"Concepts" table of contents

0.0: Using the InfoCenter to view WebSphere Application Server documentation

The IBM WebSphere Application Server InfoCenter provides central navigation and search of the entire product documentation set.

Though you will probably find it easy enough to learn about the InfoCenter simply by using it, this article describes how to get the most from the WebSphere Application Server part of the WebSphere InfoCenter. See also Using the WebSphere InfoCenter.

Select a goal:

- I want to find all of the available information
- I want to update the information installed with the product
- I want a summary of what has changed when I update my InfoCenter
- I want to print the InfoCenter content and take it with me
- I am going to be working offline a lot, but need access to information
- I want to search the InfoCenter content
- I want to find my way around quickly
- I want to keep track of what I am reading, and have already read
- I want to read less and skim more!
- I need operating system and prerequisites information
- Troubleshooting the InfoCenter and GUI help systems

I want to find all of the available information

Key product Web sites are documented in the README file.

The following table summarizes the location of the product documentation.

What is it?	Where can you find it?	How current is it?
README and Getting Started book	Product CD root, and in InfoCenter	Current. Rarely needs an update
"Starter" InfoCenter	Installed with the product. Several topics are omitted, but link to reminders to download the "full" InfoCenter. If youdo not see the reminders, then the full InfoCenter is already installed.	Not that current
"Full" InfoCenter download	Download it from product Web site Install it locally tooverwrite and add to your back-level, existing InfoCenter	Very current
"Full" InfoCenter online version	Browse it from the product Web site	Could be slightly more current than InfoCenter download

I want to update the information installed with the product

If you installed the InfoCenter from the product CD (or as part of the productdownload), or if you have not updated the original version in a while, it ishighly recommended that you download the latest InfoCenter to replace yourexisting files.

The InfoCenter installation instructions are posted on the Web page fromwhich you can download the InfoCenter updates:
`Updating a locally installed InfoCenter`

I want a summary of what has changed when I update my InfoCenter

For a summary of the changes to each InfoCenter version, see the online revision history:
`http://www.ibm.com/software/webservers/appserv/doc/v35/ae/infocenter/was/000007.html`

A local copy is installed with your InfoCenter, but could be somewhat back-level, but only to the extent that your locally installedInfoCenter is back-level.

I want to print the InfoCenter content and take it with me

To print the InfoCenter, use the following options.

- Print an entire collection of related articles by printing a PDF file providedwith the InfoCenter, such as the Systems Administation guide. The PDF files are available from various navigational viewson the welcome page in the right side of the InfoCenter. The views are described above.
- Print an individual article displayed on the right side of the InfoCenter HTML frame set by right-clicking the article to make its window the active window, then using the print option provided by your Web browser.
- Use the print option in your Web browser to print an individual article displayed by itself in a browser window.

I am going to be working offline a lot, but need access to information

As discussed earlier, the latest InfoCenter is available foronline viewing from the IBM Web site. But what if your machine is not connected tothe Internet, or even an intranet?

If you are going to be offline for a while, or will be using a very slow connection, you can view a locally installed InfoCenter to avoid needing network access. Before you disconnect, download the latest InfoCenter from the WebSphere Application Server Website and install it, as described above.

The locally installed "full" InfoCenter is quite self-contained, providing accessto documentation, help files, and the product Javadoc without requiring networkconnectivity. You need only a Web browser and an optional Adobe Acrobat Viewer foraccessing PDF files.

Nearly everything is available in one place, although you **will**need a connection to view the online prerequisites information and some Websites referenced in the documentation, usually as supplemental information.

An additional option is to print the needed information and take it with you, as described in the previous section.

I want to find my way around quickly

To find information quickly, use the following features.

Feature	Description
Consult the Getting Started book for a customized path	The Getting Started book suggestsoptimal paths through the InfoCenter for users with various roles,such as evaluator, administrator, and developer. The book suggests a combination of reading and product use for each role, directing users to the most relevant activities for their purposes.
Perform a search	(For details, click "Perform a search.")
Use the WebSphere Application Server welcome page navigational views to find any topic in two mouse clicks	The welcome page is displayed on the right side of the InfoCenter. It contains links to navigational views based on: ● Roles (planner, installer, programmer, and such) ● Functional areas (Servlet support, data access, and such) ● Operating system types Each navigational view is a table of contents that contains links to the relevant articles. Because articles of all numbering levels are listed in these views,you can reach articles in just two mouse clicks, instead of clickingthrough several layers of articles. (For example, you will find direct links both to article 6.6 and article 6.6.0.1.1.4). Many of the views contain links to PDF versions of the same documentation.

Use the left side navigation to drill down through information asyou are learning	The left side of the InfoCenter contains another table of contents thatis organized according to the main activities you will need to performto install the product, develop new applications or migrate existingones, place the applications on an application server, and so on. When you click a topic listed on the left side, a general overview ofthe topic is displayed. You can follow the "Related information" and other linksin each article to drill down gradually to more detailed or low-level information. Use the left side navigation when you are first learning about atopic and want to start at the beginning, or if you are not sure whatinformation you are trying to find and would like some decision-makingguidance. For example, maybe you know that you need to secure your applications,but are not sure what kind of activities or product features this involves.
Use the sitemap to make sure thatyou have seen everything.	The sitemap is an exhaustive list of links to InfoCenter articles, incase you still have not found the topic that you are seeking. It is a "master table of contents" for the entire documentation set.

I want to keep track of what I am reading, and have already read

Most InfoCenter articles have a unique identifying number (ID). Hereis some information for understanding and utilizing the numbering scheme:

- Articles starting with the same numbers are related. Articles with shorter numbers (such as 1.1) provide general or overview information about a topic. Articles with longer numbers (such as 1.1.1.1.1.) provide more detailed information about particular facets of the topic.
- Articles with longer numbers assume the user has read related articles withshorter numbers (article 6.6.7.1 assumes knowledge of article 6.6.7, for example).
- The order among sibling articles can be fairly arbitrary, but itdoes not hurt to read sibling articles according to their numbering sequence.

  For example,it does not really matter whether you read 6.1.2 before or after its sibling6.1.3. It *is* advised that you read 6.1 (the parent to both articles) before reading either article.
- The numbers can be handy for using the InfoCenter Search. If you notethe ID number of an article, you can use the Search to find the article quickly again, whenever you need it. If you do so, include the colon (:) afterthe article number (such as 6.6:) to find a single article. To find entire sets of related articles, such as all articles beginning with 6.6, omitthe colon.
- Some IDs are omitted, particularly in the Standard Edition of the product.Because the Standard Edition information is a subset of the Advanced Edition information, the Standard Edition numbering sequence contains gaps.
- In addition, in both the Advanced and Standard Editions, some ID numbers are intentionally left unused to preserve parallel structure among topics. Do not assume that a file with a particularID exists. Consult the InfoCenter site map for a list of all valid files and their IDs.
- Most articles are numbered, but a small percentage are not, because theyare used in other documentation sets that do not impose a numbering scheme. Standalone books, such as IBM WebSphere Application Server Getting Started,are not numbered according to the InfoCenter scheme.

I want to read less and skim more!

Here is a brief legend to the main InfoCenter graphics that are used toidentify information and help you to distinguish quickly among search results. Throughout the documentation, you can view the text description of any graphic by holding your cursor over it.

Graphic	Usual location	Meaning
🖨	Any article	The article can be printed as part of as part of PDF file.
PDF	Navigational views	The file to which you are linking is in Portable Document Format. Use Adobe Acrobat Viewer to view and print it.
WAS AE	Any article	The article applies to the Advanced Edition of the product.

WAS SE	Any article	The article applies to the Standard Edition of the product.
FP 3.5.1	Any article	All or part of the article pertains specifically to a product fixpack, as identified by the graphic. The Version 3.5.1 graphic, displayed to theleft, indicates information that applies only to Fix Pack 1 and later.
C:\ 🌐 ☕	User assistance	The article discusses information pertaining to all administrative clients
☕	User assistance	The article applies only to the Java-based administrative console (WebSphere Administrative Console)
C:\	User assistance	The article applies only to the command line administrative clients
C:\ 🌐	User assistance	The article applies only to the Web-based administrative console

I need operating system and prerequisites information

See the "Operating systems..." list on the WebSphere Application Server welcome page for linksto information that applies to particular operating systems, such as theinstallation instructions.

The "Operating systems..." list also contains a link to the IBM WebSphereApplication Server Prerequisites Web site:
`http://www.ibm.com/software/webservers/appserv/doc/v35/prereq.html`

The site is the ultimate source of supporting software information for current product releases. Because the site provides centralized access to detailedprerequisite information, the InfoCenter documentation is written in a more general way with respect to citing specific prerequisite brands and versions.

Troubleshooting the InfoCenter and GUI help systems

The following problems and workarounds apply to the InfoCenter and therelated help systems for the product interfaces.

Problem	Op System	Workaround
The readme.html file incorrectly states that the WebSphere Application Server Standard Edition installation guides are included in the product installation.	All	Obtain the Standard Edition installation guides as part of the full InfoCenter download available from the product Web site. Details for this download are in the readme.html file. The installation guides are also available from the online InfoCenter.

When opening the InfoCenter or its files using a Netscape browser, you might receive this message:
`file not found`
This message refers to the name and path of the file you are attempting to open, for example:
`'c:\WebSphere\AppServer\web\doc\begin_here\filename.html' not found`
	Windows NT/2000	Do the following:
		1. Start Windows Explorer.
		2. Click **View**, then select **Options**.
		3. Click the **File Types** tab.
		4. Scroll the Registered File Types list to **Netscape Hypertext Document** and double click.
		5. Under Actions, double click **Open**.
		6. Uncheck the Use DDE option.
		7. Click **OK**, then close.
		With the Use DDE (Dynamic Data Exchange) option off, the window errors that complain that a local file (like the Readme) is not found should not appear.

Using the Netscape browser, the InfoCenter and Help display blank pages or give Java/Javascript errors when opening InfoCenter or Help topics.	Solaris	Set up the Netscape browser environment to allow use of the InfoCenter and Help system (setup consists of putting all Netscape .jar files in the *CLASSPATH*, followed by the required WebSphere JDK):
		1. Configure Netscape:
		    1. Set your *CLASSPATH* to include (in this order) each Jar file in the *MOZILLA_HOME/Java/classes* directory; the Java classes.zip file found in /usr/java1.1/lib.
		    2. In Netscape, select **Preferences**, then **Appearance**, then **Fonts** and change the font settings. The recommended setting for variable width font is New Century Schoolbook; and for fixed width font, it is Courier. Note that many of the font choices included with the Netscape Navigator product will work. Choose a set you like.
		2. Access the InfoCenter and Help from a browser on a non-SUN system using the URLs:
		`http://<server>/IBMWebAS/doc/begin_here`
		`http://<server>/IBMWebAS/help/helpcon.htm`

If Netscape Navigator (or Communicator) is the system default browser and a browser is already open for a purpose other than displaying WebSphere Application Server help or documentation, the open browser might ignore a request to display the WebSphere Application Server help or documentation. The help or documentation is not displayed in the open browser, and no new browser instance is launched for displaying the requested file.	Windows NT/2000	If instances of the default browser are already open for other purposes, close these browser instances, if possible, before requesting help or documentation from WebSphere Application Server using the Help menu on the WebSphere Administrative Console or the Start menu on Windows NT.
The online help for the Administrative Console accessible from the Help menu choices **What is it?**, **How do I?**, and **Property Help** sometimes does not work.	Solaris, HP-UX	Fixes and workarounds are being investigated so that you can access context-sensitive help reliably from the product interfaces. Note, as a workaround the help files are available from the InfoCenter, with a help file quick reference providing easy access.

WAS AE

0.0.7: InfoCenter revision history ("What's New")

The full InfoCenter is periodically updated on the Library page of the product Web site. You can browse the InfoCenter from the Web site or download the updates to a localmachine that already contains a version of the InfoCenter.

This article helps you determine the "freshness" of the information you are viewing or downloading, including:

- Differences between online InfoCenters and downloadable InfoCenter JAR files
- Differences between English InfoCenters and other National Language Version InfoCenters
- Determining when to update your locally installed InfoCenter
- Detailed revision histories for each available InfoCenter version

This article highlights the main changes in each revision of the InfoCenter, startingwith the most recent revision. Information applies to both the Standard and Advanced Editions,unless otherwise stated. It also describes how to tell when your locally installed InfoCenterneeds to be updated.

Differences between the online InfoCenter versions and downloadable JAR files

- The InfoCenter Web page announces the availability of new JAR files and online versions.
- The English online versions of the InfoCenter will always contain the latest information available.
- The corresponding InfoCenter JAR files will *usually* match the online versions. In some cases, JAR files will be updated only with the last of several small,frequent InfoCenter updates.

  For example, online updates #5, #6 (described below),and #7 (upcoming) will be combined into the new JAR files available with update #7, rather than offering three separate JAR file updates in such a short period.
- The English online InfoCenters might contain more documentation sets than the corresponding JAR files.

  For example, IBM HTTP Server documentation integrated into the online English InfoCenter isnot included in the English InfoCenter JAR file. However, the JAR file will contain, at minimum,documentation for IBM WebSphere Application Server, Distributed Debugger, and OLT.

  The IBM HTTP Server documentation is integrated into the online InfoCenter for your convenience,and as part of prototyping efforts for a common InfoCenter for WebSphere products.
- The National Language Version (NLV) InfoCenters other than English will match the correspondingNLV JAR files.

  For example, the Spanish online InfoCenter will match the Spanish JAR file. Boththe online and JAR file versions of the National Language InfoCenters might be down-level comparedto the English online InfoCenter and/or the English JAR files, as described in the next section.

National language differences

Differences between the English and other National LanguageVersions of InfoCenters are also noted by the revision history below.

English documentation is updated fairly frequently, usuallycoinciding with code releases that may or may not be offered in multiple languages, such as Fix Packs. English documentation changes are delivered quickly to the Web site.

In a timely manner, the InfoCenter documentation is offered to global contacts for retranslation. Therefore, the English documentationwill usually be somewhat more updated than the translated versions.

- Most recent English version available: #6
- Most recent translated version available: #1

The following refreshes of National Language Version InfoCenters are anticipated (but not guaranteed):

- Early to mid-January, 2001: NLV InfoCenters at #6 with exception of Fix Pack 2 information
- Early to mid-March, 2001: NLV InfoCenters to include information for Fix Pack 2 "plus"

When to update your locally installed InfoCenter

Here is a recommended procedure:

1. Determine the version of your locally installed InfoCenter by viewing its revision history article (0.0.7).
2. Check the revision history of the "live" InfoCenter that is currently posted on the Web site for online browsing:

`http://www-4.ibm.com/software/webservers/appserv/doc/v35/ae/infocenter/was/000007.html`

3. Compare ... If your locally installed 0.0.7 article does not match the 0.0.7 article currently posted on the Web site in terms of the InfoCenter levels mentioned, then your local InfoCenter is back-level.

   For example, yourlocal 0.0.7 article describes the change history through version 4 of the InfoCenter,but the 0.0.7 article on the Web site describes a version 5, then your local InfoCenterneeds to be updated to version 5.
4. Determine whether an updated JAR file is available for you to download.

   Usually, the JAR files available from the Web site for downloading an updatedInfoCenter are at the same level as the "live" version of the InfoCenter that youcan browse from the Web site.

   View the Web page from which you can browse or downloadthe InfoCenter to confirm whether the "live" (browsable) and downloadable InfoCentersmatch.
   `http://www-4.ibm.com/software/webservers/appserv/infocenter.html`

   If you do not see any notes on that page about a mismatch, then assumethat the "live" version and the JAR files are at the same level.
5. If an updated JAR is available, and your local InfoCenter is back-level comparedto the "live" version, then download and install a JAR file, referring to the download instructions.

InfoCenter versions, starting with most recent

The remainder of this article provides a detailed revision (change) history ofthe InfoCenter, starting with the latest version.

Version 6 for IBM WebSphere Application Server Version 3.5.x (Coinciding with late December IBM WebSphere Application Server for Linux availability)

The InfoCenter now contains information about:

- Version 3.5
- Fix pack 1 (3.5.1)
- Fix pack 2 (3.5.2)
- AIX, HP-UX, Solaris, Windows NT support through Version 3.5.2
- Linux (Intel), Linux on S/390, and NetWare support through Version 3.5.1

The 12/22 refresh of the English Standard and Advanced Edition InfoCentersbuilds on # 1 through 5 below.

Planning and installation	● Added installation and configuration information for product for Linux on S/390 ● Added installation and configuration information for product on NetWare ● Added Linux for S/390 and NetWare pages to "Operating systems..." views to Welcome page ● Updated article 1.2.3.6 with Linux for S/390 and NetWare database driver information
Programming	● No changes
Systems Administration	● No changes
Tuning and Troubleshooting	● No changes
Other/General	● Updated Release Notes for Linux; expanded to include Linux on S/390 support ● Added Release Notes for NetWare support ● Updated article 0.0.7 to reduce length of InfoCenter version headings, provide more detail about which product fix packs and ports are covered by each InfoCenter version

Important notes about this InfoCenter version:

- There is no downloadable version of this refresh. Only the English online InfoCenters posted on the Web site have been refreshed.
- The JAR files will be updated with InfoCenter version #7 due in middle to late January, 2001.

Version 5 for IBM WebSphere Application Server Version 3.5.x (Coinciding with late November electronic code delivery of Linux trial code support)

The InfoCenter now contains information about:

- Version 3.5

- Fix pack 1 (3.5.1)
- Fix pack 2 (3.5.2)
- AIX, HP-UX, Solaris, Windows NT support through Version 3.5.2
- Linux (Intel) trial code support through Version 3.5.1

The 11/21 refresh of the English Standard and Advanced Edition InfoCentersbuilds on # 1 through 4 below.

Planning and installation	• Added installation and configuration information for product on Linux (Intel) • Added "Operating systems..." views to Welcome page • Deleted "Selecting installation steps" article from Section 2 (file 022.html) • Distributed former 022.html material among new operating systems views • Updated article 1.2.3.6 with Linux database driver information and general improvements
Programming	• No changes
Systems Administration	• No changes
Tuning and Troubleshooting	• Updated Tuning Guide
Other/General	• Added Release Notes for Linux (Intel) trial code • Introduced prototype of "WebSphere InfoCenter" (new look and feel) • Restored online Search capability (English InfoCenters for now)

⚠ Important notes about this InfoCenter version:

- There is no downloadable version of this refresh. Only the English online InfoCenters posted on the Web site have been refreshed.
- The JAR files will be updated with InfoCenter version #7 due in middle to late January, 2001.

Version 4 for IBM WebSphere Application Server Version 3.5.x
(Coinciding with 10/13 electronic code delivery of Fix Pack 2)

The InfoCenter now contains information about:

- Version 3.5
- Fix pack 1 (3.5.1)
- Fix pack 2 (3.5.2)
- AIX, HP-UX, Solaris, Windows NT support through Version 3.5.2

The 10/13 refresh of the English Standard and Advanced Edition InfoCentersbuilds on # 1 through 3 below. It contains the following updates, geared toward Fix Pack 2 code changes.

Planning and installation	• Added case-specific instructions for IBM UDB DB2 Version 6.1 Fix Pack 5 • Applied usability improvements to case-specific installation and configuration instructions • Corrected and extended article 1.2.4.2, the remote OSE configuration documentation (Advanced Edition only)
Programming	• Added article 4.2.1.2.1a describing new Java Servlet 2.2. support • Added articles 4.2.5.* describing Bean Scripting Framework support • Added article 4.2.10 describing a script for running a Web application from XML on a servlet engine inside a development runtime environment such as IBM VisualAge for Java • Added article 4.2.2.2.4 describing JSP 1.1 batch compiling • Updated articles 3.3.2.* regarding migration to the newly supported Servlet 2.2 and JSP 1.1 levels • Updated article 4.3.1 with note that EJB clients need ioser.dll • Added article 4.3.2 describing JNDI caching support new in 3.5.2 • Updated article 4.4.1.1.2 with tip regarding the use of sync() in locking and unlocking sessions

Systems Administration	Added article 6.6.8.1.3 describing WAR file conversions using the Java console (WebSphere Administrative Console) Updated article 6.6.10 to describe how to configure JSP file reloading options Updated articles 6.6.8, 6.6.7.1.4, 6.6.8.1.4, and 6.6.16.1.4 with descriptions of new properties in Java console pertaining to extended Servlet and JSP file support Added article 6.6.0.2.1.4 describing WAR file conversion using XMLConfig Updated articles 6.6.0.2.* to include WebSphere XML changes pertaining to extended Servlet and JSP file support Updated article 1.2.3.6 with new driver information; reorganized article for increased usability Updated article 6.6.0.1.14 to better explain Stop for Restart setting Updated article 5.2.5, removing outdated security information regarding Domino users Updated article 0.18.8 with slight clarification about the use of directory services (change applies to Advanced Edition only) Updated articles 1.4.1 and 6.6.0.1a to clarify that Java console requires a JDK in order to run, and that a JDK is not automatically installed with the console Added link from transport type descriptions in 6.6.7.1.4 and 6.6.13.1.4 (latter is for Advanced Edition only) to Tuning Guide, which provides advice for configuring the transport type setting Modified articles 6.6.0.2.* to address WSCP documentation correction Added missing *.tcl.txt WSCP example scripts to documentation set (accessible from WSCP documentation in InfoCenter)
Tuning and Troubleshooting	Updated WebSphere Performance Tuning Guide further Added article 8.7.1 describing new Log Analyzer, available from Web site Updated article 8.1.3 with Invocation Cache Size tuning information Updated article 1.2.3.7 with InstantDB tracing information
Other/General	Added 0.8.1 and 0.8.2 providing conceptual information about WAR file support and conversion Miscellaneous link fixes

Version 3 for IBM WebSphere Application Server Version 3.5.x
(Coinciding with 9/10 electronic code delivery of Fix Pack 1)

The InfoCenter now contains information about:

- Version 3.5
- Fix pack 1 (3.5.1)
- AIX, HP-UX, Solaris, Windows NT support through Version 3.5.1

The 9/10 refresh of the English Standard and Advanced Edition InfoCenters buildson #1 and 2 below. It contains the following updates, geared toward Fix Pack 1 code changes and general documentation improvements.

Planning and installation	Updated "Planning and installation" to include DB2 Version 7.1 informationin the case-specific installation and configuration instructions Updated "Planning and installation" with installation and configuration instructions for new cases Added database migration assistant documentation (articles 1.2.3.2.*) Added JDK migration assistant documentation (articles 1.2.12.*) Updated articles 3.1.* regarding the Migration Assistant (Advanced Edition only) Updated article 1.2.3.8 to contain Sybase session persistence tip; arranged article1.2.3.8 by database type to increase usability; modified session help article 6.6.11 to refer to 1.2.3.8 Updated article 1.4.2.4 regarding remote OSE sample topology, numerous small changes Updated article 1.4.1 with installation workaround affecting remote OSE topology; addedlinks to various sample topologies from their entries in the installable components table Updated article 6.6.0.1.14 with improved descriptions of Stop for Restart and Restart actionsin WebSphere Administrative Console (Java console)

Programming	● Added article 4.2.8 regarding Web application performance, including ServletContext.log() usage tip ● Added article 4.2.9 regarding Web application language encodings, including instructions for settingJSP file and servlet encodings, and encodings for data access ● Added link to product Javadoc index from bottom of each Programming file (articles 4.*)
Systems Administration	● Added article 6.6.18.2 about securing cloned applications (Advanced Edition only);modified some other 6.6.18.* articles to link to it ● Added link to InfoCenter from bottom of each help file (articles 6.6.*) for navigating toInfoCenter interface from a help file that has been opened in a standalone browser window from a product administrative client ● Updated article 6.6.8.1.4 to indicate that the Use Shared Context setting for Webapplications should remain set to false
Tuning and Troubleshooting	● Updated and expanded WebSphere Performance Tuning Guide to include tuning methodology, tuning reference, and summary of tuning differencesfrom IBM WebSphere Application Server Version 3.0x. Access the Tuning Guidethrough "Tuning and Troubleshooting" on the Welcome page
Other/General	● Added Version 3.5 Fix Pack 1 Release Notes to Section 0 (Introduction) ● Added this revision history (article 0.0.7) to Section 0 (Introduction) ● Corrected typos in article 3.3.10, XML configuration migration ● Removed Release Notes links from "Planning and Installation," instead pointing to InfoCenterSection 0, which will offer an archive of Release Notes for base and subsequent Fix Packs andports of the base ● Modified file headers to prevent Javascript error when file is opened outside of the InfoCenter interface ● Other, trivial updates (link fixes and such)

Version 2 for IBM WebSphere Application Server Version 3.5 - English ZIP file downloads (Coinciding with 8/31 code delivery for AIX, HP-UX, Solaris, and Windows NT)

In late August, ZIP files containing the English InfoCenters were postedto the Web site for temporary use while the InfoCenter "smart" downloadsin ten languages were being finalized.

These English InfoCenters contained these main changes with respectto the documentation posted for the previous delivery:

- Content updates throughout sections 1 (articles 1.*) and 8 (articles 8.*)
- Note added: Disregard WebSphere Peformance Tuning Guide document until updated
- Improvements to layout of "Planning and installation" for easier retrieval of PDF and HTML versions of case-specific installation and configuration documents
- Javadoc link added to Welcome page for accessing product API documentation
- Public API documentation published as part of InfoCenter (not included in Search)
- Search facility enabled
- Other, trivial updates (link fixes and such)

⚠ Important notes about this InfoCenter version:

- Discard the ZIP file after the "smart" InfoCenter download becomes available. TheZIP file must be installed prior to installing WebSphere Personalization documentation.The smart download will not have the same limitation.

Version 1 for IBM WebSphere Application Server Version 3.5 base (Coinciding with 8/31 code delivery in 10 languages for AIX, HP-UX, Solaris, and Windows NT)

For the 8/31 delivery (and 8/17 electronic delivery) of IBM WebSphere Application ServerVersion 3.5, the Web site was updated to include "live" InfoCenter versions for supportedlanguages. The live versions can be browsed from the Web site, but cannot be downloaded.Their Search facilities were disabled due to a technical problem currently underinvestigation.

This InfoCenter version is the baseline against which this article (0.0.7) documents changes to the InfoCenter in subsequent refreshes.

⚠ Important notes about this InfoCenter version:

- Disregard WebSphere Peformance Tuning Guide document until updated

Related information...

- 0.0: Becoming an InfoCenter super user

0.1: What are enterprise applications?

An enterprise applicationor J2EE applicationis a combination of Java components(building blocks) that work together to perform a business function. The code components can include:

- enterprise beans in an EJB module
- servlets, JSP files, HTML, and XML in a Web module
- application clients

0.1.1 What are Enterprise Archive (EAR) files?

An Enterprise Archive file represents a J2EE application that can bedeployed in a WebSphere application server. EAR files are standard Javaarchive files and have the file extension .ear. An EAR file canconsist of the following:

- One or more Web modules
- One or more EJB modules
- One or more application client modules
- Any combination of the above

The modules that make up the EAR file are themselves packaged in archivefiles specific to their types--for example, a Web module contains Webarchive files and an EJB module contains Java archive files. EAR filesalso contain a deployment descriptor (an XML file) that describes the contentsof the application and contains instructions for the entire application, suchas security settings to be used in the run-time environment.

When an EJB or Web module is installed as a stand-alone application, it isautomatically wrapped in an Enterprise Archive (EAR) file.

0.2: What are administrative server nodes?

A node is a physical machine in a topology comprised of one or moremachines. An administrative server node is a node that contains a WebSphere administrative server.

WebSphere Application Server Advanced Edition provides centralized administration of multiple nodes. That is, the administrative server on each machine (node) can be configured to use a shared administrativedatabase. See article 0.14.3 for more information.

Nearly every resource in the WebSphere administrative domainresides on one node or another. A few resource types, such as virtualhosts and enterprise applications, are not associated with particularnodes.

For its contents to be managed bythe IBM WebSphere Application Server product, a node must containan administrative server. Therefore, for all practical purposes,a node and administrative server node are equivalent in the worldof WebSphere administration.

- 0.14.3: What is administrative data?

0.14.3: What is administrative data?

IBM WebSphere Application Server has an administrative server thatmanages data about application configurations, application servers, and other resources known to the product.

The administrative server keeps its data in an administrativedatabase, which can be the same database in which applications keeptheir data, or a different database.

The administrative database allows centralized administration -- several administrative serverson various machines can share the data.

- 6.1.1: Administrative elements

0.3: What are application servers?

Application servers extend a Web server's capabilities to handleapplication requests. Given an application, the application server makes thefollowing exchange possible:

1. A user at a Web browser on the public Internet visits acompany Web site. The user requests to use an application thatprovides access to data in a database.
2. The user request flows to the Web server.
3. The Web server determines that the request involves anapplication containing resources not handled directly by the Web server (such as servlets). It forwardsthe request to IBM WebSphere Application Server.
4. The IBM WebSphere Application Server product forwards the request toone of its application servers on which the application is running.
5. The application processes the user request.

   For example, a servlet prepares the user request for processing by an enterprisebean that performs the database access.

   The application produces a Web page containing the results of the user query.
6. The application server collaborates with the Web server to return the results to the user at the Web browser.

The IBM WebSphere Application Server Advanced Edition product provides multiple application servers that can be either separatelyconfigured processes or nearly identical clones.

- 0.3.1: Server terminology
- 0.1: What are enterprise applications?
- 0.5: What are enterprise beans?
- 0.8: What are applications?
- 0.9: What are servlets?
- 0.22: What are server groups?
- 0.45: What are Web server plug-ins?

0.3.1: Server terminology

This section defines a few frequently confused terms:

WebSphere Application Server

An IBM product providing one or more application servers for deploying enterpriseapplications.

To support the application servers, it provides a complete environment,including an administrative server and clients, tracing and debugging, performancemonitoring, and many additional features.

WebSphere application server

One of possibly several application server processes within the WebSphereApplication Server product. Its parts include an EJB server-container runtime and servlet engine support. Advanced Edition supports multiple application servers on a machine.

EJB server

A generic term for a server that handles components coded to the Enterprise JavaBeans specification.

WebSphere administrative server

The server that handles administrative data for the WebSphereApplication Server product. It is not the same thing as an application server. Rather, it maintains data about the configurations of application servers and their contents.

Enterprise JavaBeans (EJB) is a trademark of Sun Microsystems, Inc.

- 0.3: What are application servers?
- 0.4: What are enterprise bean containers?
- 0.7: What are web containers?

0.4: What are enterprise bean containers?

The enterprise beans installed in an application server do not communicate directlywith the server; instead, an enterprise bean *container* provides an interface between the enterprise beans and the server. Together, the container and server provide the bean runtime environment.

The container provides many low-level services, including threading and transactionsupport. Perhaps most important from an administrative viewpoint, the container manages data storage and retrieval for the beans within.

One or more JAR files can be installed in a single container.

Related information...

- 0.3: What are application servers?
- 0.5: What are enterprise beans?

0.5: What are enterprise beans?

An enterprise bean is a Java component that can be combined with other resources to create distributed client/server applications.

There are two types of enterprise beans, entity beans and session beans:

- Entity beans require database connections because they store permanent data.
- Session beans do not *require* database access, though they can obtain it indirectly (as needed) by accessing entity beans.

Beans requiring data access use *data sources*, administrativeresources defining pools of database connections.

All beans reside in enterprise bean containers, which provide an interfacebetween the beans and the application server on which they reside.

- 0.4: What are enterprise bean containers?

0.7: What are web containers?

A *web container* handles requests for servlets, JSP files, and other types of server-side include coding. The web container creates servlet instances, loads and unloads servlets, creates and manages request and response objects, and performs other tasks for managing servlets effectively.

In the WebSphere administrative domain, each application server contains zero or one web containers. The Web server plug-ins provided by the WebSphere Application Server product help supported Web servers pass servlet requests to web containers.

- 0.3: What are application servers?
- 0.9: What are servlets?
- 0.45: What are Web server plug-ins?

0.9: What are servlets?

Similar to the way applets run on a browser and extend the browser's capabilities,HTTP *servlets* run on a Java-enabled Web server and extend the Web server's capabilities.

For example, servlets can support dynamic Web page content, provide database access,serve multiple clients at one time, and filter data by MIME type.

Servlets are Java programs that use the Java Servlet Application Programming Interface (API). They exist as *ServletName*.class files, or could beincluded in a JAR (Java ARchive) file.

For the purposes of IBM WebSphere Application Server, discussions ofservlets focus on HTTP servlets, which serve Web-based clients. Non-HTTP servlets are possible, but are not central to solutions providing Web-based e-business transactions.

- 0.7: What are web containers?

0.8: What are Web modules?

A Web module represents a Web application. It is used to assemble servlets, JSP files, Web pages, and otherstatic content into a single deployable unit. Web modules are stored inWeb archive files (WAR) files, which are standard Java archive files.The standard file extension for WAR files is .war. A Web modulecontains the following:

- One or more servlets, JavaServer Pages (JSP) files, and otherfiles.
- A deployment descriptor, stored in an Extensible Markup Language (XML)file. This file contains information about the structure and externaldependencies of Web components, and information about how the components areto be used at run-time.

A Web module can be used as a stand-alone application, or it can becombined with other Web modules, or with EJB modules, to create a J2EEapplication. A Web module is installed and run in a Webcontainer.

- 0.9: What are servlets?
- 0.10: What are JSP files?

0.10: What are JavaServer Pages (JSP) files?

The Application Server supports a powerful approach to dynamic Web pagecontent: JavaServer Pages (JSP) files. JSP files are application buildingblocks coded to the Sun Microsystems JSP specification.

JSP files enable the separation of the HTML coding from thebusiness logic in Web pages, allowing HTML programmers and Javaprogrammers to more easily collaborate in creating and maintaining pages.

The IBM extensions to the JSP specification include JSP tags that resembleHTML, making it easy for HTML authors to add the power of Java to Web pageswithout being experts in Java.

JSP files support a division of roles:

Team member	Role
HTML authors	Develop JSPs that access databases and reusable Java components, such asservlets and JavaBeans
Java programmers	Create the reusable Java components and provide the HTML authors with the componentnames and attributes
Database administrators	Provide the HTML authors with the name of the database access and table information

0.22: What are server groups and clones?

A *server group* is a template for creating additional, nearly identical copies of an administrative resource and its contents. Usually, the administrative resource is an application server. The resource is referred to as the *original*. The copies are called *clones*.

The clones can be used for workload management purposes. That is, a requestfor the original resource can be handled by the original or any of its clones. The clones might be distributed (located on different machines). Cloning also supports failover and vertical scaling.

- 0.22.1: What is workload management?

0.22.1: What is workload management?

Workload management optimizes the distribution of client processingtasks. Incoming work requests are distributed to the applicationservers, enterprise beans, servlets, and other objects that can mosteffectively process the requests. Workload management also providesfailover when servers are not available, improving applicationavailability.

In the WebSphere Application Server environment, workloadmanagement is implemented by using servlet groups and clones; transports; and workload management-enabled Java Archive (JAR) filesfor enterprise beans. Administrative servers can also participate inworkload management.

0.3.2: What are application server configuration files?

Application server configuration files define the administrative configuration of your WebSphere Application Server product installation. That is, applicationconfiguration files describe the available application servers, their configurations, andtheir contents.

WebSphere Application Server stores administrative data in a database. Think of the application server configuration files of this product version as flattened, XML versions of the administrative database. When you start the product, the XML is parsedto determine the contents of the application server being started.

Location of the application server configuration files

One or more configuration files are provided for you to copy and modify. The followingfile contains a configuration for a default application server and several other applications and resources that you can use as examples or defaults:

`product_installation_root``/config/server-cfg.xml`

A second file contains a very basic configuration on which you can build:

`product_installation_root``/config/admin-server-cfg.xml`

Use the administrative console to modify configurationfiles, rather than modifying them directly.

How can I tell which one I am using?

To determine which configuration file is in use, start the administrative console. Look for the server configuration file name,under the banner, towards the top left corner of the screen.

- 0.3.2a: Tour an application server configuration file
- 6.2.2: Backing up and restoring configurations

0.3.2a: Tour an application server configuration file

This article discusses the typical contents of an application serverconfiguration file. The article "What are application configuration files?" describeswhere to find such a file. Consider viewing the configuration file as you reviewthis description. Because the contents of the configuration file are subject to change, particularly as you customize the configuration, the following descriptionmight not be identical to the actual file contents. Also, in the followingdescription, lines have been wrapped in order to fit on the printed page.

Recent versionsof the Internet Explorer Web browser support a hierarchical view of the XML-based application server configuration file, allowing you to expandand collapse the various XML elements. See your browser documentation for more information about its abilities to display XML, or simply attempt to openthe application server configuration file in your browser.

Note that directory paths are described throughout the server-cfg.xml filewith UNIX-style forward slashes, not DOS-style backslashes.

Administrative server node

A node is defined in an XML structure that looks like this:

```
<applicationserver:Node xmi.id="Node_1" name="localhost">
```

Application servers

Within your node, you will see application servers. Here is the default server:

```
<applicationserver:ApplicationServer xmi.id="ApplicationServer_1" name="Default
Server">
```

Within the server, you will find your ORB settings, including the bootstrap host and port:

```
<applicationserver:ApplicationServer.orbSettings>  <applicationserver:ORBConfig
xmi.id="ORBConfig_1" bootstrapHost="localhost" bootstrapPort="900">    ...
</applicationserver:ORBConfig></applicationserver:ApplicationServer.orbSettings>
```

Also within the server, you will find Web container and EJB container configurations:

```
<applicationserver:ApplicationServer.webContainer>  <applicationserver:WebContainer
xmi.id="WebContainer_1" installedWebModules="WebModuleRef_1">
<applicationserver:WebContainer.transports>      <applicationserver:HTTPTransport
xmi.id="HttpTransport_1" hostname="localhost" port="80"/>
</applicationserver:WebContainer.transports>...
</applicationserver:WebContainer></applicationserver:ApplicationServer.webContainer><applicationserver:ApplicationServer.ejbContainer>
<applicationserver:EJBContainer xmi.id="EJBContainer_1"
installedEJBModules="EJBModuleRef_1">
<applicationserver:EJBContainer.cacheSettings>      <applicationserver:EJBCache
xmi.id="EJBCache_1" cleanupInterval="50" cacheSize="2000"/>
</applicationserver:EJBContainer.cacheSettings>
</applicationserver:EJBContainer></applicationserver:ApplicationServer.ejbContainer>
```

Installed applications

Notice the **installed*Modules** attributes as shown previously. The values of theseattributes are pointers to modules within the installed applications section:

```
<applicationserver:Node.installedApps>  <applicationserver:ApplicationRef
xmi.id="ApplicationRef_1" name="sampleApp"
archiveURL="${APP_INSTALL_ROOT}/sampleApp.ear">
<applicationserver:ApplicationRef.modules>     <applicationserver:WebModuleRef
xmi.id="WebModuleRef_1" uri="default_app.war"/>     <applicationserver:WebModuleRef
xmi.id="WebModuleRef_2" uri="examples.war"/>     <applicationserver:EJBModuleRef
xmi.id="EJBModuleRef_1" uri="Increment.jar"/>
</applicationserver:ApplicationRef.modules>    ...
</applicationserver:ApplicationRef></applicationserver:Node.installedApps>
```

Notice that the path to each application is described symbolically, (althoughsymbolic representation is not required). The symbols are defined in the path-map section:

```
<applicationserver:Node.pathMap>  <server:PathMap xmi.id="PathMap_1">
<server:PathMap.entries>      <server:PathMapEntry xmi.id="PathMapEntry_1"
symbolicName="APP_INSTALL_ROOT"
path="${WAS_ROOT}/installedApps"                            description="The
filesystem path to the directory                                which will contain
installed enterprise applications."/>      <server:PathMapEntry
xmi.id="PathMapEntry_2" symbolicName="LOG_ROOT"
path="${WAS_ROOT}/logs"                            description="The filesystem path
to the directory which                                will contain server log
files."/>      ...  </server:PathMap.entries></server:PathMap>
```

Resources

Resource provider configurations (such as data source definitions) are displayed in two places:

```
      <applicationserver:Domain.resourceProviders>       <resources:JDBCDriver
xmi.id="JDBC1" name="Db2Drvr"
implementationClassName="COM.ibm.db2.jdbc.app.DB2Driver" ... />
<resources:J2EEResourceProvider.factories>          <resources:DataSource
xmi.id="DS_1" name="Gremlin Datasource"
jndiName="jdbc/GremlinDataSource"  ... />            ...
</resources:J2EEResourceProvider.factories>       </resources:JDBCDriver>
...      </applicationserver:Domain.resourceProviders>      ...
<applicationserver:Node.installedResourceProviders>
<resources:ResourceProviderRef xmi.id="Ref_1" classpath="c:/sqllib/java/db2java.zip"
resourceProvider="JDBC1"/>        ...
</applicationserver:Node.installedResourceProviders>
```

Notice that, at the domain level, there are resource providers (such as JDBC drivers) that have zero or more associated factories (such as data sources). Although they are not shown here, you might also see providers and factories for other types of J2EE resources (including JavaMail, JMS, and URL). Because general resource configuration information is specified at the domain level, you will also notice that each node has a list of installed providers. The list provides classpath information about where to find the implementation of the particuar resources on the particularnode.

- 0.3.2: What are application server configuration files?

0.24 What are application client modules?

An application client is a standalone Java program (in contrast to a Webbrowser-based program). An application client module is used toassemble the files that make up the application client into a singleunit. An application client module is stored in a standard Java archivefile and contains the following:

- One or more classes.
- A deployment descriptor, stored in an Extensible Markup Language (XML)file. This file contains information used to set up security and tolook up references to enterprise beans needed by the application.

An application client is installed and run in an application clientcontainer. The application client container uses the deploymentdescriptor information to initialize the JNDI namespace with the referencesand resources needed by the client.

0.5.1 What are EJB modules?

An EJB module is used to assemble one or more enterprise beans into asingle deployable unit. An EJB module is stored in a standard Javaarchive file. An EJB module contains the following:

- One or more deployable enterprise beans.
- A deployment descriptor, stored in an Extensible Markup Language (XML)file. This file contains information about the structure and externaldependencies of the beans in the module, and information about how the beansare to be used at run-time.

An EJB module can be used as a stand-alone application, or it can becombined with other EJB modules, or with Web modules, to create a J2EEapplication. An EJB module is installed and run in an enterprise beancontainer.

0.8.1: What are Web Archive (WAR) files?

A Web Archive (WAR) file is a Java archive file used to store one or moreof the following:

- Servlets
- JavaServer Pages (JSP) files
- Utility classes
- Static documents, such as HTML files, images, and sound.
- Client-side applets, beans, and classes
- Descriptive metainformation

Its standard file extension is .war. WAR files are used topackage Web modules. A Web module can represent a stand-alone Webapplication, or it can be combined with other modules (for example, EJBmodules) to form a J2EE application. The J2EE application can then beinstalled and run in a WebSphere application server.

0.9.1: What are .servlet configuration files?

This servlet configuration method involves creating an XML servletconfiguration file (which is an XML document named *servlet_instance_name*.servlet) that contains:

- The name of the servlet class file
- A description of the servlet
- The servlet initialization parameters
- A page list that contains the URIs (universal resource identifiers) of the JavaServer Pages (JSP) files that the servlet can call. The page list can include a default page, an error page, and one or more target pages that are loaded if their name appears in the HTTP request.

IBM WebSphere Application Server allows the administrator to store an XML servletconfiguration file (*servlet_instance_name*.servlet) in the classpath of anapplication containing a servlet. When the application server hostingthe application receives a request for a servlet instance, it obtains the servlet configuration information from the .servlet file.

0.9.2: What are page lists?

Some servlets (or JSP files) process data related to a user request,then forward the user request to one of several Web pages, based on the outcome of processing the data. For each outcome, the "page list" partof the servlet configuration specifies the location to which to forward the request.

As a simple example, consider a logon servlet that:

1. Prompts for and reads a user ID and password
2. Submits these for authentication by a user registry or directory service
3. Waits to find out whether the directory service approves or denies the user
4. Based on the outcome, forwards the user either to the page that is the entry point to the requested resource, or to a "logon denied" page

A page list allows developers to avoid hard coding the locations (URIs or URLs) of servlets and JSP files. In the above example, thedeveloper would not need to hard code URLs specifying the two possible pagesto which the user could be directed.

By avoiding hard coding, a developer does not need to recompile a servlet in order to change the URLs to which the servlet refers users.

Instead, the administrator tending the servlet can simply change the page list in a servlet configuration file, which is separate from the servlet class file andis named *servlet_instance_name*.servlet. The list can contain a default page, error page, and other JavaServer Pages that are called, depending on the HTTP request. Whenever the .servlet file changes, the application server automatically reloads the corresponding servlet instance.

Servlets do not automatically have configuration files -- implementing a pagelist requires some effort. With WebSphere Application Server,you can either manually create a servlet configuration file (.servlet file), oruse one generated by IBM WebSphere Studio. In addition, the servlet developermust ensure that the servlet supports page list functionality.

0.11: What are sessions and Session Managers?

A session is a series of requests to a servlet, originating from the same user at the samebrowser. Sessions allow servlets running on a servlet engine to keep track of individual users, aconcept known as personalization.

For example, a servlet might use sessions to provide "shopping carts" to on-line shoppers. Suppose the servlet is designed to record the items each shopper indicates he or she will purchase from the Web site. It is important that the servlet be able to associate incoming requests withparticular shoppers. Otherwise, the servlet might mistakenly add Shopper_1's choices to the cart of Shopper_2.

A servlet distinguishes users by their unique session IDs. The session ID arrives with each request. If the user's browser is cookie-enabled, the session ID can be stored as a cookie. As an alternative, the session ID can conveyed to the servlet by URL rewriting, in which the session ID is appended to the URL of the servlet or JavaServer Pages (JSP) file from which the user is making requests.One additional alternative for secure sessions is theusage of SSL information to pass a session ID between server and browser.

The product provides facilities, grouped under the heading *Session Manager*, that support the javax.servlet.http.HttpSessioninterface described in the Servlet API specification.

Session Manager allows for session scoping only by Web application.Only servlets in the same Web application can access the data associated with a particular session.Multiple requests from the same browser, each specifying a unique Web application,result in multiple sessions with a shared session ID.Any of the sessions that share a session ID can beinvalidated without affecting the other sessions.

A session time-out can be configured for each Web application.A Web application time-out value of 0 (the default value)means that the global time-out value from the Session Manager is used.For details, see the Related information.

- 0.9: What are servlets?
- 0.14.4: What is application data?
- Index to API documentation (Javadoc)
- 4.4.1: Tracking sessions

0.14.4: What is application data?

Application data is simply data that applications access. Applicationbuilding blocks such as servlets can establish database connections to:

- Query databases, possibly modifying their contents
- Store session and user profile data
- Store other data, such as application results

0.11.2: What is session clustering?

The Session Manager session support allows multiple application server instances to share a common pool of sessions, known as a *session cluster*. A *session cluster* is the binding of the session to more than one active application server Java virtual machine (JVM) sharing a common HTTP session table.

The implementation of clustering in IBM WebSphere Application Server allows failover. This preserves session data integrity and the common pool of sessions in the event of a system failure in one or more of the clustered JVMs running servlets within a group of servers.

The session clustering implementation also allows load balancing, whereby the session workload is distributed among the JVMs that comprise the cluster.

Session clustering requires an affinity mechanism so thatall requests for a particular session are directed to the sameJava Virtual Machine (JVM) in the cluster. This conforms to the Servlet 2.2 Specification in that multiple requests for a sessioncannot coexist in multiple JVMs. One such solution provided by IBM WebSphere Application Serveris Session Affinity in a model/clone environment; this solution is available as part of the WebSphereplug-ins for Web servers.If one of the servers in the cluster fails, it is possible for the request to be rerouted toanother server in the cluster. The new server can access session data from the common SESSIONS table. This is transparent to the servlet, browser, and user.

If the JVM fails during the writing of session information in the session database,the update to the database does not get committed. Still, the common pool of sessions continues to function, including the session being processed at the time of failure.

For non-catastrophic failures (such as when the JVM remains functional), any session changes that cannot be completed are rolled back. The session reverts to its state prior to the start of the persistence operation. If instead the write operation is completed successfully and the changes are committed, the session is still accessible, regardless of the failure.

0.11.3: What are cookies?

Netscape defines a cookie as "a general mechanism which server side connections (such as CGI scripts) can use to both store and retrieve information on the client side of the connection." See the Netscape cookie specification for additional information.

0.12: What are user profiles and User Profile Managers?

Some applications collect data about the users with which they interact. Thedata is stored in a database. The next time the user interacts with theapplication, the application recalls the data.

Because the application already "knows" something about the user, it can provide the user with a more personalized experience.

Basically, user profiles allow a company to maintain and manage database tables containing fields for demographic data, and use those tables to interact with a database of individual customers or other users on the company system.

For example, when a repeat user logs onto a Web site that supports userprofiles, the Web site can display headlines and advertising tailored to the shopping preferences of that user. The site can address the user by hisor her logon name.

An application implementing user profiles requires database access for storing the user profile data it gathers.

WebSphere Application Server provides user profile support in theform of programming APIs (for the developer) and User Profile Managers that specify the Java classes and data sources providing user profile support (for the administrator).

- 0.14.4: What is application data?
- 4.4.2: Keeping user profiles

0.13: What are transports?

Web clients request service from applications. They make their requests to Web servers. If a requested application resides on an application server remote to a Web server, the Web server must use a *transport* to relay the request to the remote application server.

The transport is available in these varieties:

- Remote OSE redirector
- Thin servlet redirector
- Thick servlet redirector

Article 1.4.2.9 compares the transportoptions, as well as some additional configurations for accomplishing a separation of Web servers from application servers.

When a Web server and application server handling servlet requests reside on the same machine, it is not necessary to use a transport.

0.14: What is data access?

The following table summarizes data access concepts central to IBM WebSphere Application Server:

data store or database	A relational database such as DB2 or Oracle, or another product that manages and accesses data
application database	Holds data accessed by applications managed with IBM WebSphere Application Server
administrative database	Holds data for the IBM WebSphere Application Server administrative server
JDBC resource provider	Software that enables Java applications, suchas those supported by the product, to connect toJDBC-compliant databasesthrough the use of DataSource objects
JDBC resource provider configuration	An administrative configuration that specifies the location of the JDBC DataSource codefor connecting Java applications to a database
data source configuration	An administrative configuration that establishes a pool of connections to a database, for use by Java components. It also specifies connection pooling parameters, such as the maximum number of connections to maintain in the pool. A data source instance (and the associated connection pool)is created for every application server instance that uses the data source. The connection pool associated with a data sourceis in turn shared by all application components(servlets, JSP files, and enterprise beans)that are running in an application server.
connection pooling	A scheme that addresses inefficiencies in obtaining and releasing database connections

0.14.1: What is connection pooling?

Originally, JDBC 1.0 defined the Java APIs for access to relational databases.

With the introduction of JDBC 2.0, the APIs were split into two parts:

The JDBC 2.0 Core API

> This contains evolutionary improvements but has been kept small and focused like the JDBC 1.0 API in order to promote ease of use. Code written for the 1.0 API continues to work with the 2.0 API. The 2.0 API classes remain in the java.sql package.

The JDBC 2.0 Optional Package API

> This supports integration with additional Java standards, including JNDI (Java Naming and Directory Interface), JTA (Java Transaction API), and EJB (Enterprise JavaBeans), as well as providing support for connection pooling and Java beans. IBM implemented these extensions in Version 3.0.2 and then again in versions 3.5 and later.

Connection pooling is the maintenanceof a group of database connections for reuse by applications on anapplication server.It is part of the JDBC 2.0 Optional Package API. Another part of the Optional Package API provides for the use of the Java Naming and Directory Interface (JNDI) and DataSource objects instead of JDBC 1.0 DriverManager objects to access relational data.

Benefits of connection pooling

Each time a resource attempts to access a database, it must connect to that database. A database connection incurs overhead -- it requires resources to create the connection, maintain it, and then release it when it is no longer required. The overhead is particularly high for Web-based applications, because Web users connect and disconnect more frequently. In addition, user interactions are typically shorterbecause of the surfing nature of Internet users. Often, more effort is spent connecting and disconnecting than is spent during the interactions themselves. Further, because Internet requests can arrive from virtually anywhere, usage volumes can be large and difficult to predict.

To address the problem, WebSphere Application Server enables administrators to establish a pool of database connections that can be shared by applications on an application server. Connection pooling spreads the connection overhead across several user requests, thereby conserving resources for future requests.

Connection pooling can improve the response time of any application that requires connections, especially Web-based applications. When a user makes a request over the

Web to a resource, the resource accesses a datasource. Most user requests do not incur the overhead of creating a new connection because the datasource may locate and use an existing connection from the pool of connections. When the request is satisfied and the response is returned to the user, the resource returns the connection to the connection pool for reuse. Again, the overhead of a disconnect is avoided. Each user request incurs a fraction of the cost of connection or disconnecting. After the initial resources are used to produce the connections in the pool, additional overhead is insignificant because the existing connections are reused.

Caching of prepared statements is another mechanism by which WebSphere connection pooling improves Web-based application response times. A cache of previously prepared statements is available on a connection. When a new prepared statement is requested on a connection, the cached prepared statement is returned, if available. This caching reduces the number of costly prepared statements created, which improves response times. The cache is useful for applications that tend to prepare the same statement time and again.

In addition to improved response times, WebSphere connection pooling provides a layer of abstraction from the database that can buffer the client application and make different databases appear to work in the same way relative to an application. This buffering makes it easier to switch application databases because the application code does not have to deal with vendor-specific SQL exceptions but, rather, with a WebSphere connection pooling exception.

- 0.14.2: How the product manages connection pools
- 4.2.4.2: Obtaining and using database connections

0.14.2: How the product manages connection pools

IBM WebSphere Application Server establishes and maintains pools of connections asspecified by the administrator.

After the connections are set up, the product manages them byparceling out connections in response to user requests and then performing housekeepingoperations to maintain a balance between available connections and demand for connections. This ensures that an existing connection is available when a servlet or application server needs a connection.

For example, the connection pool periodically identifies idle or orphaned connections.It terminates idle connections and returns orphaned connections to the connection pool. This means that fewer connections are available (and fewer resources are used) when demand for connections is low.

An idle connection is one that has been free in the connection pool for the amount of time specified in the idleTimeout property of the data source. Idled connections are removed from the pool and closed to the database.

A connection is orphaned when its owning servlet or application has not used the connection for longer than the amount of time specified in the orphanTimeout property of the data source.Orphaned connections are returned to the pool of free connections. Any use of an orphaned connection or its associated JDBC resources results ina StaleConnectionException being thrown.

At the end of a transaction, the transaction manager closes all connections that have enlisted in that transaction. This enables the transaction manager to ensure that connections are not held for long periods of time, which tends to force the pool size up. As a result,an application cannot obtain a connection in one transaction and try to use it in another transaction. If the application tries this, a StaleConnectionException is thrown, because the connection is already closed.

However, if the application must hold a connection outside the scope of the transaction, the administrator can configure a datasource not to automatically clean up the connections it has allocated.To do this, the administratorsets the disableAutoConnectionCleanup property on a datasource to true.This property should be enabled only if the application will always close its own connections. Otherwise, the pool will quickly run out of connections.

Multiple-connection scenarios

If your application typically requirestwo or more connections to the same database manager,consider the following:

- If no transaction is in force(servlets usually run with no transactions in force), the application can get as many connections as it requires,whether or not the data source is JTA-enabled.Each of these connection objects corresponds toseparate connections to the database.
- If a local transaction is in force, the application can get as many connections as it requires,whether or not the data source is JTA-enabled.If the application gets multiple connections from the same data source by using the same user ID, password, and transaction identifier(that is, same transaction with same user), all connection objects share one database connection.If the application requests multiple connections from separate data sources,a new database connection is obtained each time.
- If a global transaction (two-phase commit protocol) is in force and the application requiresmultiple connections from different data sources, these data sources *must* be JTA-enabled. If the application requests two connections from non-JTA data sources in the same global transaction, an error will occur. If the application gets multiple connections from the same JTA data source by using the same user ID, password, and transaction identifier, all connection objects share one database connection.
- 0.14.1: What is connection pooling?

0.14.7: What are data access beans?

The data access beans are Java classes written tothe JavaBeans specification. Data access beans provide a rich set of features andenhanced function with respect to the java.sql package, while hiding much of the complexity associated with accessing relational databases.

You can use the data access beans in JavaBeans-compliant tools, such as the IBM product VisualAge for Java. Because the data access beans are also Java classes, you can use them like ordinary classes as well.

Beginning with WebSphere Application Server Version 3.x, the data access beans have used the JDBC 2.0 Optional Package APIs to take advantage of DataSource objects and connection pooling.

The data access beans (in the package com.ibm.db) offer the following capabilities:

Feature	Details
Caching of query results	SQL query results can be retrieved all at once and placed in a cache. Programs using the result set can move forward and backward through the cache or jump directly toany result row in the cache. In contrast, java.sql classes retrieve rows from the databaseone at a time, in the forward direction only.Each newly retrieved row overlays the previously retrieved row unless additional code is included to expand functionality. For large result sets, the data access beans provideways to retrieve and manage *packets*, subsets of the complete result set.
Updates through result cache	Programs can use standard Java statements (rather than SQL statements) to change,add, or delete rows in the result cache. Changes to the cache can bepropagated to the underlying relational table.
Query parameter support	The base SQL query is defined as a Java String, with parameters replacing some of the actual values. When the query is run, the data access beans provide a way to replace theparameters with values made available at run time.Default mappings for common datatypesare provided, but you can specify whatever your Java program and database require.

Metadata support	A StatementMetaData object contains the base SQL query.Information about the query (*metadata*)enables the object to pass parameters into the query asJava datatypes. Metadata in the object maps Java datatypes to SQL datatypes(as well as the reverse).When the query is run,the Java-datatyped parameters are automaticallyconverted to SQL datatypes as specified in the metadata mapping. When results are returned,the metadata object automatically converts SQL datatypes back into the Java datatypesspecified in the metadata mapping.

- 4.2.4.2.3: Using IBM data access beans to access relational databases
- 0.14.1: What is connection pooling?

0.15: What are custom services?

A *custom service* is a server being managed in theWebSphere administrative domain although it is not a serversupplied by the WebSphere Application Server product.

The custom service can be any server or process necessaryto support the application server environment, including a:

- Java server
- C or C++ server or process
- CORBA server
- Remote Method Invocation (RMI) server

0.16: What are virtual hosts?

A virtual host is a configuration enabling a single host machine to resemble multiple host machines. Resources associated with one virtual host cannot share data with resources associated with another virtual host, evenif the virtual hosts share the same physical machine.

Each virtual host has a logical name and a list of one or more DNS aliases by which it is known. A DNS alias is the TCP/IP hostname and port number usedto request the servlet, for example *yourHostName*:80. When no portnumber is specified, 80 is assumed.

When a servlet request is made, the server name and port number enteredinto the browser are compared to a list of all known aliases in an effort tolocate the correct virtual host and serve the servlet. If no match is found,an error is returned to the browser.

Application Server provides a default virtual host with some common aliases, such asthe machine's IP address, short host name, and fully qualified host name. The alias comprises the first part of the path for accessing a resource such as a servlet. For example, it is *localhost:80* in the request http://localhost:80/myServlet.

A virtual host is not associated with a particular node (machine). It is a configuration, rather than a "live object,"explaining why it can be created, but not started or stopped. For many users,virtual host creation will be unnecessary because the default_host is provided.

- 0.16.1: Why and when to use virtual hosting
- 0.16.2: How Web paths associate resources with virtual hosts
- 0.16.3: The default virtual host, default_host
- 0.16.4: How requests map to virtual hosts

0.16.1: Why and when to use virtual hosting

Virtual hosts allow the administrator to isolate, and independently manage, multiple sets of resources on the same physical machine.

Suppose an Internet Service Provider(ISP) has two customers whose Internet sites it would like to host on the samemachine. The ISP would like to keep the two sites isolated from one another,despite their sharing a machine.

The ISP could associate the resources of the first company with *VirtualHost1*and the resources of the second company with *VirtualHost2*. Now suppose bothcompany's sites offer the same servlet. Each site has its own instances of the servlet,which are unaware of the other site's instances.

If the company whose site is organized on *VirtualHost2* is past due in paying its account with the ISP, the ISP can refuse all servlet requests that are routed to VirtualHost2. Even though the same servlet is availableon VirtualHost1, the requests directed at VirtualHost2 will not be routed there.

The servlets on one virtual host do not share their context with the servlets on the other virtual host. Requests for the servlet on VirtualHost1 can continue as usual, even though VirtualHost2 is refusing to fill requests for the sameservlet.

- 0.9: What are servlets?

0.16.2: How Web paths associate resources with virtual hosts

The administrator can associate the Web paths of resources, such as servlets, Web pages, and JavaServer Pages (JSP) files, with virtual hosts. It is common to say that the resources are "on" the virtual host,even though the virtual host is a configuration, *not* a physical machinethat can hold files.

The Web path of a resource, such as a servlet, is a path by which users canrequest the resource. For example, an administrator might specify two Web paths for a servlet class named *Animals*. This allows users to specify either http://www.companyname.com/Animals or http://www.companyname.com/AnimalsToo to request the servlet.

Because the administrator associates the Web path of a resource, and not the resourceitself, with a virtual host, the administrator can associate one Web path of a servletwith one virtual host, and another Web path of the servlet with a differentvirtual host. WebSphere Application Server provides the flexibility to setup virtual hosting in the way that best suits your needs.

- 0.9: What are servlets?

0.16.3: The default virtual host (default_host)

The product provides a default virtual host (named *default_host*. Thedefault uses port 80. The default_host has these aliases:

- The IP address of the local machine (*yourIPAddress*:80)
- The "localhost" alias, meaning the local machine (localhost:80)
- The DNS name (such as *software*:80)
- The fully qualified host name (such as www.software.ibm.com:80)
- The loopback address (127.0.0.1:80)

Once in a while, the fully qualified name cannot be constructed. If servable paths containing the fully qualified name do not seem to be working,use the WebSphere Administrative Console to check the virtual host's Aliases property to ensure the fully qualified name is registered as an alias.

Unless the administrator specifically wants to isolate resources from one another onthe same node (physical machine), he or she probably do not need any virtual hostsin addition to the default host.

- 6.1.5.1: The default resources

0.16.4: How requests map to virtual host aliases

When a user requests a resource, WebSphere Application Server tries to map the request to an alias of a defined virtual host. The mapping is case insensitive, but the match must be alphabetically exact. Also, differentport numbers are treated as different aliases.

For example, the request

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

maps successfully to

`http://WWW.MYHOST.COM/MYSERVLET`

and to

`http://Www.Myhost.Com/Myservlet`

But it does **not** map successfully to

`http://myhost/myservlet`

or to

`http://myhost:9876/myservlet`

If a user requests a resource using an alias that cannot be mappedto an alias of a defined virtual host, the user will receive a 404 error in the browser used to issue therequest. A message will state that the virtual host could not be found.

0.18: What is security?

Basically, a security system gives an administrator the power to determine who can access applications and their resources. The WebSphere security system enables the administrator to define security *policy* to establish control of resources. The system provides security *services* to enforce the policy.

The articles labeled 0.18.* discuss security concepts underlyingthe product. For comprehensive information about administering and programming for WebSphere security, see section 5.

- 0.18.1: What are authentication and authorization?
- 0.18.2: What is basic authentication?
- 0.18.3: What are digital certificates?
- 0.18.4: What are principals?
- 0.18.8: What are user registries and directory services?
- 0.5: Securing applications

0.18.1: What are authentication and authorization?

A security system gives an administrator the power to protect resourcesby setting up restrictions on who can do what with the system. All securitysystems are based on two fundamental concepts: authentication and authorization.

Authentication

Authentication means proving that you are who you claimto be. Authentication requires that the users of the system have someway of identifying themselves to the system. A familiar example isthe identifier/password combination. If you can provide a valid passwordfor an identifier (also called a *principal*), then you havedemonstrated that you are a valid user of that identifier.

The identifier/password combination is probably the most familiarauthentication strategy, but it is by no means the only one. Eachsystem can define an authorization strategy to meet its needs. Forexample, having a valid user's card (like a library card) is sufficientauthentication for some systems. Some systems may require no authenticationat all, and others may require something as sophisticated as a retinalscan or a DNA match. Each of these strategies is simply a way of provingan identity.

Authentication generally requires two steps:

1. The user submits the required information (like an identifier/password combination) to an authentication service.
2. The service validates the information. If the information is valid, the server grants the user the status as an authenticated user.

   The validation mechanism varies with the type of information needed. For example, an identifier and password combination is validated against a database of information (a *registry*), and an ID card can be visually inspected for a photograph and an expiration date.

Successful authentication is necessary, but not sufficient,for gaining access to protected resources. After a security systemknows *who you are*, it must use that information to determine what*what you are allowed to do*.

Authorization

Authorization is the process of determining what a user is permittedto do. Different classes of users can be given different privileges.For example, everyone can read the online card catalogue of apublic library. You don't even have to be an authenticated user of the system.In other words, all users are authorized to read the catalogue.But the system probably restricts the right to borrow books to authenticatedusers, where *authenticated* means holding a valid card for this library.Depending on the sophistication of the authentication mechanism,the system can restrict your privileges based on the card you carry.For example, certain users can be authorized to borrow an unlimitednumber of books, and others are limited in the number theycan borrow.

The concepts of authentication and authorization allow the constructionof security systems that can be tailored to meet the needsof any class of users and any set of privileges. They can beas simple or as complex as necessary. For example, you can usethe built-in authentication and authorization services providedby WebSphere Application Server, and you can implement others ifyour needs require it.

0.18.2: What is basic authentication?

The *challenge* or *basic authentication* approach is familiar to many Web server administrators and users. It represents a basic form of authentication, in which the security service requests an identifier and password combination from a user when the user attempts to access a resource.

After the user offers an identifier and password, the security service compares them to the database of known users. If the user-provided information is valid, the security system considers the user authenticated.

The database of user information can take one or more of the forms described in article 0.18.8.

- 0.18.8: What are user registeries and directory services?

0.18.8: What are user registries and directory services?

WebSphere Application Server authenticates users against either:

- The operating system's *user registry* (for example, the `/etc/passwd` file on a UNIX system). If you choose to authenticate against the local user registry, WebSphere Application Server ensures that the principal/password combination provided by the user matches a valid combination in the user registry.

  You cannot use the local registry in scenarios involving more than one machine and more than application server. In such scenarios, LTPA is the only supported security solution. LTPA uses a directory service that supports the Lightweight Directory Access Protocol (LDAP) as the user registry, providing a centralized registry that is valid across machines.
- A third-party authentication mechanism, LTPA, which uses a separate product (a *directory service*) for storing authentication data.

  WebSphere Application Server supports certain directory services. See the product prerequisites for a list. An additional option is available for tailoring any of the default filters to fit a directory service not listed.

  If you choose the LTPA authentication mechanism, authentication is delegated to the directory service. The actual authentication process can take one of several forms. For example, if the user can provide only an identifier and a password, authentication is performed by verifying these against the LDAP registry. If the user has an LTPA certificate (or token), then authentication is performed by validating the certificate. If the user has a certificate and is able to set up an secure communication channel (SSL) between the Web client and server, the user is trusted to be the owner of the certificate, and user information in the certificate is mapped to the LDAP registry.

0.18.3: What are digital certificates?

A *digital certificate* is an electronic document that providesproof of identity for a user. In some ways, using a certificate in theelectronic world is like using an ID card in the physical world. Thecertificate is issued to you, and you need only present it. Forexample, authentication typically takes place without interventionby the user. There are no identifiers and passwords to remember andenter.

A certificate contains encrypted security data. The user must keepa private key (used only by the user) to decrypt the information.If this key is lost or stolen, security can be compromised.

Certificates require knowledge and effort to administer. Some administratorsmust issue and maintain large numbers of certificates. Certificates can bemanaged partially by certificate-authority server software, such as theIBMVault Registry product orVerisign.

0.18.4: What are principals?

A principal represents a human user or system entity (for example, a server process).

When a principal requests a protected resource from a server such as ApplicationServer or the Web server, the server attempts to authenticate the principal. A directory service or user registry provides the mechanism necessary for validating the data presented by the principal.

0.19: What are messages, logs, and traces?

Messages, logs, and traces are important diagnostic tools for investigating the behavior of WebSphere Application Server product code, including application servers and administrative servers.

Information	Description	Performance impact when fully enabled
Messages	Provide high-level view of important events, such as successful completions and fatal errors	Minimal to none
Logs	Provide information about administrative and application servers as they initialize and run. After an error or problem condition occurs, logs can be reviewed for clues as to what happened	Low to medium
Traces	Collections of data from trace statements placed throughout the WebSphere product code. As the code executes, tracing informationis sent to a specified file or stream, so that the administrator and IBM support personnel can analyze it	Medium to high

The trace system can provide the most extensive information. It supports:

- Tracking the occurrence of specific events within the product or its components. This is generally known as event tracing and can include method entry/exit tracing and debug tracing among other types
- Dumping the state of the product or its components. This is generally known as performing a state dump

Tracing provides the most detail, but it also can drain performance. Typically,the administrator should enable tracing only when directed to so by IBM supportpersonnel.

- 0.19.1: Message events
- 0.19.2: Trace events

0.19.1: Message events

Message events are generated by IBM WebSphere Application Server code in response to system events occurring in the application server environment. They are always collected. The administrator can decidewhether to look at them.

Message events include, in order of increasing severity:

Audit

Indicates a significant event that must be recorded. Are shown when audit messages are enabled.

Terminate

Indicates that a process has terminated normally (a 0 is returned to the shell). Are shown when audit messages are enabled.

Warning

Indicates that a problem has occurred that must be fixed to prevent a more serious problem

Error

Indicates that the application has encountered a severe error but has not terminated processing

Fatal

Indicates that a process has encountered a fatal error and has terminated abnormally. When a process terminates in this way, the trace service writes its internal ring buffer to a local file and returns a -1 to the shell.

0.19.2: Trace events

Trace events are generated by system or user code. They are collected only if you specify to collect them.

The Application Server product has built-in tracing statements. The console has settings for enabling or disabling the collection of these categories of trace events:

Entry

Indicates that a process has entered a method.

Exit

Indicates that a process has exited a method.

Debug

Provides information for debugging purposes.

Event

Indicates that a significant event took place, such as a state change.

0.20: What are transactions?

A *transaction* is a set of operations that must be executed as a single unit.The operations are used to move data between consistent states.

A *distributed transaction* runs in multiple processes, potentially on many machines. Distributed transactions are available with the Advanced Edition of the product.

- 0.20.1: Transaction identifiers and states

0.20.1: Transaction identifiers and states

The following identifiers are used to track transactions:

- A *local* transaction identifier (TID) is an identifier for a transaction in a specific server. It is known only to that server.
- A *global* TID ties a transaction to all of its participating applications across different servers. The global TID is unique across all servers.
- An *application* TID identifies a participating application.

The administrator tracks transactions through the following states:

Active

Indicates that transaction processing is still in progress.

Committed

Indicates that a transaction has been committed and the effects of the transaction have been made permanent.

Committing

Indicates that a transaction is in the process of committing (that is, the transaction has started committing but has not completed the process).

Marked Rollback

Indicates that a transaction is marked to be rolled back.

No Transaction

Indicates that a transaction does not exist in the current transaction context.

Prepared

Indicates that a transaction has been prepared but not completed.

Preparing

Indicates that a transaction is in the process of preparing (that is, the transaction has started preparing but has not completed the process).

RolledBack

Indicates that a transaction has been rolled back.

RollingBack

Indicates that a transaction is in the process of rolling back (that is, the transaction has started rolling back but has not completed the process).

Unknown

Indicates that the status of a transaction is unknown.

When a transaction **commits**, all actions associated with that transaction are written toa log. In the event of system problems, those actions are repeated if necessary when thesystem's recovery mechanism replays the log.

When a transaction **aborts**, any changes madeby the transaction are undone. After a transaction is undone (rolled back), no evidencethat the transaction was ever attempted remains outside of records in the transactionprocessing system's log.

0.21: What is the Resource Analyzer?

The Resource Analyzer is a performance monitor for WebSphere Application Server Advanced Edition. The Analyzer retrieves performance data by periodically polling the administrative server. Data is collected continuously and retrieved as needed from within the Analyzer. An administrator can set instrumentation levels to control the effect that data collection has on application servers.

The Resource Analyzer provides a range of performance data for two kinds of resources: WebSphere resources, such as enterprise beans and servlets, and runtime resources, such as Java Virtual Machine (JVM) memory application server thread pools and database connection pools.

Depending on which aspects of performance are being measured, the Resource Analyzer can manipulate data to accomplish the following tasks:

- View data in real time, or view historical data from log files.
- View data in chart form, allowing comparisons of one or more statistical values for a given resource on the same chart. In addition, different units of measurement can be scaled to enable meaningful graphic displays.
- Record current performance data in a log, and replay performance data from previous sessions.
- Compare data from a single resource to a group of resources on a single node.
- 0.31: What is the Prformance Monitor?

0.24.1 What are client containers?

Client containers support the execution of J2EE application clients.A client container is an execution shell that launches J2EE applicationclients, providing services to them, such as a `java:comp/env`name space and connections to J2EE resources.

0.25: What are resource providers?

Content coming soon!

0.28: What are Object Level Trace and Debugging?

Object Level Trace (OLT) is a distributed object tracing anddebugging facility. You use OLT to see the interactions among objects involved in servicing an application request. The objects can be servlets or JavaServer Pages (JSP) files. The tracing facility provides an object-level view of interactions among objects deployed in an application server and across application servers. For example, when tracing a servlet that calls another servlet in an application server, the trace interface shows a representation of both servlets and shows arrows forthe method calls between the servlets.

Object Level Debugging (OLD) provides a debugger for removing codingerrors from distributed applications. With the debugger you can step through servlet or JSP file code to debug an application, step through source code, set breakpoints, inspect variables, and perform other debugging functions.

In the WebSphere environment, OLT/OLD consists of the following:

- Server-side components that are installed with the WebSphere Application Server product. They include the Java interpreted debugger and the OLT trace process.
- An OLT Controller in the WebSphere administrative console that provides information to the OLT runtime, including:
  - The location of the OLT/OLD client interface executables. By default, the OLT Controller is configured to assume the OLT and OLD interfaces will run on the same machine as the application server on which tracing or debugging is being performed. However, you can install the OLT and OLD interfaces remotely and change the OLT Controller settings to point to the remote machine
  - The trace mode, which specifies whether to run tracing or debugging
- An OLT Controller interface that confirms the OLT Controller settings and is visible on the machine containing the WebSphere administrative server
- OLT and OLD client interfaces that can be installed on machines local or remote to the WebSphere Application Server product
- Debug settings for each application server managed in the WebSphere administrative console, enabling OLT and OLD for the server

To trace or debug a servlet, you use the administrative console to enable the application server where the servlet is deployed. Thus, OLT and debug enablement is done at the application server level. After enablement, all servlets and JSPs deployed in the application server are OLT enabled.

0.33: What is XML?

Extensible Markup Language (XML) is a framework for defining document markup languages and is predicted to become the primary approach to document exchange over the Internet. In simple terms, a document markup language is a set of elements (frequently called *tags*) that support one or more of the following document characteristics:

- Structure
- Content
- Rendering

Benefits of XML

XML can provide the following benefits:

- **You can accurately describe document content by enabling an extensible tag set**

  XML implementers can define their own tag sets to describe document content. The precision of those descriptions is left to the implementer. For example, one implementation might use a <name> tag and another might find it more useful to use a <city_name> tag.
- **XML enables validating document contents against a standardized grammar**

  The content and structure of an XML document is defined by its grammar. The Document Type Definition (DTD) is an example of such a grammar. Other XML-based schemas are evolving. *Grammar* is used in this documentation as a generic reference to such schemas.

  The grammar describes the valid tags, attributes (characteristics of tags, such as identifiers), and other content for the XML document. Whether an XML document is created as a static file or dynamically generated, the author is responsible for ensuring compliance with the grammar.
- **XML promotes interchange of documents among users and applications**

  Since the early days of computer networking, there's been a need to facilitate the exchange of information among users. The size of potential user populations expands with the use of large networks, such as the Internet, and with the increase in computer-based communication. XML is not the first common document format, but it has advantages over comparable document exchange formats.

XML is the best format for source documents, because it enables the delivery of content in the most appropriate output format (such as HTML, Portable Document Format, and PostScript) and formats for applications (EDI, electronic data interchange).

- **XML supports advanced searching**

  The structure and meaning of XML document content are known (as defined in the document's grammar). The tag structure of XML enables unambiguous parsing of XML documents.

  Grammars define structure and content of XML documents and can be used in performing more efficient searches. For example, user agents could support searching a collection of documents that were authored against a particular grammar. Searching by tag names, tag attributes, data content, and location within a document are other search strategies that XML documents enable.

- **You can separate document structure, content, and presentation**

  Separating document content from document structure is especially important when the content for Web documents must be dynamically generated using programs. Such separation enables Web team members (Web page authors, business logic programmers, and graphics designers) to work in parallel with limited impact on one another's work.

  XSL stylesheets can manipulate XML content in order to present different material to different users in different forms. For example, an auto parts catalog can be presented to a shopper as a view that includes the prices, descriptions, and order numbers for parts. The catalog view for the auto mechanic could include the information available to shoppers plus schematics that show the position of the installed part. The manufacturer's view could include information about subcomponents and materials.

  In addition, XSL syntax supports the use of Cascading Style Sheets (CSS) to control the presentation of XML documents. Putting the presentation controls in a separate file from content enables XML implementers to create multiple views for an XML document without changing the document itself.

- **XML was designed to work over a network**

  XML implementations can have the Web server send an XML document and its associated XSL stylesheets to the client once. Each stylesheet can provide a different view of portions or all of the document data. The user selects the stylesheet to apply. Changing from one view (stylesheet) to the next would not involve sending another request to the server.

- **XML supports Unicode**

XML applications support many document encodings, including Unicode double-byte character set (UTF-16) and a compressed version (UTF-8). Therefore, XML documents can include virtually any languages and scripts.

- **XML extends the definition of linking among documents**

  In HTML, the <A> tag links a document to another document or to a target within the same document. Those links are unidirectional (from the source document to the target). The <A> tag includes the address (URL) of the target link and a text label for the link.

  In contrast, XML supports two types of advanced linking: XLink and XPointer. The XLink and XPointer standards are evolving. With XLink, any tag can be a link. Optional XLink attributes provide additional information about the link itself and about the target document. Other attributes control how the link is activated and what happens when the link is activated. A single link (called an extended link) can even point to multiple targets.

  In HTML, an <A> tag can point to a heading, paragraph, or list within a document. The target section must be a named anchor or tag that permits the ID attribute. In XML, XPointer links can refer to any part of an XML document, even those without identifiers. XPointers link to points in the Document Object Model (DOM) tree (that represents the XML document) and consist of object references, such as root().child(2, address). XPointers can also point to ranges within a document.
- 4.2.3: Incorporating XML

0.33.1: XML constructs

This topic outlines the components of an XML implementation.

Document Type Definition (DTD)

The XML recommendation does not specify a set of tags that can appear in an XML document. Instead, XML implementers determine the actual tags that will be permitted in their documents. Those tags are defined formally in an XML-based grammar, such as a DTD. A DTD can be within the XML document or in a separate file that is referenced in the document's <!DOCTYPE> statement.

DTDs are optional but are almost essential in cases when XML documents must be exchanged among a large number of users. A DTD can be placed inline within the XML document or in an external file.

Well-formed versus valid XML documents

The structure of an XML document is governed by syntax rules for its tag set. There are general rules that are applied to all XML documents to determine whether the document is *well-formed*. To be parsed, an XML document must be well-formed. The XML 1.0 Recommendation describes the rules for well-formed documents, some of which are:

- The first line of the document must be the XML document declaration.
- The document must contain at least one *element* (or *tag*, the more popular term).
- Every starting tag must have a closing tag, such as <tag></tag>. The format <tag/> is also permitted for tags that do not have content (do not contain data).
- The document must contain a unique opening and closing tag within which all other tags in the document must be nested. For example, in the state XML document, the <state> and </state> tags are the unique opening and closing tags. All of the other tags are nested within the two tags.
- Tags cannot overlap. For example, for the state XML document, <name><population></name></population> is not valid.

An XML document that is well-formed and conforms to the rules specified in its grammar is a valid XML document. XML document validating processors (validators) must report validation errors, but they are not required to end the processing.

Stylesheets

XML documents can be exchanged among enabled applications without human intervention. In such cases, XML implementers do not need to be concerned about document presentation. In contrast, if users need to display a formatted XML document, implementers must determine the presentation format.

XSL stylesheets control how XML documents are presented. Stylesheets control whether all or parts of the document are displayed, as well as aspects of the appearance, such as fonts, color, and alignment.

XML also supports Cascading Style Sheet (CSS) language, which was developed for use with HTML. CSS is ideal for controlling actual rendering in the browser, while XSL is designed for more complex document subsetting and formatting. XSL supports advanced features, such as including JavaScript in style sheets, controlling formatting of tag content, and hiding content.

- 4.2.3.2.2: Creating or obtaining DTDs
- www.w3.org site XSL information

0.33.2: XML compared to HTML

XML and Hypertext Markup Language (HTML) are derived from the more complex Standard Generalized Markup Language (SGML). SGML's complexity and high cost of implementation spurred the interest in developing alternatives.

HTML is the most widely used markup language for Web-based documents. As the popularity of HTML increases, the limitations of the language have become more apparent. Those limitations include restricting the user to a relatively small set of tags. HTML authors cannot create their own HTML tags, because commercially available Web browsers have no knowledge of tags that are not part of the HTML standards.

Another limitation of HTML is tags that control presentation are in the same file with tags that describe the document content. Although HTML 4 and Cascading Style Sheets enable HTML authors to separate content from presentation, HTML 4 remains weak in its ability to describe the content of a document.

XML overcomes limitations of HTML and other markup languages, while providing capabilities that are not a part of the earlier languages. Here's a simple XML document and an HTML document that contains the same data:

XML document	**HTML document**
`<?xml version="1.0" standalone="yes" ?><state stateid="MN"><city cityid="12"> <name>Johnson</name> <population>5000</population></city><city cityid="15"> <name>Pineville</name> <population>60000</population></city><city cityid="20"> <name>Lake Bell</name> <population>20</population></city></state>`	`<html><h1 id="MN">State</h1><h2 id="12">City</h2><dl> <dt>Name</dt> <dd>Johnson</dd> <dt>Population</dt> <dd>5000</dd></dl><h2 id="15">City</h2><dl> <dt>Name</dt> <dd>Pineville</dd> <dt>Population</dt> <dd>60000</dd></dl><h2 id="20">City</h2><dl> <dt>Name</dt> <dd>Lake Bell</dd> <dt>Population</dt> <dd>20</dd></dl></html>`

In the XML document, the tag names convey the meaning of the data they contain. The structure of the document is easily discerned and follows a pattern. In contrast, the HTML tag names reveal little about the meaning of their content and the structure is not particularly useful for manipulating the document and exchanging it between applications.

0.33.3: What is the Document Object Model (DOM)?

The Document Object Model (DOM) is a language-independent API for XML and HTML documents. The API represents XML and HTML documents as objects that can be accessed by object-oriented programs (such as Web browsers, document search engines, conversion tools, business logic, and scripting languages). By using the DOM, these programs can create, navigate, manipulate, and modify the documents.

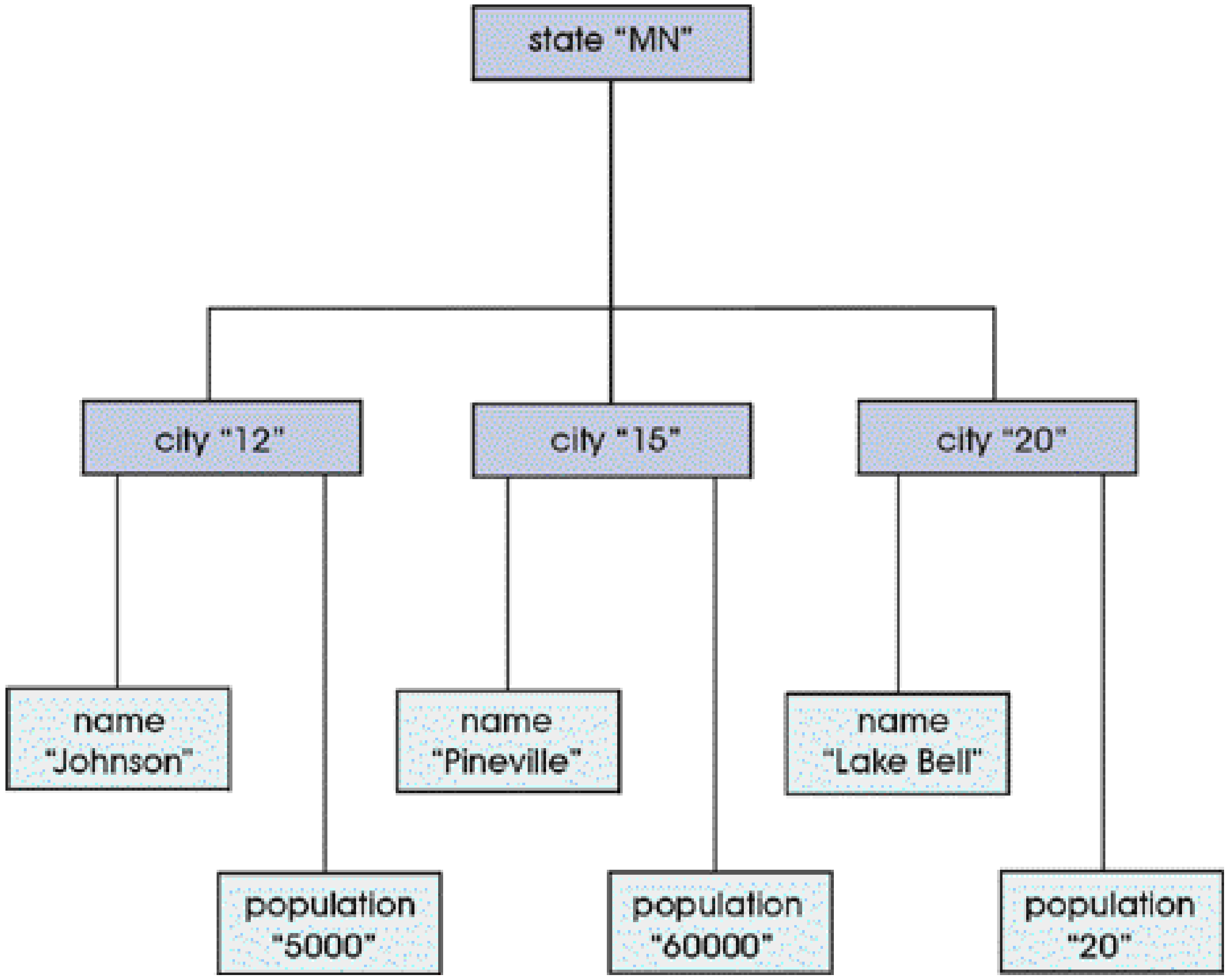
See article 4.1.1.2 to learn about the supported DOM specification.

DOM objects

The DOM can be used to represent an existing XML document or generate an XML document. The document is stored in computer memory. However, it is not persistent.

In the DOM, a document consists of a collection of Nodes that have parent/child relationships. The Node is the primary object and can be of different types (such as, Document, Element, Attribute, Text, Processing Instruction, CDATA Section, and Comment).

The logical structure of each document has a single Document node, which has no parent and zero or more children that are Element nodes. The following figure shows the DOM for the sample XML state document:

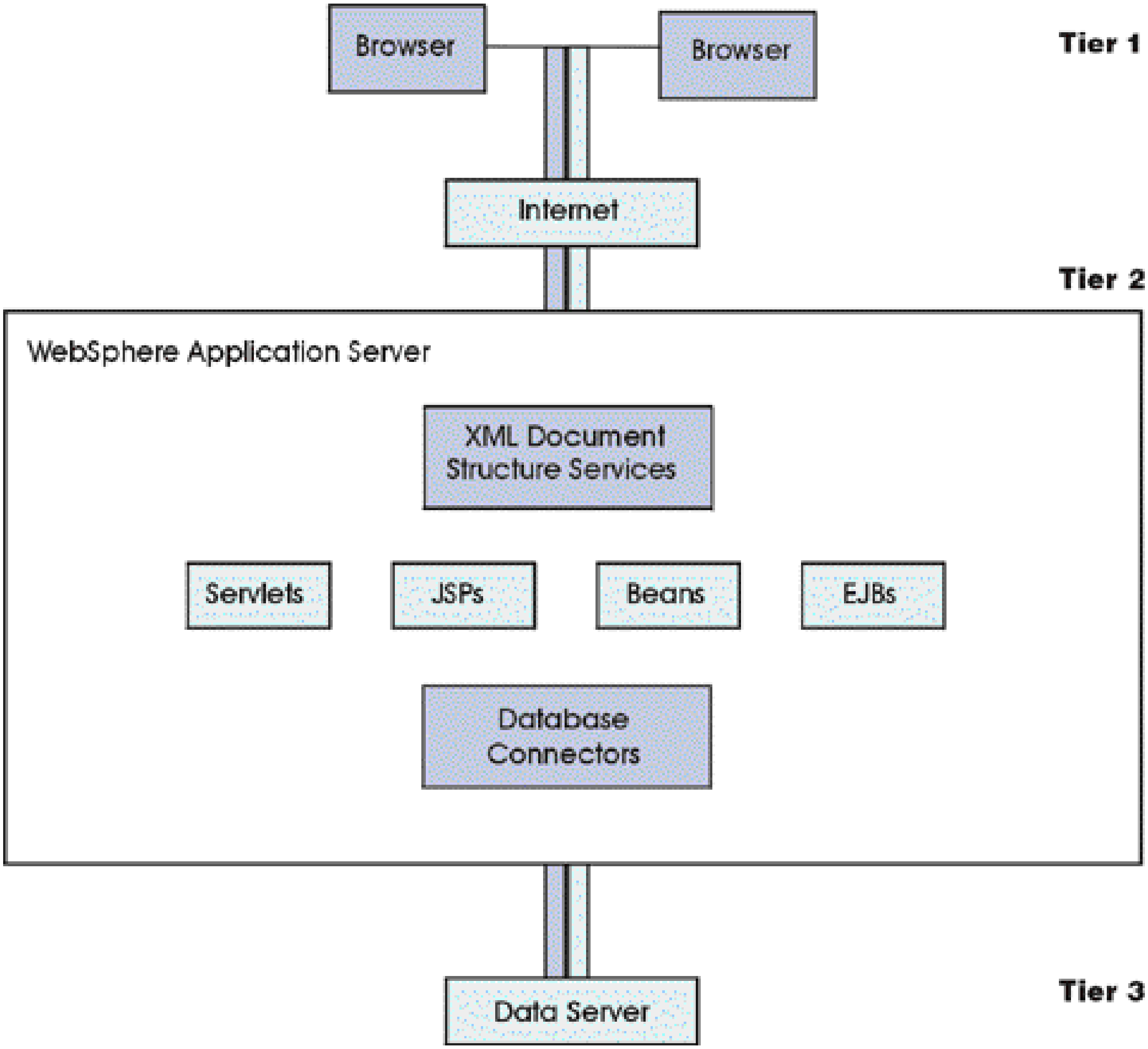


The Document node (state) is the root of the tree. Each of the Element nodes is a tag in the state document. The tag attributes are Attribute nodes. The text (data) within each tag is a Text node.

- 4.1.1.2: Supported XML/XSL APIs and specifications

0.33.4: XML application model

XML applications can be deployed in a logical three-tier environment as follows:



Tier 1

XML-enabled user agents parse an XML document, apply stylesheets, and present the document contents. Some user agents convert the XML document to HTML for presentation. Microsoft Internet Explorer 5 is an example of an XML-enabled user agent that can display XML documents without first converting them to HTML. Although only a small number of XML-enabled user agents are available today, their number is expected to grow as the popularity of XML increases.

Tier 2

XML-based servlets and applications provide server-side XML processing. A

Web server (an HTTP server) can be configured to serve static XML documents. However, to process and dynamically generate XML documents, the Web server base function must be extended. The XML Document Structure Services in the Application Server provide such an extension of the Web server and enables Tier 2 servlets (database connectors and integration applications) to parse, generate, manipulate, and validate XML-based dynamic content. This content is sent to Tier 1 and interchanged with other servlets. Tier 2 can also be used to selectively apply stylesheets to XML documents when the Tier 1 devices do not support the application of XSL stylesheets to XML documents.

Tier 3

The content for dynamically generated XML documents can be retrieved from data servers. Depending on the XML application, the extracted data can be returned as an XML document or returned in JDBC or some other format to a servlet that converts the data to an XML document. In the future, XML-capable Java beans for accessing databases should be commercially available.

0.34 What is application assembly?

Application assembly is the process of creating a Java archive (JAR) filethat bundles all of the files belonging to an application, including.class files, HTML files, GIF files, and so on. A single JARfile can represent an application, or multiple JAR files can be furtherbundled into a higher-level JAR file if the application has manycomponents. Each bundled entity is called a *module*.The assembly process consists of selecting all of the files to be included inthe module, creating a deployment descriptor (XML file) for the module, andthen packaging these files into a single archive file. The deploymentdescriptor lists the contents and characteristics of the module and containsinstructions for how the module is to be deployed in the run-timeenvironment.

The archive files used are in the standard JAR file format but are referredto as follows:

- Web modules, which containing Web application files, are packaged in Webarchive (WAR) files.
- EJB modules, which contain enterprise beans, are packaged in JARfiles.
- Application clients are packaged in JAR files.
- Enterprise (J2EE) applications are packaged in Enterprise Archive (EAR)files.

0.34.1 What are Java Archive (JAR) files?

A Java Archive (JAR) file is a platform-independent file format thatcollects many files into one. For example, multiple applets written inthe Java programming language, and their requisite components (.classfiles, images, sounds, and other resource files) can be bundled in a JAR fileand subsequently downloaded to a browser in a single HTTP transaction.JAR files support file compression and digital signatures. Theirstandard file extension is .jar.

WebSphere Application Server uses the standard JAR file format to packagemodules, regardless of the type of module. However, the archive file isreferred to according to its content:

- An EJB module--packaged in a JAR file
- A Web module--packaged in Web archive (WAR) file
- An application-client module--packaged in a JAR file
- Enterprise application--packaged in an Enterprise Archive (EAR) file

0.35 What is generating code for deployment?

Deployment is the process of making an application available foruse. The Application Assembly Tool is used to prepare an applicationfor deployment. It assembles the modules that make up the applicationand creates deployment descriptor files for the application. Forapplications that contain enterprise beans (EJB modules), the ApplicationAssembly Tool invokes the EJBDeploy tool to generate the containerimplementation classes, and the client-side stubs and server-side skeletonsneeded for remote method invocation (RMI). The container-specific codeis needed to handle services for the application, such as security,transactions, and persistence.

After a module is created, it must be configured and installed in acontainer. A single module (JAR or WAR file) or an entire application(EAR file) can be configured. (If a single module is installed, it isautomatically wrapped in an EAR file.) The process of configurationconsists of preparing the application for the run-time environment. Forexample, external resources needed by the application (and not bundled withthe module), such as other enterprise beans or data stores, must be locatedand bound to the application. Security must be set up based on theaccess control instructions defined in the module's deploymentdescriptor.

0.35.1 What are deployment descriptors?

A deployment descriptor contains configuration data that the run-timeenvironment uses for an application. A deployment descriptor caninclude information about the following:

- The structure and content (enterprise beans or servlets, for example) ofthe application.
- References to internal and external dependencies of theapplication. For example, an enterprise bean in an EJB module canrequire another enterprise bean that is not bundled in the same module.
- References to resource factory objects, such as URLs, JDBC DataSources,JavaMail Sessions, JMS Connection Factories, JMS Destinations, and J2CConnection Factories.
- Security roles that the container uses when implementing the requiredaccess control for the application.
- Transactional information about how (and whether) the container is tomanage transactions for the application.
- Environment variables that the application requires.

Deployment descriptors are XML files packaged with the application'sfiles in a Java archive file. A J2EE application contains oneapplication-level deployment descriptor file, governing the application as awhole. It also contains several component-level deployment descriptors,one for each module in the application. The values of theapplication-level deployment descriptor file override any values at thecomponent level.

Deployment descriptors also include information on binding and IBMextensions. Binding information maps a logical name of an externaldependency or resource to an actual JNDI name. For example, thecontainer uses binding information to locate a remote bean atinstallation. IBM extensions are additions to the standard descriptorsfor J2EE applications, Web applications, and enterprise beans. Theextensions enable Enterprise Edition or legacy (older) systems to work in theWebSphere Application Server environment. They are also used to specifyapplication behavior that is vendor specific, undefined in a currentspecification, or expected to be included in a future specification.Both binding and extension descriptors are stored in XMI files.

0.35.2 What are bindings?

A binding file contains location information necessary to resolve externalreferences for an application or module. For example, bindinginformation is needed to map an external resource from its logical name (inthe deployment descriptor) to its actual JNDI name. Bindings are storedin the `META-INF\ebm-ejb-jar-bnd.xmi` file contained in theJAR file for the module.

Binding files are automatically generated by the Application AssemblyTool. However, it can be useful to understand their structure andcontent in the event that you want to or need to manually edit them.

The Hello enterprise bean in the default configuration contains thefollowing simple binding:

```
 <ejbbnd:EJBJarBinding.ejbBindings>      <ejbbnd:EnterpriseBeanBinding
xmi.id="EnterpriseBeanBinding_1" jndiName="tests/Hello">
<ejbbnd:EnterpriseBeanBinding.enterpriseBean>               <ejb:Session
href="META-INF/ejb-jar.xml#Hello"/>
</ejbbnd:EnterpriseBeanBinding.enterpriseBean>      </ejbbnd:EnterpriseBeanBinding>
... <ejbbnd:EJBJarBinding.ejbBindings>
```

In the example, the bean named `Hello` (as linked to in the EJB1.1 deployment descriptor `META-INF/ejb-jar.xml#Hello`)is to be bound under the global JNDI name `tests/Hello` via the JNDIname entry.

CMP beans have more complicated bindings because they also have a bindingto a data source. The following example is from the Incrementbean:

```
<ejbbnd:EnterpriseBeanBinding xmi.id="EnterpriseBeanBinding_1" jndiName="IncBean">
<ejbbnd:EnterpriseBeanBinding.datasource>            <commonbnd:ResourceRefBinding
xmi.id="ResourceRefBinding_1" jndiName="jdbc/SampleDataSource">
<commonbnd:ResourceRefBinding.defaultAuth>
<commonbnd:BasicAuthDataxmi.id="BasicAuthData_1" userId="" password=""/>
</commonbnd:ResourceRefBinding.defaultAuth>
</commonbnd:ResourceRefBinding>  </ejbbnd:EnterpriseBeanBinding.datasource>
<ejbbnd:EnterpriseBeanBinding.enterpriseBean>            <ejb:ContainerManagedEntity
href="META-INF/ejb-jar.xml#Inc"/>  </ejbbnd:EnterpriseBeanBinding.enterpriseBean>
```

Notice that the above configuration declares both the global JNDI name ofthe bean (`IncBean`) as well as the JNDI name of the data source ofthe CMP bean. It declares the latter by using the ResourceRefBindingentry (`jdbc/SampleDataSource`). The latter JNDI name maps toa configured resource in the application server:

```
<applicationserver:Domain.resourceProviders>      <resources:JDBCDriver
xmi.id="JDBCDriver_1"name="Db2JdbcDriver" ... >
<resources:J2EEResourceProvider.factories>               <resources:DataSource
xmi.id="DataSource_3" name="MyDS" jndiName="jdbc/SampleDataSource"... />
</resources:J2EEResourceProvider.factories>      </resources:JDBCDriver>
```

0.35.3 What are IBM extensions?

The IBM deployment descriptor extensions are additions to the standarddescriptors defined in the J2EE specification for J2EE applications, Webapplications, and enterprise beans. The extensions allow you to specifyproperties that enable Enterprise Edition or legacy (older) systems to work inthe WebSphere Application Server environment. They also allow you tospecify behavior that is vendor specific, undefined in a currentspecification, or expected to be included in a future specification.Examples of extensions are the MIME-filtering properties of Web applicationsand the transaction isolation level of enterprise bean methods.Extensions can be found in the ibm-*-ext.xmi files of themodule.

WAS SE

0.36: What are resource providers?

This information is not available yet.

Related information...

- 0.14: What is data access?
- 0.37: What are mail providers?
- 0.38: What are URL providers?
- 0.39: What are JMS providers?

0.37: What are JavaMail and mail sessions?

JavaMail models an e-mail and messaging service.The JavaMail APIs provide a platform and protocol independent framework to build Java technology-based mail and messaging applications. JavaMail requires the Java Activation Framework (JAF) to deal with complex data types as, for example,MIME (Multipurpose Internet Mail Extensions).

Mail sessions are represented by the `javax.mail.Session` class.The "session" object validates a JavaMail user, and controls the user's access to the message storageand transport services.

To create platform-independent applications,a JavaMail program uses a resource factory reference to obtain a JavaMail session.A resource factory is an object that provides access to resources in a program's deployed environment using the naming conventions defined by JNDI (Java Naming and Directory Interface).

See the Related information section for more JavaMail topics.

- 4.6.1: JavaMail programming
- 6.6.37: JavaMail administration

0.38: What are URLs and URL providers?

A *Uniform Resource Locator (URL)* is an identifier that points toa resource that is accessible electronically, such as a file in a directory on a machine on a network or documents stored in databases.

URLs are in the format *scheme*:*scheme_information*.

A *scheme* might be `http`, `ftp`, `file`, oranother term that identifies the type of resource and the mechanism by which theresource can be accessed. In a World Wide Web browser's location or address box, a URL for a file available using HyperText Transfer Protocol (HTTP) starts with `http:`. An example is `http://www.ibm.com`. Files available using File Transfer Protocol (FTP) start with `ftp:`. Files available locally start with `file:`.

The *scheme_information* commonly identifies the Internet machine making a resource available, the path to that resource, and the resource name. The scheme_information for HTTP, FTP and file generally starts with two slashes (//), then provides the Internet address separated from the resource's path name with one slash (/). For example, `http://www-4.ibm.com/software/webservers/appserv/library.html`.For HTTP and FTP, the path name ends in a slash when the URL points to a directory.In such cases, the server generally returns the default index for the directory.

A *URL provider*, or an Internet Service Provider (ISP), gives you a connection to the Internet with a high-speed link.

- 6.6.36: Administering URL providers

0.39: What is Java Messaging Service (JMS)?

Java Messaging Service (JMS) supports the development of message-based applications in the Java programming language, allowing for the asynchronous exchange of data and events throughout an enterprise. JMS provides an enterprise messaging API developed by Sun Microsystems, Inc., with input from other enterprise messaging vendors. The JMS API defines a common set of messaging concepts and programming strategies that are supported by JMS technology-compliant messaging systems. The API defines an interface for message services but does not define an implementation.

As to WebSphere Application Server, JMS offers the following:

- Connection factories to help create connections to specific JMS providers. Each factory has defined parameters that configure a connection.
- Destination objects that a client uses to specify the target of messages it produces and the source of messages it consumes. A JMS destination provides a specific end-point for messages. A JMS application may use multiple messages.
- Support for distributed transactions.
- 0.39.1: What are JMS providers?
- 6.6.39: JMS administration

0.39.1: What are JMS providers?

A JMS vendor provides an implementation of the JMS API on a JMS client application. A JMS vendor also provides a messaging server that implements the routing and delivery of messages. The client application and the messaging server are collectively referred to as a *JMS provider*.

- What is Java Messaging Service (JMS)?

0.37.1: What are mail providers?

Mail, or also known as service, providers implement specific protocols.Only the SMTP (Simple Mail Transfer Protocol) and IMAP (Internet Messsage Access Protocol) service providers are included with WebSphere Application Server.

- 0.37: What is mail?

WAS *SE*

0.38.1: What are URL providers?

Related information...

- 0.36: What are resource providers?

View the PDF file containing this article for easy printing

0.43: Introducing the J2EE Connectors (J2C)

J2EE Connectors (J2C) provide a J2EE-compliant way for Java applications to interact with non-relational backend systems such as CICS and IMS. TheWebSphere implementation is based on the Proposed Final Draft #2 versionof the J2EE Connector Architecture specification (http://java.sun.com/aboutJava/communityprocess/jsr/jsr_016_connect.html).

IBM WebSphere Application Server supports J2EE Connectors (J2C) through the following components:

- A J2EE Connector (Beta) runtime environment.
- A specific J2EE connector (also known as a J2C (Beta) Resource Adapter) for each non-relational backend system.

The J2EE Connector (Beta) runtime environment is supplied with IBM WebSphere Application Server.
The J2C (Beta) Resource Adapters will be available for free download from the Web. Subsequent production versions will need to be purchased when available.

0.43.1: Installation

Before you can use J2EE connectivity, you must install the J2EE Connector (Beta) runtime and at least one specific J2C (Beta) Resource Adapter.

The J2EE Connector (Beta) runtime is installed separately from the base IBM WebSphere Application Server product. For more information, see 6.6.42.6.1: Installing and uninstalling the J2EE Connector (Beta) runtime.

For information on installing a specific J2C (Beta) Resource Adapter, see 6.6.42.6.2: Installing the J2C (Beta) Resource Adapters .

0.43.2: Tooling

The J2EE Connector (Beta) runtime is tuned to support the Enterprise Access Builder (EAB) Session Beans that are produced by VisualAge for Java version 3.5.3 and later. You can create your beans outside of VisualAge for Java, but it's a more manual process. Using VisualAge for Java makes it much easier to develop beans which work with the J2EE Connectors (J2C).

For more information on creating and exporting EAB Session Beans, see 4.3.4: Developing enterprise beans for use with the J2EE Connectors (J2C).

To deploy and assemble applications that use J2EE Connectors, you use a combination of three tools which are supplied with IBM WebSphere Application Server:

- The batch EJB deploy tool.
- The Application Assembly Tool (AAT).
- The WebSphere Administrative Console.

For more J2C-specific information on deploying and assembling applications, see 6.6.42: Administering J2C related administrative objects (overview).

0.43.3: Supported platforms

J2EE Connectors (J2C) will be available for all platforms supported by IBM WebSphere Application Server. However, for the beta version of the product, they are only available for Windows, AIX, and Solaris.

Related information...

- 4.3.4: Developing enterprise beans for use with the J2EE Connectors (J2C)
- 6.6.42: Administering J2C related administrative objects (overview)
- 6.6.42.6: Installing the J2EE Connectors (J2C)