Servers, nodes and node agents, cells and the deployment manager are fundamental concepts in the administrative universe of the product. It is also important to understand the various processes in the administrative topology and the operating environment in which they apply.

For more information, refer to “Welcome to basic administrative architecture” on page 6.

- **Scripting**

  The WebSphere administrative (wsadmin) scripting program is a powerful, non-graphical command interpreter environment enabling you to run administrative operations in a scripting language. You can also submit scripting language programs to run. The wsadmin tool is intended for production environments and unattended operations.

  For more information, refer to “Introduction: Administrative scripting (wsadmin).”
- **Commands**

  Command-line tools are simple programs that you run from an operating system command-line prompt to perform specific tasks, as opposed to general purpose administration. Using the tools, you can start and stop application servers, check server status, add or remove nodes, and complete similar tasks.

  For more information, refer to “Introduction: Administrative commands” on page 5.
- **Programming**

  The product supports a Java programming interface for developing administrative programs. All of the administrative tools supplied with the product are written according to the API, which is based on the industry standard Java Management Extensions (JMX) specification.

  For more information, refer to “Introduction: Administrative programs” on page 5.
- **Data**

  Product configuration data resides in XML files that are manipulated by the previously-mentioned administrative tools.

  For more information, refer to “Introduction: Administrative configuration data” on page 6.

## Introduction: Administrative console

The administrative console is a graphical interface for performing deployment and system administration tasks. It runs in your Web browser. Your actions in the console modify a set of XML configuration files.

You can use the console to perform tasks such as:

- Add, delete, start, and stop application servers
- Deploy new applications to a server
- Start and stop existing applications, and change application configurations
- Add and delete Java 2 Platform, Enterprise Edition (J2EE) resource providers for applications that require data access, mail, URLs, and so on
- Manage variables, shared libraries, and other configurations that can span multiple application servers
- Configure product security, including access to the administrative console
- Collect data for performance and troubleshooting purposes
- Find the product version information. It is located on the front page of the console.

See the *Using the administrative clients* PDF for information on how you begin using the console. See also the **Reference > Administrator > Settings** section of the Information Center navigation. It lists the settings or properties you can configure.

## Introduction: Administrative scripting (wsadmin)

The WebSphere administrative (wsadmin) scripting program is a powerful, non-graphical command interpreter environment enabling you to run administrative operations in a scripting language. The wsadmin tool is intended for production environments and unattended operations. You can use the wsadmin tool to perform the same tasks that you can perform using the administrative console.

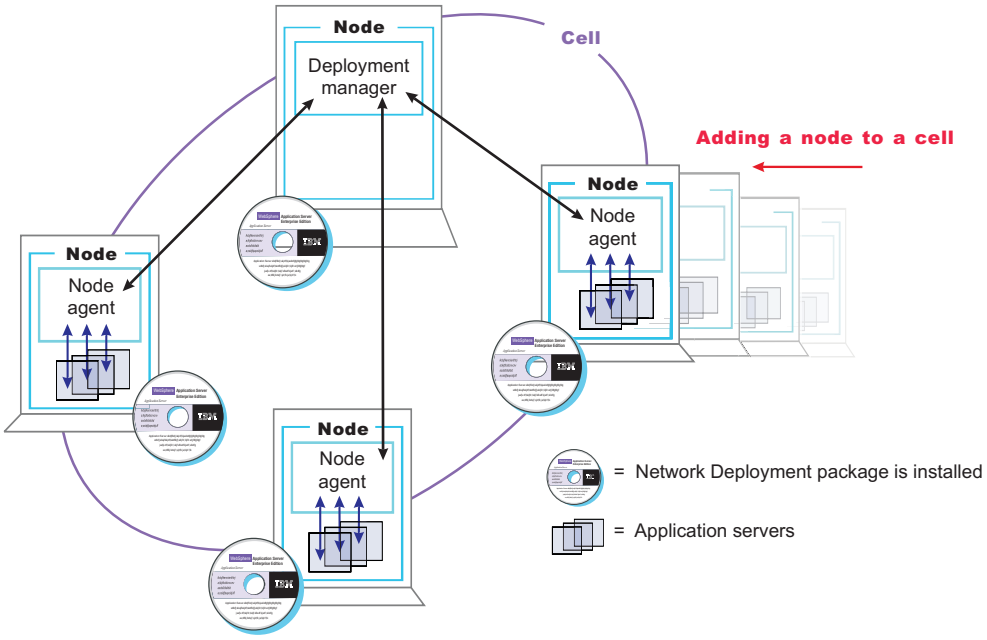
**Application Server:** A WebSphere Application Server provides the functions that are required to support and host user applications. An application server runs on only one node, but one node can support many application servers.

**Node agent:** When a node is federated, a node agent is created and installed on that node. The node agent works with the deployment manager to perform administrative activities on the node.

**Deployment manager:** With the deployment manager, you can administer multiple application servers from one centralized manager. The deployment manager works with the node agent on each node to manage all the servers in a distributed topology.

The following diagram depicts the concepts that are discussed in this article.

**IBM WebSphere Application Server Network Deployment package**




The concepts that are discussed in this article form the basis of WebSphere Application Server administration. More detailed descriptions can be found in other sections.

# Introduction: Servers

### Application servers

Application servers provide the core functionality of the WebSphere Application Server product family. They extend the ability of a Web server to handle Web application requests, and much more. An application server enables a server to generate a dynamic, customized response to a client request.

For additional overview, refer to “Introduction: Application servers” on page 8.

### Clusters

*Workload management* optimizes the distribution of client processing tasks. Incoming work requests are distributed to the application servers that can most effectively process the requests. Workload management also provides failover when servers are not available, improving application availability.

Specifies the name of the node for the deployment manager. The default is the name of the host computer on which the deployment manager is installed with `CellManager##` appended, where *##* is a two-digit number.

| | |
|---|---|
| **Data type** | String |
| **Default** | *host_name*CellManager01 |

***State:***

Indicates the state of the deployment manager. The state is *Started* when the deployment manager is running and *Stopped* when it is not running.

| | |
|---|---|
| **Data type** | String |
| **Default** | Started |

## Node

A node is a logical grouping of managed servers.

A node usually corresponds to a logical or physical computer system with a distinct IP host address. Nodes cannot span multiple computers. Node names usually are identical to the host name for the computer.

Nodes in the network deployment topology can be managed or unmanaged. Two types of managed nodes exist while one type of unmanaged node exists.

One type of managed node has a node agent which manages all servers on a node, whether the servers are WebSphere Application Servers, Java Message Service (JMS) servers (on Version 5 nodes only), Web servers, or generic servers. The node agent represents the node in the management cell. A deployment manager manages this type of managed node. The other type of managed node has no node agent. This type of managed node is defined on a standalone Application Server. The deployment manager cannot manage this standalone Application Server. An standalone Application Server can be federated. When it is federated, a node agent is automatically created. The node becomes a managed node in the cell. The deployment manager manages this node.

An unmanaged node does not have a node agent to manage its servers. Unmanaged nodes can have server definitions such as Web servers in the WebSphere Application Server topology. Unmanaged nodes can never be federated. That is, a node agent can never be added to an unmanaged node.

A supported Web server can be on a managed node or an unmanaged node. You can define only one Web server to a standalone WebSphere Application Server node. This Web server is defined on an unmanaged node. You can define Web servers to the deployment manager. These Web servers can be defined on managed or unmanaged nodes.

WebSphere Application Server supports basic administrative functions for all supported Web servers. For example, generation of a plug-in configuration can be performed for all Web servers. However, propagation of a plug-in configuration to remote Web servers is supported only for IBM HTTP Servers that are defined on an unmanaged node. If the Web server is defined on a managed node, propagation of the plug-in configuration is done for all the Web servers by using node synchronization. The Web server plug-in configuration file is created according to the Web server definition and is created based on the list of applications that are deployed on the Web server. You can also map all supported Web servers as potential targets for the modules during application deployment.

# Chapter 5. Starting and stopping quick reference

This topic describes how to start and stop the main operations in your application serving environment. It also provides a quick guide to accessing the main tools that are provided with this product.

- Use commands to start and stop servers.

| **Quick reference: Issuing commands to start and stop servers** |
| --- |
| These examples are for starting and stopping the default profile on a server. Otherwise, you might need to be in the `install_root/profiles/profile_name/bin`directory to start and stop the server. |
| **Deployment manager**<br><br>Go to the *install_root* `/bin`directory of a Network Deployment installation and run the following command. See "startManager command" on page 855 for details and variations<br>`startManager` |
| **Node**<br><br>Go to the *install_root* `/bin`directory of a WebSphere Application Server installation and run the following command. See "startNode command" on page 858 for details and variations<br>`startNode` |
| **Application server**<br><br>Go to the *install_root* /bin directory of a WebSphere Application Server or Network Deployment installation and run the following command. See "startServer command" on page 853 for details and variations<br>`startServer server`<br><br>where *server* is the application server that you want to start. |
| **Stopping the servers**<br><br>Use the same command as to start, except substitute stop for start. For example, to stop an application server, issue the command:<br>`stopServer server`<br><br>To start and stop application server clusters, see "Starting clusters" on page 282. |

- Use administrative clients and tools.

| **Quick reference: Opening the administrative console** |
| --- |
| To open the console, enter this Web address in your Web browser:<br>`http://your_fully_qualified_server_name:9060/ibm/console`<br><br>Depending on your configuration, your Web address might differ. Other factors can affect your ability to access the console. See "Starting and stopping the administrative console" on page 349 for details, as needed. |

  - To launch a scripting client, see "Starting the wsadmin scripting client" on page 429.
  - To learn about all available administrative clients, see Chapter 4, "Using the administrative clients," on page 349.
  - For performance monitoring, see ″Monitoring performance with Tivoli Performance Viewer (TPV)″ in the information center.

  See the administrator commands that 1 e **Reference** section of the information center.
- Use development and deployment tools. Use the following tools to edit deployment descriptors. A deployment descriptor is an Extensible Markup Language (XML) file that describes how to deploy a module or application by specifying configuration and container options. The tools are available for use on distributed operating systems.

# Installable module versions

The contents of a module affect whether you can install the module on a WebSphere Application Server Version 6.0 and later (6.x) deployment target, or must install the module on a Version 5.0 and later (5.x) deployment target.

### Installable application modules

You can install an application, enterprise bean (EJB) module or Web module developed for a Version 5.x product on a 5.x or 6.x deployment target, provided the module--

- Does not support Java 2 Platform, Enterprise Edition (J2EE) 1.4;
- Does not call any 6.x runtime application programming interfaces (APIs); and
- Does not use any 6.x product features.

If the module supports J2EE 1.4, calls a 6.x API or uses a 6.x feature, then you must install the module on a 6.x deployment target.

1

Selecting options such as **Pre-compile JSP**, **Use Binary Configuration**, **Deploy Web services** or **Deploy enterprise beans** during application installation to a 6.x server or a 6.x deployment manager indicates that the application uses 6.x product features. You cannot deploy such applications on a 5.x deployment target. You must deploy such applications on a 6.x deployment target.

Similarly, you must deploy an application that uses J2EE 1.4 features such as Java Authorization Contract for Containers (JACC) provided by an application server on a 6.x deployment target.

### Installable RAR files

You can install a standalone resource adapter (connector) module, or RAR file, developed for a Version 5.x product to a 5.x or 6.x deployment target, provided the module does not call any 6.x runtime APIs. If the module calls a 6.x API, then you must install the module on a 6.x deployment target.

### Deployment targets

A *5.x deployment target* is a server or a cluster with at least one member on a WebSphere Application Server Version 5 product.

A *6.x deployment target* is a server or cluster with all members on a WebSphere Application Server Version 6 product.

*Table 14. Compatible deployment target versions for 5.x and 6.x modules*

| Module type | Module Java support | Module calls 6.x runtime APIs or uses 6.x features? | Client versions that can install module | Deployment target versions |
|---|---|---|---|---|
| Application, EJB, Web, or client | J2EE 1.3 | No | 5.x or 6.x | 5.x or 6.x |
| Application, EJB, Web, or client | J2EE 1.3 | Yes | 6.x | 6.x |
| Application, EJB, Web, or client | J2EE 1.4 | Yes or No | 6.x | 6.x |
| Resource adapter | JCA 1.0 | No | 5.x or 6.x | 5.x or 6.x |
| Resource adapter | JCA 1.0 | Yes | 6.x | 6.x |
| Resource adapter | JCA 1.5 | Yes or No | 6.x | 6.x |

# Chapter 8. Developing 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. Also use this section to learn about developing WebSphere applications.

See Learn about WebSphere applications: Overview and new features for an introduction to each technology.

| | | | | |
|---|---|---|---|---|
| Web applications | How do I?... | Overview | | Samples |
| EJB applications | How do I?... | Overview | Tutorials | Samples |
| Client applications | How do I?... | Overview | | Samples |
| Web services | How do I?... | Overview | Tutorials | Samples |
| Data access resources | How do I?... | Overview | Tutorials | Samples |
| Messaging resources | How do I?... | Overview | Tutorials | Samples |
| Mail, URLs, and other J2EE resources | How do I?... | Overview | | |
| Security | See the *Securing WebSphere applications* PDF | See the *Administering applications* PDF | See the *Securing WebSphere applications* PDF | See the *Securing WebSphere applications* PDF |
| Naming and directory | How do I?... | Overview | | |
| Object Request Broker | How do I?... | Overview | | |
| Transactions | How do I?... | Overview | | Samples |
| ActivitySessions | How do I?... | Overview | | Samples |
| Application profiling | How do I?... | Overview | | Samples |
| Asynchronous beans | How do I?... | Overview | | Samples |
| Dynamic caching | How do I?... | Overview | | |
| Dynamic query | How do I?... | Overview | | Samples |
| Internationalization | How do I?... | Overview | | Samples |
| Object pools | How do I?... | Overview | | |
| Scheduler | How do I?... | Overview | | Samples |
| Startup beans | How do I?... | Overview | | |
| Work areas | How do I?... | Overview | | |

## Web applications 1

## Task overview: Developing and deploying Web applications

A developer creates the files comprising a Web application, and then assembles the Web application components into a Web module. Next, the deployer (typically the developer in a unit-testing environment or the administrator in a production environment) installs the Web application on the server.

1. **(Optional)** Migrate existing Web applications to run in the new version of WebSphere.
2. Design the Web application and develop its code artifacts: Servlets, JavaServer Pages (JSP) files, and static files, as for example, images and Hyper Text Markup Language (HTML) files. See the ″Resources for learning″ article for links to design documentation.