and verifies core functionality of the product installation.

The Profile creation wizard creates profiles. After creating a profile, the Profile creation wizard displays a prompt for starting the First steps console. The First steps console is unique for each profile. See “firststeps command” on page 228 for more information.

Installation verification is the first option on the First steps console.




The IVT program for an application server profile starts and monitors the application server process, which is the server1 process. The installation verification for a deployment manager profile starts and monitors the deployment manager process, which is the dmgr process. The IVT works differently for the deployment manager profile than for a stand-alone application server. On a stand-alone application server, the IVT queries servlets from the ivtApp application. However, the deployment manager does not have the ivtApp application, so the IVT looks at log files only.

1. Start the First steps console and select **Installation verification** after creating a deployment manager profile or an application server profile.

No installation verification is possible for a custom profile. After federating the node and using the deployment manager to create a server, you can start the server process to verify its functionality.

Select the check box to launch the First steps console at the end of profile creation. You can also start the First steps console from the command line, as described in “firststeps command” on page 228.

You can also start the “ivt command” on page 291 directly from the bin directory of the profile:

-   `install_root/profiles/profile_name/bin/ivt.sh`
-  `install_root\profiles\profile_name\bin\ivt.bat`

If you create profiles in another location, the ivt script location is within the *profile\_home/bin* directory.

2. Observe the results in the First steps status window.

The default log file for installation verification is the *install\_root/profiles/profile\_name/logs/ivtClient.log*. If you create profiles in another location, the file path is *profile\_home/logs/ivtClient.log*.

The IVT provides the following useful information about the application server:

- The application server name
- The name of the profile
- The profile file path

- The type of profile
- The node name
- The current encoding
- The port number for the administrative console
- Various informational messages that include the location of the `SystemOut.log` file and how many errors are listed within the file
- A completion message

As the IVT starts the application server on a Windows platform, the IVT attempts to start the Windows service for the application server, if a Windows service exists. This is true even though the Windows service might have a manual startup type.

If you federate a stand-alone application server, you can still run the IVT on the server.

The IVT provides the following useful information about the deployment manager:

- The deployment manager server name: `dmgr`
- The name of the profile
- The profile file path
- The type of profile: `dmgr`
- The cell name
- The node name
- The current encoding
- The port number for the administrative console
- Various informational messages that include the location of the `SystemOut.log` file and how many errors are listed within the file
- A completion message

As the IVT starts the deployment manager on a Windows platform, the IVT attempts to start the Windows service for the deployment manager if a Windows service exists. This is true even though the Windows service might have a manual startup type.

3. If the log shows that errors occurred during the installation verification, correct the errors and run the IVT again. If necessary, create a new profile after correcting the error, and run the IVT on the new profile.

The IVT program starts the server process automatically if the server is not running. Once the server initializes, the IVT runs a series of verification tests. The tool displays pass or fail status in a console window. The tool also logs results to the `profile_home/logs/ivtClient.log` file. As the IVT verifies your system, the tool reports any detectable errors in the `SystemOut.log` file.

## ivt command

The `ivt` command starts the installation verification test (IVT) program. The IVT verifies that the installation of the application server or deployment manager profile was successful. A *profile* consists of files that define the run-time environment for a deployment manager or an application server. Each profile has its own IVT command.




The IVT program starts the application server or deployment manager automatically if the server process is not already running. After the server process initializes, the IVT runs a series of verification tests and displays pass or fail status in a console window.

The IVT program scans the `SystemOut.log` file for errors and verifies core functionality of the profile.

You can start the IVT program from the command line or from the First steps console.

## Location of the command file

The location of the `ivt.sh` or `ivt.bat` script for any profile is:

-   `install_root/profiles/profile_name/bin/ivt.sh`
-  `install_root\profiles\profile_name\bin\ivt.bat`

## Parameters

The following parameters are associated with this command.

### **server\_name**

Required parameter that identifies the name of the server process, such as `server1` or `dmgr`.

### **profile\_name**

Required parameter that identifies the name of the profile that contains the server definition.

### **-p** *server\_port\_number*




Optional parameter that identifies the `default_host` port when the port is not 9080, which is the default.

### **-host** *machine\_host\_name*

Optional parameter that identifies the host machine of the profile to test. The default is `localhost`.

## Syntax for the ivt command

Use the following syntax for the command:

-   `install_root/profiles/profile_name/bin/ivt.sh`
-  `install_root\profiles\profile_name\bin\ivt.bat`

## Logging

The `ivt` command logs results to the `install_root/profiles/profile_name/logs/ivtClient.log` file.

## Example

The following examples test the `server1` process in the `profile01` profile on the `myhost` machine using the `default_host` on port 9081.

### 

```
ivt.bat server1 profile01 -p 9081 -host myhost
```

### 

```
ivt.sh server1 profile01 -p 9081 -host myhost
```

---

## Chapter 11. Communicating with Web servers

The WebSphere Application Server works with a Web server to route requests for dynamic content, such as servlets, from Web applications. The Web servers are necessary for directing traffic from browsers to the applications that run in WebSphere Application Server. The Web server plug-in uses the XML configuration file to determine whether a request is from the Web server or the Application Server.

The Web server plug-ins for distributed platform Web servers are provided on a separate CD from the WebSphere Application Server products. A Web Server Plug-in Installation Wizard is also provided on that CD. “Installing Web server plug-ins” on page 136 describes how to install a Web server plug-in and create a Web server definition.

1. Install your Web server if it is not already installed. If you want to use an IBM HTTP Server, see “Installing IBM HTTP Server” on page 134. Otherwise, see the installation information provided with your Web server.
2. Ensure that your Web server is configured to perform operations required by Web applications, such as GET and POST. Typically, this involves setting a directive in the Web server configuration file (such as the `httpd.conf` file for an IBM HTTP Server). Refer to the Web server documentation for instructions. If an operation is not enabled when a servlet or JSP file requiring the operation is accessed, an error message displays, such as this one from the IBM HTTP Server:  
HTTP method POST is not supported by this URL.
3. Use the Plug-in Installation wizard to install the appropriate plug-in file to your Web server and run the script `configureWeb_server_name` created by the wizard to create and configure the Web server definition in the WebSphere configuration repository. The following substeps are performed during the plug-in installation process. See the Plug-in Installation Roadmap for additional information.
  - a. A node is created. An unmanaged node is created when the Web server is on a different computer from the Application Server. An unmanaged node is a node that does not have a WebSphere node agent running on it. Using unmanaged nodes, WebSphere Application Server can represent servers that are not application servers within its configuration topology. This representation enables connection information between those servers and application servers to be maintained. The *Administration: Operations* PDF describes how to create a node.
  - b. A Web server definitions is created. You can also use either use the administrative console or use the `ConfigureWebServerDefintion.jacl` script to create a Web server definition.  
If you use the administrative console:
    - 1) Select the node that was created in the preceding step, and in the Server name field, enter the local name of the Web server for which you are creating a Web server definition.
    - 2) Use the wizard to complete the Web server definition.
  - c. An application or modules are mapped to a Web server. If an application that you want to use with this Web server is already installed, the application is automatically mapped to the Web server. If the application is not installed, select this Web server during the Map modules to servers step of the application installation process.
  - d. Master repository is updated and saved.
4. **Optional:** Configure the plug-in. Use either the administrative console, or issue the **GenPluginCfg** command to create your `plugin-cfg.xml` file.

When setting up your Web server plug-in, you must decide whether or not to have the configuration automatically generated in response to a configuration change. When the Web server plug-in configuration service is enabled and any of the following conditions occur, the plug-in configuration file is automatically generated:

- When the Web server is created or saved.
- When an application is installed.
- When an application is uninstalled.

- When the virtual host definition is updated

Generating or regenerating the configuration file might take a while to complete. After it finishes, all objects in the administrative cell use their newest settings, which the Web server can access. If the Application Server is on the same physical machine as the Web server, the regeneration usually takes about 30 to 60 seconds to complete. The regeneration takes longer if they are not both on the same machine.

**Important:** When the plug-in configuration file is first generated, it does not include `admin_host` on the list of virtual hosts. “Allowing Web servers to access the administrative console” on page 216 describes how to add it to the list.

To use the administrative console:

- Select **Servers > Web Servers > *webserver* > plug-in properties**.
- Select **Automatically generate plug-in configuration file** or click on one or more of the following topics to manually configure the `plugin-cfg.xml` file:
  - Caching
  - Request and response
  - Request routing
  - Service

Web server plug-in configuration properties maps each property to one of these topics.

- Click **OK**.
  - You might need to stop the application server and then start the application server again to enable the Web server to locate the `plugin-cfg.xml` file.
- If you want to use Secure-socket layer (SSL) with this configuration, use the plug-in’s installation wizard to install the appropriate GSKIT installation image file on your workstation. See the *Security Guide* PDF for information on how to configure GSKIT.
  - If you want to enable the Web server plug-in to use private headers, define an SSL configuration repertoire that defines a trust file. Then in the administrative console, select **Application servers > server1 > Web Container Settings > Web Container Transport Chains > *transport\_chain* > SSL Inbound Channel (SSL\_2)** and specify this repertoire for that transport chain. If you try to use private headers without setting up an SSL configuration repertoire that does not include a trust file definition, the private headers will be ignored. If the private headers are ignored, the application server might not locate the requested application.

After you enable the use of private headers, the transport chain’s SSL inbound channel trusts all private headers it receives. Therefore, you must ensure that all paths to the transport chain’s SSL inbound channel are trusted.

- Tune your Web server with Web server tuning parameters.
- Propagate the plug-in configuration. The plug-in configuration file (`plugin-cfg.xml`) is automatically propagated to the Web server if the Web server plug-in configuration service is enabled, and one of the following is true:
  - The Web server is a local Web server. (It are located on the same machine as an application server.)
  - The Web server is a remote IBM HTTP Server (IHS) Version 6.0 that has a running IHS Administrative server.

If neither of these conditions is true, the `plugin-cfg.xml` file must be manually copied to the remote Web server’s installation location.

The remote Web server installation location is the location you specified when you created the node for this Web server.

The configuration is complete. To activate the configuration, stop and restart the Web server. If you encounter problems restarting your Web server, check the `http_plugin.log` file for information on what portion of the `plugin-cfg.xml` file contains an error. The log file states the line number on which the error

occurred along with other details that might help you diagnose why the Web server did not start. You can then use the administrative console to update the `plugin-cfg.xml` file.

If applications are infrequently installed or uninstalled, which is usually the situation in a production environment, or if you can tolerate the performance impact of generating and distributing the plug-in configuration file each time any of the previously listed actions occur, you should consider enabling this service.

If you are making a series of simultaneous changes, like installing numerous applications, you might want the configuration service disabled until after you make the last change. The Web server plug-in configuration service is enabled by default. To disable this service, in the administrative console click **Servers > Application Servers > *server\_name* > Administration Services > Web server plug-in configuration service** and then unselect the Enable automated Web server configuration processing option.

**Tip:** If your installation uses a firewall, make sure you configure the Web server plug-in to use a port that has been opened. (See your security administrator for information on how to obtain an open port.)

---

## Web server plug-in properties settings

Use this page to view or change the settings of a Web server plug-in configuration file. The plug-in configuration file, `plugin_cfg.xml`, provides properties for establishing communication between the Web server and the Application Server.

To view this administrative console page, click **Servers > Web Servers > *Web\_server\_name* Plug-in Properties**.

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 this Web server has accessed applications running on application servers and there is an `http_plugin.log` file.

### Plug-in log file name

Specifies the fully qualified path to the log file to which the plug-in will write error messages. The default file path is `plugin_install_root/logs/web_server_name/http_plugin.log`.

If the file does not exist then it will be created. If the file already exists, it will be opened in append mode and the previous plug-in log messages will remain.

This field corresponds to the RequestMetrics loggingEnabled element in the `plugin-cfg.xml` file.

<b>Data type</b>	String
<b>Default for Linux and UNIX platforms</b>	<code>plugin_install_root/logs/web_server_name/http_plugin.log</code>
<b>Default for Windows platforms</b>	<code>plugin_install_root/logs/web_server_name/http_plugin.log</code>

### Plug-in installation location

Specifies the fully qualified path to where the plug-in configuration file is installed.

<b>Data type</b>	String
<b>Default</b>	The default value is the installation root directory.



If you select a Web server plug-in during installation, the installer program configures the Web server to identify the location of the `plugin-cfg.xml` file, if possible. The Web server is considered installed on a local machine if it is on the same machine as the application server. It is considered installed on a remote machine if the Web server and the application server are on different machines.

- If the Web server is installed on a remote machine, the plug-in configuration file, by default, will be installed in the `plugin_install_root/config/web_server_name` directory.
- If the Web server is installed on a local standalone machine, the plug-in configuration file, by default will be installed in the `profile_install_root/config/cells/cell_name/nodes/web_server_name_node/servers/web_server_name` directory.
- If the Web server is installed on a local distributed machine, the plug-in configuration file, by default will be installed in the `profile_install_root/config/cells/cell_name/nodes/node_name/servers/web_server_name` directory.

The installer program adds a directive to the Web server configuration that specifies the location of the `plugin-cfg.xml` file.

For remote Web servers, you must copy the file from the local directory where the Application Server is installed to the remote machine. This is known as propagating the plug-in configuration file. If you are using an IBM HTTP Server (IHS) V6 for your Web server, WebSphere Application Server can automatically propagate the plug-in configuration file for you to remote machines provided there is a working HTTP transport mechanism to propagate the file.

## Plug-in configuration file name

Specifies the file name of the configuration file for the plug-in. The Application Server generates the `plugin-cfg.xml` file by default. The configuration file identifies applications, Application Servers, clusters, and HTTP ports for the Web server. The Web server uses the file to access deployed applications on various Application Servers.

<b>Data type</b>	String
<b>Default</b>	<code>plugin-cfg.xml</code>

If you select a plug-in during installation, the installer program configures the Web server to identify the location and name of the `plugin-cfg.xml` file, if possible.

You can change the name of the plug-in configuration file. However, if you do change the file name, you must also change the Web server configuration to point to the new plug-in configuration file.

## Automatically generate plug-in configuration file

To automatically generate a plug-in configuration file to a remote Web server:

- This field must be checked.
- The plug-in configuration service must be enabled

When the plug-in configuration service is enabled, a plug-in configuration file is automatically generated for a Web server whenever:

- The WebSphere Application Server administrator defines new Web server.
- An application is deployed to an Application Server.
- An application is uninstalled.
- A virtual host definition is updated and saved.

Clear the check box if you want to manually generate a plug-in configuration file for this Web server.



**Important:** When the plug-in configuration file is generated, it does not include `admin_host` on the list of virtual hosts. “Allowing Web servers to access the administrative console” on page 216 describes how to add it to the list.

## Automatically propagate plug-in configuration file

To automatically propagate a copy of a changed plug-in configuration file to a Web server:

- This field must be checked.
- The plug-in configuration service must be enabled
- A WebSphere Application Server node agent must be on the node that hosts the Web server associated with the changed plug-in configuration file.

**Note:** The plug-in configuration file can only be automatically propagated to a remote Web server if that Web server is an IHS V6.0 Web server and its administration server is running.

## Ignore DNS failures during Web server startup

Specifies whether the plug-in ignores DNS failures within a configuration when starting.

This field corresponds to the `IgnoreDNSFailures` element in the `plugin-cfg.xml` file.

When set to **true**, the plug-in ignores DNS failures within a configuration and starts successfully if at least one server in each `ServerCluster` is able to resolve the host name. Any server for which the host name can not be resolved is marked **unavailable** for the life of the configuration. No attempts to resolve the host name are made later on during the routing of requests. If a DNS failure occurs, a log message is written to the plug-in log file and the plug-in initialization continues rather than causing the Web server not to start. When **false** is specified, DNS failures cause the Web server not to start.

<b>Data type</b>	String
<b>Default</b>	false

## Refresh configuration interval

Specifies the time interval, in seconds, at which the plug-in should check the configuration file to see if updates or changes have occurred. The plug-in checks the file for any modifications that have occurred since the last time the plug-in configuration was loaded.

In a development environment in which changes are frequent, a lower setting than the default setting of 60 seconds is preferable. In production, a higher value than the default is preferable because updates to the configuration will not occur so often. If the plug-in reload fails for some reason, a message is written to the plug-in log file and the previous configuration is used until the plug-in configuration file successfully reloads. If you are not seeing the changes you made to your plug-in configuration, check the plug-in log file for indications of the problem.

<b>Data type</b>	Integer
<b>Default</b>	60 seconds.

## Plug-in logging

Specifies the location and name of the `http_plugin.log` file. Also specifies the scope of messages in the log.

This field corresponds to the `RequestMetrics traceLevel` element in the `plugin-cfg.xml` file.

The log describes the location and level of log messages that are written by the plug-in. If a log is not specified within the configuration file, then, in some cases, log messages are written to the Web server error log.

On a distributed platform, if the log file does not exist then it will be created. If the log file already exists, it will be opened in append mode and the previous plug-in log messages will remain.

**Log file name** - The fully qualified path to the log file to which the plug-in will write error messages.

<b>Data type</b>	String
<b>Default</b>	<code>plugin_install_root/logs/web_server_name/http_plugin.log</code>

Specify the file path of the `http_plugin.log` file.

**Log level**- The level of detail of the log messages that the plug-in should write to the log. You can specify one of the following values for this attribute:

- Trace. All of the steps in the request process are logged in detail.
- Stats. The server selected for each request and other load balancing information relating to request handling is logged.
- Warn. All warning and error messages resulting from abnormal request processing are logged.
- Error. Only error messages resulting from abnormal request processing are logged.

If a Log level is not specified, the default value **Error** is used.

Be careful when setting the level to **Trace**. A lot of messages are logged at this level which can cause the disk space/file system to fill up very quickly. A **Trace** setting should never be used in a normally functioning environment as it adversely affects performance.

<b>Data type</b>	String
<b>Default</b>	Error

## Web server plug-in request and response optimization properties settings

Use this page to view or change the request and response optimization properties for a Web server plug-in.

To view this administrative console page, click **Servers > Web Servers > web\_server\_name Plug-in Properties > Request and Response**.

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 this Web server has accessed applications running on application servers and there is an `http_plugin.log` file.

### Maximum chunk size used when reading the response body

Specifies the maximum chunk size the plug-in can use when reading the response body.

This field corresponds to the `ResponseChunkSize` element in the `plugin-cfg.xml` file.

The plug-in reads the response body in 64K chunks until all of the response data is read. This approach causes a performance problem for requests whose response body contains large amounts of data.

If the content length of the response body is unknown, the values specified for this property is used as the size of the buffer that is allocated. The response body is then read in this size chunks, until the entire body

is read. If the content length is known, then a buffer size of either the content length or the specified size (whichever is less) is used to read the response body.

<b>Data type</b>	Integer
<b>Default</b>	64 kilobytes
	Specify the size in kilobytes (1024 byte blocks).

### **Enable Nagle algorithm for connections to the Application Server**

When checked, the Nagle algorithm is enabled for connections between the plug-in and the Application Server.

This field corresponds to the ASDisableNagle element in the plugin-cfg.xml file.

The Nagle algorithm is named after engineer John Nagle, who invented this standard part of the transmission control protocol/internet protocol (TCP/IP). The algorithm reduces network overhead by adding a transmission delay (usually 20 milliseconds) to a small packet, which lets other small packets arrive and be included in the transmission. Because communications has an associated cost that is not as dependent on packet size as it is on frequency of transmission, this algorithm potentially reduces overhead with a more efficient number of transmissions.

Clear the check box to disable the Nagle algorithm.

### **Enable Nagle Algorithm for the IIS Web Server**

When checked, the Nagle algorithm is used for connections from the Microsoft Internet Informations Services (IIS) Web Server to the Application Server.

This field corresponds to the IHSDisableNagle element in the plugin-cfg.xml file. It only appears if you are using the Microsoft Internet Informations Services (IIS) Web server. Clear the check box to disable the Nagle algorithm for this connection.

### **Chunk response to the client**

When checked, responses to the client are broken into chunks if a Transfer-Encoding : Chunked response header is present in the response.

This field corresponds to the ChunkedResponse element in the plugin-cfg.xml file. It only appears if you are using a Microsoft Internet Informations Services (IIS) Web Server, a Java System Web server, or a Domino Web server. The IHS Web server automatically handles breaking the response into chunks to send to the client.

Clear the check box to if you do not want responses broken into chunks.

### **Accept content for all requests**

This field corresponds to the AcceptAllContent element in the plugin-cfg.xml file.

When selected, users can include content in POST, PUT, GET, and HEAD requests when a Content-Length or Transfer-encoding header is contained in the request header.

### **Virtual host matching**

When selected, virtual host mapping is performed by physically using the port number for which the request was received.

This field corresponds to the VHostMatchingCompat element in the plugin-cfg.xml file.

If this field is not selected, matching is done logically using the port number contained in the host header.

Use the radio buttons to make your physical or logical port selection.

### **Application server port preference**

Specifies which port number the Application Server should use to build URI's for a sendRedirect.

This field corresponds to the AppServerPortPreference element in the plugin-cfg.xml file.

Specify:

- webserverPort if the port number from the host header of the HTTP request coming in is to be used.
- hostHeader if the port number on which the Web server received the request is to be used.

The default is webserverPort.

### **Priority used by the IIS Web server when loading the plug-in configuration file**

Specifies the priority in which the Microsoft Internet Informations Services (IIS) Web server loads the WebSphere Web server plug-in.

This field corresponds to the IISPluginPriority element in the plugin-cfg.xml file. It only appears if you are using the IIS Web server. Because the IIS Web server uses this value during startup, the Web server must be restarted before a change to this field takes effect.

Select one of the following priorities:

- High
- Medium
- Low

The default value of **High** ensures that all requests are handled by the Web server plug-in before they are handled by any other filter/extensions. If problems occur while using a priority of **Medium** or **Low**, you will have to rearrange the order or change the priority of the interfering filter/extension.

### **Web server plug-in caching properties settings**

Use this page to view or change the caching properties for a Web server plug-in.

To view this administrative console page, click **Servers > Web Servers > Web\_server\_name Plug-in Properties > Caching Properties**.

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 this Web server has accessed applications running on application servers and there is an http\_plugin.log file.

### **Enable Edge Side Include (ESI) processing to cache the responses**

Specifies whether to enable Edge Side Include processing to cache the responses.

This field corresponds to the esiEnable element in the plugin-cfg.xml file.

When selected, Edge Side Include (ESI) processing is used to cache responses. If ESI processing is disabled for the plug-in, the other ESI plug-in properties are ignored. Clear the checkbox to disable Edge Side Include processing.

### **Enable invalidation monitor to receive notifications**

When checked, the ESI processor receives invalidations from the Application Server.

This field corresponds to the ESIInvalidationMonitor element in the plugin-cfg.xml file. It is ignored if Edge Side Include (ESI) processing is not enabled for the plug-in.

## Maximum cache size

Specifies, in 1K byte units, the maximum size of the cache. The default maximum size of the cache is 1024K bytes (1 megabyte). If the cache is full, the first entry to be evicted from the cache is the entry that is closest its expiration time.

This field corresponds to the esiMaxCacheSize element in the plugin-cfg.xml file.

<b>Data type</b>	Integer
<b>Default</b>	1024 kilobytes
	Specify the size in kilobytes (1024 byte blocks).

## Web server plug-in request routing properties settings

Use this page to view or change the request routing properties for a Web server plug-in.

To view this administrative console page, click **Servers > Web Servers > *Web\_server\_name* Plug-in Properties > Plug-in *server\_cluster\_name* Properties**.

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 this Web server has accessed applications running on application servers and there is an http\_plugin.log file.

## Load balancing option

Specifies the load balancing option that the plug-in uses in sending requests to the various application servers associated with that Web server.

This field corresponds to the LoadBalanceWeight element in the plugin-cfg.xml file.

Select the appropriate load balancing option:

- Round robin
- Random

The Round Robin implementation has a random starting point. The first application server is picked randomly. Round Robin is then used to pick application servers from that point forward. This implementation ensures that in multiple process based Web servers, all of the processes don't start up by sending the first request to the same Application Server.

The default load balancing type is Round Robin.

## Retry interval

Specifies the length of time, in seconds, that should elapse from the time an application server is marked down to the time that the plug-in retries a connection.

This field corresponds to the ServerWaitforContinue element in the plugin-cfg.xml file.

<b>Data type</b>	Integer
<b>Default</b>	60 seconds

## Maximum size of request content

Select whether there is a limit on the size of request content. If limited, this field also specifies the maximum number of bytes of request content allowed in order for the plug-in to attempt to send the request to an application server.

This field corresponds to the PostSizeLimit element in the plugin-cfg.xml file.

When a limit is set, the plug-in fails any request that is received that is greater than the specified limit.

Select whether to limit the size of request content:

- No limit
- Set limit

If **Set limit** is selected, specify a limit size.

<b>Data type</b>	Integer
<b>Default</b>	-1, which indicates there is no limit for the post size.  Specify the size in kilobytes (1024 byte blocks).

## Remove special headers

When checked, the plug-in will remove any headers from incoming requests before adding the headers the plug-in is supposed to add before forwarding the request to an application server.

This field corresponds to the RemoveSpecialHeaders element in the plugin-cfg.xml file.

The plug-in adds special headers to the request before it is forwarded to the application server. These headers store information about the request that will need to be used by the application. Not removing the headers from incoming requests introduces a potential security exposure.

Clear the check box to retain special headers.

## Clone separator change

When checked, the plug-in expects the plus character (+) as the clone separator.

This field corresponds to the ServerCloneID element in the plugin-cfg.xml file.

Some pervasive devices cannot handle the colon character (:) used to separate clone IDs in conjunction with session affinity. If this field is checked, you must also change the configurations of the associated application servers such that the application servers separates clone IDs with the plus character as well.

Clear the checkbox to use the colon character to separate clone IDs.

---

## Web server plug-in custom properties

If you are using a Web server plug-in, you can add the following custom property to the configuration settings for that plug-in.

To add a custom property:

1. In the administrative console, click **Servers > Web Servers > *Web\_server\_name* > Plug-in properties > 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 is a list of Web server plug-in custom properties that are provided with the Application Server. These properties are not shown on the properties settings pages for the plug-in.

### **StashfileLocation**

Use this element to set a value for the stashfile initialization parameter.

**Data type** String

### **KeyringLocation**

Use this element to set a value for the keyring initialization parameter.

**Data type** String

---

## **Web server plug-in configuration service properties settings**

Use this page to view or change the configuration settings for the Web server plug-in configuration service.

To view this administrative console page, click **Application Servers** > *server\_name* > **Administration Services** > **Web server plug-in configuration service**.

On the **Configuration** tab, you can edit fields. On the **Runtime** tab, you can look at read-only information.

### **Enable automated Web server configuration processing**

When selected, the Web server plug-in configuration service automatically generates the plug-in configuration file whenever the Web server environment changes. For example, the plug-in configuration file is regenerated whenever one of the following activities occurs:

- A new application is deployed on an associated application server.
- The Web server definition is saved.
- An application is removed from an associated application server.
- A new virtual host is defined.

---

## **Application Server property settings for a Web server plug-in**

Use this page to view or change application server settings for a Web server plug-in.

To view this administrative console page, click **Application Servers** > *server\_name* > **Web Server Plug-in**.

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 this Web server has accessed applications running on application servers and there is an `http_plugin.log` file.

### **Server role**

Specifies the role this application server is assigned.

Select **Primary** to add this application server to the list of primary application servers. The plug-in initially attempts to route requests to the application servers on this list.

Select **Backup** to add this application server to the list of backup application servers. The plug-in does not load balance across the backup application servers. A backup server is only used if a primary server is not

available. When the plug-in determines that a backup application server is required, it goes through the list of backup servers, in order, until no servers are left in the list or until a request is successfully sent and a response received from one of the servers on this list.

## Connect timeout

Specifies whether or not there is a limited amount of time the Application Server will maintain a connection with the Web server.

You can select either **No timeout** or **Set timeout**. If you select **Set timeout** you, must specify, in seconds, the length of time a connection with the Web server is to be maintained.

This property enables the plug-in to perform non-blocking connections with the application server. Non-blocking connections are beneficial when the plug-in is unable to contact the destination to determine whether or not the port is available. If no value is specified for this property, the plug-in performs a blocking connect in which the plug-in sits until an operating system times out (which could be as long as 2 minutes depending on the platform) and allows the plug-in to mark the server unavailable.

A value of 0 causes the plug-in to perform a blocking connect. A value greater than 0 specifies the number of seconds you want the plug-in to wait for a successful connection. If a connection does not occur after that time interval, the plug-in marks the server unavailable and fails over to another application server defined for the requested application.

**Data type** Integer

## Maximum number of connections that can be handled by the Application Server

Specifies the maximum number of pending connections to an Application Server that can be flowing through a Web server process at any point in time.

This field corresponds to the ServerMaxConnections element in the plugin-cfg.xml file.

You can select either **No limit** or **Set limit**. If you select **Set limit** you, must specify the maximum number of connections that can exist between the Web server and the Application Server at any given point in time.

For example, assuming that:

- The application server is fronted by 5 nodes that are running an IHS Web server.
- Each node starts 2 processes.
- This property is set to 50.

In this example, the application server could potentially get up to 500 connections. (You take the number of nodes, 5, multiply it by the number of processes, 2, and then multiply that number by the number specified for this property, 50, for a total of 500 connections.)

If this attribute is set to either zero or -1, there is no limit to the number of pending connections to the Application Servers.

**Data type** Integer  
**Default** -1



## Use extended handshake to check whether Application Server is running

When selected, the Web server plug-in will use an extended handshake to check whether or not the Application Server is running.

This field corresponds to the ServerExtendedHandshake element in the plugin-cfg.xml file.

Select this property if a proxy firewall is between the plug-in and the application server.

The plug-in marks a server as down when the connect() fails. However, when a proxy firewall is in between the plug-in and the application server, the connect() will succeed, even though the back end application server is down. This causes the plug-in to not failover correctly to other application servers.

If the plug-in performs some handshaking with the application server to ensure that it is started before it sends a request it can failover to another application server if it detects that the application server with which it is attempting to perform a handshake is down.

## Send the header "100 Continue" before sending the request content

This field corresponds to the WaitForContinue element in the plugin-cfg.xml file.

When selected, the Web server plug-in will send the header "100 Continue" to the Application Server before it sends the request content.

---

## Web server plug-in configuration properties

The following table indicates which panel in the administrative console you need to use to manually configure a Web server plug-in property.

Table 21. Web server plug-in configuration properties

Administrative console panel	Field name	Configuration property name
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties</b>	Refresh configuration interval	RefreshInterval
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties</b>	Plug-in log file name	Log->name
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties</b>	Plug-in logging	Log->LogLevel
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties</b>	Ignore DNS failures during Web server startup	IgnoreDNSFailures
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Custom properties &gt; New</b>	KeyringLocation	Keyring

Table 21. Web server plug-in configuration properties (continued)

In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Custom properties &gt; New</b>	StashfileLocation	Stashfile
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request routing</b>	Load balancing option	LoadBalance
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request Routing</b>	Clone separator change	CloneSeparatorChange
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request Routing</b>	Retry interval	RetryInterval
In the administrative console, click <b>Servers &gt; Web server_name &gt; Plug-in properties &gt; Request routing</b>	Maximum size of request content	PostSizeLimit
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request routing</b>	Remove special headers	RemoveSpecialHeaders
In the administrative console, click <b>Application Servers &gt; server_name &gt; Web server plug-in properties</b>	Server role	PrimaryServers and BackupServers list
In the administrative console, click <b>Application Servers &gt; server_name &gt; Web server plug-in properties</b>	Connect timeout	Server ConnectTimeout
In the administrative console, click <b>Application Servers &gt; server_name &gt; Web server plug-in properties</b>	Use extended handshake to check whether Application Server is running	Server Extended Handshake
In the administrative console, click <b>Application Servers &gt; server_name &gt; Web server plug-in properties</b>	Send the header "100 Continue" before sending the request content	WaitForContinue
In the administrative console, click <b>Application Servers &gt; server_name &gt; Web server plug-in properties</b>	Maximum number of connections that can be handled by the Application Server	Server MaxConnections
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request and Response</b>	Application server port preference	AppServerPortPreference
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request and Response</b>	Enable Nagle algorithm for connections to the Application Server	ASDisableNagle

Table 21. Web server plug-in configuration properties (continued)

In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request and Response</b>	Enable Nagle Algorithm for the IIS Web Server	IISDisableNagle
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request and Response</b>	Virtual host matching	VHostMatchingCompat
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request and Response</b>	Maximum chunk size used when reading the response body	ResponseChunkSize
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request and Response</b>	Accept content for all requests	AcceptAllContent
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request and Response</b>	Chunk response to the client	ChunkedResponse
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Request and Response</b>	Priority used by the IIS Web server when loading the plug-in configuration file	IISPluginPriority
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Caching</b>	Enable Edge Side Include (ESI) processing to cache the responses	ESIEnable
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Caching</b>	Maximum cache size	ESIMaxCacheSize
In the administrative console, click <b>Servers &gt; Web Servers &gt; Web_server_name &gt; Plug-in properties &gt; Caching</b>	Enable invalidation monitor to receive notifications	ESIInvalidationMonitor

## Web server plug-in connections

The WebSphere Application Server Web server plug-ins are used to establish and maintain persistent HTTP and HTTPS connections to Application Servers .

When the plug-in is ready to send a request to the application server, it first checks its connection pool for existing connections. If an existing connection is available the plug-in checks its connection status. If the status is still good, the plug-in uses that connection to send the request. If a connection does not exist, the plug-in creates one. If a connection exists but has been closed by the application server, the plug-in closes that connection and opens a new one.

After a connection is established between a plug-in and an application server, it will not be closed unless the application server closes it for one of the following reasons:

- If the **Use Keep-Alive** property is selected and the time limit specified on the **Read timeout** or **Write timeout** property for the HTTP inbound channel has expired.
- The maximum number of persistent requests which can be processed on an HTTP inbound channel has been exceeded. (This number is set using the HTTP inbound channel's **Maximum persistent requests** property.)
- The Application Server is shutting down.

Even if the application server closes a connection, the plug-in will not know that it has been closed until it tries to use it again. The connection will be closed if one of the following events occur:

- The plug-in receives a new HTTP request and tries to reuse the existing connection.
- The number of **httpd** processes drop because the Web server is not receiving any new HTTP requests. (For the IHS Web server, the number of **httpd** processes that are kept alive depends on the value specified on the Web server's **MinSpareServers** directive.)
- The Web server is stopped and all **httpd** processes are terminated, and their corresponding sockets are closed.

**Note:** Sometimes, if a heavy request load is stopped or decreased abruptly on a particular application server, a lot of the plug-in's connections to that application server will be in **CLOSE\_WAIT** state. Because these connections will be closed the first time the plug-in tries to reuse them, having a large number of connections in **CLOSE\_WAIT** state should not affect performance

---

## Web server plug-in remote user information processing

You can configure your Web server with a third-party authentication module and then configure the Web server plug-in to route requests to the Application Server. If an application calls the `getRemoteUser()` method, it relies on a private HTTP header that contains the remote user information and is parsed by the plug-in. The plug-in sets the private HTTP header value whenever a Web server authentication module populates the remote user in the Web server data structure. If the private HTTP header value is not set, the application's call to `getRemoteUser()` returns a null value.

- In the case of an Apache and IBM HTTP Server (IHS) Web server, the plug-in builds the private header from the information contained in the associated request record.
- In the case of a Sun One Web server, the plug-in builds the private header from the information contained in the **auth\_user** property associated with the request. The private header is usually set to the name of the local HTTP user of the Web browser, if HTTP access authorization is activated for the URL.
- In the case of a Domino Web server, the plug-in builds the private header from the information contained in the **REMOTE\_USER** environment variable. The plug-in sets this variable to **anonymous** for users who have not logged in and to the *username* for users who are logged into the application.
- In the case of an Internet Information Services (IIS) Web server, the plug-in builds the private header from the information contained in the **REMOTE\_USER** environment variable. The plug-in sets this variable to the name of the user as it is derived from the authorization header sent by the client.

If the private header is not being set in the Sun One, IIS, or Domino Web server plug-in, make sure the request record includes information about the user requesting the data.

**Note:** If an application's call to `getRemoteUser()` returns a null value, or if the correct remote user information is not being added to the Web server plug-in's data structure, make sure the remote user parameter within the `WebAgent` is still set to **YES**. (Sometimes this parameter gets set to **NO** when service is applied.)

---

## Web server plug-ins

Web server plug-ins enable the Web server to communicate requests for dynamic content, such as servlets, to the application server. A Web server plug-in is associated with each Web server definition. The configuration file (plugin-cfg.xml) that is generated for each plug-in is based on the applications that are routed through the associated Web server.

A Web server plug-in is used to forward HTTP requests from a supported Web server to an application server. Using a Web server plug-in to provide communication between a Web server and an application server has the following advantages:

- XML-based configuration file
- Standard protocol recognized by firewall products
- Security using HTTPS, replacing proprietary Open Servlet Engine (OSE) over Secure Sockets Layer (SSL)

Each of the supported distributed platform Web server plug-ins run on a number of operating systems. See Supported Hardware and Software for the product for the most current information about supported Web servers.

---

## Checking your IBM HTTP Server version

At times, you might need to determine the version of your IBM HTTP Server installation.

1. Change directory to the installation root of the Web server. For example, this is /opt/IBMIHS on a Solaris machine.
2. Find the subdirectory that contains apache.exe (on a Windows platforms) or apachectl (on a UNIX-based platforms, such as z/OS, Solaris, Linux, and HP-UX)
3. On a Windows platform, issue:  
apache.exe -V
4. On a UNIX-based platforms issue:  
./apachectl -V 4

The version is shown in the "Server version:" field and will look something like the following:

```
Server version: IBM_HTTP_Server/2.0.47 Apache/2.0.47
Server built: July 2 2004 20:38:36
Server's Module Magic Number: 20020903:4.
```

---

## Web server tuning parameters

WebSphere Application Server provides plug-ins for several Web server brands and versions. Each Web server operating system combination has specific tuning parameters that affect the application performance.

- **IBM HTTP Server**

The IBM HTTP Server V6.0 is a multi-process, multi-threaded server.

- **Access logs**

- **Description:** Collects all incoming HTTP requests. Logging degrades performance because IO operation overhead causes logs to grow significantly in a short time.
- **How to view or set:**
  1. Open the IBM HTTP Server httpd.conf file, located in the directory *IBM\_HTTP\_Server\_root\_directory/conf*.
  2. Search for a line with the text **CustomLog**.
  3. Comment out this line by placing # in front of the line.
  4. Save and close the httpd.conf file.
  5. Stop and restart the IBM HTTP Server.
- **Default value:** Logging of every incoming HTTP request is enabled.

- **Recommended value:** Disable the access logs.
- **MaxClients**
  - **Description:** The MaxClients directive controls the maximum number of simultaneous connections or users that the web server can service at any one time. If, at peak usage, your web server needs to support 200 active users at once, you should set MaxClients to 220 (200 plus an extra 10% for load growth). Setting MaxClients too low could cause some users to believe the web server is not responding. You should have sufficient RAM in your web server machines to support each connected client. For IBM HTTP Server V6.0 on UNIX, you should allocate around 1.5MB MaxClients of RAM for use by the IBM HTTP Server. For IBM HTTP Server V6.0 on Windows, you should allocate around 300KB MaxClients of RAM for use by the IBM HTTP Server. Some third party modules can significantly increase the amount of RAM used per connected client.
  - **How to view or set:** Edit the MaxClients directive in the IBM HTTP Server httpd.conf file, located in the directory *IBM\_HTTP\_Server\_root\_directory/conf*.
  - **Default value:** 150
  - **Recommended value:** The maximum number of users normally simultaneously connected to your web server, plus an additional 10% for buffer. Note: The KeepAliveTimeout setting can affect how long a user is connected to the webserver.
- **MinSpareServers, MaxSpareServers, and StartServers**
  - **Description:** Pre-allocates and maintains the specified number of processes so that few processes are created and destroyed as the load approaches the specified number of processes. Specifying similar values reduces the CPU usage for creating and destroying HTTPD processes. Adjust this parameter if the time waiting for IBM HTTP Server to start more servers, so that it can handle HTTP requests, is not acceptable.
  - **How to view or set:** Edit the MinSpareServers, MaxSpareServers and StartServers directives in the httpd.conf file located in the *IBM\_HTTP\_Server\_root\_directory/conf* directory.
  - **Default value:** MinSpareServers 5, MaxSpareServers 10, StartServers 5
  - **Recommended value:** For optimum performance, specify the same value for the MinSpareServers and the StartServers parameters. If MaxSpareServers is set to less than MinSpareServers, IBM HTTP Server resets MaxSpareServer=MinSpareServer+1. Setting the StartServers too high can cause swapping if memory is not sufficient, degrading performance.
- **ListenBackLog**
  - **Description:** Sets the length of a pending connections queue. When several clients request connections to the IBM HTTP Server, and all threads used, a queue exists to hold additional client requests. However, if you use the default Fast Response Cache Accelerator (FRCA) feature of IBM HTTP Server V6.0 on Windows, the ListenBackLog directive is not used since FRCA has its own internal queue.
  - **How to view or set:** For non-FRCA: Edit the IBM HTTP Server httpd.conf file. Then, add or view the ListenBackLog directive.
  - **Default value:** For HTTP Server V6.0: 1024 with FRCA enabled, 511 with FRCA disabled
  - **Recommended value:** Use the defaults.
- **IBM HTTP Server - Linux**
  - **MaxRequestsPerChild**
    - **Description:** Sets the limit on the number of requests that an individual child server process handles. After the number of requests reaches the value set for the MaxRequestsPerChild parameter, the child process dies. Adjust this parameter if destroying and creating child processes is degrading your Web server performance.
    - **How to view or set:**
      1. Edit the IBM HTTP server httpd.conf file located in the *IBM\_HTTP\_Server\_root\_directory/conf* directory.
      2. Change the value of the parameter.
      3. Save the changes and restart the IBM HTTP server.
    - **Default value:** 500
    - **Recommended value:** Should normally be set to 0. Non-zero settings can be useful if child memory usage is observed to steadily increase over time. Memory leaks have occasionally been observed in third party modules and various OS runtime libraries used by the IBM HTTP Server.
- **IBM HTTP Server - Windows 2000 and Windows 2003**



– **ThreadsPerChild**

- **Description:** Sets the number of concurrent threads running at any one time within the IBM HTTP Server.
- **How to view or set:** Edit the IBM HTTP Server file `httpd.conf` file located in the directory `IBM_HTTP_Server_root_directory/conf`. Change the value of the parameter. Save the changes and restart the IBM HTTP Server.

There are two ways to find how many threads are used under load:

1. Use the Windows 2000 and Windows 2003 Performance Monitor under the desktop Start menu:
  - a. Right-click the status bar on your desktop. Click **Task Manager**.
  - b. Select the **Processes** tab.
  - c. Click **View > Select Columns**.
  - d. Select **Thread Count**.
  - e. Click **OK**.

The **Processes** tab shows the number threads for each process under the column name **Threads**, including Apache.

2. Use the IBM HTTP Server `server-status` (this choice works on all platforms, not just Windows):
  - a. Edit the IBM HTTP Server `httpd.conf` file as follows: Remove the comment character `#` from the following lines: `#LoadModule status_module, modules/ApacheModuleStatus.dll, #<Location/server-status>, #SetHandler server-status, and #</Location>`.
  - b. Save the changes and restart the IBM HTTP Server.
  - c. In a Web browser, go to the URL: `http://yourhost/server-status`. Alternatively,
  - d. Click **Reload** to update status.
  - e. (Optional) If the browser supports refresh, go to `http://your_host/server-status?refresh=5` to refresh every five seconds. You will see five requests currently processing 45 idle servers.
- **Default value:** 250 for IBM HTTP Server V6.0.
- **Recommended value:** Set this value to prevent bottlenecks, allowing just enough traffic through to the application server.

– **Web server configuration reload interval**

- **Description:** Tracks a variety of configuration information about WebSphere Application Server resources. The Web server needs to understand some of this information, such as Uniform Resource Identifiers (URIs) pointing to WebSphere Application Server resources. This configuration data is pushed to the Web server through the WebSphere Application Server plug-in at intervals specified by this parameter. Periodic updates add new servlet definitions without having to restart any of the WebSphere Application Server servers. However, the dynamic regeneration of this configuration information is costly in terms of performance. Adjust this parameter in a stable production environment.
- **How to view or set:** Use the Refresh configuration interval Web server plug-in property to change the current setting for this parameter. In the administrative console, click **Servers > Web Servers > Web\_server\_name > Plug-in properties**.
- **Default value:** The default reload interval is 60 seconds.
- **Recommended value:** Increase the reload interval to a value that represents an acceptable wait time between the servlet update and the Web server update.

For more information about the `plugin-cfg.xml` file see the topic “Web server plug-ins” on page 309.

• **Sun Java System Web server, Enterprise Edition (formerly Sun ONE) - Solaris operating environment**

The default configuration of the Sun ONE Web server, Enterprise Edition provides a single-process, multi-threaded server.

– **Active threads**

- **Description:** Specifies the current number of threads active in the server. After the server reaches the limit set with this parameter, the server stops servicing new connections until it finishes old

connections. If this setting is too low, the server can become throttled, resulting in degraded response times. To tell if the Web server is being throttled, consult its perfdump statistics. Look at the following data:

- **WaitingThreads count:** If WaitingThreads count is getting close to zero, or is zero, the server is not accepting new connections.
- **BusyThreads count:** If the WaitingThreads count is close to zero, or is zero, BusyThreads is probably very close to its limit.
- **ActiveThreads count:** If ActiveThreads count is close to its limit, the server is probably limiting itself.
- **How to view or set:** Use the Maximum number of simultaneous requests parameter in the Enterprise Server Manager interface to control the number of active threads within Sun ONE Web server, Enterprise Edition. This setting corresponds to the RqThrottle parameter in the magnus.conf file.
- **Default value:** 512
- **Recommended value:** Increase the thread count until the active threads parameters show optimum behavior.
- **Microsoft Internet Information Server (IIS) - Windows NT and Windows 2000**
  - **IIS permission properties**
    - **Description:** The Web server has several properties that dramatically affect the performance of the application server. The default settings are usually acceptable. However, because other products can change the default settings without user knowledge, make sure to check the IIS settings for the Home Directory permissions of the Web server. The permissions should be set to Script and not to Execute. If the permissions are set to Execute, no error messages are returned, but the performance of WebSphere Application Server is decreased.
    - **How to view or set:** To check or change these permissions, perform the following procedure in the Microsoft management console:
      1. Select the Web site (usually default Web site).
      2. Right-click and select the **Properties** option.
      3. Click the **Home Directory** tab. To set the permissions of the Home Directory:
        - a. In the **Application** settings, select the **Script** check box in the **Permissions** list and clear the **Execute** check box.
        - b. (Optional) Check the permissions of the sePlugin:
          - 1) Expand the Web server.
          - 2) Right-click the sePlugin and select **Properties**.
          - 3) Confirm that the **Execute** permissions are set to **Execute**.
    - **Default value:** Script
    - **Recommended value:** Script
  - **Number of expected hits per day**
    - **Description:** Controls the memory that IIS allocates for connections.
    - **How to view or set:** Using the performance window, set the parameter to More than 100000 in the Web site properties panel of the Microsoft management console.
    - **Default value:** Fewer than 100000
    - **Recommended value:** More than 100000
  - **ListenBackLog parameter**
    - **Description:** Alleviates failed connections under heavy load conditions, if you are using IIS on Windows NT and Windows 2000. Failure typically occurs when you are using more than 100 clients. ListenBackLog increases the number of requests that IIS keeps in its queue. Consider raising this value if you see intermittent Unable to locate server errors in the Netscape browser.
    - **How to view or set:**
      1. Use a command prompt to issue the **regedit** command to access the operating system registry.
      2. In the registry window, locate the parameter in the HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters\ListenBackLog directory.
      3. Right-click the parameter to adjust the setting according to the server load.



- **Default value:** 25 (decimal)
- **Recommended value:** You can set the ListenBackLog parameter can be set as high as 200, without negative impact on performance and with an improvement in load handling.

### Modifying the WebSphere plug-in to improve performance

You can improve the performance of IBM HTTP Server V6.0 (with the WebSphere Web server plug-in) by modifying the plug-in's RetryInterval configuration. The RetryInterval is the length of time to wait before trying to connect to a server that has been marked temporarily unavailable. Making this change can help the IBM HTTP Server V6.0 to scale higher than 400 users.

The plug-in marks a server temporarily unavailable if the connection to the server fails. Although a default value is 60 seconds, it is recommended that you lower this value in order to increase throughput under heavy load conditions. Lowering the RetryInterval is important for IBM HTTP Server V6.0 on UNIX operating systems that have a single thread per process, or for IBM HTTP Server 2.0 if it is configured to have fewer than 10 threads per process.

How can lowering the RetryInterval affect throughput? If the plug-in attempts to connect to a particular application server while the application server threads are busy handling other connections, which happens under heavy load conditions, the connection times out and the plug-in marks the server temporarily unavailable. If the same plug-in process has other connections open to the same server and a response is received on one of these connections, the server is marked again. However, when you use the IBM HTTP Server V6.0 on a UNIX operating system, there is no other connection since there is only one thread and one concurrent request per plug-in process. Therefore, the plug-in waits for the RetryInterval before attempting to connect to the server again.

Since the application server is not really down, but is busy, requests are typically completed in a small amount of time. The application server threads become available to accept more connections. A large RetryInterval causes application servers that are marked temporarily unavailable, resulting in more consistent application server CPU utilization and a higher sustained throughput.

**Note:** Although lowering the RetryInterval can improve performance, if all the application servers are running, a low value can have an adverse affect when one of the application servers is down. In this case, each IBM HTTP Server V6.0 process attempts to connect and fail more frequently, resulting in increased latency and decreased overall throughput.

---

## Gskit install images files

The Global Security Kit (GSKit) installation image files for the WebSphere Web server plug-ins are packaged on the CD with the Web server plug-in files. You can download the appropriate GSKIT file to the workstation on which your Web server is running. Use the following table to assist you in selecting the correct GSKIT installation image file.

Operating system	GSKit 7 Installation image file
Windows	No image name
AIX	gskta.rte
HP-UX	gsk7bas
Solaris Operating Environment	gsk7bas
Linux	gsk7bas_7.0.3.1.i386.rpm
Linux390	gsk7bas-7.0.3.1.s390.rpm
LinuxPPC	gsk7bas-7.0.3.1.ppc.rpm

---

## Plug-ins: Resources for learning

Use the following links to find relevant supplemental information about Web server plug-ins. 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 Redbooks that supplement the broad coverage of the release documentation with in-depth examinations of particular product areas.

View links to additional information about:

- Programming model and decisions
- Programming instructions and examples

### Programming model and decisions

- Best Practice: WebSphere Plug-in Configuration Regeneration

### Programming instructions and examples

- IBM HTTP Server documentation
- WebSphere Application Server education
- Listing of all IBM WebSphere Application Server Redbooks

---

## Web server plug-in tuning tips

### Balancing workloads:

During normal operation, the backlog of connections pending to an application server is bound to grow. Therefore, balancing workloads among application servers in a network fronted by a Web server plug-in helps improve request response time.

In a distributed environment, you can limit the number of connections that can be handled by an applications server. To do this:

1. Go to the **Servers > Application Servers > *server\_name***.
2. Under Additional Properties, click > **Web Server Plug-in properties** .
3. Select **Set limit** for the Minimum number of connections that can be handled by the Application Server field.
4. Specify in the Connections field the maximum number of connections you want to allow.
5. Then click **Apply** and **Save**.

When this maximum number of connections is reached, the plug-in, when establishing connections, automatically skips that application server, and tries the next available application server. If no application servers are available, an HTTP 503 response code will be returned to the client. This code indicates that the server is currently unable to handle the request because it is experiencing a temporary overloading or because maintenance is being performed.

The capacity of the application servers in the network determines the value you specify for the maximum number of connections. The ideal scenario is for all of the application servers in the network to be optimally utilized. For example, if you have the following environment:

- There are 10 application servers in a cluster.
- All of these application servers host the same applications (that is, Application\_1 and Application\_2).
- This cluster of application servers is fronted by five IBM HTTP Servers.
- The IBM HTTP Servers get requests through a load balancer.

- Application\_1 takes approximately 60 seconds to respond to a request
- Application\_2 takes approximately 1 second to respond to a request.

Depending on the request arrival pattern, all requests to Application\_1 might be forwarded to two of the application servers, say Appsvr\_1 and Appsvr\_2. If the arrival rate is faster than the processing rate, the number of pending requests to Appsvr\_1 and Appsvr\_2 can grow.

Eventually, Appsvr\_1 and Appsvr\_2 are busy and are not able to respond to future requests. It usually takes a long time to recover from this overloaded situation.

If you want to maintain 2500 connections, and optimally utilize the Application Servers in this example, set the number of maximum connections allowed to 50. (This value is arrived at by dividing the number of connections by the result of multiplying the number of Application Servers by the number of Web servers; in this example,  $2500/(10 \times 5) = 50$ .)

Limiting the number of connections that can be established with an application server works best for Web servers that follow the threading model instead of the process model, and only one process is started.

The IBM HTTP Server V6.0.x follows the threading model. To prevent the IBM HTTP Server from starting more than one process, change the following properties in the Web server configuration file (`httpd.conf`) to the indicated values:

```
ServerLimit      1
ThreadLimit     4000
StartServers    1
MaxClients      1024
MinSpareThreads 1
MaxSpareThreads 1024
ThreadsPerChild 1024
MaxRequestsPerChild 0
```

### Improving performance in a high stress environment:

If you use the default settings for a Microsoft Windows operating system, you might encounter Web server plug-in performance problems if you are running in a high stress environment. To avoid these problems, consider tuning the the TCP/IP setting for this operating system. Two of the keys setting to tune are `TcpTimedWaitDelay` and `MaxUserPort`.

To tune the `TcpTimedWaitDelay` setting, change the value of the `tcp_time_wait_interval` parameter from the default value of 240 seconds, to 30 seconds:

1. Locate in the Windows Registry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\tcpip\Parameters\TcpTimedWaitDelay
```

If this entry does not exist in your Windows Registry, create it by editing this entry as a new `DWORD` item.

2. Specify, in seconds, a value between 30 and 300 inclusive for this entry. (It is recommended that you specify a value of 30. )

To tune the `MaxUserPort` setting:

1. Locate in the Windows Registry:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\tcpip\Parameters\MaxUserPort
```

If this entry does not exist in your Windows Registry, create it by editing this entry as a new `DWORD` item.

2. Set the maximum number of ports to a value between 5000 and 65534 ports, inclusive. (It is recommended that you specify a value of 65534.)

See the Microsoft Web site for more information about these settings.

---

## **Tuning Web servers**

“Web server tuning parameters” on page 309 lists tuning parameters specific to Web servers. The listed parameters may not apply to all of the supported Web servers. Check your Web server documentation before using any of these parameters.

---

## Appendix A. Uninstalling the product

This task describes how to uninstall WebSphere Application Server Network Deployment.

The “uninstall command” on page 318 calls the uninstaller program that is created during installation. The uninstaller program is customized for each product installation, with specific disk locations and routines for removing installed features.

The uninstaller program removes registry entries, uninstalls the product, and removes all related features. The uninstaller program does not remove log files in the installation root directory.

The uninstaller program removes all profiles, including all of the configuration data and applications in each profile. Before you start the uninstall procedure, back up the `config` folder, the `installableApps` folder, and the `installedApps` folder of each profile, if necessary. See the *Administration: Operations* PDF for a description of managing configuration files. Back up all applications that are not stored in another location.

This procedure uninstalls the WebSphere Application Server product.

See the following topics to uninstall other components on the product disc:

- “Uninstalling the Web server plug-ins for WebSphere Application Server” on page 200
- Uninstalling IBM HTTP Server
- “Uninstalling Application Client for WebSphere Application Server” on page 222

**Estimating the time required to uninstall** The time required to uninstall is dependent on the processing speed of your machine. As a rough guideline, uninstalling the core product files and one application server profile takes approximately 10 minutes when using the **uninstall** command.

1. Log on as root on a Linux or UNIX system, or as a user who belongs to the administrator group on a Windows system.
2. Run the uninstaller program for the Web server plug-ins for WebSphere Application Server.  
If a Web server is configured to run with the Application Server, uninstall the plug-ins to remove the configuration from the Web server. See “Uninstalling the Web server plug-ins for WebSphere Application Server” on page 200.
3. Stop each running Application Server with the **stopServer** command.  
If security is disabled, the uninstaller program can stop all WebSphere Application Server processes automatically. If servers are running and security is enabled, the uninstaller program cannot shut down the servers and the uninstall procedure fails. Manually stop all servers before uninstalling.

Stop all server processes in all profiles on the machine. For example, issue the following command from the `/opt/IBM/WebSphere/AppServer/profiles/app_server_profile/bin` directory on a Linux machine to stop the `server1` process in the profile:

```
./stopServer.sh server1
```

If servers are running and security is enabled, use the following commands:

```
./stopServer.sh server1 -user user_ID -password password
```

4. Stop the `nodeagent` process with the **stopNode** command.

Stop the `nodeagent` process that might be running on the machine. For example, issue the following command from the `/opt/IBM/WebSphere/AppServer/profiles/app_server_profile/bin` directory of a federated node on a Linux machine to stop the `nodeagent` process:

```
./stopNode.sh
```

If servers are running and security is enabled, use the following command:

```
./stopNode.sh -user user_ID -password password
```

5. Stop the deployment manager dmgr process with the **stopManager** command.  
Stop all dmgr processes that are running on the machine. For example, issue this command on a Linux machine from the `/opt/IBM/WebSphere/AppServer/profiles/dmgr_profile/bin` directory:  

```
./stopManager.sh -user user_ID -password password
```
6. **Optional:** Back up configuration files and log files to refer to them later, if necessary.  
The uninstaller program does not remove log files in the installation root directory. The uninstaller program removes all profiles and all of the data in all profiles.  
Back up the `config` folder and the `logs` folder of each profile to refer to it later, if necessary. You cannot reuse profiles so there is no need to back up an entire profile.
7. Uninstall the product.  
“Uninstalling Network Deployment” on page 322  
After running the **uninstall** command, the directory structure has only a few remaining directories. The `logs` directory is one of the few directories with files.
8. Review the `install_root/logs/uninstlog.txt` file.  
The `install_root/logs/uninstlog.txt` file records file system or other unusual errors. Look for the **INSTCONFSUCCESS** indicator of success in the log:  

```
Uninstall, com.ibm.ws.install.ni.ismp.actions.  
  ISMPLogSuccessMessageAction, msg1,  
  INSTCONFSUCCESS
```
9. Uninstall manually before reinstalling.  
The uninstaller program leaves some log files, including the `install_root/logs/uninstlog.txt` file.  
Manually uninstall the product to remove all artifacts of the product so that you can reinstall into the same installation root directory. If you do not plan to reinstall, you do not need to manually uninstall.  
See “Uninstalling manually” on page 325 for more information.

This procedure results in uninstalling the product.

After uninstalling the product and cleaning the system with the manual uninstall procedure, you can reinstall.

---

## uninstall command

The **uninstall** command uninstalls the product. When you uninstall the product, the uninstaller program removes the core product files and all of the profiles.

### Before you begin uninstalling the product

If servers are running and security is enabled, the uninstaller program cannot shut down the servers and the uninstall procedure fails. Manually stop all servers before uninstalling the product.

You can disable security in the administrative console before uninstalling the product. Then the uninstaller program can stop all server processes. Click **Security > Global security** and clear the check box for enabling global security in the administrative console.

You can also modify the `install_root/profiles/profile_name/properties/soap.client.props` file in each profile to allow the uninstaller program to issue commands against the profile. After modifying the file, you can run the uninstaller program in a secure environment without having to manually stop each server process.

See Appendix A, “Uninstalling the product,” on page 317 for more information about how to edit the `properties/soap.client.props` file.

## Location of the command file




The command file is located in the `_uninst` directory in the installation root directory.

The command file is named:

-   `uninstall`
-  `uninstall.exe`

## Command syntax

Command parameters are described at the end of this topic. This section describes the recommended command syntax, which starts the uninstaller wizard.

-   Issue the **uninstall** command:  
`install_root/_uninst/uninstall`
-  Run the uninstaller program in the installation root of the product that you intend to uninstall. Issue a fully qualified command or change directories to the `install_root\_uninst` directory and issue the **uninstall** command. For example, if the installation root is `C:\IBM\WebSphere\AppServer`, issue the following command:  
`"C:\IBM\WebSphere\AppServer\_uninst\uninstall.exe"`

## Uninstalling the product

See “Uninstalling Network Deployment” on page 322 for information about how to uninstall the product.

## Logging

Examine the `install_root/logs/uninstlog.txt` file to verify that there were no file system or other unusual errors while uninstalling. If there are problems, correct them, and uninstall manually before reinstalling the product.

After uninstalling a WebSphere Application Server product, delete the installation root directory before reinstalling into the same directory. Deleting the old directory avoids XML configurations that are not valid that result from retaining old files. To delete all files so that you can reinstall with a clean system, see “Uninstalling manually” on page 325.

---

## vpd.properties file

The installer program for any WebSphere Application Server product uses the Install Shield for Multiplatforms (ISMP) program to install code. ISMP uses the `vpd.properties` file to track WebSphere products that it installs on all platforms but Solaris and HP-UX. The `vpd.properties` file lists program components that are currently installed. The file helps ISMP and the installer programs of WebSphere Application Server products to recognize previous installations of WebSphere Application Server products and to control options for new installations.

## Situations that require you to edit the vpd.properties file

Certain situations require you to edit the `vpd.properties` file before reinstalling a WebSphere Application Server product. The uninstaller programs for WebSphere Application Server products edit the `vpd.properties` while uninstalling a product, to remove entries for the product and any of its features that might have entries in the file.



It is necessary to remove product entries in the vpd.properties file before you can reinstall the product. Some situations that occur require manually removing entries from the vpd.properties file before you can reinstall a product. These situations include:

- Bypassing the uninstaller program to uninstall a product manually
- Uninstalling a product manually when the uninstaller program is not present or is not working

If the vpd.properties file has entries for a product that you uninstalled, you must edit the file and remove the entries. If you do not edit the vpd.properties file to remove entries for a product or features of a product, you cannot reinstall the product into the same directory structure. If product entries in the vpd.properties file are present, the installer program reads the vpd.properties file, determines that the product is already installed, and displays the panel that prompts you to install additional features into the existing product or to install the binaries a second time. Unfortunately, the existing binaries might not be valid at that point. The installer program does not verify the products that it finds listed in the vpd.properties file.

### Location of the vpd.properties file

The location of the vpd.properties file varies per operating platform:

- AIX platforms: The root directory or the usr/lib/objrepos directory
- Linux platforms: The root directory
- Windows platforms: installation directory of the operating system, such as the C:\WINNT directory or the C:\windows directory.

### Example of the vpd.properties file

The following example shows all of the entries for the vpd.properties file for Version 6.0.0 of the Network Deployment product on a Windows platform. The example shows entire lines but breaks each line for formatting purposes. The identifier for the first component is highlighted. The installation location of the first component is also highlighted.

```

WSBAA60|6|0|0|0|6.0.0.0|1=WebSphere Application Server|
  IBM WebSphere Application Server| IBM| 6.0.0.0|
  C:\Program Files\IBM\WebSphere\AppServer|0|0|1|
  WSBAA60|6|0|0|0|6.0.0.0|1|0|false|
  "_uninst" "uninstall.jar" "uninstall.dat" ""|true|3|
  WSBAA60|6|0|0|0|6.0.0.0|1
WSBAA60NDOnlyComponent| | | | |1=NDOnly.ismp.component|
  ND Only| IBM| 6.0.0.0|
  C:\Program Files\IBM\WebSphere\AppServer\properties\version\install\
  6.0.0.0\componentmaps|0|0|1|
  WSBAA60NDOnly|6|0|0|0|6.0.0.0|1|0|false| |true|3|
  WSBAA60NDOnlyComponent| | | | |1
WSBAA60JavadocsComponent|6|0|0|0|6.0.0.0|1=Javadocs.ismp.component|
  Javadocs| IBM| 6.0.0.0|
  C:\Program Files\IBM\WebSphere\AppServer\properties\version\install\
  6.0.0.0\componentmaps|0|0|1|
  WSBAA60Javadocs|6|0|0|0|6.0.0.0|1|0|false| |true|3|
  WSBAA60JavadocsComponent|6|0|0|0|6.0.0.0|1
WSBAA60Javadocs|6|0|0|0|6.0.0.0|1=Javadocs|
  Javadocs|$L(com.ibm.ws.install.resourcebundle.WSResourceBundle,
  Features.javadoc.description)|IBM| 6.0.0.0|
  C:\Program Files\IBM\WebSphere\AppServer|0|0|1|
  WSBAA60CoreRuntime|6|0|0|0|6.0.0.0|1|0|false| |true|3|
  WSBAA60Javadocs|6|0|0|0|6.0.0.0|1
WSBAA60LicensingComponent| | | | |1=LAP Component|
  LAP Component|This bean will work with the licensing files| | | |
  C:\Program Files\IBM\WebSphere\AppServer|0|0|1|
  WSBAA60|6|0|0|0|6.0.0.0|1|0|false| |true|3|
  WSBAA60LicensingComponent| | | | |1
WSBAA60CoreRuntime|6|0|0|0|6.0.0.0|1=CoreRuntime|
  Core Runtime|$L(com.ibm.ws.install.resourcebundle.WSResourceBundle,

```



```

Features.coreruntime.description)|IBM| |6.0.0.0|
C:\Program Files\IBM\WebSphere\AppServer|0|0|1|
WSBAA60|6|0|0|0|6.0.0.0|1|0|false| |true|3|
WSBAA60CoreRuntime|6|0|0|0|6.0.0.0|1
WSBAA60SamplesComponent|6|0|0|0|6.0.0.0|1=Samples.ismp.component|
Samples| |IBM| |6.0.0.0|
C:\Program Files\IBM\WebSphere\AppServer\properties\version\install\
6.0.0.0\componentmaps|0|0|1|
WSBAA60Samples|6|0|0|0|6.0.0.0|1|0|false| |true|3|
WSBAA60SamplesComponent|6|0|0|0|6.0.0.0|1
WSBAA60CoreRuntimeComponent|6|0|0|0|6.0.0.0|1=CoreRuntime.ismp.component|
Core Runtime| |IBM| |6.0.0.0|
C:\Program Files\IBM\WebSphere\AppServer\properties\version\install\
6.0.0.0\componentmaps|0|0|1|
WSBAA60CoreRuntime|6|0|0|0|6.0.0.0|1|0|false| |true|3|
WSBAA60CoreRuntimeComponent|6|0|0|0|6.0.0.0|1
WSBAA60NDOnly|6|0|0|0|6.0.0.0|1=NDOnly|
Invisible ND Only Feature| |IBM| |6.0.0.0|
C:\Program Files\IBM\WebSphere\AppServer|0|0|1|
WSBAA60CoreRuntime|6|0|0|0|6.0.0.0|1|0|false| |true|3|
WSBAA60NDOnly|6|0|0|0|6.0.0.0|1
WSBAA60Samples|6|0|0|0|6.0.0.0|1=Samples|
Samples|$L(com.ibm.ws.install.resourcebundle.WSResourceBundle,
Features.samples.description)|IBM| |6.0.0.0|
C:\Program Files\IBM\WebSphere\AppServer|0|0|1|
WSBAA60CoreRuntime|6|0|0|0|6.0.0.0|1|0|false| |true|3|
WSBAA60Samples|6|0|0|0|6.0.0.0|1

```

## Identifying entries in the vpd.properties file

Use the following table to help identify product entries.

Table 22. Identifiers in the vpd.properties file for WebSphere products

Identifier	Product
WSB...60	All Version 6 products use this identifier to identify the core product files: <ul style="list-style-type: none"> <li>WebSphere Application Server - Express, Version 6</li> <li>WebSphere Application Server, Version 6</li> <li>WebSphere Application Server Network Deployment, Version 6</li> </ul>
WSP...60	Web server plug-ins for WebSphere Application Server, Version 6
WSC...	WebSphere Application Server Clients, Version 6

Web server plug-ins for WebSphere Application Server installs the following packages:

Operating system	Registry keys
AIX, Linux, and Windows	<ul style="list-style-type: none"> <li>WSPAA60</li> <li>WSPAA60DefineglobalconstantsComponent</li> <li>WSPAA60DefinelocalvariablesComponent</li> <li>WSPAA60LicensingComponent</li> <li>WSPAA60Webserverplugins</li> <li>WSPAA60WebserverpluginsComponent</li> <li>WSPAA60AddBytes</li> <li>WSPAA60gskit</li> <li>WSPAA60gskitComponent</li> </ul>

---

## Uninstalling Network Deployment

This task describes uninstalling the WebSphere Application Server Network Deployment product.

The uninstaller program removes registry entries, uninstalls the product, and removes all related features. The uninstaller program does not remove log files in the installation root directory.

The uninstaller program removes all profiles, including all of the configuration data and applications in each profile. Before you start the uninstall procedure, back up the `config` folder, the `installableApps` folder, and the `installedApps` folder of each profile, if necessary. See the *Administration: Operations* PDF for a description of managing configuration files. Back up all applications that are not stored in another location.

This procedure uninstalls the WebSphere Application Server Network Deployment product.

See the following topics to uninstall other components on the product disc:

- “Uninstalling the Web server plug-ins for WebSphere Application Server” on page 200
  - Uninstalling IBM HTTP Server
  - “Uninstalling Application Client for WebSphere Application Server” on page 222
1. Log on as root on a Linux or UNIX system, or as a user who belongs to the administrator group on a Windows system.
  2. Run the uninstaller program for the Web server plug-ins for WebSphere Application Server.  
If a Web server is configured to run with the Application Server, uninstall the plug-ins to remove the configuration from the Web server. See “Uninstalling the Web server plug-ins for WebSphere Application Server” on page 200.
  3. Stop the deployment manager `dmgr` process with the **stopManager** command.  
Stop all `dmgr` processes that are running on the machine. For example, issue this command on a Linux machine from the `/opt/IBM/WebSphere/AppServer/profiles/dmgr_profile/bin` directory:  

```
./stopManager.sh -user user_ID -password password
```
  4. Stop the `nodeagent` process with the **stopNode** command.  
Stop the `nodeagent` process that might be running on the machine. For example, issue the following command from the `/opt/IBM/WebSphere/AppServer/profiles/app_server_profile/bin` directory of a federated node on a Linux machine to stop the `nodeagent` process:  

```
./stopNode.sh
```

  
If servers are running and security is enabled, use the following command:  

```
./stopNode.sh -user user_ID -password password
```
  5. Stop each running Application Server with the **stopServer** command.  
If security is disabled, the uninstaller program can stop all WebSphere Application Server processes automatically. If servers are running and security is enabled, the uninstaller program cannot shut down the servers and the uninstall procedure fails. Manually stop all servers before uninstalling.  
Stop all server processes in all profiles on the machine. For example, issue the following command from the `/opt/IBM/WebSphere/AppServer/profiles/app_server_profile/bin` directory on a Linux machine to stop the `server1` process in the profile:  


```
./stopServer.sh server1
```




  
If servers are running and security is enabled, use the following commands:  


```
./stopServer.sh server1 -user user_ID -password password
```
  6. **Optional:** Back up configuration files and log files to refer to them later, if necessary.  
The uninstaller program does not remove log files in the installation root directory. The uninstaller program removes all profiles and all of the data in all profiles.  
Back up the `config` folder and the `logs` folder of each profile to refer to it later, if necessary. You cannot reuse profiles so there is no need to back up an entire profile.

7. Issue the **uninstall** command. The command file is named:



-   `uninstall`
-  `uninstall.exe`

 Issue the **uninstall** command:

-   `install_root/_uninst/uninstall`
-  `install_root\_uninst\uninstall.exe`

 Call the program directly from the `install_root \_uninst` directory. For example, if the installation root is `C:\IBM\WebSphere\AppServer`, issue the following command:

```
C:\IBM\WebSphere\AppServer\_uninst> uninstall.exe
```

  Issue the **uninstall** command from the `install_root/_uninst` directory:

```
./uninstall
```

The uninstaller wizard begins and displays the Welcome panel.

8. Click **Next** to begin uninstalling the product. The uninstaller wizard displays a confirmation panel that lists a summary of the product and features that you are uninstalling.

9. Click **Next** to continue uninstalling the product.

The uninstaller deletes all profiles before it deletes the core product files. Before using the uninstaller, back up any profile data that you intend to preserve. The uninstaller does not retain data files or configuration data that is in the profiles.

After uninstalling profiles, the uninstaller program deletes the core product files in component order.




10. Click **Finish** to close the wizard after the wizard removes the product.

11. Remove any configuration entries in the managed node that describe a deleted deployment manager.

A common topology is to install the core product files on multiple machines. One machine has the deployment manager and other machines have managed nodes created from custom profiles or federated application server profiles. If you delete a Network Deployment installation where you created an application server profile or a custom profile and federated the node into a deployment manager cell in another installation, you must remove the configuration from the deployment manager.

The official statement of support for a node configuration problem in the managed node is that you use the **backupConfig** command after the initial installation. Use the command again whenever you make significant changes to the configuration that you must save. With a valid backup of the configuration, you can always use the **restoreConfig** command to get back to a previously existing state in the configuration.

You can also use the following command to remove the node when the deployment manager is not running. Issue the command from the `install_root/profiles/managed_node_profile/bin` directory on the machine with the managed node:

-   `removeNode.sh -force`
-  `removeNode.bat -force`

If you must manually clean up the configuration on the managed node, you can attempt the following unsupported procedure:

a. Rename the `cell_name` directory for the node to the original name if the current name is not the original name.

Go to the `install_root/profiles/node_profile_name/config/cells/` directory. Rename the `cell_name` directory to the original name.

b. Delete the `dmgr_node_name` directory if it exists.




Go to the `install_root/profiles/node_profile_name/config/cells/original_cell_name/nodes` directory to look for the `dmgr_node_name` directory that you must delete.

- c. Edit the `setupCmdLine.sh` file on a Linux or UNIX operating system, or the `setupCmdLine.sh` file on a Windows system and change the cell name to the original cell name.

The file is in the `install_root/profiles/node_profile_name/bin` directory. Change the value of the `WAS_CELL` variable to the original cell name.

12. Remove any configuration entries in the deployment manager that describe a deleted managed node. Open the administrative console of the deployment manager and click **System administration > Nodes > node\_name > Remove node**.

If the administrative console cannot successfully remove the node, run the following command with the deployment manager running:

-   `install_root/bin/cleanupNode.shnode_name`
-  `install_root/bin/cleanupNode.batnode_name`

The official statement of support for a node configuration problem in the deployment manager is that you use the **backupConfig** command after the initial installation. Use the command again whenever you make significant changes to the configuration that you must save. With a valid backup of the configuration, you can always use the **restoreConfig** command to get back to a previously existing state in the configuration.

If you must manually clean up the configuration, you can attempt the following unsupported procedure:

- a. Within the nodes directory of the deployment manager, remove the configuration directory for the node that you deleted.

Go to the `install_root/profiles/dmgr_profile_name/config/cells/cell_name/nodes` directory to find the `deleted_node_name` file.

- b. Within the buses directory of the deployment manager, remove the configuration directory for the node that you deleted.

Go to the `install_root/profiles/dmgr_profile_name/config/cells/cell_name/buses` directory to find the `deleted_node_name` file.

- c. Edit the `coregroup.xml` file in each subdirectory of the coregroups directory of the deployment manager. Look for elements of type `coreGroupServers`. Remove any `coreGroupServers` elements that have a reference to the node that you deleted.

Go to the

`install_root/profiles/dmgr_profile_name/config/cells/cell_name/coregroups/deleted_node_name` directory to find the file.

- d. Edit the `nodegroup.xml` file in each subdirectory of the nodegroups directory of the deployment manager. Look for elements of type `members`. Remove any `members` elements that have a reference to the node that you deleted.

Go to the

`install_root/profiles/dmgr_profile_name/config/cells/cell_name/coregroups/deleted_node_name` directory to find the file.

This procedure uninstalls the Network Deployment product.

After running the **uninstall** command, the directory structure has only a few remaining directories. The `logs` directory is one of the few directories with files.

The uninstaller program leaves some log files, including the `install_root/logs/uninstlog.txt` file.

Manually uninstall the product to remove all artifacts of the product so that you can reinstall into the same installation root directory. If you do not plan to reinstall, you do not need to manually uninstall.

See “Uninstalling manually” for more information.

---

## Uninstalling manually

The uninstaller program leaves some files that can prevent you from reinstalling into the original directory. Delete files and registry entries to *clean* the machine so that you can reinstall into any directory. If you are not planning to reinstall, do not uninstall manually.

You can reinstall without a clean machine. Such an installation creates a coexistence scenario that can prevent you from installing into the original directory.

The uninstaller program removes all profiles, including all of the configuration data and applications in each profile. Before you start the uninstall procedure, back up the `config` folder, the `installableApps` folder, and the `installedApps` folder of each profile, if necessary. See the *Administration: Operations* PDF for a description of managing configuration files. Back up all applications that are not stored in another location.

Cleaning the machine means deleting everything from the previous installation, including log files that are left behind by the **uninstall** command. Before you start the procedure, back up log files, if necessary. See Appendix B, “Troubleshooting installation,” on page 337 for the location of log files.

This procedure uninstalls your WebSphere Application Server product.

Run one of the following procedures to produce a clean machine on your operating system.

- Uninstall manually on an AIX system.
- Uninstall manually on an HP-UX system.
- Uninstall manually on a Linux system.
- Uninstall manually on a Solaris system.
- Uninstall manually on a Windows system.

Manually uninstalling produces a clean system. A clean system has no evidence of a previously deleted installation.

### Manually uninstalling on an AIX system

This procedure uninstalls a WebSphere Application Server product from an AIX system. After running the uninstaller program, manual steps remove log files and registry entries that can prevent you from reinstalling the product into the original directory. If you are not planning to reinstall, do not uninstall manually.

The uninstaller program removes all profiles, including all of the configuration data and applications in each profile. Before you start the uninstall procedure, back up the `config` folder, the `installableApps` folder, and the `installedApps` folder of each profile, if necessary. See the *Administration: Operations* PDF for a description of managing configuration files. Back up all applications that are not stored in another location.

Determine the installation root directory for the product so that you remove the correct product and produce a clean system.

Reinstalling the product into a new directory when files remain from a previous installation can create a coexistence scenario. However, you can delete all files and registry entries to completely remove a WebSphere Application Server product. A clean system lets you reinstall the product into the original directory without coexistence.

Default directories are shown in the following planning table:

Identifier	Directory	Actual location
<i>install_root</i>	/usr/IBM/WebSphere/AppServer	
<i>profiles_install_root</i>	/usr/IBM/WebSphere/AppServer/profiles	
<i>plug-ins_install_root</i>	/usr/IBM/WebSphere/Plugins	

The Installation wizard and the Profile creation wizard let you specify your own locations for installation root directories. Examine the following files to determine the actual locations:

- The `~/ .WASRegistry` file identifies the installation root for all installed WebSphere Application Server products.
- The `install_root/logs/wasprofile/wasprofile_create_profile_name.log` file for each created profile identifies the installation location in the stanza with the `<method>invokeWSProfile</method>` tag.

Uninstalling the product leaves the `profiles_root/profile_name` directory with the `pctLog.txt` file.

Perform the following procedure to produce a clean system.

1. Log on as root.
2. Run the uninstaller program for the Web server plug-ins for WebSphere Application Server.  
If a Web server is configured to run with the application server, uninstall the plug-ins to remove the configuration from the Web server. See “Uninstalling the Web server plug-ins for WebSphere Application Server” on page 200.

3. Issue the **uninstall** command.

If you have already run the uninstaller program or if you cannot run the uninstaller program, simply skip this step.

```
install_root/_uninst/uninstall
```

See the description of the “uninstall command” on page 318 for more information.

The Uninstaller wizard begins and displays the Welcome panel.

4. Use the **kill** command to kill all Java processes that are running.

If running Java processes are not related to WebSphere Application Server products and it is not possible to stop them, stop all WebSphere Application Server product-related processes. Use the following command to determine all processes that are running:

```
ps -ef | grep java
```

Stop all WebSphere Application Server-related processes with the **kill -9 java\_pid\_1 java\_pid\_2...java\_pid\_n** command.

5. List WebSphere Application Server components that are installed.

Type the following command to search for related packages:

```
lslpp -l | grep -i WS
```

Do not remove packages for WebSphere Application Server products that you are not uninstalling. Version 6 package names have a prefix of WSB or WSP and a suffix of 60. WSC package names do not have a suffix of 60.

6. Use the **geninstall** command to remove registry information for each component that displays in the list.

For example, issue the follow command to display packages with a prefix of WS:

```
lslpp -l | grep -i WS
```

The system displays the list of matching packages:

```
WSBAA60CoreRuntime      6.0.0.0  COMMITTED  Installs the core product run
```

Issue the follow command to remove the registry entry:

```
geninstall -u WSBAA60CoreRuntime
```

7. Change directories to the `/usr/IBM` directory.



8. Type `rm -rf WebSphere` to delete this WebSphere Application Server-related directory, but only if the `AppServer` directory is the only directory within the `WebSphere` directory. Delete the directory if the only products contained in the directory are products that you intend to delete.
9. Edit the `.WASRegistry` file.

The file location is the home directory, `~/ .WASRegistry`.

The `.WASRegistry` file contains a one-line entry for each WebSphere Application Server product installation.

You can delete the file if there is just one line that identifies the product that you are removing. Otherwise, use a flat-file editor to remove the line that identifies the installation root directory of the product that you are removing. Leave the other lines intact. Do not delete the `.WASRegistry` file unless you are removing all of the installations listed in the file.

The following example shows a `.WASRegistry` file for a system with five installations:

```
/usr/IBM/WebSphere/AppServer
/usr/IBM/ND/AppServer
/usr/ExpressV6/IBM/WebSphere/AppServer
/usr/ND/IBM/WebSphere/AppServer
/usr/Express/IBM/WebSphere/AppServer
```

10. Run the `odmclean.sh` and `aixclean.sh` scripts.

Do not run the `aixclean.sh` script if the WebSphere MQ product is installed.

- a. Obtain the scripts from the technote document titled, `Manual uninstall On AIX requires odmclean.sh and aixclean.sh` on the WebSphere Application Server Support site.
- b. Edit the `odmclean.sh` script and replace every instance of the string `/usr/WebSphere/AppServer` with the actual installation root directory.
- c. Run the `aixclean.sh` script from the command line:
 

```
./aixclean.sh
```
- d. Run the `odmclean.sh` script from the command line:
 

```
./odmclean.sh
```

This procedure results in having a clean system. You can reinstall into the same directories now. A clean system has no trace of a previously deleted installation.

### Example of displaying package names beginning with `WS`, for WebSphere Application Server-related products

```
==>ls1pp -l | grep WS
```

```
WSBAA60          6.0.0.0  COMMITTED  ISMP installed entry
WSBAA60AddBytesNonHP 6.0.0.0  COMMITTED  ISMP installed entry
WSBAA60CoreRuntime 6.0.0.0  COMMITTED  Installs the core product run
WSBAA60CoreRuntimeComponent
WSBAA60Javadocs   6.0.0.0  COMMITTED  Installs the public Java API
WSBAA60JavadocsComponent 6.0.0.0  COMMITTED  ISMP installed entry
WSBAA60License    6.0.0.0  COMMITTED  ISMP installed entry
WSBAA60LicenseComponent 0.0.0.0  COMMITTED  ISMP installed entry
WSBAA60LicensingComponent 0.0.0.0  COMMITTED  This bean will work with the
WSBAA60Samples    6.0.0.0  COMMITTED  Includes Samples with source
WSBAA60SamplesComponent 6.0.0.0  COMMITTED  ISMP installed entry...
WSPAA60           6.0.0.0  COMMITTED  ISMP installed entry...
WSPAA60DefineglobalconstantsComponent 6.0.0.0  COMMITTED  ISMP installed entry...
WSPAA60DefinelocalvariablesComponent 6.0.0.0  COMMITTED  ISMP installed entry...
WSPAA60LicensingComponent 6.0.0.0  COMMITTED  ISMP installed entry...
WSPAA60Webserverplugins 6.0.0.0  COMMITTED  ISMP installed entry...
WSPAA60WebserverpluginsComponent 6.0.0.0  COMMITTED  ISMP installed entry...
WSPAA60AddBytes    6.0.0.0  COMMITTED  ISMP installed entry...
WSPAA60gskit       6.0.0.0  COMMITTED  ISMP installed entry...
WSPAA60gskitComponent 6.0.0.0  COMMITTED  ISMP installed entry...
```

## Manually uninstalling on an HP-UX system

This procedure uninstalls a WebSphere Application Server product from an HP-UX system. After running the uninstaller program, manual steps remove log files and registry entries that can prevent you from reinstalling the product into the original directory. If you are not planning to reinstall, do not uninstall manually.

The uninstaller program removes all profiles, including all of the configuration data and applications in each profile. Before you start the uninstall procedure, back up the `config` folder, the `installableApps` folder, and the `installedApps` folder of each profile, if necessary. See the *Administration: Operations* PDF for a description of managing configuration files. Back up all applications that are not stored in another location.

Determine the installation root directory for the product so that you remove the correct product and produce a clean system.

Reinstalling the product into a new directory when files remain from a previous installation can create a coexistence scenario. However, you can delete all files and registry entries to completely remove a WebSphere Application Server product. A clean system lets you reinstall the product into the original directory without coexistence.

Default directories are shown in the following planning table:

Identifier	Directory	Actual location
<i>install_root</i>	/opt/IBM/WebSphere/AppServer	
<i>profiles_install_root</i>	/opt/IBM/WebSphere/AppServer/profiles	
<i>plug-ins_install_root</i>	/opt/IBM/WebSphere/Plugins	

The Installation wizard and the Profile creation wizard let you specify your own locations for installation root directories. Examine the following files to determine the actual locations:

- The `~/ .WASRegistry` file identifies the installation root for all installed WebSphere Application Server products.
- The `install_root/logs/wasprofile/wasprofile_create_profile_name.log` file for each created profile identifies the installation location in the stanza with the `<method>invokeWSProfile</method>` tag.

Uninstalling the product leaves the `profiles_root/profile_name` directory with the `pctLog.txt` file.

Perform the following procedure to produce a clean system.

1. Log on as root.
2. Run the uninstaller program for the Web server plug-ins for WebSphere Application Server.  
If a Web server is configured to run with the application server, uninstall the plug-ins to remove the configuration from the Web server. See “Uninstalling the Web server plug-ins for WebSphere Application Server” on page 200.
3. Issue the **uninstall** command.  
If you have already run the uninstaller program or if you cannot run the uninstaller program, simply skip this step.  
`install_root/_uninst/uninstall`  
See the description of the “uninstall command” on page 318 for more information.  
The Uninstaller wizard begins and displays the Welcome panel.
4. Use the **kill** command to kill all Java processes that are running.



If running Java processes are not related to WebSphere Application Server products and it is not possible to stop them, stop all WebSphere Application Server product-related processes. Use the following command to determine all processes that are running:

```
ps -ef | grep java
```

Stop all WebSphere Application Server-related processes with the **kill -9 java\_pid\_1 java\_pid\_2...java\_pid\_n** command.

5. Use HP-UX System Administration Manager (SAM) to remove packages.
  - a. Start the SAM utility and verify that your DISPLAY and TERM environment variables are set properly.
  - b. Click **Software management**.
  - c. Click **View installed software**.
  - d. Look for WebSphere Application Server entries or IBM HTTP Server entries in the SD list.
  - e. Close the SD list.
  - f. Click **Remove local host software**.
  - g. Click any of the following instances that display in the SD Remove List:
    - **IBM HTTP Server**
    - **WSBAA60**
    - **WSPAA60**
    - **WSCAA**
    - **gsk7bas**
  - h. Click **Actions > Mark for remove**.
  - i. Click **Actions > Remove**.
  - j. Click **OK** in the Remove analysis dialog box.
  - k. Click **Logs** to display real-time removal of selected packages.
  - l. Click **Done** when all packages are removed.
  - m. Exit SAM.

6. Search for the packages to verify their removal.

Type **swlist | grep WS** to show packages for WebSphere Application Server.

Package names for Web server plug-ins for WebSphere Application Server are:

```
WSPAA60
WSPAA60DGCC
WSPAA60DLVC
WSPAA60LC
WSPAA60WSPC
WSPAA60AddBytesHS
WSPAA60gskithHP
WSPAA60gskithPC
WSPAA60jdkHP
```

7. Type **rm -rf /opt/IBM/WebSphere/AppServer/** to remove WebSphere Application Server directories in the /opt/IBM/WebSphere/AppServer installation root. Do not remove installation root directories for products that you intend to keep. Remove all of the profile directories as well.
8. Edit the .WASRegistry file.

The file location is the home directory, ~/.WASRegistry.

The .WASRegistry file contains a one-line entry for each WebSphere Application Server product installation.

You can delete the file if there is just one line that identifies the product that you are removing. Otherwise, use a flat-file editor to remove the line that identifies the installation root directory of the product that you are removing. Leave the other lines intact. Do not delete the .WASRegistry file unless you are removing all of the installations listed in the file.

The following example shows a .WASRegistry file for a system with five installations:

```
/opt/IBM/WebSphere/AppServer
/opt/IBM/P1/AppServer
/opt/V6/IBM/WebSphere/AppServer
/opt/P2/IBM/WebSphere/AppServer
/opt/P3/IBM/WebSphere/AppServer
```

This procedure results in having a clean system. You can reinstall into the same directories now. A clean system has no trace of a previously deleted installation.

### Example of displaying package names for WebSphere Application Server products

Issue the following command:

```
swlist | grep WSBA60
```

The following package name is returned:

```
WSBA60      6.0.0      IBM WebSphere Application Server
```

## Manually uninstalling on a Linux system

This procedure uninstalls a WebSphere Application Server product from a Linux system. After running the uninstaller program, manual steps remove log files and registry entries that can prevent you from reinstalling the product into the original directory. If you are not planning to reinstall, do not uninstall manually.

The uninstaller program removes all profiles, including all of the configuration data and applications in each profile. Before you start the uninstall procedure, back up the config folder, the installableApps folder, and the installedApps folder of each profile, if necessary. See the *Administration: Operations* PDF for a description of managing configuration files. Back up all applications that are not stored in another location.

Determine the installation root directory for the product so that you remove the correct product and produce a clean system.

Reinstalling the product into a new directory when files remain from a previous installation can create a coexistence scenario. However, you can delete all files and registry entries to completely remove a WebSphere Application Server product. A clean system lets you reinstall the product into the original directory without coexistence.

Default directories are shown in the following planning table:

Identifier	Directory	Actual location
<i>install_root</i>	/opt/IBM/WebSphere/AppServer	
<i>profiles_install_root</i>	/opt/IBM/WebSphere/AppServer/profiles	
<i>plug-ins_install_root</i>	/opt/IBM/WebSphere/Plugins	

The Installation wizard and the Profile creation wizard let you specify your own locations for installation root directories. Examine the following files to determine the actual locations:

- The `~/ .WASRegistry` file identifies the installation root for all installed WebSphere Application Server products.
- The `install_root/logs/wasprofile/wasprofile_create_profile_name.log` file for each created profile identifies the installation location in the stanza with the `<method>invokeWSProfile</method>` tag.

Uninstalling the product leaves the `profiles_root/profile_name` directory with the `pctLog.txt` file.

Perform the following procedure to produce a clean system.

1. Log on as root.
2. Run the uninstaller program for the Web server plug-ins for WebSphere Application Server.  
If a Web server is configured to run with the application server, uninstall the plug-ins to remove the configuration from the Web server. See “Uninstalling the Web server plug-ins for WebSphere Application Server” on page 200.
3. Issue the **uninstall** command.  
If you have already run the uninstaller program or if you cannot run the uninstaller program, simply skip this step.  
`install_root/_uninst/uninstall`  
See the description of the “uninstall command” on page 318 for more information.  
The Uninstaller wizard begins and displays the Welcome panel.
4. Use the **kill** command to kill all Java processes that are running.  
If running Java processes are not related to WebSphere Application Server products and it is not possible to stop them, stop all WebSphere Application Server product-related processes. Use the following command to determine all processes that are running:  
`ps -ef | grep java`  
Stop all WebSphere Application Server-related processes with the **kill -9 java\_pid\_1 java\_pid\_2...java\_pid\_n** command.
5. Search for related packages. Issue the following command to show packages for the WebSphere Application Server products:  
`rpm -qa | grep WS`  
For example, after issuing the command, the following list of packages might display:  
WSBAA60CoreRuntimeComponent-6.0-0  
WSBAA60LicenseComponent-6.0-0  
WSBAA60JavadocsComponent-6.0-0  
WSBAA60NDOnlyComponent-6.0-0  
WSBAA60AddBytesNonHP-6.0-0  
WSBAA60LicensingComponent-6.0-0  
WSBAA60SamplesComponent-6.0-0  
WSPAA60  
WSPAA60DefineglobalconstantsComponent  
WSPAA60DefinelocalvariablesComponent  
WSPAA60LicensingComponent  
WSPAA60Webserverplugins  
WSPAA60WebserverpluginsComponent  
WSPAA60AddBytes  
WSPAA60gskit  
WSPAA60gskitComponent  
Do not remove packages for WebSphere Application Server products that you are not uninstalling. Version 6 package names have a prefix of WSB or WSP and a suffix of 60. WSC package names do not have a suffix of 60.
6. If there are packages to delete, type **rpm -e packagename** to remove any packages for the product that you are uninstalling.  
Alternatively, you can search for packages to verify that everything in the list is something to delete:  
`rpm -qa | grep WS`  
If the list contains packages that you intend to delete and no others, remove all of the packages with the following command:  
`rpm -qa | grep WS | xargs rpm -e`  
If there is a problem with package dependencies, you can use the following command to remove the packages:  
`rpm -e packagename --nodeps --justdb`

The `nodeps` option skips the dependency check. The `justdb` option updates only the package database, and not the file system. Using only the `nodeps` option can cause a failure in package removal if there is any mismatch in the dependent file system (files and directories).

7. Type `rm -rf /opt/IBM/WebSphere/AppServer/` to remove WebSphere Application Server directories in the `/opt/IBM/WebSphere/AppServer` installation root. Do not remove installation root directories for products that you intend to keep. Remove all of the profile directories as well.
8. Edit the `.WASRegistry` file.

The file location is the home directory, `~/ .WASRegistry`.

The `.WASRegistry` file contains a one-line entry for each WebSphere Application Server product installation.

You can delete the file if there is just one line that identifies the product that you are removing. Otherwise, use a flat-file editor to remove the line that identifies the installation root directory of the product that you are removing. Leave the other lines intact. Do not delete the `.WASRegistry` file unless you are removing all of the installations listed in the file.

The following example shows a `.WASRegistry` file for a system with five installations:

```
/opt/IBM/WebSphere/AppServer
/opt/IBM/P1/AppServer
/opt/V6/IBM/WebSphere/AppServer
/opt/P2/IBM/WebSphere/AppServer
/opt/P3/IBM/WebSphere/AppServer
```

This procedure results in having a clean system. You can reinstall into the same directories now. A clean system has no trace of a previously deleted installation.

## Manually uninstalling on a Solaris system

This procedure uninstalls the WebSphere Application Server product from a Solaris system. After running the uninstaller program, manual steps remove log files and registry entries that can prevent you from reinstalling the product into the original directory. If you are not planning to reinstall, do not uninstall manually.

The uninstaller program removes all profiles, including all of the configuration data and applications in each profile. Before you start the uninstall procedure, back up the `config` folder, the `installableApps` folder, and the `installedApps` folder of each profile, if necessary. See the *Administration: Operations* PDF for a description of managing configuration files. Back up all applications that are not stored in another location.

Determine the installation root directory for the product so that you remove the correct product and produce a clean system.

Reinstalling the product into a new directory when files remain from a previous installation can create a coexistence scenario. However, you can delete all files and registry entries to completely remove a WebSphere Application Server product. A clean system lets you reinstall the product into the original directory without coexistence.

Default directories are shown in the following planning table:

Identifier	Directory	Actual location
<code>install_root</code>	<code>/opt/IBM/WebSphere/AppServer</code>	
<code>profiles_install_root</code>	<code>/opt/IBM/WebSphere/AppServer/profiles</code>	
<code>plug-ins_install_root</code>	<code>/opt/IBM/WebSphere/Plugins</code>	

The Installation wizard and the Profile creation wizard let you specify your own locations for installation root directories. Examine the following files to determine the actual locations:

- The `~/ .WASRegistry` file identifies the installation root for all installed WebSphere Application Server products.
- The `install_root/logs/wasprofile/wasprofile_create_profile_name.log` file for each created profile identifies the installation location in the stanza with the `<method>invokeWSProfile</method>` tag.

Uninstalling the product leaves the `profiles_root/profile_name` directory with the `pctLog.txt` file.

Perform the following procedure to produce a clean system.

1. Log on as root.

2. Run the uninstaller program for the Web server plug-ins for WebSphere Application Server.

If a Web server is configured to run with the application server, uninstall the plug-ins to remove the configuration from the Web server. See “Uninstalling the Web server plug-ins for WebSphere Application Server” on page 200.

3. Issue the **uninstall** command.

If you have already run the uninstaller program or if you cannot run the uninstaller program, simply skip this step.

```
install_root/_uninst/uninstall
```

See the description of the “uninstall command” on page 318 for more information.

The Uninstaller wizard begins and displays the Welcome panel.

4. Use the **kill** command to kill all Java processes that are running.

If running Java processes are not related to WebSphere Application Server products and it is not possible to stop them, stop all WebSphere Application Server product-related processes. Use the following command to determine all processes that are running:

```
ps -ef | grep java
```

Stop all WebSphere Application Server-related processes with the **kill -9 java\_pid\_1 java\_pid\_2...java\_pid\_n** command.

5. Search for related packages. Type the following command to search for packages for WebSphere Application Server products:

```
pkginfo | grep WS
```

If no packages appear when using these commands, skip the next step. The resulting list of packages has the following format:

```
application WSBA60          WebSphere Application Server
```

6. Change directories to the directory where package information is registered.

```
cd /var/sadm/pkg
```

7. Issue the following command to remove any WebSphere Application Server-related packages.

```
pkgrm packagename1 packagename2 packagename3 ...
```

Do not remove packages for WebSphere Application Server products that you are not uninstalling. Version 6 package names have a prefix of WSB or WSP and a suffix of 60. WSC package names do not have a suffix of 60.

Issue the following commands from the `/var/sadm/pkg` directory to search for and remove any WebSphere Application Server product-related packages that are registered in the `/var/sadm/pkg` directory:

a. Change directories to the correct directory: `cd /var/sadm/pkg`

b. `ls |grep WSB|xargs -i pkgrm -n {}` for WebSphere Application Server products

c. `ls |grep WSC|xargs -i pkgrm -n {}` for WebSphere Application Server Clients

d. `ls |grep WSP|xargs -i pkgrm -n {}` for Web server plug-ins for WebSphere Application Server

Package names for Web server plug-ins for WebSphere Application Server are:

```
WSPAA60
WSPAA60AC
WSPAA60BC
WSPAA60CC
```

WSPAA60DC  
WSPAA60FC  
WSPAA60FB  
WSPAA60GC  
WSPAA60HC

If there is a problem removing the packages, remove the related package directories in the `/var/sadm/pkg` directory, including the preremove files. For example, remove the following file before issuing the `pkgm -n WSBAA60` command:

```
/var/sadm/pkg/WSBAA60/install/preremove
```

8. Type `rm -rf /opt/IBM/WebSphere/AppServer/` to remove WebSphere Application Server directories in the `/opt/IBM/WebSphere/AppServer` installation root. Do not remove installation root directories for products that you intend to keep. Remove all of the profile directories as well.
9. Edit the `.WASRegistry` file.

The file location is the home directory, `~/.WASRegistry`.

The `.WASRegistry` file contains a one-line entry for each WebSphere Application Server product installation.

You can delete the file if there is just one line that identifies the product that you are removing. Otherwise, use a flat-file editor to remove the line that identifies the installation root directory of the product that you are removing. Leave the other lines intact. Do not delete the `.WASRegistry` file unless you are removing all of the installations listed in the file.

The following example shows a `.WASRegistry` file for a system with five installations:

```
/opt/IBM/WebSphere/AppServer  
/opt/IBM/P1/AppServer  
/opt/V6/IBM/WebSphere/AppServer  
/opt/P2/IBM/WebSphere/AppServer  
/opt/P3/IBM/WebSphere/AppServer
```

This procedure results in having a clean system. You can reinstall into the same directories now. A clean system has no trace of a previously deleted installation.

## Manually uninstalling on a Windows system

This procedure uninstalls a WebSphere Application Server product from a Windows system. After running the uninstaller program, manual steps remove log files and registry entries that can prevent you from reinstalling the product into the original directory. If you are not planning to reinstall, do not uninstall manually.

The uninstaller program removes all profiles, including all of the configuration data and applications in each profile. Before you start the uninstall procedure, back up the `config` folder, the `installableApps` folder, and the `installedApps` folder of each profile, if necessary. See the *Administration: Operations* PDF for a description of managing configuration files. Back up all applications that are not stored in another location.

Determine the installation root directory for the product so that you remove the correct product and produce a clean system.

Reinstalling the product into a new directory when files remain from a previous installation can create a coexistence scenario. However, you can delete all files and registry entries to completely remove a WebSphere Application Server product. A clean system lets you reinstall the product into the original directory without coexistence.

Default directories are shown in the following planning table:



Identifier	Directory	Actual location
<i>install_root</i>	C:\ Program Files\ IBM\ WebSphere\ AppServer	
<i>profiles_install_root</i>	C:\ Program Files\ IBM\ WebSphere\ AppServer\ profiles	
<i>plug-ins_install_root</i>	C:\ Program Files\ IBM\ WebSphere\ Plugins	

**2000** **Windows 2000 consideration:** If you move either installation root directory, move them to a shorter path name. Windows 2000 has a length restriction of 258 characters for a command. A problem can occur that prevents the successful creation of a profile when either path is too long. The maximum length for the installation root directory is 60 characters. The maximum length for the profiles installation root directory is 80 characters.

The Installation wizard and the Profile creation wizard let you specify your own locations for installation root directories. Examine the following files to determine the actual locations:

- The %USERPROFILE%\ .WASRegistry file identifies the installation root for all installed WebSphere Application Server products.
- The *install\_root*\logs\wasprofile\wasprofile\_create\_profile\_name.log file for each created profile identifies the installation location in the stanza with the <method>invokeWSPProfile</method> tag.

Perform the following procedure to produce a clean system.

1. Log on with a user ID that belongs to the administrator group.
2. Run the uninstaller program for the Web server plug-ins for WebSphere Application Server.  
If a Web server is configured to run with the application server, uninstall the plug-ins to remove the configuration from the Web server. See “Uninstalling the Web server plug-ins for WebSphere Application Server” on page 200.
3. Issue the **uninstall** command.  
If you have already run the uninstaller program or if you cannot run the uninstaller program, simply skip this step.  
`install_root\_uninst\uninstall`  
See the description of the “uninstall command” on page 318 for more information.  
The Uninstaller wizard begins and displays the Welcome panel.
4. Verify that you have an Emergency Recovery Disk. Instructions for creating this disk are in the Windows help documentation.  
This step is a safeguard. This procedure does not require the recovery disk.
5. Use the regback.exe program from the Windows Resource Kit to back up the registry.  
This step is a safeguard. This procedure does not require the backup copy of the registry.
6. Delete product registry entries for the WebSphere Application Server product that you are uninstalling.  
Invoke **regedit.exe** from a command prompt, to edit the Windows system registry.

#### Handle the Registry with care

**Note:** You can easily make a mistake while using the registry editor to view and edit registry contents. The editor does not warn you of editing errors, which can be extremely dangerous. A corrupt registry can disrupt your system to the point where your only option is to reinstall the Windows operating system.

- a. Use **Ctrl-F** to search for all instances of **WebSphere**, to determine whether you should delete each entry. You might not be able to remove all of the entries related to WebSphere Application Server, which is not a problem.



- b. Expand and select keys related to **WebSphere Application Server** products.

Delete the following keys for the WebSphere Application Server product:

- HKEY\_CURRENT\_USER\ Software\ Microsoft\ Windows\ CurrentVersion\ Explorer\ MenuOrder\ Start Menu\ Programs\ IBM WebSphere\ Application Server Network Deployment v6
- HKEY\_LOCAL\_MACHINE\ Software\ IBM\ WebSphere Application Server Network Deployment\ 6.0.0.0
- HKEY\_LOCAL\_MACHINE\ Software\ IBM\ Web server Plug-ins for IBM WebSphere Application Server\ 6.0.0.0

- c. Click **Edit > Delete** from the menu bar for each related key.
- d. Click **Yes** when asked to confirm deletion of the key.
- e. Click **Registry > Exit** from the menu bar when you are finished.

7. Delete the installation root directory for the product that you are uninstalling.
8. Determine all of the profile directories and remove the directories.
9. Open a Windows Explorer window and browse to the C:\Documents and Settings\All Users\Start Menu\Programs\IBM WebSphere directory.

If you have one installation of a WebSphere Application Server product, delete the following folder:

Application Server Network Deployment v6

10. Delete the %WINDIR%\IsUninst.exe file.
11. Edit the .WASRegistry file.

The file location is the home directory of the user ID who installed the product, %USERPROFILE%\ .WASRegistry.

The .WASRegistry file contains a one-line entry for each WebSphere Application Server product installation.

You can delete the file if there is just one line that identifies the product that you are removing. Otherwise, use a flat-file editor to remove the line that identifies the installation root directory of the product that you are removing. Leave the other lines intact. Do not delete the .WASRegistry file unless you are removing all of the installations listed in the file.

The following example shows a .WASRegistry file for a system with five installations:

```
C:\Program Files\IBM\WebSphere\AppServer
C:\IBM\WebSphere\AppServer
C:\ExpressV6\IBM\WebSphere\AppServer
C:\ND_Install_9_25_2004_build\IBM\WebSphere\AppServer
C:\Express_09_25_2004\IBM\WebSphere\AppServer
```

12. Restart your machine if a prompt displays that directs you to restart.

This procedure results in having a clean system. You can reinstall into the same directories now. A clean system has no trace of a previously deleted installation.

---

## Appendix B. Troubleshooting installation

This topic describes troubleshooting the installation of the WebSphere Application Server Network Deployment product.

Use this topic after installing your WebSphere Application Server product.

The successful installation of the Network Deployment product is a two-part process:

- The first step is using the installation wizard to install a shared set of core product files.
- The second step is using the *Profile creation wizard* to create a *deployment manager profile*, an *Application Server profile*, or a *custom profile*.

If an installation is not successful, use this troubleshooting information to correct the problems.

The *install\_root/logs/log.txt* file, the *install\_root/logs/wasprofile/wasprofile\_create\_profile\_name.log* file, and the *profiles\_root/profile\_name/logs/pctLog.txt* file record installation status.

The installer program records the following indicators of success in the logs:

- INSTCONFSUCCESS
- INSTCONFPARTIALSUCCESS
- INSTCONFFAILED

If the error happens early in the installation, look for the *log.txt* file in the system temporary area. The installation program copies the log from the temporary area to the *logs* directory at the end of the installation.

Perform the following procedure to troubleshoot an installation.

1. Check the installation log files for errors after installing:

During installation, a single entry in the *install\_root/logs/log.txt* file points to the temporary log file, either *%TEMP%\log.txt* on Windows platforms, or */tmp/log.txt* on Linux and UNIX platforms. The installation program copies the file from the temporary directory to the *install\_root/logs/log.txt* location at the end of the installation.

If the installation fails and the *install\_root/logs/log.txt* has only this one pointer to the temporary directory, open the *log.txt* file in the temporary directory. The log might have clues to the installation failure. Uninstalling creates the *install\_root/logs/uninstlog.txt* file.

### **Log more information when InstallShield for MultiPlatforms (ISMP) cannot start the Installation wizard.**

Verify or troubleshoot the installation if the *install\_root/logs/log.txt* file or the *profiles\_install\_root/profile\_name/logs/pctLog.txt* file does not contain a record of any problems, but problems exist. If the *profiles\_install\_root/profile\_name* directory exists, the *pctLog.txt* file is in the *logs* directory. If the error happens early in the installation, look for the logs in the system temporary area. The installation program copies the logs from the system temporary area to the *logs* directory at the end of the installation.

If the *profiles\_install\_root/profile\_name* directory does not exist, the *pctLog.txt* file is in the *USER\_HOME* directory and is named *.\$~pctLog.txt*. Issue the **env** command to display the *USER\_HOME* directory on Linux and UNIX systems. See the environment variable settings on Windows system to display the value of the variable.

Certain events can prevent ISMP from starting the Installation wizard. Such an event is not enough disk space to launch the Installation wizard, for example. If your installation fails and there is no information in the installation logs, use the *-log* parameter to record entries for events that cause the ISMP program to fail to start the installation wizard. The syntax of the **install** command for logging such events is:

```
install -options fully_qualified_options_response_file_name
-silent
-log # !fully_qualified_log_file_name @ALL
```

The following example is for AIX systems:

```
install -options "/usr/IBM/WebSphere/silentFiles/myresponsefile.txt"
-silent -log # !/usr/IBM/WebSphere/myOptionFiles/log.txt @ALL
```

The following example is for Linux systems, HP-UX systems, and Solaris systems:

```
install -options "/opt/IBM/WebSphere/silentFiles/myresponsefile.txt"
-silent -log # !/opt/IBM/WebSphere/myOptionFiles/log.txt @ALL
```










The following example is for Windows systems:


```
install.exe -options "C:\IBM\WebSphere\silentFiles\myresponsefile.txt"
-silent -log # !C:\IBM\WebSphere\silentFiles\log.txt @ALL
```




### Attention:

The following examples show how to use the `-log` parameter when creating a deployment manager profile.

The command is in the `install_root/bin/ProfileCreator` directory. The name of the command varies per platform:

-  `pctAIX.bin`
-  `pctHPUX.bin`
-  64-bit platforms: `pctHPUXIA64.bin`
-  `pctLinux.bin`
-  64-bit platforms: `pctLinux390.bin` S/390 platforms: `pctLinux390.bin`
-  Power platforms: `pctLinuxPPC.bin`
-  `pctSolaris.bin`
-  `pctWindows.exe`
-  64-bit platforms: `pctWindowsIA64.exe`

```

pctAIX.bin
-options /usr/IBM/WebSphere/silentFiles/responsefile.pct.NDstandAloneProfile.txt
-silent
-log # !/usr/IBM/WebSphere/silentFiles/pctlog.txt @ALL
```

   This example shows the command for Linux on 32-bit platforms. Use the appropriate command for your Linux or UNIX operating system. The options and attributes are the same with only file path differences.

```
pctLinux.bin
-options "/opt/IBM/WebSphere/silentFiles/responsefile.pct.NDstandAloneProfile.txt"
-silent
-log # !/opt/IBM/WebSphere/silentFiles/log.txt @ALL
```

 The following command is for Windows platforms:

```
pctWindows.exe -options
"C:\IBM\WebSphere\silentFiles\responsefile.pct.NDstandAloneProfile.txt"
-silent -log # !C:\IBM\WebSphere\silentFiles\log.txt
@ALL
```









### Log file names and locations

The following information shows the log files for all of the installable components on the product disc.

#### Log files for IBM HTTP Server

The following table shows the installation log locations when installing IBM HTTP Server V6.






Table 23. Installation log locations when installing IBM HTTP Server

Windows system log path name	Linux and UNIX operating system log path name
	 /usr/ IBMHttpServer/ log.txt
	 /usr/ IBMHttpServer/ ihsv6_install.log
	 /opt/ IBMHttpServer/ log.txt
C:\Program Files\ IBM HTTP Server\ log.txt	 /opt/ IBMHttpServer/ ihsv6_install.log
C:\Program Files\ IBM HTTP Server\ ihsv6_install.log	 /opt/ IBMHttpServer/ log.txt
	 /opt/ IBMHttpServer/ ihsv6_install.log
	 /opt/ IBMHttpServer/ log.txt
	 /opt/ IBMHttpServer/ ihsv6_install.log

#### Log files for Application Client for WebSphere Application Server

The following table shows the installation log locations when installing the application clients for Version 6.

Table 24. Installation log locations when installing the Application Clients for WebSphere Application Server

Windows system log path name	Linux and UNIX operating system log path name
	 usr/ IBM/ WebSphere/ AppClient/ logs/ WAS.Client.install.log
 C:\Program Files\ IBM\ WebSphere\ AppClient\ logs\ WAS.Client.install.log	 /opt/ IBM/ WebSphere/ AppClient/ logs/ WAS.Client.install.log
	 /opt/ IBM/ WebSphere/ AppClient/ logs/ WAS.Client.install.log
	 /opt/ IBM/ WebSphere/ AppClient/ logs/ WAS.Client.install.log

**Log files for WebSphere Application Server products:** The following table shows the installation logs, content, and indicators of success and failure for WebSphere Application Server products for Version 6.

Table 25. Installation logs for WebSphere Application Server products

Log	Content	Indicators
<i>install_root</i> / logs / log.txt	Logs all installation events	<p><b>INSTCONFFAIL</b> Total installation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful installation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Installation errors occurred but the installation is still usable. Additional information identifies the errors.</p>
<i>install_root</i> / logs / wasprofile / wasprofile _ create _ profile_name.log	<ul style="list-style-type: none"> <li>Traces all events that occur during the creation of the named profile</li> <li>Created when using the Profile creation wizard or the <b>wasprofile</b> command</li> </ul>	<p><b>INSTCONFFAIL</b> Total profile creation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile creation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile creation errors occurred but the profile is still functional. Additional information identifies the errors.</p>
<i>install_root</i> / logs / wasprofile / wasprofile _ delete _ profile_name.log	<ul style="list-style-type: none"> <li>Traces all events that occur during the deletion of the named profile</li> <li>Created when using the Profile creation wizard or the <b>wasprofile</b> command</li> </ul>	<p><b>INSTCONFFAIL</b> Total profile deletion failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile deletion.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile deletion errors occurred but the profile is still deleted. Additional information identifies the errors.</p>
<i>install_root</i> / profiles / profile_name / logs / pctLog.txt	Logs all profile creation events that occur when using the Profile creation wizard	<p><b>INSTCONFFAIL</b> Total profile creation failure.</p> <p><b>INSTCONFSUCCESS</b> Successful profile creation.</p> <p><b>INSTCONFPARTIALSUCCESS</b> Profile creation errors occurred but the profile is still functional. Additional information identifies the errors.</p>

**Description of the wasprofile\_create\_profile\_name.log file**

The wasprofile\_create\_profile\_name.log file is an XML file that contains a record of the events that occur during the creation of the last profile.

In addition to the date tag at the beginning of the file, other tags of interest in the log files include the sequence tag, the level tag, the method tag, and the message tag:

- The sequence tag records the sequence of events that occur during the creation of the profile.
- The level tag is an early indicator of event status:

**INFO** Indicates a normal event.

**WARNING**

Indicates an event that occurred with errors that do not prevent the creation of the profile.

**ERROR**

Indicates an event that prevents the creation of the profile.

- The method tag indicates the name of the routine that recorded the event.

- The message tag describes the event and contains any data returned by the method.

The following stanza is an example of how an event is documented in each log file:

```
<record>
  <date>2004-09-08T11:51:39</date>
  <millis>1094658699225</millis>
  <sequence>0</sequence>
  <logger>com.ibm.ws.profile.WSProfile</logger>
  <level>INFO</level>
  <class>com.ibm.ws.profile.WSProfile</class>
  <method>getRegistryFile</method>
  <thread>10</thread>
  <message>Returning registry file at:
    C:\NDV6\IBM\WebSphere\AppServer\properties\profileRegistry.xml
  </message>
</record>
```

### Log files created during the creation of the Application Server profile

In addition to the logs created within the core product files, the following logs are created in the *install\_root/profiles/default/logs* directory when the Profile creation wizard or the **wasprofile** command creates an Application Server profile:

#### **activity.log**

Compiled activity log from various installation activities

#### **amjrte\_config.log**

Tivoli Access Manager configuration log for its Java Runtime Environment

#### **collect\_metadata.log**

Collects metadata information about managed objects in the system to evaluate and prevent potential installation conflicts

#### **createDefaultServer.log**

A log from wsadmin recording the creation of the server1 process in the default profile

#### **createshortcutforprofile.log**

Windows tool log for creating menu entries and shortcuts

#### **defaultapp\_config.log**

JACL script log from configuring default application resources

#### **defaultapp\_deploy.log**

Application DefaultApplication installation log

#### *Node\_name* **Service.log**

Start and stop events for server1

#### **filetransfer\_config.log**

Application filetransfer installation log

#### **hamanager\_config.log**

Configuration log for the high availability application

#### **ivt\_config.log**

Application ivtApp installation log

#### **mejb\_config.log**

Application ManagementEJB installation log

#### **pctLog.txt**

Log created when using the Profile creation wizard to create a profile. This log is not created when using the **wasprofile** command directly.

This log is not created during installation of the product.

#### **query\_config.log**

Application Query installation log

**samples\_config.log**

Configuration log for the PlantsByWebSphere Samples application

**samples\_install.log**

Installation log for the SamplesGallery and PlantsByWebSphere Samples applications

**scheduler.cal\_config.log**

Application SchedulerCalendars installation log

**SIBDefineChains.log**

Creation log for service integration bus endpoints, inbound channels and channel chains, outbound thread pool, and outbound channel and channel chains

**SIBDeployRA.log**

Deployment log for the service integration bus function

**webui\_config.log**

Application administrative console installation log

**winservice\_config.log**

*Node\_name* service log for the Windows service created for server1

The following logs are created in the *install\_root/profiles/default/logs/server1* directory:

**startServer.log**

Log of start server events

**stopServer.log**

Log of stop server events

**SystemErr.log**

Record system errors

**SystemOut.log**

Log of all activity within the system

**trace.log**

Log of all traced events within the system

The following logs are created in the *install\_root/profiles/default/logs/ffdc* directory:

**server1\_exception.log**

First failure data capture log for server1 errors

**server1\_numeric\_identifier.txt**

Any first failure data capture logs

2. Verify that no files exist in the *install\_root/classes* directory.

IBM Support sometimes queues work for customers and provides test or debugging fixes. A common location for the fixes is in the *install\_root/classes* directory.

By default, the *install\_root/classes* directory is picked up first in the WebSphere Application Server class path to let it override other classes.

Putting a fix in the directory lets you verify that the fix does indeed solve your problem. After verifying that the fix solves the problem, you are supposed to delete the fix from the *install\_root/classes* directory to return the system to a working state.

If you do not remove such fixes from the *install\_root/classes* directory, you can experience errors.

3. Uninstall the product, if possible, and reinstall after turning on tracing if the error logs do not contain enough information to determine the cause of the problem.

- Report the stdout and stderr logs to the console window, by adding the `-is:javaconsole` parameter to the **install** command:

```
–  
```

```
install -is:javaconsole
```

Capture the stream to a file with the following commands:



```
install -is:javaconsole > captureFileName.txt 2>&1
```

– **Windows**

```
install.exe -is:javaconsole
```

Capture the stream to a file with the following commands:

```
install -is:javaconsole > drive:\captureFileName.txt
```

- Capture additional information to a log of your choice with the `-is:log file_name` option.
- Turn on additional installation logging by passing the `-W Setup.product.install.logAllEvents="true"` parameter to the **install** command:

– **Linux** **UNIX**

```
install -W Setup.product.install.logAllEvents="true"
```

– **Windows**

```
install -W Setup.product.install.logAllEvents="true"
```

4. If you have successfully created an Application Server profile, use the First steps console or the command line method to start the Application Server.
5. Verify whether the server starts and loads properly by looking for a running Java process and the *Open for e-business* message in the `SystemOut.log` and `SystemErr.log` files.

If no Java process exists or if the message does not appear, examine the same logs for any miscellaneous errors. Correct any errors and retry.

You can find the `SystemOut.log` and `SystemErr.log` files in the following platform-specific directory:

• **Linux** **UNIX** `install_root/profiles/profile_name/logs/server_name`

• **Windows** `install_root\profiles\profile_name\logs\server_name`

6. Use the First steps console or the command line method to stop the Application Server, if it is running, and to start the deployment manager if one exists.

**To stop server1 from the command line:**

• **Linux** **UNIX** `install_root/profiles/profile_name/bin/stopServer.sh server1`

• **Windows** `install_root\profiles\profile_name\bin\stopServer server1`

If you enable security, specify the `-user` and the `-password` parameters of the command.

**To start the deployment manager from the command line:**

• **Linux** **UNIX** `install_root/profiles/profile_name/bin/startManager.sh`

• **Windows** `install_root\profiles\profile_name\bin\startManager`

7. Verify that the server starts and loads properly by looking for a running Java process and the *Server dmgr open for e-business* message in the `install_root/profiles/profile_name/logs/server_name/SystemOut.log` file.

**Windows** Press **Ctrl+Alt+Delete** and type **T** to open the Task Manager. Click the **Processes** tab and the **Image Name** column header to sort by image name. Look for processes named `java.exe`.

**Linux** **UNIX** Open a command window and issue the **top** command to see a display of running processes. If the **top** command is not available on your UNIX system, use the **ps** command:

```
ps -ef | grep java
```

If no Java process exists or if the message does not appear, examine the same logs for any miscellaneous errors. Correct any errors and try again to start the deployment manager.

8. Refer to the plug-in configuration documentation, if you have installed plug-ins and the Web server does not come up properly.
9. Start the Snoop servlet to verify the ability of the Web server to retrieve an application from the Application Server.

Test your environment by starting your Application Server, your Web server, and using the snoop servlet with an IP address.

- a. Start the Application Server. In a Network Deployment environment, the Snoop servlet is available in the cell only if you included the DefaultApplication when adding the Application Server to the cell. The `-includeapps` option for the **addNode** command migrates the DefaultApplication to the cell. If the application is not present, skip this step.

Change directories to the `install_root/profiles/profile_name/bin` directory and run the **startServer** command:

-   `./startServer.sh server1`
-  `startServer server1`

- b. Start the IBM HTTP Server or the Web server that you are using.

Use a command window to change the directory to the IBM HTTP Server installed image, or to the installed image of your Web server. Issue the appropriate command to start the Web server, such as these commands for IBM HTTP Server:

**To start the IBM HTTP Server from the command line:**

Access the **apache** and **apachectl** commands in the `IBMHttpServer/bin` directory.

-   `./apachectl start`
-  `apache`

- c. Point your browser to `http://localhost:9080/snoop` to test the internal HTTP transport provided by the Application Server. Point your browser to `http://Host_name_of_Web_server_machine/snoop` to test the Web server plug-in.

The HTTP Transport port is 9080 by default and must be unique for every profile. The port is associated with a virtual host named `default_host`, which is configured to host the installed DefaultApplication and any installed Samples. The snoop servlet is part of the DefaultApplication. Change the port to match your actual HTTP Transport port.

- d. Verify that snoop is running.

Either Web address should display the Snoop Servlet - Request/Client Information page.

- e. **Remote IBM HTTP Server only:** Verify that the automatic propagation function can work on a remote IBM HTTP Server by using the following steps. This procedure is not necessary for local Web servers.
  - 1) Run the **htpasswd** command on the Web server machine to create the administrative user ID and password information.
  - 2) Use the administrative console of the deployment manager or the Application Server to enter the User ID and password information that you created for the administrative user of IBM HTTP Server. Go to **Servers > Web server > Web\_server\_definition > Remote Web server administration**.

10. Start the WebSphere Application Server administrative console.

- a. Start the Application Server.

- b. Point your browser to `http://localhost:9060/ibm/console`.

The HTTP Admin port is 9060 by default and must be unique for the administrative console of each stand-alone Application Server. The port is associated with a virtual host named `admin_host`, which is configured to host the administrative console, which is installed by default as a system application. Change the port to match your actual HTTP Admin port.

If you have problems accessing the administrative console after installation, check the `installAdminConsole.log` file for a failure indication. Clean up the system temporary directory and reinstall the administrative console using the `wsadmin` scripting facility.

- c. Type any ID and click **OK** at the administrative console window.




The server starts. The administrative console starts. You can access the administrative console through the browser. The administrative console accepts your login.

11. Federate the base Application Server into the cell. **To add the base Application Server into the cell:**

- **Deployment manager administrative console method:**

Click **System administration > Nodes > Add Node** and follow the wizard. The default SOAP port for the Application Server is 8880. You can use localhost as the value of the Host name field, if the Application Server is on the same machine.

- **Command-line method assuming the SOAP port of the dmgr is 8879:**

```
-   install_root/profiles/profile_name/bin/addNode.sh localhost 8879
  -includeapps
-  install_root\profiles\profile_name\bin\addNode.bat localhost 8879
  -includeapps
```

If you enable security, specify the `-user` and the `-password` parameters of the command.

12. Verify that the Application Server was incorporated into the cell. The command window displays a sequence of messages when you issue the **addNode** command:

```
Tool information is being logged in file
  C:\Program Files\IBM\WebSphere\AppServer\profiles\Profile01\logs\addNode.log
Begin federation of node AppServer01 with Deployment Manager at
  localhost:8879.
Successfully connected to Deployment Manager Server: localhost:8879
Servers found in configuration:
Server name: server1
Stopping all server processes for node AppServer01
Creating node agent configuration for node: AppServer01
Reading configuration for node agent process: nodeagent
Adding node AppServer01 configuration to cell: AdvancedDeploymentCell
Performing configuration synchronization between node and cell.
Launching node agent process for node: AppServer01
Node agent launched. Waiting for initialization status.
Node agent initialization completed successfully. Process ID is: 3012
Node AppServer01 has been successfully federated.
```

The last message is an indicator of success. A second Java process is running, which is the `nodeagent` process. The `stdout.log` file and `stderr.log` file in the `node_name` directory each contain relevant messages.

13. Resolve any IP address caching problems.

By default, the Java 2 SDK caches the IP address for the domain name service (DNS) naming lookup. After resolving the host name successfully, the IP address stays in the cache. By default, the cache entry remains forever.

This default IP caching mechanism can cause problems, as described in the following problem scenarios.

#### **Problem scenario 1**

Suppose the Application Server at `host1.ibm.com` has an initial IP address of `1.2.3.4`. When a client at `host2.ibm.com` conducts a DNS lookup of `host1.ibm.com`, the client stores the `1.2.3.4` address in the cache. Subsequent DNS name lookups return the cached value, `1.2.3.4`.

The cached value is not a problem until the `host1.ibm.com` IP address changes, to `5.6.7.8`, for example. The client at `host2.ibm.com` does not retrieve the current IP address, but always retrieves the previous address from the cache.

If this scenario occurs, the client cannot reach `host1.ibm.com` unless you stop and restart the client process.

#### **Problem scenario 2**

Suppose the Application Server at host1.ibm.com has an initial IP address of 1.2.4.5. Although the IP address of the application server does not change, a network outage can record an exception code as the IP address in the cache, where it remains until the client is restarted on a working network.

For example, if the client at host2.ibm.com disconnects from the network because of an unplugged cable, the disconnected lookup of the Application Server at host1.ibm.com fails. The failure causes the IBM Developer Kit to put the special exception code entry into the IP address cache.

Subsequent DNS name lookups return the exception code, which is `java.net.UnknownHostException`.

### **IP address caching and WebSphere Application Server process discovery**

If you change the IP address of a federated WebSphere Application Server node, processes running in other nodes cannot contact the changed node until you stop and restart them.

If a deployment manager process starts on a disconnected node, it cannot communicate with cell member processes until you stop and restart the deployment manager process. For example, plugging in an unplugged network cable does not restore proper addresses in the IP cache until the deployment manager process is restarted.

### **Using the IP address cache setting**

You can always stop and restart a deployment manager process to refresh its IP address cache. However, this process might be expensive or inappropriate.

The `networkaddress.cache.ttl` (public, JDK1.4) and `sun.net.inetaddr.ttl` (private, JDK1.3) parameters control IP caching. The value is an integer that specifies the number of seconds to cache IP addresses. The default value, -1, specifies to cache forever. A value of zero (0) is a specification to never cache.

Using a zero (0) value is not recommended for normal operation. If you do not anticipate network outages or changes in IP addresses, use the cache forever setting. The never caching setting introduces the potential for DNS spoofing attacks.

### **For more information about the Java 2 SDK**

The Java 2 SDK, Standard Edition 1.4 Web site at

<http://java.sun.com/j2se/1.4/docs/guide/net/properties.html> describes the private `sun.net.inetaddr.ttl` property, which works in both Java 2 SDK, Standard Edition 1.3 (WebSphere Application Server V5.0.0, V5.0.1, and V5.0.2) and Java 2 SDK, Standard Edition 1.4 (WebSphere Application Server V5.1 and V6).

This procedure results in using some simple procedures to debug errors that might be occurring in the installation.

The “Troubleshooting installation problems” on page 355 contains more detailed debugging and reporting instructions. See “Installation component troubleshooting tips” on page 355 for more information about troubleshooting the installation.

For current information available from IBM Support on known problems and their resolution, see the IBM Support page.

IBM Support has documents that can save you time gathering the information that you need to resolve a problem. Before opening a PMR, see the IBM Support page.

---

## **Troubleshooting Web server plug-ins installation and removal**

This topic describes troubleshooting the installation and removal of the Web server plug-ins for WebSphere Application Server.

Use this procedure if you are having problems installing or uninstalling the Web server plug-ins for WebSphere Application Server. Otherwise:

- If you are trying to install, see “Installing Web server plug-ins” on page 136.

- If you are trying to uninstall the Web server plug-ins for WebSphere Application Server, see “Uninstalling the Web server plug-ins for WebSphere Application Server” on page 200.
- If you are trying to uninstall the WebSphere Application Server, see Appendix A, “Uninstalling the product,” on page 317.

This procedure is divided into three parts:

- Troubleshooting plug-ins installation
- Troubleshooting plug-ins removal
- Miscellaneous messages, tips, and hints

## Log files for Web server plug-ins for WebSphere Application Server

The following log files are in the plug-ins `plugins_install_root/logs/install` directory:

### **log.txt**

Records all of the ISMP events that occur during the installation. The log also describes whether the installation was local or remote. Messages at the end of the file indicate whether manual configuration steps are required to complete the installation.

### **masterConfigurationLog.txt**

Records all of the configuration events that occur during the installation.

### **installGSKit.log**

Records events that occur during the installation of the GSKit code.

### **installWeb\_serverPlugin.log**

Records events that occur during the installation of a Web server plug-in. The name of the file varies to reflect the Web server:

- `installAPACHEPlugin.log`
- `installIHSPPlugin.log`
- `installIISPlugin.log`
- `installSUNONEPlugin.log`
- `installDOMINOPPlugin.log`

### **configure\_Web\_server\_webserver.log**

Records events that occur during the configuration of a Web server plug-in. The name of the file varies to reflect the Web server:

- `configure_APACHE_webserver.log`
- `configure_IHS_webserver.log`
- `configure_IIS_webserver.log`
- `configure_SUNONE_webserver.log`
- `configure_DOMINO_webserver.log`

## • Troubleshooting plug-ins installation

1. Does the installation image have the proper directories?

The following directories are required for a successful installation:

- `parent_directory/plugin`
- `parent_directory/JDK`
- `parent_directory/GSKit`

**Remember:** The `parent_directory` variable is the directory where you can unpack the images. All of the directories in the list must have the same parent directory.

If the directories are not present, or if the directories are empty, download a new installation image or discuss the product CD that you are using with someone who is knowledgeable about its creation. The IBM product discs are certified.

Reinstall using the IBM product disc.

2. Is there any sign that the installation occurred?

If not, look for the `temporaryPluginInstallLog.txt` file in the temporary directory of the user who installed the plug-ins.

For example, the `/tmp/temporaryPluginInstallLog.txt` file might exist if the root user installed the plug-ins on a Linux or UNIX system.

This log is of particular interest after a silent installation. Suppose that the silent installation `responsefile.txt` file has incorrect entries. The installation cannot succeed. The cause of the problem is recorded in the `temporaryPluginInstallLog.txt` file.

If the `responsefile.txt` does not pass validation, the failure is logged as an `INSTCONFFAILED` entry. The installation does not occur. Correct the failure and run the silent installation again.

See “`responsefile.txt`” on page 191 for information about editing the file properly. You might have to start with a new copy of the file that is on the product disc if you cannot get your copy to work.

3. Does the installation result in a `INSTCONFPARTIALSUCCESS` status?

- a. Look for errors in the `log.txt` file.
- b. Look for an entry in the `log.txt` file that shows the location of the `masterConfigurationLog.txt` file.
- c. Edit the `masterConfigurationLog.txt` file.
- d. Starting at the end of the file, scan towards the front of the file looking for configuration scripts that could not run.



For example, the following stanza shows a configuration script that could not run:

```
<record>
<date>2004-10-08T10:31:43</date>
<millis>1097245903200</millis>
<sequence>189</sequence>
<logger>com.ibm.ws.install.configmanager.ConfigManager</logger>
<level>INFO</level>
<class>com.ibm.ws.install.configmanager.ConfigManager</class>
<method>dumpNonFatalFailedActionsInfoToLogFile</method>
<thread>10</thread>
<message>This action failed to execute:
  C:\Plugins\properties\version\install\plugin\6.0.0.0\
  config\full\install\99SGSKitInstall.ant</message>
</record>
```

- e. A configuration script that fails to run is likely the cause of a partially successful installation status. To debug the example, look at the `installGSKit.log`, which is the log file for GSKit. Look for signs of a failed installation to determine if you can correct the problem.

**Configuration script log files and recovery procedures:** The following configuration scripts run during the configuration of supported Web servers.

**Configuration script:**98SConfigureWebserverDefinition.ant

Log file to investigate for error	Recovery
<p>configureWeb_server_type_webserver.log</p> <p>For example, the file might be named:</p> <p>configure_IHS_webserver.log</p> <p>Logs the Web server definition creation.</p>	<ol style="list-style-type: none"> <li>1. If the Web server definition partially exists, delete the Web server definition.</li> <li>2. Run the Web server definition script manually. The Web server definition is referenced in the script. The script is in the plug-ins_install_root/bin directory, with a name similar to the following convention: <ul style="list-style-type: none"> <li>•  Linux</li> <li>•  configureWeb_server_definition_name.sh configureWeb_server_definition_name.bat</li> </ul> </li> </ol>

**Configuration script:**99SBootStrapPluginsSunOne.ant

Log file to investigate for error	Recovery
<p>installSunOnePlugin.log</p> <p>Investigate the log file to determine what failed:</p> <ol style="list-style-type: none"> <li>1. If the error is due to plugin-cfg.xml not existing, investigate why this file is not present.</li> <li>2. If the error is due to the Web server definition creation failing, address this as discussed under the 98SConfigureWebserverDefinition.ant description.</li> <li>3. If the Web server configuration files do not update, investigate the files to verify that they exist and that the file permissions are correct.</li> </ol>	<ul style="list-style-type: none"> <li>• You can manually configure the Web server by following the procedure documented in “Configuring the Sun Java System Web Server” on page 215.</li> <li>• You can rerun the Plug-ins installation wizard. Select the existing plug-ins installation root directory. Doing so results in the Web server being configured to use the existing binaries.</li> </ul>

**Configuration script:**99SBootStrapPluginsDomino6.ant

Log file to investigate for error	Recovery
<p>installDomino6Plugin.log</p> <p>Investigate the log file to determine what failed:</p> <ol style="list-style-type: none"> <li>1. If the error is due to plugin-cfg.xml not existing, investigate why this file is not present.</li> <li>2. If the error is due to the Web server definition creation failing, address this as discussed under the 98SConfigureWebserverDefinition.ant description.</li> <li>3. If the Web server configuration files do not update, investigate the files to verify that they exist and that the file permissions are correct.</li> </ol>	<ul style="list-style-type: none"> <li>• You can manually configure the Web server by following the procedure documented in “Configuring Lotus Domino” on page 205.</li> <li>• You can rerun the Plug-ins installation wizard. Select the existing plug-ins installation root directory. Doing so results in the Web server being configured to use the existing binaries.</li> </ul>



**Configuration script:99SBootStrapPluginsDomino5.ant**

<b>Log file to investigate for error</b>	<b>Recovery</b>
<p>installDomino5Plugin.log</p> <p>Investigate the log file to determine what failed:</p> <ol style="list-style-type: none"><li>1. If the error is due to plugin-cfg.xml not existing, investigate why this file is not present.</li><li>2. If the error is due to the Web server definition creation failing, address this as discussed under the 98SConfigureWebserverDefinition.ant description.</li><li>3. If the Web server configuration files do not update, investigate the files to verify that they exist and that the file permissions are correct.</li></ol>	<ul style="list-style-type: none"><li>• You can manually configure the Web server by following the procedure documented in “Configuring Lotus Domino” on page 205.</li><li>• You can rerun the Plug-ins installation wizard. Select the existing plug-ins installation root directory. Doing so results in the Web server being configured to use the existing binaries.</li></ul>

**Configuration script:99SBootStrapPluginsIIS6.ant**

<b>Log file to investigate for error</b>	<b>Recovery</b>
<p>installIIS6Plugin.log</p> <p>Investigate the log file to determine what failed:</p> <ol style="list-style-type: none"><li>1. If the error is due to plugin-cfg.xml not existing, investigate why this file is not present.</li><li>2. If the error is due to the Web server definition creation failing, address this as discussed under the 98SConfigureWebserverDefinition.ant description.</li><li>3. Start the Internet Information Services (IIS) application and investigate the IIS configuration.</li></ol>	<ul style="list-style-type: none"><li>• You can manually configure the Web server by following the procedure documented in “Configuring Microsoft Internet Information Services (IIS)” on page 212.</li><li>• You can rerun the Plug-ins installation wizard. Select the existing plug-ins installation root directory. Doing so results in the Web server being configured to use the existing binaries.</li></ul>

**Configuration script:99SBootStrapPluginsIIS5.ant**

<b>Log file to investigate for error</b>	<b>Recovery</b>
<p>installIIS5Plugin.log</p> <p>Investigate the log file to determine what failed:</p> <ol style="list-style-type: none"><li>1. If the error is due to plugin-cfg.xml not existing, investigate why this file is not present.</li><li>2. If the error is due to the Web server definition creation failing, address this as discussed under the 98SConfigureWebserverDefinition.ant description.</li><li>3. Start the IIS application and investigate the IIS configuration.</li></ol>	<ul style="list-style-type: none"><li>• You can manually configure the Web server by following the procedure that is documented in “Configuring Microsoft Internet Information Services (IIS)” on page 212.</li><li>• You can rerun the Plug-ins installation wizard. Select the existing plug-ins installation root directory. Doing so results in the Web server being configured to use the existing binaries.</li></ul>




**Configuration script:**99SBootStrapPluginsApache.ant

Log file to investigate for error	Recovery
<p>installApachePlugin.log</p> <p>Investigate the log file to determine what failed:</p> <ol style="list-style-type: none"> <li>1. If the error is due to plugin-cfg.xml not existing, investigate why this file is not present.</li> <li>2. If the error is due to the Web server definition creation failing, address this as discussed under the 98SConfigureWebserverDefinition.ant description.</li> <li>3. If the Web server configuration files do not update, investigate the files to verify that they exist and that the file permissions are correct.</li> </ol>	<ul style="list-style-type: none"> <li>• You can manually configure the Web server by following the procedure documented in “Configuring Apache HTTP Server V2.0” on page 203.</li> <li>• You can rerun the Plug-ins installation wizard. Select the existing plug-ins installation root directory. Doing so results in the Web server being configured to use the existing binaries.</li> </ul>

**Configuration script:** 99SBootStrapPluginsIHS.ant

Log file to investigate for error	Recovery
<p>installIHSPugin.log</p> <p>Investigate the log file to determine what failed:</p> <ol style="list-style-type: none"> <li>1. If the error is due to plugin-cfg.xml not existing, investigate why this file is not present.</li> <li>2. If the error is due to the Web server definition creation failing, address this as discussed under the 98SConfigureWebserverDefinition.ant description.</li> <li>3. If the Web server configuration files do not update, investigate the files to verify that they exist and that the file permissions are correct.</li> </ol>	<ul style="list-style-type: none"> <li>• You can manually configure the Web server by following the procedure documented in “Configuring IBM HTTP Server Version 6.0” on page 210.</li> <li>• You can rerun the Plug-ins installation wizard. Select the existing plug-ins installation root directory. Doing so results in the Web server being configured to use the existing binaries.</li> </ul>

**Configuration script:**99SGSKitInstall.ant

Log file to investigate for error	Recovery
<p>installGSKit.log</p> <p>Investigate the log file to determine what failed:</p> <ol style="list-style-type: none"> <li>1. In cases where the CD layout is not correct, then the gskit installer program cannot be found. GSKit fails to install.</li> <li>2. In cases where GSKit is already installed, the installation might incorrectly report a failure. If the logs show that an installation already existed, you can safely ignore the error.</li> </ol>	<p>Manually install GSKit by running the installer program. The installGSKit.log file shows the installer program that runs to install GSKit.</p> <p>  Run the following command from the /cdrom/GSKit directory:</p> <ul style="list-style-type: none"> <li>• /cdrom/GSKit/gskit.sh</li> </ul> <p> Run the following command from the "E:\GSKit\ directory, assuming the E drive is the CD-ROM drive:</p> <ul style="list-style-type: none"> <li>• "E:\GSKit\Setup.exe" WASPLUGIN60_041008085014 "C:\Program Files" -s -z -f1"E:\GSKit\setup.iss"</li> </ul>

**Solaris** Configuration script: 99SSolarisGSKitInstall4.ant

Log file to investigate for error	Recovery
<p>On Solaris, the installation is logged to the installGSKit.log file.</p> <p>Investigate the log file to determine what failed:</p> <ol style="list-style-type: none"> <li>1. In cases where the CD layout is not correct, then the gskit installer program cannot be found. GSKit fails to install.</li> <li>2. In cases where GSKit is already installed, the installation might incorrectly report a failure. If the logs show that an installation already existed, you can safely ignore the error.</li> </ol>	<p>Manually install GSKit by running the installer program.</p> <p><b>Solaris</b> Run the following command from the /cdrom/GSKit directory:</p> <ul style="list-style-type: none"> <li>• /cdrom/GSKit/gskit4.sh</li> </ul>

**Windows** Configuration script:90SCreateWinRegPlugin.ant

Log file to investigate for error	Recovery
<p>This script does not have an associated log file.</p> <p>The script creates the Windows registry entry for the Web server plug-ins for WebSphere Application Server.</p> <p>HKEY_LOCAL_MACHINE &gt; Software &gt; IBM &gt; Web server Plug-ins for IBM WebSphere Application Server &gt; 6.0.0.0</p>	<p>Add the HKEY_LOCAL_MACHINE &gt; Software &gt; IBM &gt; Web server Plug-ins for IBM WebSphere Application Server &gt; 6.0.0.0 key manually with the actual value of the installation location.</p> <p>Set the following registry values under this key:</p> <ul style="list-style-type: none"> <li>• Name: <i>plug-ins_install_root</i></li> <li>• Type: REG_SZ</li> <li>• Data: <i>plug-ins_install_root</i></li> </ul>

Configuration script:99SModifySetupCmdLine.ant



Log file to investigate for error	Recovery
<p>This script does not have an associated log file.</p> <p>Investigate the <i>plug-ins_install_root/bin/setupCmdLine.sh</i> script to verify that the PLUGIN_HOME variable is set to the Web server plug-ins installation root directory.</p> <p>Verify that file permissions are 755 (rwxr-xr-x) on Linux and UNIX systems.</p>	<ul style="list-style-type: none"> <li>• You can edit the setupCmdLine file, replacing \$(PLUGIN_HOME) with the location to the plug-ins installation root directory.</li> <li>• Change the file permissions to 755. Issue the following command from the <i>plug-ins_install_root/bin</i> directory: chmod 755 setupCmdLine.sh</li> </ul>

If you do not find the cause of the problem, open the *plug-ins\_install\_root/logs/Web\_server\_name* directory. The directory is empty until the Web server loads the plug-in and errors occur. The plug-in then creates the *http\_plugin.log* file to record the errors. If no errors occur, the directory is empty. Examine any relevant files for error entries. Correct any errors and reinstall.

4. Does the installation result in a INSTCONFFAILED status?
  - a. This is a serious failure of the installation.
  - b. Analyze the log files to determine the problem.
  - c. Uninstall the plug-ins.
  - d. Delete the plug-ins installation root directory.
  - e. Install the plug-ins again.
5. Do errors occur when you start the snoop application?  
See "Start the Snoop servlet" in the installation troubleshooting file.

If errors occur, look at the `plug-ins_install_root/logs/Web_server_name/http_plugin.log` file for causes.

Also investigate the WebSphere Application Server installation logs and the logs for your supported Web server.

6.   Do you have the correct file permissions for the `configureWeb_server_definition_name` script?

On Linux and UNIX systems, copying the `configureWeb_server_definition_name.sh` script from one operating system, such as AIX, to another, such as HP-UX, can cause the file permissions of the script to be invalid for running the script.

Verify that file permissions are 755 (rwxr-xr-x) on Linux and UNIX systems.

- **Troubleshooting plug-ins removal**

1. Does the `plug-ins_install_root/logs/uninstall` directory have any log files?

If log files exist, examine them, correct the errors, and reinstall.

2. Is there any sign that the removal occurred?

If not, look for the `temporaryPluginUninstallLog.txt` file in the temporary directory of the user who installed the plug-ins.

For example, the `/tmp/temporaryPluginUninstallLog.txt` file might exist if the root user uninstalled the plug-ins on a Linux or UNIX system.

3. Does the installation result in a `INSTCONFPARTIALSUCCESS` status?

- a. Look for errors in the `log.txt` file.

- b. Look for an entry in the `log.txt` file that shows the location of the `masterConfigurationLog.txt` file.

- c. Edit the `masterConfigurationLog.txt` file.

- d. Starting at the end of the file, scan towards the front of the file looking for configuration scripts that could not run.

For example, the following stanza shows a configuration script that could not run:

```
<record>
<date>2004-10-08T10:31:43</date>
<millis>1097245903200</millis>
<sequence>189</sequence>
<logger>com.ibm.ws.install.configmanager.ConfigManager</logger>
<level>INFO</level>
<class>com.ibm.ws.install.configmanager.ConfigManager</class>
<method>dumpNonFatalFailedActionsInfoToLogFile</method>
<thread>10</thread>
<message>This action failed to execute:
  C:\Plugins\properties\version\install\plugin\6.0.0.0\
  config\full\uninstall\99SGSKitUnInstall.ant</message>
</record>
```





- e. A script that fails to run is likely the cause of a partially successful installation status. To debug our example, look at the `installGSKit.log`, which is the log file for GSKit. Look for signs of a failed removal to determine if you can correct the problem.

Configuration script	Log file to investigate for error
99SBootStrapPluginsSunOneUninstall <i>uniqueID</i> .ant	uninstallWeb_server_typePlugin.log
99SBootStrapPluginsDomino6Uninstall <i>uniqueID</i> .ant	For example, the file might be named:
99SBootStrapPluginsDomino5Uninstall <i>uniqueID</i> .ant	uninstallIHSPugin.log
99SBootStrapPluginsIIS6Uninstall <i>uniqueID</i> .ant	Logs the Web server deconfiguration events for the Web server.
99SBootStrapPluginsIIS5Uninstall <i>uniqueID</i> .ant	
99SBootStrapPluginsApacheUninstall <i>uniqueID</i> .ant	
99SBootStrapPluginsIHSUninstall <i>uniqueID</i> .ant	
99SGSKitUnInstall.ant	uninstallGSKit.log  On Windows only, the Web server plugins GSKit key will be unregistered. GSKit is uninstalled if no other product is registered to use GSKit.  On Linux and UNIX, it is your responsibility to uninstall GSKit when no other products are using it.
90SDeleteWinRegPlugin.ant	This script does not have an associated log file.  The script deletes the Windows registry entry for the Web server plug-ins for WebSphere Application Server.  HKEY_LOCAL_MACHINE > Software > IBM > Web server Plug-ins for IBM WebSphere Application Server > 6.0.0.0

#### 4. Did the uninstall procedure fail?

If the uninstall failed in any way, follow the manual uninstall steps to verify the system is clean and if necessary, remove Web server plug-in entries.

#### • Miscellaneous messages, tips and hints

Option	Description
<b>SAFE_BROWSER_EXCEPTION_CAUGHT</b>	This SAFE_BROWSER_EXCEPTION_CAUGHT error means that: <ul style="list-style-type: none"> <li>If you do not have a browser, this error is thrown and the roadmap is not launched when it is selected during the installation of the plug-ins.</li> <li>  If you have a supported browser that is not the Konqueror browser, the roadmap is launched. This message is thrown with the supported browser when it is not the Konqueror browser. You can safely ignore this exception and open the roadmap in a supported browser.</li> </ul>
 <b>number.tmp.sorted</b>	 A limitation in ISMP does not remove temporary working files in the / root directory on Solaris operating systems. You can safely ignore or delete the files that follow this naming convention.

This procedure results in troubleshooting the installation or removal of the Web server plug-ins for WebSphere Application Server.

For current information available from IBM Support on known problems and their resolution, see the IBM Support page.

IBM Support has documents that can save you time gathering the information that you need to resolve a problem. Before opening a PMR, see the IBM Support page.

---

## Installation component troubleshooting tips

If you are having problems installing your WebSphere Application Server product, follow these steps to resolve the problem:

- If possible, follow the steps that are outlined in Appendix B, “Troubleshooting installation,” on page 337.
- Browse the relevant log files for clues:
  - The main installation log file: *install\_root/logs/log.txt*.
  - The Profile creation wizard log file: *install\_root/profiles/profile\_name/logs/pctLog.txt* when you create a profile with the Profile creation wizard.
  - The profile creation log for creating the profile:  
*install\_root/logs/wasprofile/wasprofile\_create\_profile\_name.log*.
  - IBM HTTP Server log: *ihp\_install\_root/log.txt* and *ihp\_install\_root/ihsv6\_install.log*.
  - The log files produced when the default application .ear file is installed are:  
*install\_root/profiles/profile\_name/logs/defaultapp\_config.log* and  
*install\_root/profiles/profile\_name/logs/defaultapp\_deploy.log*.
  - Other logs for other applications for a profile in the *install\_root/profiles/profile\_name/logs* directory.
- Verify that you have installed the correct level of dependent software, such as operating system version and revision level, by reviewing <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>.

### If none of these steps solves the problem:

Check to see if the problem is identified and documented using the links in Diagnosing and fixing problems: Resources for learning.

If you do not see a problem that resembles yours, or if the information provided does not solve your problem, contact IBM support for further assistance.

For current information available from IBM Support on known problems and their resolution, see the IBM Support page.

IBM Support has documents that can save you time gathering information needed to resolve this problem. Before opening a PMR, see the IBM Support page.

---

## Troubleshooting installation problems

Select the problem that you are having with the WebSphere Application Server installation:

- I need information about Troubleshooting Web server plug-ins installation and removal.
- “Installation either completes with errors or warnings, or hangs” on page 356.
- There are “Messages issued during installation and profile creation” on page 357

If you do not see a problem that resembles yours, or if the information provided does not solve your problem, see Appendix B, “Troubleshooting installation,” on page 337.

If you did not find your problem listed, contact IBM support.

For current information available from IBM Support on known problems and their resolution, see the IBM Support page.

IBM Support has documents that can save you time gathering information needed to resolve this problem. Before opening a PMR, see the IBM Support page.

---

## Installation either completes with errors or warnings, or hangs

### Errors while installing the product

If the WebSphere Application Server installation program indicates that errors were encountered while installing the product:

- Browse the log files in the *install\_root/logs* directory and in the *install\_root/profiles/profile\_name/logs* directory for clues. Pay particular attention to the main installation log file, *log.txt*.
- Check the command prompt from which the installation panel was launched for error messages.
- Look up any error or warning messages in the message reference table by selecting the "Reference" view in the information center navigation and expanding the "Messages" heading.
- Review Appendix B, "Troubleshooting installation," on page 337.

---

## Problems installing or starting Apache or IBM HTTP Server

What kind of error are you seeing?

- "Failed to initialize GSK environment" error appears when trying to start Apache
- IBM HTTP Server (IHS) on Windows does not start, error `WSADuplicateSocket` failed for socket appears in `error.log`.

If you did not find your problem listed, contact IBM support.

### "Failed to initialize GSK environment" error appears when trying to start Apache

The following messages may appear in Apache's error log file when you attempt to start Apache:

```
[Wed Aug 27 19:21:59 2003] [error] lib_security:
  initializeSecurity: Failed to initialize GSK environment
[Wed Aug 27 19:21:59 2003] [error] ws_transport:
  transportInitializeSecurity: Failed to initialize security
[Wed Aug 27 19:21:59 2003] [error] ws_server:
  serverAddTransport: Failed to initialize security
```

These messages are most common after building Apache from the downloadable source code and installing Apache separately from WebSphere Application Server.

If SSL Transport (HTTPS) is defined in the `plugin-cfg.xml` file, the plug-in will try to load the Global Security Kit (GSKit) library `libgsk7ssl.so`. The error messages appear in the error log file if the level 7 GSKit `libgsk7ssl.so` is not installed on your system. To correct these errors, try one of the following:

- Remove the transport section that has "https" in the `plugin-cfg.xml` file
- Install the level 7 GSKit

**Note:** The GSKit installs automatically if you install either the IBM HTTP Server (IHS) product or the Web server plug-ins for WebSphere Application Server from the product disc.

For details and help on the Apache Web server, visit the Apache Web site <http://httpd.apache.org/>.

### IBM HTTP Server on Windows does not start, error `WSADuplicateSocket` failed for socket appears in `error.log`

If your installation of IBM HTTP Server (IHS) on a Windows platform will not start, you may find a message similar to the following in the `log\error.log` file of your IHS installation:



[crit] (10045)

The attempted operation is not supported for the type of object referenced:  
Parent: WSADuplicateSocket failed for socket nnnn

This problem occurs when IHS is run on a system with a Virtual Private Networking client such as Aventail Connect. Aventail Connect is a Layered Service Provider (LSP) that intercepts calls between the Winsock 2 API and Window's native Winsock 2 implementation. The failure occurs because Aventail Connect does not implement `WSADuplicateSocket()`. The interception still takes place when Aventail Connect is shut down.

To correct the problem try one of the following:

- Reboot the machine, or
- Uninstall the LSP product

For current information available from IBM Support on known problems and their resolution, see the IBM Support page.

IBM Support has documents that can save you time gathering information needed to resolve this problem. Before opening a PMR, see the IBM Support page.

---

## Messages issued during installation and profile creation


What kind of problem are you having?

- "Error: The input line is too long"
- "Note: The following file systems will be expanded during the installation" on page 358
- "The disk space is nn Mbyte less than required. The installation cannot continue." on page 358
- "Specify a different directory or perform a manual uninstall" on page 358
- "Error: /usr/opt/ibm/gskta/bin/gsk7ikm: not found" on page 359
- "Error writing file = There may not be enough temporary disk space." on page 359
- "Error: localhost is not a valid host name for remote access" on page 359
- "INFO: Created system preferences directory in java.home" on page 360
- "The installer could not successfully add the product information into the RPM database." on page 360
- "Error: java.io.IOException: konqueror: not found" on page 361

If you did not find your problem listed, contact IBM support.

### Error: The input line is too long

This is a profile creation error.

 The installation directory path must be no longer than 60 characters.

- **Option 1:** If the installation failed, reinstall the WebSphere Application Server product using a shorter directory path and a shorter node name.

The node name field is available during a custom installation. The length of the default node name is usually short enough. Avoid using a node name that is longer than 10 characters if possible.

- **Option 2:** If the installation was successful but the `ISMPCWSPprofileLaunchAction` failed, use the Profile creation wizard to create the profile. Use a shorter profile directory path and a shorter node name when creating the profile.

You can select your own profiles path. It could be `C:\profiles`, for example.

Verify from the `log.txt` file that the `ISMPCConfigManagerLaunchAction` is successful.

Examine the following messages in the `log.txt` file to determine whether the installation was successful. If so, you can run the Profile creation wizard to create the default profile.

```
(date time), Install,  
com.ibm.ws.install.ni.ismp.actions.ISMPConfigManagerLaunchAction,  
msg1, INSTCONFSUCCESS: Post-installation configuration is successful.
```


```
(date time), Install,  
com.ibm.ws.install.ni.ismp.actions.ISMPWSProfileLaunchAction,  
err, INSTCONFFAILED: Cannot complete required configuration actions  
after the installation. The configuration failed. The installation is  
not successful. Refer to  
C:\Program Files\IBM\WebSphere\AppServer1234567890\logs\wasprofile\  
wasprofile_create_default.log for more details.
```

```
(date time), Install,  
com.ibm.ws.install.ni.ismp.actions.ISMPLogFileAction,  
msg1, Profile Creation Tool (PCT) is executed.
```

```
(date time), Install,  
com.ibm.ws.install.ni.ismp.actions.ISMPLogSuccessMessageAction,  
msg1, INSTCONFFAILED
```

The following error within a log in the *install\_root*\profiles\default\logs directory indicates that a wsadmin action has failed to create a profile. The failure is because the length of the file path, the cell name, and the node name on the command string has caused the entire command to exceed the operating system limit for command length.

The input line is too long.

 The profiles directory path must be no longer than 80 characters.

If your directory path is 80 characters or less and you still have a problem, edit the *install\_root*\bin\setupCmdLine.bat file to make it use the Windows **subst** command. The **subst** command maps an entire path to a virtual drive. After editing the file, run the Profile creation wizard again. If changing the setupCmdLine.bat file does not fix the problem, you can install the WebSphere Application Server product using a shorter installation path, such as C:\WAS.

See "The input line is too long" section of Problems starting or using the wsadmin command for a description of how to edit the setupCmdLine.bat file.

## Note: The following file systems will be expanded during the installation

You can allocate expansion space for directories on AIX. If the Installation wizard does not have enough space, InstallShield for MultiPlatforms (ISMP) issues a system call for more space that increases the space allocation dynamically. The message you might see when this occurs for the /usr directory is similar to the following example:

```
NOTE: The following file systems will be expanded during the installation:  
/usr
```

## The disk space is nn Mbyte less than required. The installation cannot continue.

If the file system is not dynamically expandable, an insufficient amount of disk space results in a message that is similar to the following example:

```
The disk space is 33 Mbyte less than required. The installation cannot continue.
```

## Specify a different directory or perform a manual uninstall

This error indicates that you deleted the installation root directory before using the uninstaller program to remove the product. Now you attempting to reinstall into the same directory.

To correct the problem, perform a manual uninstall. See “Uninstalling manually” on page 325 for an introduction to manually uninstalling. The topic has a link to the manual uninstall procedure for your operating system.

The warning entry in the \$TMP/log.txt file is:

```
(Month day, year time), Install,  
com.ibm.ws.install.ni.ismp.actions.ISMPWarningDialogAction,  
wrn, Specify a different directory or perform a manual uninstall  
to remove all packages before reinstalling to the same directory.
```

## **Error: /usr/opt/ibm/gskta/bin/gsk7ikm: not found**

This error indicates that the x1C.rte 6.0 run-time code has not been installed. You must install the x1C.rte 6.0 run-time code before you install Global Security Kit 7 (GSKit7).

To correct this problem, download the run-time code from the AIX Support site at <https://techsupport.services.ibm.com/server/aix.fdc>. If you have AIX 5.2, you can install the x1C.rte 6.0 run-time code from the AIX 5.2 CD.

For a complete list of installation prerequisites, see <http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>.

## **Error writing file = There may not be enough temporary disk space.**

```
Searching for Java(tm) Virtual Machine...  
A suitable JVM could not be found.  
Please run the program again using the option  
-is:javahome <JAVA HOME DIR>  
Error writing file = There may not be enough temporary disk space.  
Try using -is:tempdir to use a temporary directory on a partition with more disk space.
```

This error can occur when you have not provided enough temp space to create a profile. Verify that you have a minimum of 40 MB of temp space available before creating a profile.

See “Using the Profile creation wizard” on page 234 for more information about creating profiles.

## **Error: localhost is not a valid host name for remote access**

This error occurs when you enter localhost as the value for the Host name field in the Profile creation wizard.

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 to communicate with this node. Selecting a host name that other machines can reach within your network is extremely important. Do not use the generic localhost identifier for this value.

If you define coexisting nodes on the same computer with unique IP addresses, 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 servers (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 advantage of being totally unambiguous and also 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 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 format disadvantage is being dependent 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 ability of being redefined in the local hosts file so that the system can run the Application Server even when disconnected from the network. Define the short name to 127.0.0.1 (local loopback) in the hosts file to run disconnected. A format disadvantage is being dependent 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 you name with a numeric IP address without DNS being available. A format disadvantage 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 format disadvantage is that you cannot use the node if the host is disconnected from the network.

### **Warning: Cannot convert string "<type\_name>" to type FontStruct**

If you install the Web server plug-ins for WebSphere Application Server, you also install the `ikeyman` utility. The `ikeyman` utility is part of the Global Services Kit 7 (GSKit7).

If you issue the `ikeyman.sh` script on a Linux system, you might see the following message:

```
Warning: Cannot convert string
"-monotype-arial-regular-r-normal---140---p---iso8859-1"
to type FontStruct
```

You can safely ignore the warning and use the `ikeyman` utility.

### **INFO: Created system preferences directory in java.home**

The Key Man Utility (`ikeyman`) on Linux Redhat Enterprise 3.0 displays the following message after your first use of the utility.

```
[root@benson12 bin]# ./ikeyman.sh
Oct 19, 2004 10:47:26 AM java.util.prefs.FileSystemPreferences$3 run
INFO: Created system preferences directory in java.home.
```

The utility is logging preferences. The message does not display when you use the utility again.

You can safely ignore the message.

### **The installer could not successfully add the product information into the RPM database.**

If the last line in the `log.txt` file is something similar to the following example, the problem might be a corrupt RedHat Package Manager (RPM) database:

```
(Oct 17, 2004 4:02:16 PM),
Plugin.Install,
com.ibm.wizard.platform.linux.LinuxProductServiceImpl,
wrn, The installer could not successfully add the product
information into the RPM database. Installation will continue
as this is not critical to the installation of the product.
```

Run the following command to verify that the problem is a corrupt RPM database:

```
rpm -q --all
```

If the command hangs, the problem is a corrupt RPM database.

## **Error: java.io.IOException: konqueror: not found**

The ISMP Launch Browser action throws the following exception on Linux and UNIX systems:

```
com.installshield.wizardx.actions.LaunchBrowserAction, err, java.io.IOException:
konqueror: not found
STACK_TRACE: 11
java.io.IOException: konqueror: not found
    at java.lang.UNIXProcess.forkAndExec(Native Method)
    at java.lang.UNIXProcess.<init>(UNIXProcess.java:72)
    at java.lang.Runtime.execInternal(Native Method)
    at java.lang.Runtime.exec(Runtime.java:602)
    at java.lang.Runtime.exec(Runtime.java:524)
    at java.lang.Runtime.exec(Runtime.java:490)
    at com.installshield.util.BrowserLauncher.openURL(BrowserLauncher.java:578)
    at com.installshield.wizardx.actions.LaunchBrowserAction.execute(LaunchBrowserAction
.java:62)
    at com.installshield.wizard.RunnableWizardBeanContext.run(RunnableWizardBeanContext.
java:21)
```

This action searches for a Netscape, Mozilla, or Konqueror browser for displaying an HTML page or a Web site. In the case of the Web server plug-ins for WebSphere Application Server, the target page is the Plug-ins roadmap. Even though the ISMP Launch Browser action might find Mozilla or Netscape, the exception is thrown and a log entry occurs.

You can safely ignore this error.



---

## Appendix C. 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 Licensing  
IBM Corporation  
500 Columbus Avenue  
Thornwood, New York 10594 USA





---

## **Appendix D. Trademarks and service marks**

For trademark attribution, visit the IBM Terms of Use Web site (<http://www.ibm.com/legal/us/>).