WebSphere® Application Server for z/OS V5.1

Getting Started

GA22-7957-02

Note

Before using this information, be sure to read the general information under “Notices” on page 171.

Compilation date: May 19, 2004

Contents

How to send your comments

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

- To send comments on articles in the WebSphere Application Server Information Center
  1. Display the article in your Web browser and scroll to the end of the article.
  2. Click on the **Feedback** link at the bottom of the article, and a separate window containing an e-mail form appears.
  3. Fill out the e-mail form as instructed, and click on **Submit feedback** .
- To send comments on PDF books, you can e-mail your comments to: **wasdoc@us.ibm.com** or fax them to 919-254-0206.

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

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

About this book

This book is organized into sections that describe planning, installing and migrating. The information in this book is also available online, which is the most current version.

Planning

Planning is an activity in which you make decisions that affect one or more subsequent activities.

Pre-installation planning helps IT architects and various administrators select and implement multiple node server environments comprised of one or more product installations.

Installing

Installing is an activity in which you put software onto machines. Some installation can be performed after the initial product setup, such as installing the complimentary application clients provided with the product.

Installation can be part of setting up a multiple node production environment.

Migrating

Migrating is an activity in which you take advantage of existing materials. Migration tasks and tools help you upgrade the product and its prerequisites, reuse existing application components when feasible, and transfer administrative configurations from your past version to a current one.

Accessing this information online

This book is compiled from information center articles that are available at Web address: http://publib.boulder.ibm.com/infocenter/ws51help/index.jsp. The information center has many intertopic links that this book is not able to replicate in its subset of information. Although some effort was made to remove such links and replace them with links to external information center articles, it is possible that some links were missed. Intertopic links to articles that are not in this PDF and that we did not change to point to the external information center do not function properly. Such links were overlooked during the production of this PDF. We apologize for any inconvenience a broken link causes and ask you to look for the information in the information center.

Link to external articles on the Web and in the external information center work correctly.

Links to articles that are within this document work correctly.

To view an article in the information center, go to the Web address and search for the title of the document.

This document is superseded by the PDF document in the online information center. Download the current Getting Started PDF at the http://www.ibm.com/software/webservers/appserv/infocenter.html Web site.

Chapter 1. Installing WebSphere Application Server products

This topic introduces the context of installing and customizing IBM WebSphere Application Server for z/OS V5.x and IBM WebSphere Business Integration Server Foundation for z/OS V5.x, including the tasks you need to perform before and after installing the code. IBM WebSphere Application Server for z/OS V5.x is provided in both U.S. English and Japanese.

To create a complete, customized WebSphere Application Server for z/OS installation, you need to prepare the operating system and subsystems, configure native products (for example, WLM, RACF and TCP/IP), physically load the product code through SMP/E, run the Customization Dialog, follow the customized instructions and run the jobs (including the Installation Verification Test) from the Dialog, and bring up your server.

Note: See "Installing a practice runtime" on page 42 for steps you can follow to set up a practice version of WebSphere Application Server for z/OS if you want to just get the feel for it or see the basics.

Perform the following tasks to create a running version of the product on your machine.

1. Prepare for WebSphere Application Server for z/OS installation and customization, as described in "Planning for WebSphere Application Server for z/OS installation and customization" on page 3. You must prepare z/OS subsystems and do other tasks, such as determine important information about WebSphere Application Server for z/OS and z/OS subsystems, before you start installation and customization.
2. Install WebSphere Application Server for z/OS, as described in "Installing WebSphere Application Server for z/OS" on page 27. You must first load the WebSphere Application Server for z/OS code onto your system, using SMP/E install, before you make it usable through customization.
3. Customize WebSphere Application Server for z/OS, as described in "Setting up a base Application Server cell" on page 37. Once you have the WebSphere Application Server for z/OS code installed on your system, you are ready to make it your own by customizing it.
4. Perform WebSphere Application Server for z/OS post-installation tasks, as described in "Performing WebSphere Application Server for z/OS post-installation tasks" on page 57. Once you have WebSphere Application Server for z/OS customized and running smoothly on your system, there are many post-installation tests and modifications you can perform.
5. Tune for performance. Performance tuning for WebSphere for z/OS becomes a complex exercise because the runtime involves many different components of the operating system and middleware. Use the Tuning index for WebSphere Application Server for z/OS to find information and parameters for tuning the z/OS operating system, subsystems, the WebSphere for z/OS runtime environment, and some J2EE application tuning tips.

New in this release

5.1+ IBM WebSphere Business Integration Server Foundation for z/OS, Version 5.1 builds on the industry leading IBM WebSphere Application Server for z/OS, Version 5.1 to deliver a next generation composite application platform optimized for building service-oriented applications that extend and integrate your existing IT assets.

5.1+ IBM WebSphere Business Integration Server Foundation for z/OS is the foundation for higher value IBM WebSphere Business Integration solutions.

5.1+ The product value proposition is a compelling feature of IBM WebSphere Business Integration Server Foundation for z/OS, Version 5.1. The product provides exciting new capabilities, such as support for BPEL4WS (Business Process Execution Language for Web Services) and a technology preview of the Common Event Infrastructure feature.

5.1 + Future releases of IBM WebSphere Business Integration Server Foundation for z/OS will focus on enhancing product integration capabilities including maintaining and enhancing the process choreography and business rule features and adding new integration capabilities and easier-to-use tools. IBM WebSphere Business Integration Server Foundation for z/OS will continue to build on the IBM WebSphere Application Server for z/OS and will maintain its commitment to J2EE and open standards.

5.1 + IBM WebSphere Studio Application Developer Integration Edition, Version 5.1 (name unchanged) is a comprehensive Eclipse-based composite application development environment for building service-oriented applications that extend and integrate your existing IT assets. IBM WebSphere Studio Application Developer Integration Edition V5.1 is optimized for building integrated applications and business processes that deploy to IBM WebSphere Business Integration Server Foundation for z/OS V5.1 and includes all of the features available in IBM WebSphere Studio Application Developer V5.1.1.

IBM WebSphere Application Server offers a world-class infrastructure for open e-business platforms. As the foundation of the WebSphere software platform, WebSphere Application Server provides a rich, e-business application deployment environment with a complete set of application services including capabilities for transaction management, security, clustering, performance, availability, connectivity and scalability.

5.1 + Version 5.1 offers full J2EE specification support (Servlet 2.3, JSP 1.2, EJB 2.0, and others), as well as a variety of extensions.

Several new and improved features are summarized here, with links to more information.

Features that are new or improved are indicated by release level. Technical updates are indicated similarly throughout the documentation. Version 3.5 and Version 4 users will know this as the documentation **Revision history** or **What's New**.

5.1 + See also the **Site Map**, available under the **Reference** container in the default view of each online information center. From the Site Map, you can determine updates relative to the documentation at the V5.0 level.

With your purchase of IBM WebSphere Application Server for z/OS, Version 5, you are entitled to use three features that ship with IBM WebSphere Application Server Network Deployment Version 5:
1. Edge Components: For Load Balancing, Custom Advisors, Consultants, and Enhanced Caching
2. Application Client: A stand-alone Java program launched from the command line or desktop, and typically accesses EJB programs running on the J2EE Application Server
3. 5.1 + Application Server Toolkit: Provides the tooling necessary to assemble and debug Web applications, JavaScript, Java, and compiled languages

If you currently own the Network Deployment product, you already have these three components. You can use the software and adopt the IBM WebSphere Application Server for z/OS license.
- The product offers production-ready J2EE 1.3 standards and Web services support.
- Multiple node environments are created by *federating* multiple WebSphere Application Server installations into *cells*, each of which is managed by a *deployment manager* provided by a Network Deployment installation.

  For more information, see the administrative agents section of Welcome to System Administration.
- The servlet redirector and remote OSE mechanisms are no longer supported. Instead, HTTP forwards Web requests from your Web server to an HTTP server running inside the appropriate Application Server.
- 5.1 + The Application Assembly Tool (AAT) is no longer a feature of the base WebSphere Application Server product. The Assembly Toolkit replaces the Application Assembly Tool. The Assembly Toolkit is

available on the IBM WebSphere Application Server Toolkit (ASTK) CD-ROM. The Assembly Toolkit consists of the J2EE Perspective of the WebSphere Studio Application Developer product, including code generation capabilities.

- PQ74053, which indicates that DB2 is no longer a prerequisite, resulted in changes to Determining WebSphere Application Server for z/OS system requirements.
- PQ76384, which specifies a 7-character limit for the Server Name (short) in the Customization Dialog, resulted in changes to Define Variables to configure base application server node - Definitions, Define Variables to configure Deployment Manager node - Definitions, and Define variables for Federate Base application server node - Definitions.
- PQ77221, which deals with adding a node that was created on one LPAR to the deployment manager on a different LPAR, resulted in changes to Setting up a base Application Server cell and Setting up a Network Deployment environment.
- PQ77310, which specifies a 7-character limit for the Procedure name in the Customization Dialog, resulted in changes to Define Variables to configure base application server node - Definitions and Define Variables to configure Deployment Manager node - Definitions.
- PQ78525, which deals with a workaround that compensates for slow performance on service level w500104, resulted in changes to Setting up workload management (WLM) and Applying or restoring off a service level.
- PQ79539, which deals with indicating the correct SDK level, resulted in changes to Determining WebSphere Application Server for z/OS system requirements.
- PQ79854, which introduces an 8-character limit for the Server Name (short) in the Customization Dialog, resulted in changes to Define Variables to configure base application server node - Definitions, Define Variables to configure Deployment Manager node - Definitions, and Define variables for Federate Base application server node - Definitions.
- PQ80728, which deals with an error related to SBBOLPA, resulted in the addition of Error when installing WebSphere Application Server for z/OS and changes to Define Variables to configure base application server node - Definitions and Define Variables to configure Deployment Manager node - Definitions.
- PQ83245, which deals with which variables data set to load to ensure that your ″WAS home directory″ value in the Customization Dialog is correct, resulted in changes to Define variables for Federate Base application server node - Definitions.

Planning for WebSphere Application Server for z/OS installation and customization

You must prepare z/OS subsystems and do other tasks in this section before you start installation and customization. Additionally, you must determine important information about WebSphere Application Server for z/OS and z/OS subsystems before you start customization. Follow these steps to do so.

1. Review the “Overview of WebSphere Application Server for z/OS configuration” on page 4. This will give you a basic understanding of what is involved with the product.
2. “Preparing the base z/OS environment” on page 23. The base z/OS environment and all of the subsystems associated with z/OS must be able to support your WebSphere Application Server installation. You need to install prerequisite products and configure z/OS subsystems, such as resource recovery services (RRS) and workload management. In particular, be sure you have followed instructions and tips for the following:
   - System requirements. See “Determining WebSphere Application Server for z/OS system requirements” on page 24.
   - TCP/IP. See background information and tips in “Updating your TCP/IP network” on page 27.
   - Security Server (RACF). See Planning to secure your environment.
   - Workload management (WLM). See “Setting up workload management (WLM)” on page 29.
   - Resource Recovery Services. See “Recommendations for Resource Recovery Services” on page 32.

3. If you do not already have one, set up a RACF user ID and authorize it to have read/write access to the WebSphere Application Server for z/OS files (BBO.* data sets and HFS files). Here you will see product data set names without high-level qualifiers, unless a full data set name is required for clarity, in which case `BBO` is used as the qualifier.
4. Review “Migrating and coexisting” on page 76 if you plan to migrate another version of WebSphere Application Server for z/OS.

   This task describes exactly what is migrated during the automatic migration. It also describes how to perform a manual migration using the migration tools.

You are done when you have successfully finished these preparations.

Overview of WebSphere Application Server for z/OS configuration

WebSphere Application Server for z/OS V5.1, hereafter referred to as WebSphere Application Server for z/OS, builds upon the function of WebSphere Application Server V4.0.1 for z/OS and OS/390 and WebSphere Application Server Advanced Edition.

This section covers planning, installing, and customizing tasks for WebSphere Application Server for z/OS.

Planning for, installing, and customizing WebSphere Application Server for z/OS includes those tasks you must perform prior to installing business applications, such as planning your system configuration and installing the WebSphere Application Server for z/OS run-time environment. This chapter:

- Gives a general overview of the tasks you must do to initially install and customize WebSphere Application Server for z/OS.
- Provides pictures and descriptions of your runtime environment after the initial installation and customization of the base and Network Deployment cells. The initial base installation and customization is performed on a monoplex or a single system in a sysplex.
- Provides a checklist of items you should consider for your initial installation of WebSphere Application Server for z/OS, your application development and client systems, and advanced system configurations.

To install the runtime environment initially, you must perform tasks in two general areas:

1. The base z/OS system. You must prepare various z/OS elements, products, and your network prior to setting up WebSphere Application Server for z/OS. For instance, you must perform such tasks as updating your TCP/IP network, setting up security controls, and defining workload management (WLM) workloads.
2. The WebSphere Application Server for z/OS runtime environment itself. This includes loading the code, changing parmlib members, creating environment files, and running configuration jobs.

If you already have a prior version of WebSphere Application Server for z/OS installed and customized, you can configure WebSphere Application Server for z/OS V5.1 to coexist with it. For more information, see the migration information.

After installation and customization, you can install application development environments for your application developers and client environments for your business applications. For more information about this, see the assembling applications information in the z/OS view of the WebSphere Application Server for z/OS information center.

When you have stabilized WebSphere Application Server for z/OS on the first system, you can enable WebSphere Application Server for z/OS in a sysplex. You may also implement other advanced system configurations, such as multi-node cells or connecting your business applications to an IMS or CICS database.

Steps for creating your implementation plan

To get started, plan to build all WebSphere Application Server for z/OS run-time servers on one system, then replicate them on other systems as you expand into a cell. This procedure guides you through initial planning and implementation of WebSphere Application Server for z/OS on a monoplex. Then it guides you through setting up your application development and client environments. Finally, the procedure guides you through planning for optional advanced system configurations.

Before you begin: We assume you have a z/OS system on which you will implement WebSphere Application Server for z/OS.

Perform the following steps to implement your plan, checking off each item as you complete it:

1. Plan WebSphere Application Server for z/OS on a monoplex or a single system in a multisystem sysplex.

Check off	Item	For more information, see . . .
	Determine the skills you need.	“Determining your skill needs” on page 23
	Determine WebSphere Application Server for z/OS system requirements.	“Determining WebSphere Application Server for z/OS system requirements” on page 24
	Understand and plan for customization changes you will need to do for your TCP/IP network.	“Updating your TCP/IP network” on page 27
	Understand security options and prepare for securing your system.	Planning to secure your environment
	Set up workload management environments for WebSphere Application Server for z/OS run-time servers.	“Setting up workload management (WLM)” on page 29
	Customize resource recovery services for use by WebSphere Application Server for z/OS.	“Recommendations for Resource Recovery Services” on page 32
	Plan for your performance and monitoring systems.	Monitoring performance
	Follow recommendations for memory utilization.	“Recommendations for using memory” on page 33
	Plan and define your problem diagnosis procedures.	“Planning for problem diagnosis” on page 34
	Consider automatic restart management before you install WebSphere Application Server for z/OS.	“Automatic restart management” on page 61

2. Install and customize WebSphere Application Server for z/OS.

Check off	Item	For more information, see . . .
	Install and customize a running WebSphere Application Server for z/OS V5.1 base configuration.	“Installing the code through SMP/E” on page 27 and “Using the Customization Dialog” on page 13

3. Perform various post-installation tasks.

Check off	Item	For more information, see . . .
	Plan and define your system backup procedures.	Backing up the WebSphere for z/OS system
	Plan and define your software service procedures.	"Installing service to WebSphere Application Server for z/OS" on page 57
	Implement automation controls and set up automatic restart management for WebSphere Application Server for z/OS, if desired.	"Automatic restart management" on page 61

4. Plan for your application development and client environments.

Check off	Item	For more information, see . . .
	Review WebSphere Application Server for z/OS requirements for application development and client environments.	Related information center articles

5. (Optional) Plan and implement advanced system configurations.

Check off	Item	For more information, see . . .
	Implement the Integrated JMS Provider, using WebSphere MQSeries.	Installing WebSphere MQ as the JMS provider
	Configure a WebSphere Application Server for z/OS Network Deployment cell.	"Using the Customization Dialog" on page 13
	Expand the WebSphere Application Server for z/OS Network Deployment cell to include additional systems in the same sysplex.	Setting up WebSphere Application Server for z/OS on multiple systems in a sysplex
	Implement additional Application Servers in the cell.	"Using the Customization Dialog" on page 13
	Plan to have multiple TCP/IP stacks, use connection optimization, use or an IBM Network Dispatcher.	Multiple TCP/IP stacks, Connection optimization, IBM Network Dispatcher.
	Implement advanced security controls such as SSL and Kerberos.	Implementing security considerations
	Set up RACF protection for DB2, if desired.	Setting up RACF protection for DB2
	Tune system performance.	Tuning the z/OS operating system
	Access DB2 resources (if applicable)	Related information center articles
	Access IMS resources: 1. Use the IMS Connector for Java. 2. Use the IMS JDBC Connector.	Related information center articles
	Access CICS resources with the CICS Transaction Gateway ECI connector.	CICS automation and recovery scenarios
	Plan for testing and production systems.	Testing and production phases

6. Plan and implement release and maintenance upgrades.

Check off	Item	For more information, see . . .
	Review code upgrade methods.	"Installing new releases and maintenance levels of WebSphere Application Server for z/OS" on page 65

You are done when you have checked all the applicable items.

The WebSphere Application Server for z/OS HTTP internal transport

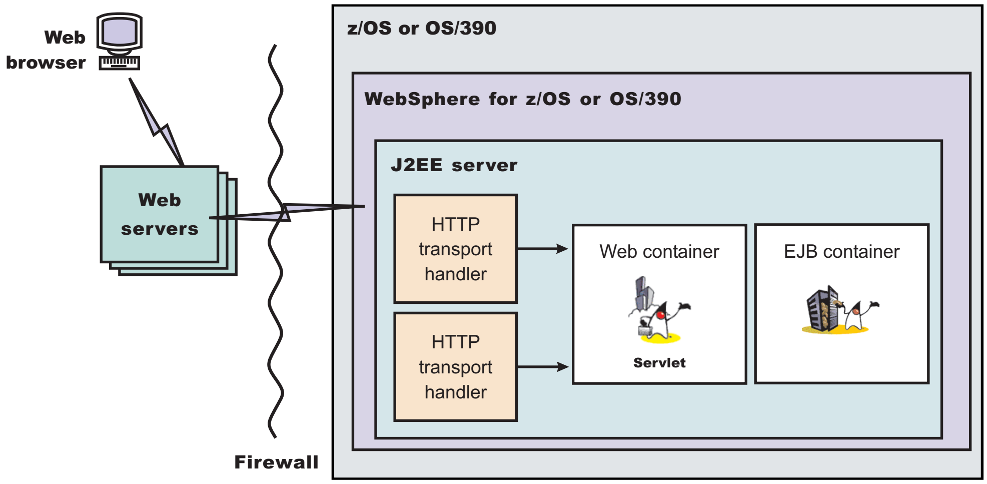
Web components, which are known as Web applications, can consist of any combination of the following parts:

- One or more Java servlets
- Any other Java classes that act as utility classes in support of the servlets
- Static files such as HTML pages and GIF or JPEG images
- JavaServer Pages (JSP) files that format dynamic output

To enable Web applications for use, your Web-serving environment requires an HTTP transport (to receive HTTP requests from a network of browsers using the HTTP access protocol) and an execution environment (to interpret the inbound request and run the appropriate servlet, based on the contents of the inbound request). The WebSphere Application Server for z/OS J2EE server includes a choice of two HTTP transports and execution environments:

1. The HTTP internal transport and/or HTTPS internal transport in combination with the Web container in the J2EE server, or
2. The IBM HTTP Server for z/OS in combination with the WebSphere HTTP plug-in for z/OS shipped with the WebSphere Application Server for z/OS product, and/or Web container in the J2EE server.

Recommendation: Because the Application Server is not designed to directly host static content (for example, HTML pages), the recommended configuration places a Web server with a plug-in in front of the Application Server.

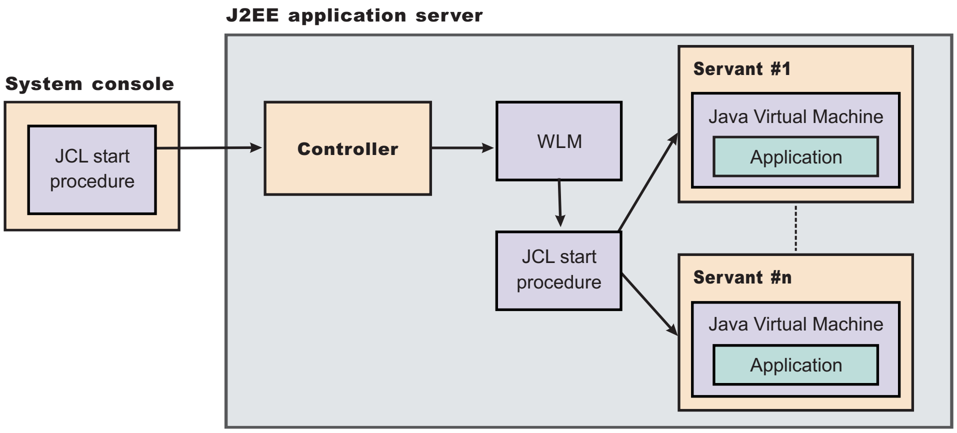


Web applications running in the Web container have direct access to resources on z/OS or OS/390, or can access them through enterprise beans running in any WebSphere Application Server for z/OS server. Web applications use the RMI/IIOP protocol to access enterprise beans running in J2EE servers on the same or different z/OS or OS/390 images.

For more information about deploying Web applications, see the assembling applications information in the WebSphere Application Server for z/OS InfoCenter.

WebSphere Application Server for z/OS topology: terminology and system configuration breakdown

In WebSphere Application Server for z/OS, the functional component on which applications run is called a *server*. Servers comprise address spaces that actually run code.



Within each server are two kinds of address spaces: controllers and servants. A *controller* runs system authorized programs and manages tasks, such as communication, for the server. Each server has one controller that you start with a JCL start procedure when you enter the appropriate start command on the MVS console.

A *servant* is the address space in which the JVM resides. It runs unauthorized programs such as business applications. Depending on the workload, a server has one or more servants running at a time. When work builds up, WLM dynamically starts additional servants to meet the demand.

Note: The location service daemon, node agent and JMS server are specialized servers and have no servants.

Here is a quick breakdown of the different server types on your system:

Unmanaged (base) Application Server
The Application Server that was set up during base configuration that hosts your J2EE applications.

Managed (Network Deployment) Application Server
The Application Server set up during Network Deployment configuration that hosts your J2EE applications.

Location service daemon
A server which is the initial point of contact for client requests in either configuration.

JMS server
Hosts the JMS function in the WebSphere Application Server for z/OS, which controls the MQ broker and queue manager in either configuration.

Deployment manager
A specialized Application Server that hosts the administrative console application (it hosts only administrative applications) and provides cell-level administrative function in a Network Deployment configuration. The administrative console application administers servers (grouped into nodes) on many different systems. The deployment manager is the sole occupant of its own node structure which does not need a node agent because there are no Application Servers in the node, and a cell may have only one deployment manager.

Note: The version of the administrative console application that runs in the deployment manager is designed to manage multinode environments, whereas the version in the base Application Server is for single node environments only.

Node agent
Provides node-level administrative function in a Network Deployment configuration.

Note: Every element of the configuration (servers, clusters, nodes and cells) has both a long and short name:

- The ″Server name″ is the server long name used in the HFS path and the principal name by which the server is known to WebSphere Application Server for z/OS. It is used to identify the server through the administrative console and scripting. It is a mixed case name and greater than 8 characters in length.
- The ″Server short name″ is the platform-specific native alias and the principal name by which the server is known to z/OS. It is used to identify the server to underlying z/OS facilities, such as the Security Server, JES, WLM and ARM. For example, the server short name is used as the MVS JOBNAME.
- The ″Cluster short name″ is used as the WLM application environment name.

A *cluster* is a *logical grouping* of like-configured servers. Clusters exist to promote scalability and availability; workload balancing occurs across the servers in a cluster. Clusters allow you to partition workloads into separate servers while still referring to them as a single unit. Clustering is typically applied to a multinode cell, where each node is configured on a separate system and the cluster has a member (server) on each node. Client requests are distributed among the cluster members based on workload manager decisions.

Note: If you intend for your cluster to span multiple systems in a sysplex, you might need to set up a shared HFS. See ″Considerations for clustered servers and stateful session beans″ for a list of conditions that would make a shared HFS necessary.

A node contains servers which may be part of a cluster. The cluster may span nodes as long as all involved nodes are in the same cell.

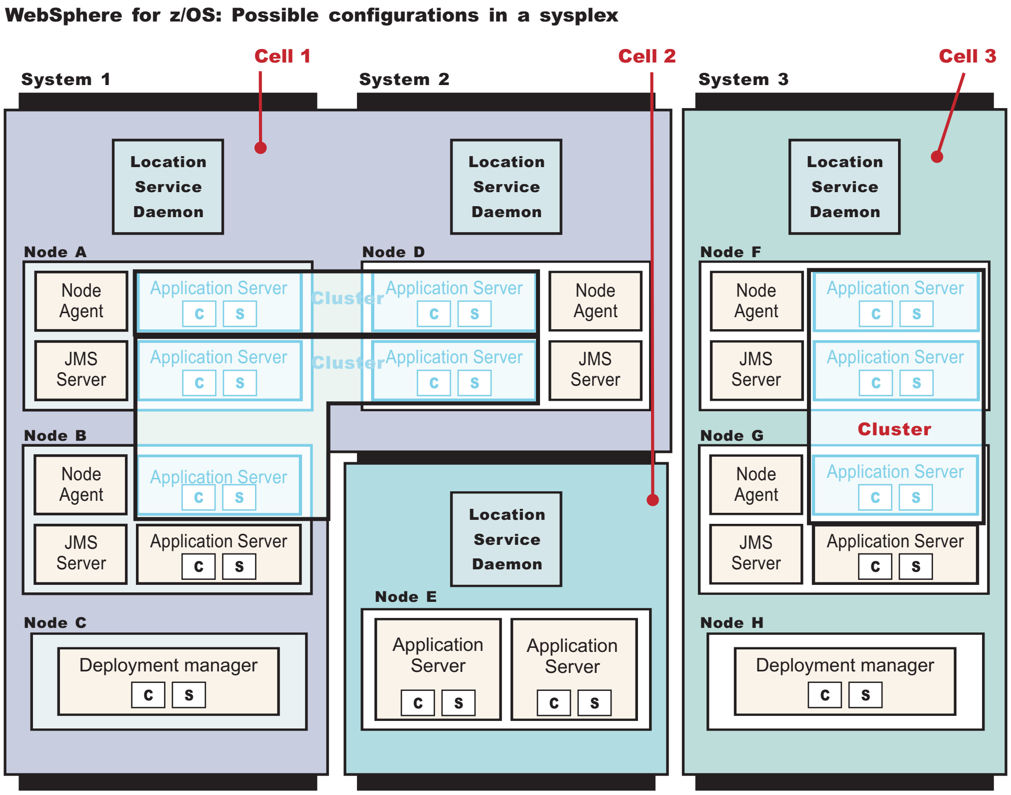
Here is a quick breakdown of clusters, nodes and cells:

cluster
A logical collection of like-configured servers. A cluster can span nodes and systems within the same cell.

node A logical collection of managed servers on a particular system in the cell. A node can contain servers that are part of clusters that span other nodes, but the node itself is confined to a single system and a single cell. You may configure multiple nodes on the same system--either in the same cell or in different cells. Nodes exist mainly for the purposes of administration, and each Application Server node managed by a deployment manager must have a node agent.

cell A logical collection of nodes and the boundary of the administrative domain. The nodes that comprise a cell must be configured on systems in the same sysplex. You may configure multiple cells in the same sysplex. Different cells may have nodes on the same systems, though a given node can be a member of only one cell. The cell is the largest unit of organization.

To help you understand the interaction between servers, clusters, nodes and cells, here is a diagram depicting various configurations you can set up in your Network Deployment sysplex:



Cells 1 and 3 in the illustration depict Network Deployment configuration cells. Cell 2 is a base configuration cell.

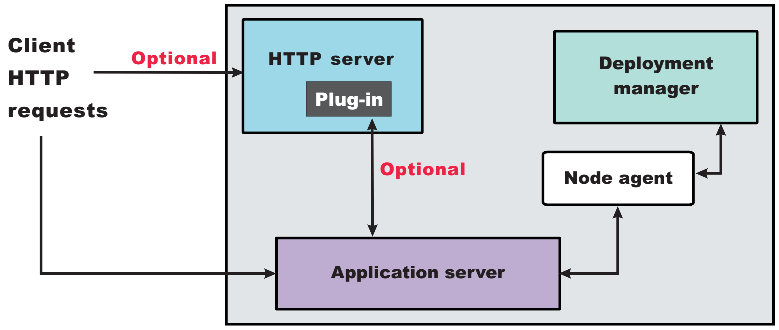
Single server topology

The following illustrations show examples of single server topologies. Each WebSphere Application Server product can run in a single server environment. The most common topology is a stand-alone base WebSphere Application Server product.

IBM WebSphere Application Server for z/OS integrates the WebSphere Application Server product and the Network Deployment product into a single package. In a case such as this, the term product really equates to function.

The WebSphere Application Server base product runs on a single machine. You can install the product in a stand-alone configuration or as part of a cell in a multimachine configuration. The stand-alone configuration is typically for developer desktops or stand-alone production computing, which involve a single Application Server instance operating independently of any other applications.

The Network Deployment function can also run on a stand-alone machine or in a multimachine configuration, as described in other topology topics. The following illustration shows a typical developer environment for Network Deployment function in a stand-alone configuration.



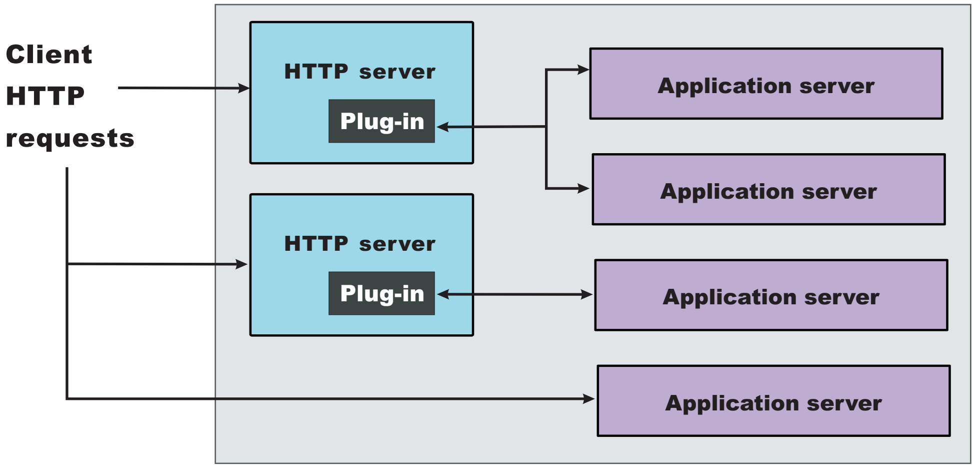
Multiple instances on one machine

The base WebSphere Application Server product contains three main topologies:

- A single installation, as described previously
- Multiple installations in a coexistence environment
- A single installation with multiple configuration instances

You can install the base WebSphere Application Server product one time as described previously, which creates a default server1 instance.

You can install the base WebSphere Application Server product multiple times in separate directories. Each resulting installation instance is a fully functional Application Server. The following illustration shows an example of such a configuration.



Planning your configuration

Before installing the product, planning your installation is necessary. Planning your WebSphere Application Server for z/OS product configuration requires you to visualize your end goal. After the end goal is firmly in place, various assumptions and other decisions are factored in.

The following four main objectives are suggested, although not required. Carefully consider these recommendations.

1. Do not use default configuration values.

This is to avoid interfering with any other WebSphere Application Server environment that may have made use of default values. It also helps avoid the problem of someone later building a configuration with default values that interfering with WebSphere Application Server for z/OS.

2. Design a naming convention that is meaningful and flexible.

   The goal to support easy horizontal and vertical scaling is really a statement about the designed naming convention. A poor naming convention might hinder the expansion of the configuration.

3. Ensure that the configuration design easily supports scaling.

   It is important to consider the ″horizontal scaling″ of the Application Server cell. Try to avoid, by planning ahead, the necessity to change things like the naming convention just because the cell was expanded.

4. Ensure that the configuration design easily supports the creation of additional servers on any of the systems.

   Consider this ″vertical scaling″ of the WebSphere Application Server for z/OS configuration within a given MVS system. Again, the objective is to design the configuration so that adding more servers will not require excessive additional work.

In addition to the basic objectives outlined above, WebSphere Application Server for z/OS Version 5 allows for three other design objectives that are also recommended, but not necessary:

1. Minimize the number of JCL start procedures One of the features of WebSphere Application Server for z/OS Version 5 is the ability to use a ″generic″ set of JCL procedures to start multiple servers. The alternative is to use separate JCL procedures for each server.

   Using the same JCL for multiple servers only works if all the servers are under the same HFS mount point. The JCL contains a `SET ROOT=` line that names the configuration HFS mount point. The value of `&ROOT` is combined with the passed-in value of `&ENV` to create the pointer to the was.env file. If two serves are under different HFS mount points, they would then require separate JCL as the `SET ROOT=` is hard-coded in the JCL.

   The recommended way to address this is to use a mount point that is shared across all MVS images. This is the cleanest and easiest solution.

2. Minimize the number of RACF userids and groups.

   Any security configurations and/or changes should always be cleared through your security administrator. With WebSphere Application Server for z/OS Version 5, you can either minimize the userids/groups or establish separate identities for each server.

3. Utilize Sysplex Distributor For clients outside the cell, it doesn’t matter which location service daemon is accessed or which node agent is used. Therefore, you can use the Sysplex Distributor to balance inbound traffic across both location service daemons with the same port values as well as node agents that share ports.

Planning for additional configuration considerations

Planning the product is fundamental for a successful installation. In addition to the obvious configuration aspects, there are other things that should be considered, including:

- Location service daemons

  Location service daemons are perhaps the least understood component of a configuration. A location service daemon is a single controller server. A Network Deployment configuration that spans MVS images in a Sysplex will have one location service daemon per system over which the cell spans.

  Note: What does a location service daemon do? Its role is to provide ″location name services″ to the node, and as part, be able to resolve indirect object requests from external clients into direct references. Suffice it to say, a location service daemon is required. Bring down the location service daemon and all servers related to that location service daemon come down as well.

  For more information on location service daemons, see Location service daemons.

- TCP/IP ports and default port assignments WebSphere Application Server for z/OS Version 5, by default, makes use of a considerable number of ports. The default values are spread out over a wide range. This makes the management of the ports for a ″test″ and ″production″ challenging. See z/OS port assignments for detailed information on default port info.
- Sysplex Distributor Sysplex Distributor is the function of z/OS TCP/IP that provides a way to intelligently distribute TCP traffic within a Sysplex environment. Clients out on the network direct their requests to the Dynamic Virtual IP Address (DVIPA) address. DVIPA is a virtual address hosted by one of the TCP stacks in the Sysplex, and capable of being taken over by another TCP stack in the event the first TCP stack fails. The TCP stack that's hosting the DVIPA intelligently balances the requests across all the processes in the Sysplex that are listening on the ports defined as eligible for this kind of routing. Not all ports are distributed by Sysplex Distributor; only those defined to TCP/IP.
- Dynamic WLM application environments Dynamic WLM application environments is a function of WLM that became available with APAR OW54622. The new function enables programs to dynamically create application environments on the fly. WebSphere Application Server for z/OS Version 5 is a program that is designed to make use of the dynamic WLM application environment if it sees the function available. Generally speaking, this is a function of the Application Server you can't turn off:

  WebSphere Application Server will use the dynamic capability of WLM if it is available, otherwise it will rely on static WLM application environments. Static application environments still in existence when APAR OW54622 is applied will simply stop being used in favor of dynamically created ones of the same name.
- Node configurations

  Allow no more than one node per system in the cell. The primary reason for imposing this limitation is because use of the Sysplex Distributor for the bootstrap port of the node agents, particularly when all the node agents in the cell are listening on the same bootstrap port. Two node agents on the same MVS image listening on the same port might imply TCP port sharing.

  It is possible to have multiple TCP stacks on the same MVS image, and to tie a port to a specific TCP stack. Multiple nodes on separate MVS images means each has its own TCP stacks with separate IP addresses. There's very little reason to configure multiple nodes in the same cell on the same MVS image. Once a node is created on an MVS image and becomes part of the configuration cell, no other nodes will be federated on that MVS image. If more servers are desired on the MVS image, servers will be added to the existing node rather than federating in more nodes.
- Log streams A parallel sysplex should have a common error log stream rather than a different log stream for each node. It just makes things easier.

  You can now configure a new log stream; the Transaction XA Partner log. For more information about the Transaction XA Partner log, see Working with diagnostic tools and controls.

Using the Customization Dialog

The Customization Dialog is intended for the system programmer or administrator responsible for installing and customizing WebSphere Application Server for z/OS. It allows you to perform the following options:

The Customization Dialog allows you to perform the following options:

1. Configure security domain

   This will produce security domain variables that you will then use in other options.
2. Configure base Application Server node

   This will give you a working base system with a single-cluster Application Server and a location service daemon.
3. Configure integral JMS provider

   This optional step will set up JMS on your system.
4. Configure deployment manager node

   This will enable the Network Deployment function of WebSphere Application Server for z/OS, creating the deployment manager and node agent.

5. Federate base Application Server node

   This optional step will add another deployment manager node to your WebSphere Application Server for z/OS configuration.

In order to use the Dialog, you must know or be able to find the system characteristics for the system on which WebSphere Application Server for z/OS will run.

The Dialog covers a portion of the customization process. Specifically, it creates tailored jobs to:

- Copy the generated jobs into your system libraries.
- Create the run-time HFS structure and the initial environment file.
- Set up WebSphere Application Server for z/OS security controls (RACF).
- Define the WebSphere Application Server for z/OS run-time configuration.
- Run the Installation Verification Test (IVT).

The Dialog also produces a set of instructions for you to follow in order to effectively run these tailored jobs.

Note the following:

- Prior to attempting to install and manage a Network Deployment configuration, it is strongly recommended that you first attempt a base Application Server configuration. Later, when you want some experience with the Network Deployment configuration, carry out those experiments separate from the base application. Setting up the runtime consists of a sequence of manual activities, jobstreams, and REXX execs. Installing the runtime takes a little practice, an understanding of the WebSphere Application Server for z/OS system structure, some systems programming skills and attention to detail.
- The V5.1 customization process is similar to that in V4, but without the DB2, LDAP, or bootstrap steps. It is recommended that you install a ″practice″ configuration before you install the first one to be used for initial testing. There are always decisions made in the Customization Dialog that you want to change once you see how they affect the runtime configuration.
- Be aware of other versions of WebSphere Application Server for z/OS that you have running on your system, as the Customization Dialog does not detect them for you. Keep this in mind when going through the Dialog and, if you are running other versions of WebSphere Application Server for z/OS, watch out for such problems as potential location service daemon port collisions or LPA issues. See the migration topics in the information center for more information.

This section outlines the steps you should follow to configure your system using the Customization Dialog. See the links at the bottom of this article for the corresponding instructions and help for each.

1. Fill in the Customization Dialog worksheets. Worksheets are provided for each task in the Dialog to help you determine what values you should enter in the ″Define Variables″ stage of customization. You need to consult with other systems programmers, specialists or administrators to determine some of the values, so it is a good idea to get it done beforehand. See the worksheets in:
   - “Define variables for security domain configuration - worksheets” on page 107
   - “Define variables to configure base Application Server node - worksheets” on page 113
   - “Define variables to configure integral JMS provider - worksheets” on page 127
   - “Define variables to configure deployment manager node - worksheets” on page 135
   - “Define variables for federate base Application Server node - worksheets” on page 144

   Note: You have several tools on your system that you can use to determine or verify this information. Those tools include MVS & TSO commands, ISPF, SDSF, and other tools such as MXI, which you can get from http://www.secltd.co.uk.
2. Start the Customization Dialog. Get the Dialog up and running so you can begin your customization. See “Steps for starting the Customization Dialog” on page 16 for instructions.

3. Determine which task you wish to complete. After pressing ″Enter″ to get past the introduction screen(s), you'll see the main Dialog panel which lists the tasks you can complete. Select the corresponding number for your desired task to start.

   Note: The remainder of the steps assume you have already chosen a task.
4. Load customization variable information saved from previous configurations, if available. If you previously configured WebSphere Application Server for z/OS and saved the variables, you can load them and use them as a base and for defining your current configuration. See "Steps for loading customization variables" on page 17 for instructions.
5. Load security domain customization variable information saved from ″Configure security domain″ task or previous configurations, if available. In the tasks that require security domain variables, first run the ″Configure security domain″ task and then load the saved variables using this option in the other tasks. See "Steps for loading security domain variables" on page 18 for instructions.
6. Allocate a new or existing target data set. Designate a target data set to store your configuration information. See "Steps for allocating the target data sets" on page 19 for instructions.
7. Define all variables for your particular customization. Using the worksheets that you completed, fill in a value for each variable. See "Steps for defining variables" on page 19 for instructions.

   Note: Ensure the accuracy of the variables and customized settings that you enter at this stage. Specification (spelling) errors and omissions are very difficult to diagnose and correct later on.
8. Save your customization variable information to a data set. This option saves both your variables and target library settings in case you want or need to use them again. Then they are available if you want to reconfigure or use the same values for another configuration or environment. This is also an important precautionary step in the case when you run into an error during the generate step--instead of having to reenter all the information, you just reload everything from the data set. See "Steps for saving the customization variables" on page 20 for instructions.
9. Save security domain customization variable information (in only the ″Configure security domain″ task) This option saves both your security domain variables and target library settings for use in the other options. See "Steps for saving the security domain variables" on page 20 for instructions.
10. Generate the customized jobs. This step creates the installation job streams you will run when stepping through the generated instructions. See "Steps for generating customization jobs" on page 21 for instructions.

    Note: At this point, you may want to save the variables to a data set again to catch any generation values you may have specified.
11. (If applicable) Correct any errors flagged by the Dialog and regenerate the customized jobs. As the jobs and data members are created (or replaced), watch for errors. If there is an ENQ on the PDS, the Dialog will stop generating the tailored jobs for that data set and you will not have a complete set of generated jobs and data.
12. Review the saved variables in the data set to which you saved them. Sort the file on the columns that contain the variable values (typically columns 12-20) using the ISPF SORT command, then scan it for the following:

    Typos in the data set high-level qualifiers
    If there is a typo in one, it may stand out because the file will list it separately from the others.

    Typos in server names and user IDs
    If you use a column-specific naming convention, these will also stand out if there is a typo.

    Unique UIDs & GIDs
    It is important that these are all unique because the Dialog does not check for UIDs and GIDs that are already assigned in the RACF database.

13. View, print and follow the generated instructions. These instructions step you through the process of running the jobs created by the Customization Dialog. This is the manual portion of actual customization. See “Steps for viewing and following the generated customization instructions” on page 22 for instructions.

After these steps are completed, you should have the desired configuration up and running smoothly. Follow the steps in this section again for whichever task you wish to perform, using the variables you saved from the first configuration as a basis for tailoring the variables for the others. (For example, you can use the same parameters and target CNTL, DATA, and SAVECFG data sets.)

After you are finished with the Customization Dialog, continue your customization with the Using the administrative console. To access the console, in a Web browser, type http://*your_fully_qualified_server_name*:*desired_console_port*/admin, where *your_fully_qualified_server_name* is the fully qualified host name for the machine that contains the administrative server and *desired_console_port* is the port on which you want the Administrative Console to run (typically port 9080 or 9090). See the article Starting and stopping the administrative console for more information.

Steps for starting the Customization Dialog

You must have the product code installed and have access to the product data sets.

Rules: Regarding your display:

- Your logon display must support 3270 emulation and be set to a minimum of 32 rows by 80 columns (32 x 80) in order for the ISPF Customization Dialog to run.
- If you have a 32-row display and use the ISPF split screen function, deselect “Always show split line” on the ISPF Settings panel and split the screen at the extreme top or bottom of the display. This prevents the split screen line from displaying and lines in the Customization Dialog from being obscured. Other uses of split screen will obscure lines in the Customization Dialog.
- If you have a 32-row display, you cannot display the PF key settings. Displaying the PF key settings will obscure lines at the bottom of the Dialog panels. Issue

  ```
  PFSHOW OFF
  ```

Perform the following steps to run the Customization Dialog:

1. From the ISPF command line, enter the following: *ex ’hlq.sbboclib(bbowstrt)’ ’options’* where:

   hlq High-level qualifier for the SBBOCLIB data set.

   options

   Command options. Enclose any and all options in a single set of quotes.

   appl(*value*)

   Specifies the ISPF application name. This option creates unique ISPF profiles, usually stored in the ″userid.ISPF.ISPFPROF″ data set, that are useful when you want to keep saved variables separate from those in other target environments. The default value is:

   ```
   BBO5
   ```

   lang(*value*)

   Specifies the national language. Values can be either `ENUS` (English) or `JAPN` (Japanese). The default is `ENUS`.

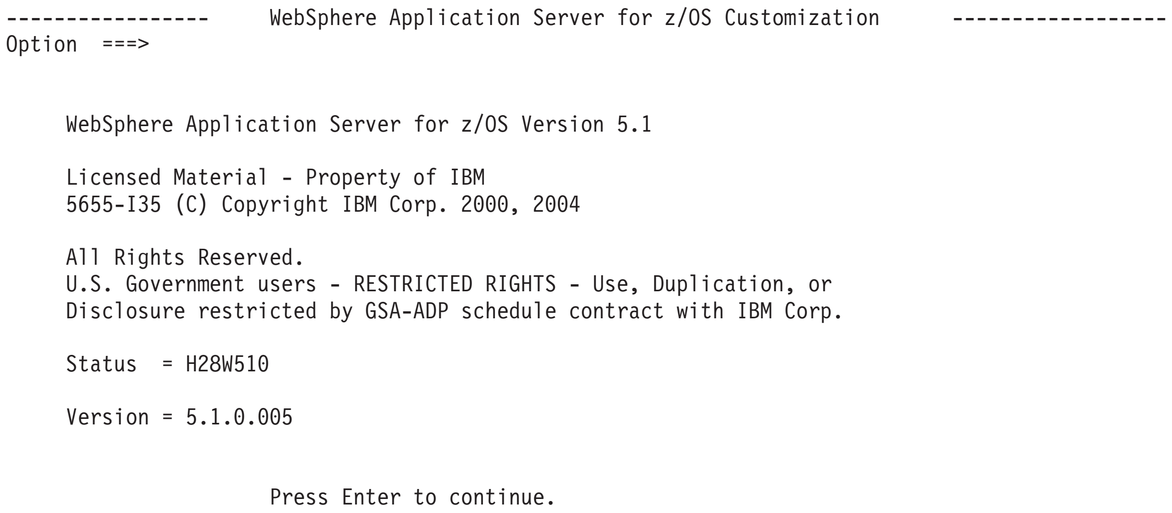
   Note: Some messages will still appear in English even if you run the Dialog in Japanese.

   Examples:

   ```
   ex 'bbo.sbboclib(bbowstrt)'
   ```

   ```
   ex 'bbo.sbboclib(bbowstrt)' 'lang(japn)'
   ```

   Result: You see the splash screen: 5.1 +

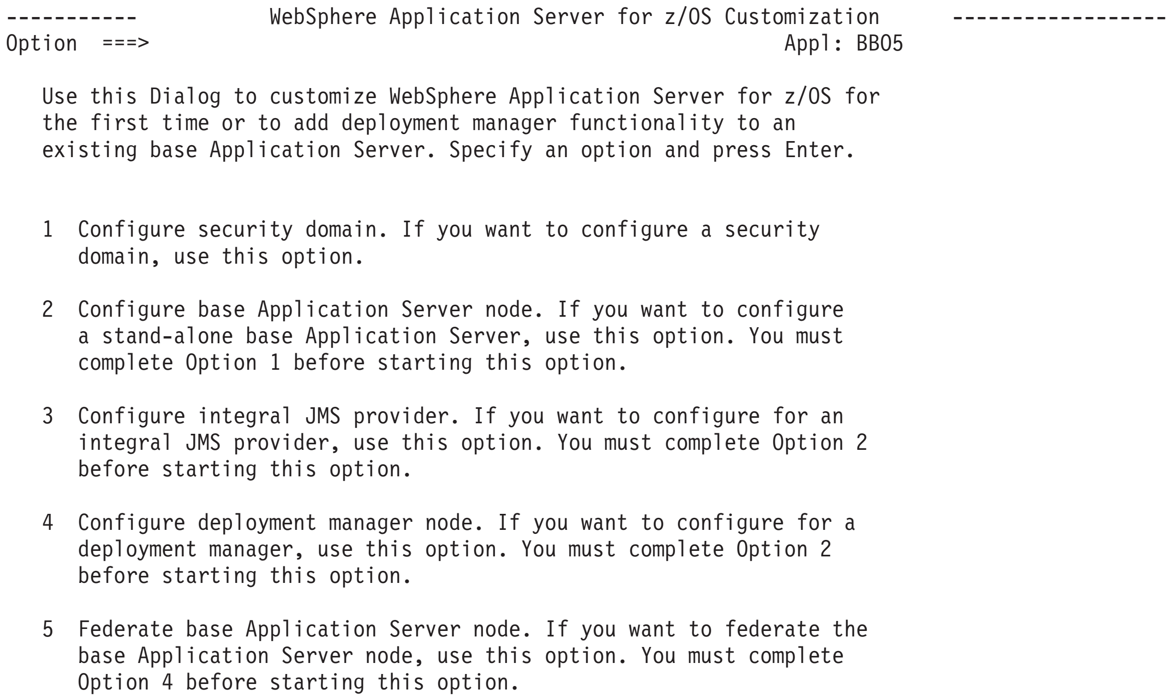


```
-----------------     WebSphere Application Server for z/OS Customization      ------------------
Option  ===>


     WebSphere Application Server for z/OS Version 5.1

     Licensed Material - Property of IBM
     5655-I35 (C) Copyright IBM Corp. 2000, 2004

     All Rights Reserved.
     U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
     Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

     Status  = H28W510

     Version = 5.1.0.005


                      Press Enter to continue.
```

2. Press Enter. **Result:** You see the ″THIRD PARTY LICENSE TERMS AND CONDITIONS, NOTICES AND INFORMATION″ panel, which is too large to reprint here.
3. Type ″END″ as instructed, then ″Y″ to finish the license section.

   Result: You see the following panel:

   5.1 + You see the following panel: 5.1 +



```
-----------           WebSphere Application Server for z/OS Customization      ------------------
Option  ===>                                                     Appl: BBO5

   Use this Dialog to customize WebSphere Application Server for z/OS for
   the first time or to add deployment manager functionality to an
   existing base Application Server. Specify an option and press Enter.


   1  Configure security domain. If you want to configure a security
      domain, use this option.

   2  Configure base Application Server node. If you want to configure
      a stand-alone base Application Server, use this option. You must
      complete Option 1 before starting this option.

   3  Configure integral JMS provider. If you want to configure for an
      integral JMS provider, use this option. You must complete Option 2
      before starting this option.

   4  Configure deployment manager node. If you want to configure for a
      deployment manager, use this option. You must complete Option 2
      before starting this option.

   5  Federate base Application Server node. If you want to federate the
      base Application Server node, use this option. You must complete
      Option 4 before starting this option.
```

You have finished starting the Customization Dialog and can now choose the option for the path you would like to follow.

Steps for loading customization variables

You must start the Customization Dialog and select your desired task option.

This section tells you how to complete the ″Load customization variables″ option that is in all of the main tasks you can perform.

When you first run the Customization Dialog, the Dialog loads initial default customization variables. If you have previously run the Dialog and saved your variables, you can follow these instructions to reload those saved variables.

1. On the main Dialog panel, type the appropriate number or letter in the *Option* field to select ″Load customization variables″.
2. Press Enter. **Result:** You see the Load Customization Variables panel, which looks similar to this:

   ```
   -----------------     WebSphere Application Server for z/OS Customization      ------------------
   Option  ===>

   Load Customization Variables

    Specify the name of a data set containing the customization variables,
    the press Enter to continue.

    IBM-supplied defaults are in

    Data set name:


    If this data set is not cataloged, specify the volume.

    Volume:
   ```
3. Follow the options in order and enter information from the Load customization variables worksheet (be sure to enclose your data set name in single quotes), then press Enter.

You are done when you successfully load the variables.

Steps for loading security domain variables

You must start the Customization Dialog and select your desired task option.

This section tells you how to complete the ″Load security domain variables″ option that is in some of the main tasks you can perform.

When you first run the Customization Dialog, you need to run through the task that configures your security domain. You then use the resulting variables in the other tasks. If you have previously run the Dialog and saved your security domain variables, you can follow these instructions to reload those saved variables.

1. On the main Dialog panel of affected tasks, type the appropriate number in the *Option* field to select ″Load security domain variables″.
2. Press Enter. **Result:** You see the Load Security Domain Variables panel:

   ```
   -----------------     WebSphere Application Server for z/OS Customization      ------------------
   Option  ===>

   Load Security Domain Variables

    Specify the name of a data set containing the security domain variables,
    then press Enter to continue.

    IBM-supplied defaults are in ''

    Data set name:


    If this data set is not cataloged, specify the volume.

    Volume:
   ```
3. Follow the options in order and enter information from the Load security domain variables worksheet (be sure to enclose your data set name in single quotes), then press Enter.

You are done when you successfully load the security domain variables.

Steps for allocating the target data sets

You must start the Customization Dialog and select your desired task option.

This section tells you how to complete the Allocate target data sets option that is in all of the main tasks you can perform. Complete this step if you didn't specify an HLQ (high-level qualifier for the WebSphere Application Server for z/OS installation data sets) when you first started the Customization Dialog or if you want to change it.

1. On the main Dialog panel, type the appropriate number in the *Option* field to select ″Allocate target data sets″.
2. Press Enter. **Result:** You see a panel that looks similar to the following:

```
-----------------     WebSphere Application Server for z/OS Customization      ------------------
Option  ===>

 Allocate Target Data Sets

 Specify a high level qualifier (HLQ) and press Enter to allocate the
 data sets to contain the generated jobs and instructions. You can
 specify multiple qualifiers (up to 39 characters).

 High level qualifier:                                         .CNTL
                                                               .DATA

 The Dialog will display data set allocation panels. You can make
 changes to the default allocations, however you should not change
 the DCB characteristics of the data sets.

    .CNTL  - a PDS with fixed block 80-byte records to
             contain customization jobs.

    .DATA  - a PDS with variable length data to contain
             other data produced by the Customization Dialog.
```

3. Follow the options in order and enter information from the Allocate Target Data Sets worksheet, then press **Enter**.

You are done when the data set allocation succeeds.

Steps for defining variables

You must start the Customization Dialog and select your desired task option.

This section describes how to complete the ″Define variables″ option that is in all of the main tasks you can perform.

1. On the main Dialog panel, type the appropriate number in the *Option* field to select ″Define variables″.
2. Press **Enter**.

   Result: Depending on what you are configuring, you see a panel similar to the following:

```
-----------------     WebSphere Application Server for z/OS Customization      ------------------
Option  ===>

Define Variables to configure base Application Server node

   Specify a number and press Enter to define the WebSphere Application
   server for z/OS variables. You should review all of the variables in
   each of the sections, even if you are using all of the IBM-supplied
   defaults.
   After you complete all sections, press PF3 to return to the main menu.
```

```
                                                   Completed?
  1 - System Locations (directories, HLQs, etc)
  2 - System Environment Customization
  3 - Server Customization
  4 - View Security Domain Configuration Panels
```

This example is from the ″Configure base Application Server node″ option, but the steps are relatively the same no matter which option you choose.

3. Follow the options in order and enter information from the define variables worksheets. **Recommendation:** Use the IBM default names the first time you customize WebSphere Application Server for z/OS to make the customization instructions easier to follow.

You are done when you finish all the ″Define Variables to configure...″ panels and a `Y` is displayed in the Completed? column next to each option. You may then move ahead and save the variable definitions.

Steps for saving the customization variables

You must have completed the definition of the variables, which is outlined in the article Steps for defining variables.

This section tells you how to complete the ″Save customization variables″ option that is in all of the main tasks you can perform.

1. On the main Dialog panel, type the appropriate number or letter in the *Option* field to select ″Save customization variables″.
2. Press Enter. **Result:** You see the Save Customization Variables panel, which looks similar to this:

```
-----------------     WebSphere Application Server for z/OS Customization      ------------------
Option  ===>

Save Customization Variables

  Specify the name of a sequential data set to contain the
  customization variables, then press Enter to continue. If the
  data set does not exist, the Dialog displays the Allocate New
  Data Set panel, with which you can allocate a data set.

    Data set name:
```

3. Follow the options in order and enter information from the Save customization variables worksheet (be sure to enclose your data set name in single quotes), then press Enter.

You are done when you successfully save the variables.

Steps for saving the security domain variables

You must have completed the definition of the security domain variables in the Configure Security Domain task.

This section tells you how to complete the ″Save security domain variables″ option that is in the Configure Security Domain task.

1. On the main Dialog panel, type the appropriate number in the *Option* field to select ″Save security domain variables″.
2. Press Enter. **Result:** You see the Save security domain variables panel, which looks similar to this:

```
-----------------     WebSphere Application Server for z/OS Customization      ------------------
Option  ===>

Save Security Domain Variables

  Specify the name of a sequential data set to contain the security domain
  variables, then press Enter to continue. If the data set does not exist,
 the Dialog displays the Allocate New Data Set panel, with which you
 can allocate a data set.

Data set name:
```

3. Follow the options in order and enter information from the Save security domain variables worksheet (be sure to enclose your data set name in single quotes), then press Enter.

You are done when you successfully save the variables.

Steps for generating customization jobs

You must complete the process of Defining variables, which is outlined in “Steps for defining variables” on page 19.

This article describes how to complete the ″Generate customization jobs″ option that is in all of the main tasks you can perform.

Recommendation: When you have finished entering all your customization data, before you generate the customization jobs, use the ″Save customization variables″ option to save your customization variables for future reference.

1. On the main Dialog panel, type the appropriate number in the *Option* field to select ″Generate customization jobs″.
2. Press Enter.

   Result: If all variables are defined correctly, you see the Specify Job Cards panel, which looks similar to this:

   ```
   -----------------     WebSphere Application Server for z/OS Customization      ------------------
   Option  ===>

   Generate Customization Jobs

    This portion of the Customization Dialog generates the jobs you must
    run after you complete this Dialog process. You must complete the
    customization process before you generate the jobs with this step.
    If you have not done this, please return to that step.

    Jobs and data files will get generated into data sets:
      'hlq.CNTL'
      'hlq.DATA'
    If you wish to generate customization jobs using other data sets, then
    exit from this panel and select option 1 (Allocate target data sets).

    All the jobs that will be tailored for you will need a job card.
    Please enter a valid job card for your installation below. The
    file tailoring process will update the job name for you in all the
    generated jobs, so you need not be concerned with that portion of
    the job cards below. If continuations are needed, replace the
    comment cards with continuations.

    Specify the job cards, then press Enter to continue.

    //jobname  JOB (ACCTNO,ROOM),'userid',CLASS=A,REGION=0M
    //*
    //*
    //*
   ```

   Note: Pay particular attention to the displayed target data sets.
3. Fill in the job card information using the Generate customization jobs worksheet, then press **Enter**.

You are done when all the jobs are generated. You may then move ahead to viewing the generated jobs.

Steps for viewing and following the generated customization instructions

You must complete the generation of the customization jobs.

This section tells you how to complete the ″View instructions″ option that is in all of the main tasks you can perform. Once you view the steps and verify your information, you can then follow the steps.

Note: It is very helpful to also print the instructions for reference and to use as a checklist for all the installation and customization steps you need to follow.

There are four phases you go through when following the generated instructions:

1. Perform manual system updates:
   - WLM
   - SCHEDxx
   - APF
   - SMF
   - BPXPRMxx
   - LPA
   - Link list.
2. Run system-related jobs:
   - Logstream
   - RRS
   - CTRACE
   - RACF profiles and resources.
3. Run your server’s HFS-related jobs:
   - Create and mount the HFS file system.
   - Populate the HFS with directories and XML.
   - Create the was.env files for the location service daemon and servers.
   - Batch install the Administration application.
   - Clean up the HFS structure.
4. Start your servers:
   - Start CTRACE.
   - Start your controller. (The controller automatically starts the location service daemon, and WLM starts the servants.)

1. On the main Dialog panel, type the appropriate number in the *Option* field to select ″View instructions″.
2. Press Enter. You will then see the generated instructions file.
3. View the instructions.
4. Check the instructions for any typographical errors you may have introduced when entering your variables or other information. If necessary, modify your variables using the Dialog, regenerate the instructions, and re-review the results.

   Note: Do **not** attempt to fix a typo or make a change by modifying the generated output. Many of the variables are used in multiple members of the target data sets, so, if you do not change them all, you will run into very difficult-to-diagnose problems.
5. Print the instructions according to your local print procedures.

You are done when you view or print the instructions. You can now go on to follow them, using the worksheet for your task as a guide.

Choosing between a base or Network Deployment configuration

Presented is a table that contains specifics on the differences between a WebSphere Application Server for z/OS base cell and Network Deployment cell.

	Base cell	Network deployment cell
Configuration:	Set up each base server node through the ISPF Customization Dialog. Set up additional servers within the node also through the ISPF Customization Dialog.	Set up each deployment manager node through the ISPF Customization Dialog. Add base server nodes to the deployment manager cell through the ISPF Customization Dialog.
Address spaces:	Min: 3 (location service daemon, controller, servant)	Min: 6 (location service daemon, Application Server controller, Application Server servant, deployment manager controller, deployment manager servant, node agent)
	Max: Limited only by resources.	Max: Limited only by resources.
Administrative isolation:	Each base server node is a separate administrative domain.	All nodes in the cell are in the same administrative domain.
Operational isolation:	You may start and stop servers independently. Each server has an independent, unshared JNDI namespace.	You may start and stop servers independently. The JNDI namespace is shared among all servers in the cell.
Application servers allowed to have multiple servants?	Only if the Administration application is not present	Yes
Clustering available?	No	Yes

Preparing the base z/OS environment

You must ensure you have an adequate skill level to support the base z/OS environment.

Some z/OS function customization steps you need to do for WebSphere Application Server for z/OS can be done before you install and customize WebSphere Application Server for z/OS itself. We have put those tasks into this chapter, allowing you to segment your work.

1. "Determining WebSphere Application Server for z/OS system requirements" on page 24.
2. "Updating your TCP/IP network" on page 27.
3. Planning to secure your environment.
4. "Setting up workload management (WLM)" on page 29.

Once you complete these steps, you need only to verify completeness:

- With the *WebSphere Application Server V5.1: Program Directory* that all prerequisites have been met. Also verify with the PSP bucket to ensure the latest adjustments have been met.
- With your z/OS system programmers, DB2 (or other DB administrators), and other technical staff to ensure your environment is ready

Determining your skill needs

In assembling your project team, you should consider the skills you need to implement WebSphere Application Server for z/OS. This topic discusses the recommended skill set necessary to support the following configurations:

- Basic configurations
- Advanced configurations
- Production environments

Documentation to support the z/OS skills described in this article can be found at this Web site: z/OS Internet Library

Note: Along with the skills listed below and depending on the needs of the applications you deploy, you may also need skills to configure the resource managers your applications require. Such skills are CICS, DB2 (if applicable), IMS, and MQ.

For basic configurations:

Below are the recommended skills necessary to support a basic configuration:

- z/OS UNIX System Services and the hierarchical file system (HFS) - to set up a functional HFS and UNIX environment
- eNetwork Communications Server (TCP/IP) or equivalent - to configure connectivity for WebSphere Application Server for z/OS clients and servers
- Resource recovery services (RRS) - to implement resource recovery services and to support two-phase commit transactions
- Security Server (RACF), or the security product you use - to authenticate WebSphere Application Server for z/OS clients and servers, and authorize access to resources
- Workload management (WLM)
- SMP/E and JCL
- System logger - to set up logstreams for RRS and the WebSphere Application Server for z/OS error log
- Java

For advanced configurations:

When dealing with advanced configurations, you need all the same skills as for basic configurations, plus the following:

- Parallel sysplex - to implement multisystem configurations
- CICS
- IMS
- MQ

For production environments:

As you move your system toward a production environment, you need to have the following system skills available:

- Automatic restart management (ARM)
- System Automation, if you have it installed, or whichever automation you prefer to use
- Sysplex, if you plan to use WebSphere Application Server for z/OS in a cell that spans systems
- Secure Sockets Layer (SSL) or Kerberos, if you plan to have security in a distributed network
- Advanced environment SSL if you enable security
- RMF or other performance measurement systems
- Webserver, if you plan to support HTTP clients

Determining WebSphere Application Server for z/OS system requirements

The following sections describe system requirements for WebSphere Application Server for z/OS.

Hardware requirements

The hardware requirements for this product are any hardware that supports z/OS V1R2 or later. However, there are significant performance advantages for those applications doing floating point arithmetic if the machine has binary floating point hardware, such as S/390 Parallel Enterprise Server-Generation 5 and later systems.

The LPAR in which the WebSphere Application Server for z/OS runtime and initial Application Servers run requires a minimum of 512 MB of real storage. You may need to increase the real storage size depending on the size and number of Application Servers you deploy. In addition, you may want to increase your JES spool space if you use WebSphere Application Server for z/OS tracing options to the SYSPRINT DD dataset.

You should plan on an extra 3390-3 volume for the distribution and configuration HFS datasets, two additional 3390-3 volumes for the WebSphere product code, and two further 3390-3 volumes if you will use the Integrated JMS Provider.

Recommendation: Increase your paging subsystem by one 3390-3 volume if your storage is constrained.

Software requirements

You must install, enable, and configure the following z/OS or OS/390 elements, features, and components. Consult the Program Directory and PSP bucket for the required corrective service.

All of the z/OS sources referenced are available at this Web site:
http://www.ibm.com/servers/eserver/zseries/zos/bkserv/

- z/OS configured as a sysplex (at minimum, you need a monoplex). For details, see *z/OS MVS Setting Up a Sysplex*.

  If you want to utilize the dynamic application environment functionality of WebSphere Application Server for z/OS V5.1, you must run at least z/OS Version 1.2.
- z/OS UNIX System Services (z/OS UNIX) with a hierarchical file system (HFS). For details, see *z/OS UNIX System Services Planning*.

  Note: When setting up the UNIX Systems Services environment, ensure you specify enough threads, files, and processes in your BPXPRMxx member of parmlib. Here is a sample of recommended values:

  ```
  MAXTHREADS:10000
  MAXTHREADTASKS:5000
  MAXFILEPROC:10000
  MAXSOCKETS in the AF_INET domain:12000
  ```
- WebSphere Application Server for z/OS V5.1 makes additional use of the /tmp directory and the file system that supports it. You may need to increase the size of the file system before you attempt to configure WebSphere Application Server for z/OS. One of the symptoms of an undersized /tmp file system is failure to copy the .ear file for the administrative console application in the BBOWIAPP job.

  Determine the size of the file system by entering the `ISHELL` command at the root ″/″ level in ISPF and using the line command `U` on the line for /tmp. This will display a box with information on the file system. Multiply the ″Block Size″ field by the ″Total Blocks″ field to see the total space in the file system in bytes. Multiply the ″Block Size″ field by the ″Available Blocks″ field to see the available space in bytes.

  Note: The administrative console application .ear file is over 10 megabytes, so if there is not at least that much available space, action must be taken to increase the amount of space.

  To increase the size of your /tmp directory file system, perform one of the following, depending on your file system type:
  - If the file system is an HFS (which is non-temporary), then you must unmount the filesystem, delete it, reallocate it with a larger amount of space, and then remount it.
  - If the file system is a TFS (a temporary file system which is essentially a memory-resident file system), the file system must be unmounted and then remounted with more space. See *UNIX Systems Services Planning manual*, GA22-7800, for more information on TFS specification.

  Example: Here is an example mount command:

  ```
  MOUNT FILESYSTEM('/TMP') TYPE(TFS) MOUNTPOINT('/tmp') PARM('-s 20')
  ```

  where `20` represents the desired amount of space in megabytes. If you do not specify a space value, the undesirable small default value of 1 megabyte will be assumed.

- eNetwork Communications Server (TCP/IP) or equivalent. In this manual, we refer to eNetwork Communications Server, but you may substitute an equivalent product. For details, see *z/OS Communications Server: IP Migration*.
- Workload management (WLM) set up in goal mode. For details, see *z/OS MVS Planning: Workload Management*.

  If you are not running z/OS V1.2 or above with the WLM-DAE support PTF (APAR OW54622), you must complete some additional steps for WLM.
- Resource recovery services (RRS). For details, see *z/OS MVS Programming: Resource Recovery*.
- A security product such as SecureWay Security Server (RACF). In this manual we refer to Security Server in examples, but you may substitute an equivalent security product. For details, see *z/OS Security Server RACF Migration*.
- IBM Developer Kit for OS/390 Java 2 Technology Edition Version 1.1, an element of WebSphere Application Server for z/OS, but also available separately. The SDK level supported by this product is 1.4.1.

  Later releases of the IBM Developer Kit for OS/390 Java 2 Technology Edition are not supported.

Regarding optional functions, consult the following table:

Table 1. Software requirements for optional functions

If you plan to use . . .	Then you need . . .	Notes . . .
Kerberos security	OS/390 SecureWay Security Server Network Authentication and Privacy Service for OS/390	For z/OS, this support is part of SecureWay Security Server.
Cryptography	Cryptographic Services System SSL, a component of Cryptographic Services Base, an element of z/OS.	For details, see *z/OS System Secure Sockets Layer Programming*.
EJB roles to secure Web components or enterprise beans	z/OS V1R2 Security Server (RACF) or equivalent	
Java Message Service (JMS)	• Integrated JMS provider • Either of the following: – MQ stack from WebSphere Application Server for z/OS V5.1 – Full function MQ using MQ 5.3.1	MQSI, a pub sub product that supports z/OS, does not support new function like MDBs.
WebSphere Application Server for z/OS IMS Connect V8 support	IMS/TM 6.1.0	
WebSphere Application Server for z/OS CICS Transaction Gateway 5.0.1 support	CICS/TS 1.3	
DB2	DB2 V7.1	DB2 is no longer a prerequisite for WebSphere Application Server for z/OS, though some functions may still require it. Check your individual configuration.
DB2 SQLJ in J2EE application components	DB2 V7.1 PTF UQ59527	DB2 is no longer a prerequisite for WebSphere Application Server for z/OS, though some functions may still require it. Check your individual configuration.
Logging	z/OS system logger	For details, see *z/OS MVS Setting Up a Sysplex*.
Connectors	WebSphere Application Server for z/OS V5.1 supports any resource adapter that is designed to use the 1.0 level of the J2EE Connector Architecture (JCA). Install the listed IBM connectors if you want to access your IMS or CICS legacy data.	

Table 1. Software requirements for optional functions (continued)

If you plan to use . . .	Then you need . . .	Notes . . .
	For the CICS Transaction Gateway ECI connector: • CICS Transaction Gateway V5.0.1 • CICS Transaction Server V1.3 • WebSphere Studio Application Developer IE V5.0	
	For the IMS Connector for Java: • IMS Connect for z/OS V2.1 • IMS V8 • WebSphere Studio Application Developer IE V5.0.1	
	For the IMS JDBC Connector: • IMS V8	

Installing WebSphere Application Server for z/OS

The term ″installing,″ in regard to WebSphere Application Server for z/OS, refers to only the SMP/E process of loading the code onto the system. This article discusses that installation. To get the product actually up and running, you need to go on and ″customize″ it. Customization is discussed in “Setting up a base Application Server cell” on page 37.

Installing the code through SMP/E

To install the code through SMP/E, follow one of two documents depending on what you use:

- If you use CBPDO, follow the *WebSphere Application Server for z/OS V5.0: Program Directory*.
- If you use ServerPac, follow *ServerPac: Installing Your Order*.

You can find further information on the eSupport Web site at http://www.ibm.com/software/webservers/appserv/zos_os390/support.html, or check the PSP buckets or contact the IBM Software Support Center.

- You can change the high-level qualifier of the installed data sets (not recommended) or the middle-level qualifier.
- If you are installing from a driving system, make sure the maintenance level of the target system meets requirements for WebSphere Application Server for z/OS.
- Make sure the product code HFSes are mounted at `/usr/lpp/java` and `/usr/lpp/zWebSphere/V5R1M0`, or at similar mount points of your choice.

Preparing your subsystems

Before you install and configure WebSphere Application Server for z/OS, you need to properly prepare any related subsystems. Follow the steps below to ensure you take care of everything.

1. Update your TCP/IP network
2. Set up workload management (WLM)
3. Set up resource recovery services
4. Set up Java Database Connectivity
5. Verify DB2 settings for WebSphere concurrency control management

Updating your TCP/IP network

WebSphere Application Server for z/OS follows the CORBA standard, Internet Inter-ORB Protocol (IIOP), for communications. Accordingly, you must consider changes to your TCP/IP network and modify the TCP/IP configuration.

This section provides background information about changes you will need to make to your Domain Name Server (DNS) and TCP/IP. The actual steps to perform are in the customized instructions of the Customization Dialog.

Tips on TCP/IP and WebSphere Application Server for z/OS

Consider the following for your TCP/IP network on z/OS.

- You can get started with a simple Domain Name Server (DNS) name server and a single z/OS image, but you should design your initial configuration with growth in mind. You may, for instance, intend to expand your business applications beyond the monoplex to a full sysplex configuration for performance reasons or to prevent a single point of failure. Several considerations come to bear here.

  Several DNS implementations and network router implementations allow the use of a generic location service daemon IP name while dynamically routing network traffic to like-configured servers. If you intend to expand your system beyond a monoplex, it might be worthwhile to use one of these implementations from the start. Non-round-robin DNS name servers limit your ability to expand without retrofitting a name server that allows dynamic network traffic routing.

  Recommendation: If you are running in a sysplex, set up your TCP/IP network with Sysplex Distributor. This makes use of dynamic virtual IP addresses (DVIPAs), which increase availability and aid in workload balancing.

  Beyond Sysplex Distributor, you have your choice of the following DNS and router implementations on or off z/OS:
  – Non-round-robin DNS name servers.
  – Round robin DNS name servers.
  – Network routers, such as the IBM Network Dispatcher.
- Select the location service daemon IP name. For your base Application Server, choose the host name of the server under which you are running. For your deployment manager, choose a generic IP name that can resolve to any or all of the systems where location service daemons run.

  You must define the location service daemon host IP name during installation and customization. Use the location service daemon IP name you chose.

  Note: The administrative console has a location service daemon configuration page on which you set location service daemon variables.
- Select the port for the location service daemon server. If you change the location service daemon port number, you can access existing objects after you recycle all your servers. You cannot, however, access the following:
  – Any object handles your application stored to disk
  – Any object references your application stored in the persistent contexts of the name space.

  If you install WebSphere Application Server for z/OS V5.1 on a system that already contains WebSphere Application Server V4.0.1 for z/OS and OS/390, double-check your current location service daemon port value (default for WebSphere Application Server V4.0.1 for z/OS and OS/390 is 5555).
- Set location service daemon port numbers and IP addresses. These are initially set in the Customization Dialog, but you can subsequently change them in the Administrative Console. Access the location service daemon configuration page through the Administrative Console navigation bar (on the left side of screen) under System Administration. If you need to use the Internet Inter-ORB Protocol (IIOP) through a firewall, ensure that your firewall supports IIOP.

  When recovering a server somewhere other than its configured system, ensure that the same port is not already in use on the system on which it is recovering. If it is, configure the server with a unique port to avoid a conflict.

  If comparing WebSphere Application Server for z/OS V5.1 and WebSphere Application Server V5.1 for other platforms, realize that only WebSphere Application Server for z/OS V5.1 has an ORB SSL port.

  HTTP and HTTPS ports are found in individual servers under the Web container transports, which are in the administrative console as ’additional properties’ on the Web container configuration page (which is off the server configuration page).

Watch for HTTP transport port conflicts if you previously installed WebSphere Application Server V4.0.1 for z/OS and OS/390.

Ensure you set up the following port assignments (along with those in the z/OS port assignments chart) on servers that require them in the administrative console:

– ORB port
– ORB SSL port
– Web container transport port
– Web container transport SSL port

See the administrative console and the InfoCenter for more information on the WebSphere variables and how to set their values.

You define ports differently depending on whether they are for the first server or subsequent servers. The first server you create is defined, along with its ports, through the Customization Dialog. You have the ability to explicitly specify the ports as you define the server. Subsequent servers and their ports are defined through the administrative console. This means that you define the server first and the ports are automatically assigned. Then, once defined, you can inspect and change the port definitions through the administrative console.

- Some ports, such as the ORB SSL port and the server startup status port, are obtained dynamically.
- Other TCP/IP-related activities include setting up NFS, Web server and Kerberos, which are all optional.
- If you use the DNS on z/OS, you may wish to change the refresh timer interval (-t value) associated with the named location service daemon. The -t value specifies the time (nn, in seconds) between refreshes of cell names and addresses and of the weights associated with those names and addresses. The default is sixty seconds. Reducing the -t value will shorten the lapse time required to register the location service daemon IP name with the DNS, but will also increase DNS processing overhead. In our testing, we used an interval of 10 seconds.

  If you use the z/OS DNS, you have to set a location service daemon variable. Do this by setting WebSphere Variable at cell level:

  ```
  daemon_wlmable=1
  ```

  Note: You can perform this for only one cell in a sysplex at a time.

  For details, see *z/OS Communications Server: IP Configuration Reference*.

Setting up workload management (WLM)

WebSphere Application Server for z/OS uses the workload management (WLM) function in z/OS to manage workloads. This section helps you get started and is sufficient to get a functioning WebSphere Application Server for z/OS system.

WebSphere Application Server for z/OS requires that z/OS run workload management in goal mode. If your system runs in compatibility mode, you must implement goal mode. For details on workload management, see *z/OS MVS Planning: Workload Management*, which is available on the z/OS Internet Library Web site.

In addition to setting up workload management in goal mode, you need to define workload management policies for WebSphere Application Server for z/OS servers and your business application servers. This section discusses specifics for the run-time servers. For details on workload management and business applications, see the assembling applications information in the z/OS view of the WebSphere Application Server information center, which you can access via the WebSphere Application Server for z/OS library Web site.

To get started, you do not need to define special classification rules and work qualifiers, but you might want to do this for your production system.

Note: After you apply service level W500104 (PTF UQ79825), you might want to set up a rule to combat performance problems related to post-install. See Workload management (WLM) tuning tips for z/OS for more information.

Each J2EE Application Server must have a WLM application environment so it can start servant regions. The application environment name is specified as the ″cluster transition name″ in the Customization Dialog.

WebSphere Application Server for z/OS servers (with the exception of the location service daemon and node agent) have a controller and one or more servants. The regions are started by the respective start procedures.

Tip: If you are running z/OS V1.2 or above with the WLM-DAE support PTF (APAR OW54622), you do not need to continue.

You have to start the controllers for the WebSphere Application Server for z/OS servers yourself. This in turn starts the location service daemon. Workload manager dynamically starts the servants as work requests arrive. Thus, you must create WLM application environments that name servant start procedures to start. For example, specify BBO5ASR as the start procedure name that workload management starts for the server.

Each new server that you create for a business application also must be defined to workload management. For more information, see the assembling applications information in the z/OS view of the WebSphere Application Server information center, which you can access via the WebSphere Application Server for z/OS library Web site.
Use the ISPF application IWMARIN0 to define WLM application environments.

Tip: If you are running z/OS V1.2 or above with the WLM-DAE support PTF (APAR OW54622), you can skip this step.

During the customization process, you will be instructed to create a WLM application environment. This section shows how to use the WLM ISPF Dialog to accomplish this.

You must have access to the IWMARIN0 application and be able to update the workload management policies to complete this process.

Use the ISPF application IWMARIN0 to define WLM application environments according to Default server values for WebSphere Application Server for z/OS V5.

Using IWMARIN0 (example of steps for defining an application environment):

Workload management must be running in goal mode, and you must have access to a WLM definition, either saved in a WLM definition data set, or active in the WLM couple data set. The user of IWMARIN0 must have update access to the RACF FACILITY class profile MVSADMIN.WLM.POLICY. Perform the following steps to create the BBOC001 application environment.

1. Open the main panel by issuing IWMARIN0. Either load a WLM goal mode definition from a WLM definition data set, or extract a working goal mode definition from the WLM couple data set. Then choose option 9:

```
 File Utilities Notes Options Help ---------------------------
 Functionality LEVEL003     Definition Menu            WLM Appl LEVEL004
Command ===> ______________________________________________________________
Definition data set . . : CB.MYCB.WLM
Definition name . . . . .CB390 (Required)
Description . . . . . . . WLMSetup for WebSphere Application Server for z/OS
Select one of the following options. . . . . 9__ 1. Policies
                                                 2. Workloads
                                                 3. Resource Groups
                                                 4. Service Classes
                                                 5. Classification Groups
                                                 6. Classification Rules
                                                 7. Report Classes
```

```
                                                   8. Service Coefficients/Options
                                                   9. Application Environments
                                                   10. Scheduling Environments
```

2. Fill in the field on the next panel as shown:

   ```
   Application-Environment Notes Options Help --------------------------------------------------
         Create an Application Environment
   Command ===> ______________________________________________________________
   Application Environment . . . BBOC001_________________________
   Required Description . . . . . . . . .WebSphere Application Server for z/OS IVT server___
   Subsystem Type . . . . . . . .CB__
   Required Procedure Name . . . . . . . . .BBO5ASR
   Start Parameters . . . . . . .JOBNAME=&IWMSSNM.S,ENV=CELL1.&SYSNAME..
                                          &IWMSSNM_________________________________
                                           _________________________________
   Limit on starting server address spaces for a subsystem instance:
   2        1. No limit
            2. Single address space per system
            3. Single address space per sysplex
               .----------------------------------------------------------------------.
               | Selection List empty. Define an application environment. (IWMAM600) |
               .----------------------------------------------------------------------.
   ```

   When filling out the previous panel, follow these guidelines:

   - The Application Environment name is either the ClusterTransitionName (if the server is not clustered) or the cluster short name (if the server is clustered).
   - The recommended JOBNAME value is `&IWMSSNM.S`. For example, if the server short name is BBOS001, then set `JOBNAME=BBOS001S`.
   - `ENV=...` must be of form *cell_short_name.node_short_name.server_short_name*
     - In a base server, the location service daemon default is that both *cell_short_name* and *node_short_name* are the same as *system_name*.
     - In a Network Deployment server, the location service daemon default is that both *cell_short_name* and *node_short_name* are the same as *sysplex_name*. As you add base server nodes to the deployment manager server's cell, you have to update the ENV= parameter of each in the WLM application environment definition. The reason for this change is because a base server node assumes the cell name of the deployment manager server once the base server node is added to the deployment manager server's cell.
   - Change `CELL1` to your cell_short_name and `&SYSNAME` to your node_short_name as needed. Otherwise, the defaults will be assumed.
   - The value of the `&IWMSSNM` variable (defined by WLM) is the server short name.

3. Save the application environment. The following panel appears:

   ```
   Application-Environment Notes Options Help
   ---------------------------------------------------------------------------
            Application Environment Selection List Row 1 to 12 of 12
   Command ===> ______________________________________________________________
   Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete,
        /=Menu Bar
   Action  Application Environment  Name           Description
       __ BBOC001                                  WebSphere Application Server for z/OS IVT server
   ******************************* Bottom of data ********************************
   ```

4. From the Utilities menu, select Install definition.
5. From the Utilities menu, select Activate service policy.
6. From the File menu, select Exit.

You are done when BBOC001 has an application environment.

Recommendations for Resource Recovery Services

WebSphere Application Server for z/OS requires the use of the Resource Recovery Services Attach Facility (RRSAF) of DB2, which in turn requires that Resource Recovery Services (RRS) be set up. If you do not have RRS set up, the Customization Dialog helps you do this. See “Using the Customization Dialog” on page 13.

When setting up RRS, consider the following:

1. You may have already configured RRS for z/OS to exploit WLM-managed DB2 Stored Procedures address spaces. However, if DB2 is the only RRS-compliant resource manager participating in transactional commits, optimizations will cause the system to bypass RRS usage of the system logger. This means that, while your installation may have configured RRS, your log streams might have just minimal activity. WebSphere Application Server for z/OS is an RRS-compliant resource manager and will participate in transactional commits with DB2. Thus, WebSphere Application Server for z/OS will require RRS to start writing data to its system logger log streams. You might need to adjust the size of your log streams.
   - WebSphere Application Server for z/OS has no significant impact on the RM.DATA log.
   - Depending on the transaction policies of both the client and container, you may not see any activity in the MAIN.UR log. This lack of activity is not a problem.
   - Depending on the transactional policy defined for your containers, you may see much more activity in your DELAYED.UR log stream than in the MAIN.UR log stream.

     All RRS transaction logging for WebSphere Application Server for z/OS will occur solely in the DELAYED.UR log stream. You may still want to configure your MAIN.UR log stream so that it can handle a production workload, in case you deploy a new container or the WebSphere Application Server for z/OS infrastructure changes.
   - WebSphere Application Server for z/OS has no significant impact on the RESTART log.
   - There is no reason to change your policy about the ARCHIVE log. Though optional, we suggest you use the ARCHIVE log. It has a small negative effect on performance. Set the retention period for the log as you would normally.
2. The Object Transaction Service in WebSphere Application Server for z/OS cannot detect when it has been restarted in a different logging group, which affects transaction recovery. We recommend you use automatic restart management (ARM) to control restart locations.
3. For structure sizes, we recommend the following for initial setup values. Through experience, you may need to adjust these:

Table 2. Recommended size of log streams

Log stream	Initial size	Size
RM.DATA	1 MB	1 MB
MAIN.UR	5 MB	50 MB
DELAYED.UR	5 MB	50 MB
RESTART	1 MB	5 MB
ARCHIVE	5 MB	50 MB

Check the MAXBUFSIZE on your log streams. If the size is too small, you may encounter DB2 failures.

Details about resource recovery are in *z/OS MVS Programming: Resource Recovery*. Details about the RRS Attach Facility are in *DB2 for OS/390 Application Programming and SQL Guide*.

Guidelines for Java Database Connectivity

Java Database Connectivity (JDBC) provides an interface for Java application programs to access relational data in a database by using dynamic SQL. DB2 supports this application programming interface. For complete information about JDBC and DB2, see *DB2 for OS/390 Application Programming Guide and Reference for Java*. This topic covers guidelines related to WebSphere Application Server for z/OS’s use of JDBC.

- You may use JDBC (dynamic SQL) in your server applications.

- Record the location of the run-time properties file, db2sqljjdbc.properties. You will use the location during the WebSphere Application Server for z/OS customization process. If you customize this file, you may want to keep the customized version in a separate directory such as `/etc` and record its location.
- All J2EE servers and the node must be granted EXECUTE authority on the DSNJDBC plan. If your installation allows public access to the DSNJDBC plan, all you need to do is issue:

  ```
  GRANT EXECUTE ON PLAN DSNJDBC TO PUBLIC
  ```

  If your installation does not allow public access to the DSNJDBC plan, then you must grant EXECUTE authority to all J2EE servers and the node. If you use DB2 secondary authorization IDs, then you can grant the authority to the groups to which the server IDs belong.
- You must use the RRSAF attachment interface (not CAF).

For more information about setting up JDBC and the implications for application programs, see *DB2 for OS/390 Application Programming Guide and Reference for Java*.

Guidelines for DB2 settings for WebSphere concurrency control management

If your installation uses typical DB2 defaults for U-lock management and lock size, certain WebSphere applications that use container-managed Enterprise beans (CMP beans) may encounter deadlocks. The likelihood of encountering deadlocks is entirely dependent on the design and execution pattern of the application. The potential for deadlocks increases with the number and frequency of applications driving concurrent transactions that update the same areas of the DB2 database. If, given your installation's application workload, the potential for deadlocks is high, consider using the following DB2 settings:

- RRULOCK(YES)
- LOCKSIZE(ROW)

For additional details, see the information about settings for the internal resource lock manager (IRLM) in *DB2 Installation Guide*.

Alternative: Your applications may qualify for the optimistic approach to concurrency control management. To determine whether your applications can use optimistic concurrency control, see the topic about controlling concurrent access to persistent data in the assembling applications information in the z/OS view of the WebSphere Application Server information center, which you can access via the WebSphere Application Server for z/OS library Web site.

Recommendations for using memory

WebSphere Application Server for z/OS differs from previous application servers in its use of memory. WebSphere Application Server for z/OS's implementation takes advantage of z/OS's efficient memory management, but, like many of today's newer application servers and languages, it is a large consumer of memory. You may experience some changes from your existing memory usage patterns. This section outlines changes you might need to make:

- For real storage requirements, see related concepts in the InfoCenter.
- In your production cell, we recommend you dynamically load certain WebSphere product load modules into the link pack area (LPA) because the size of the load modules is large, and many address spaces need to refer to those load modules. The load modules for the runtime comprise about 40 MB in size.

  Note: WebSphere Application Server for z/OS V5.1 allows for multiple cells to run on the same system at the same time. Since only one level of WebSphere Application Server for z/OS code can reside in LPA, any WebSphere Application Server for z/OS address spaces using a different WebSphere Application Server for z/OS code level must load the code (SBBOLPA, SBBOLOAD and SBBOLD2) from STEPLIB. This is set in the WebSphere Application Server for z/OS Customization Dialog. However, the BBORTSS5 module must always run from LPA or the link list. This module is specific to WebSphere Application Server for z/OS V5.1.

  Because you are using dynamic LPA, you may run out of ECSA after an IPL if you do not increase CSA at IPL time. Add 40 MB to ECSA in support of WebSphere Application Server for z/OS. You should monitor ECSA after dynamically loading the runtime into LPA. Remember to increase the size of your CSA page data set accordingly.

Note: Do not dynamically load the runtime in the LPA in your development cell. Load the DLLs out of private memory instead of using ECSA.

- If you choose to place the load modules in steplib or in the link list, you must allow for the additional 40 MB as part of each address space's region. A typical WebSphere Application Server for z/OS base configuration runs 3 address spaces, and a typical WebSphere Application Server for z/OS Network Deployment configuration runs 8 address spaces. Each of these reference most of the 40 MB of load modules.
- In addition to placing the load modules in the link pack area, give each address space a region size of at least 512 MB.
- Check to see whether your installation limits region sizes through the IEFUSI exit, JES exits, or TSO segment defaults. All of the WebSphere Application Server for z/OS JCL procedures are shipped with a default REGION=0M, which means you should give them as large a region as possible. If you choose to run from the link pack area, you will need a minimum of 128 MB for the dynamic area. If you choose to run from the link list you will need a minimum of 168 MB (40 MB for load modules and 128 MB for the dynamic area).

  If your IEFUSI exit routine limits the maximum region to a size smaller than what you need (128 MB minimum when you run from the link pack area or 168 MB minimum when you run from the link list), you will get an abend. To fix the problem, either change the IEFUSI exit routine to allow a larger default region, or change the JCL REGION= parameter to the size needed.

  Your installation may limit (control) the specification of REGION=, usually through the JES2 EXIT06 exit or the JES3 IATUX03 exit. If so, relax this restriction for the WebSphere Application Server for z/OS JCL procedures.

  Finally, check your TSO segment default region size and make sure it is at least 128 MB.

Planning for problem diagnosis

This section describes:

- Component trace
- Error log stream
- Dump data sets

Overview of problem diagnosis

WebSphere Application Server for z/OS uses component trace (CTRACE) to capture and display trace data in trace data sets. WebSphere Application Server for z/OS identifies itself to CTRACE with a dynamic component name determined by the short cell name. CTRACE allows you to:

- Merge multiple traces through the browse tool, including other components such as TCP/IP and z/OS UNIX.
- Write trace data to a data set rather than sysprint, keeping spool space free.
- Better manage system resources by allowing trace data to wrap or not wrap.
- Use CTRACE to funnel trace data from multiple address spaces to one data set, or have CTRACE send the trace data from each address space to separate data sets.
- Start and stop tracing without stopping and restarting WebSphere Application Server for z/OS address spaces.
- Use one or more data sets for capturing trace data, thus allowing you to manage I/O more effectively.

WebSphere Application Server for z/OS also has an error log stream that records error information when WebSphere Application Server for z/OS detects an unexpected condition or failure within its own code, such as:

- Assertion failures
- Unrecoverable error conditions
- Vital resource failures, such as memory
- Operating system exceptions
- Programming defects in WebSphere Application Server for z/OS code

Use the error log stream in conjunction with other facilities available to capture error or status information, such as an activity log, trace data, system logrec, and job log.

The WebSphere Application Server for z/OS error log stream is a system logger application. Because the error log stream uses the system logger, you can:

- Have error information written to a coupling facility log stream, which provides sysplex-wide error logging, or to a DASD-only log stream, which provides single system-only error logging.

  Note: There is a significant performance penalty when using DASD-only error logging.
- Set up either a common log stream for all of WebSphere Application Server for z/OS or individual log streams servers. Local z/OS or OS/390 client ORBs can also log data in log streams. Because the system logger APIs are unauthorized, any application can use them. You should control access to the log streams through a security product such as RACF.

WebSphere Application Server for z/OS provides a REXX EXEC (BBORBLOG) that allows you to browse the error log stream. By default, the EXEC formats the error records to fit a 3270 display.

This topic describes the error log stream and how to set it up. Information about using the error log stream to diagnose problems is in the Troubleshooting section of the WebSphere Application Server information center. General information and guidance about the system logger is in *z/OS MVS Setting Up a Sysplex*.

Table 3. Finding WebSphere Application Server for z/OS Error Log Stream Information

What is your goal?	You should read:
Learn about the system logger and understand its requirements	*z/OS MVS Setting Up a Sysplex*
Learn about the WebSphere Application Server for z/OS error log stream	This topic
Size the coupling facility structure space needed for the WebSphere Application Server for z/OS error log stream	*z/OS MVS Setting Up a Sysplex*
Define the WebSphere Application Server for z/OS error log stream	“Using the Customization Dialog” on page 13
View the WebSphere Application Server for z/OS error log stream	The Troubleshooting section of the WebSphere Application Server information center
Learn about how Java applications can log messages and trace data in the error log stream	The Applications section of the WebSphere Application Server information center

For details about problem diagnosis, see the Troubleshooting section of the WebSphere Application Server information center.

Planning for Component Trace

To use CTRACE, you:

- Specify trace options for identifying trace data sets and connecting WebSphere Application Server for z/OS address spaces to the data sets in parmlib members.
- Update WebSphere Application Server for z/OS WebSphere variables to allow for initial trace parameters.
- Use IPCS-CTRACE to view the trace data because you cannot read the trace data in an ordinary editor.

Recommendation for dumps

Plan as you would normally for system dumps. Due to the size of WebSphere Application Server for z/OS address spaces, you may need to re-size your system dump data sets and use dynamic dump data sets.

Updating the CFRM policy

As stated in the BBOSSINS instructions generated by the ISPF Customization Dialog, you must ″Update the CFRM Policy ... prior to using log streams that have been indicated as CF-resident.″ These log streams include the WebSphere Application Server for z/OS error log and RRS logs. Here is a step-by-step process for updating the CFRM policy.

1. Find the active policy by issuing the command `D XCF,POL`. You will get output similar to this partial example:

   ```
   -D XCF,POL
   IXC364I 10.57.49 DISPLAY XCF 061
   TYPE: ARM
   POLICY NOT STARTED
   TYPE: CFRM
   POLNAME: POLCF1N1
   STARTED: 03/14/2003 11:32:22
   LAST UPDATED: 03/14/2003 11:31:52
   . . .
   ```

   Note: Do not make any changes to the active policy--you will create a new one later.
2. List the existing CFRM policies with the following utility:

   ```
   //STEP1 EXEC PGM=IXCMIAPU
   //SYSPRINT DD SYSOUT=*
   //SYSABEND DD SYSOUT=*
   //SYSIN DD * DATA TYPE(CFRM) REPORT(YES)
   /*
   ```
3. Modify the SYSPRINT from the previous job, which currently lists all the policies and has embedded page headings, so you can use it to define a new policy in the next job. Follow these steps to gain a usable SYSPRINT:
   a. Extract the definitions for the ACTIVE policy only.
   b. Delete the heading lines.
   c. Add the new structure definition using the BBOWCFRM member of the target CNTL data set as a model.
   d. Copy the new structure definition into a FB-LRECL(80) data set that you will use as SYSIN for the following job.
4. Create or update the new policy. Use a utility such as the following:

   ```
   //STEP20 EXEC PGM=IXCMIAPU
   //SYSPRINT DD SYSOUT=*
   //SYSABEND DD SYSOUT=*
   //SYSIN DD *
   DATA TYPE(CFRM) REPORT(YES)
   DEFINE POLICY NAME(NEWPOLC) REPLACE(YES)
   CF NAME(CF1LPAR) DUMPSPACE(5000) PARTITION(0E) CPCID(00)
   TYPE(009672) MFG(IBM) PLANT(02) SEQUENCE(000000051205)
   CF NAME(CF2LPAR) DUMPSPACE(5000) PARTITION(0F) CPCID(00)
   TYPE(009672) MFG(IBM) PLANT(02) SEQUENCE(000000051205)
   STRUCTURE NAME(CTS130_DFHLOG) SIZE(24000) INITSIZE(12000)
   REBUILDPERCENT(15) PREFLIST(CF1LPAR, CF2LPAR)
   . . .
   <== Insert your new structure here
   ```

   Where ″NEWPOLC″ is the name of the new policy.
5. Switch CFRM to the new policy. Use a command such as the following:

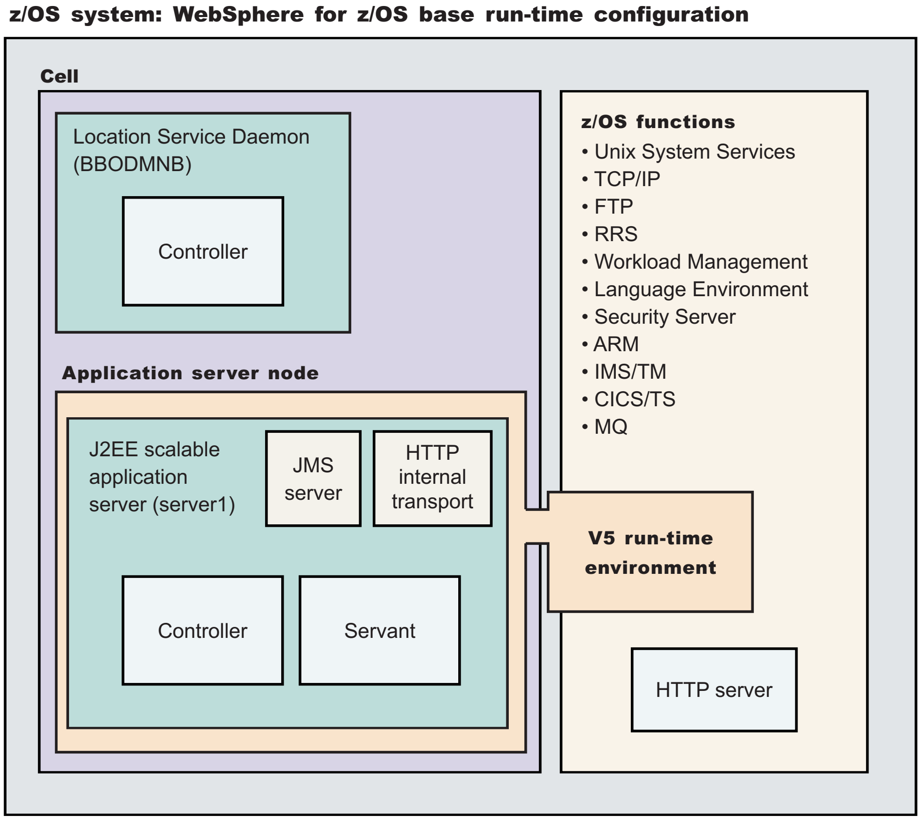
   ```
   SETXCF START,POLICY,TYPE=CFRM,POLNAME=NEWPOLC
   ```

Setting up a base Application Server cell

A typical WebSphere Application Server for z/OS base run-time includes a cell with a location service daemon (BBODMNB) and one node which, in turn, includes an Application Server (server1) with a controller and any number of servants. This article leads you through the background of a base Application Server cell as well as the tasks involved in setting up a WebSphere Application Server for z/OS base Application Server environment.

The installation and customization procedure is the same for a node to be federated into a cell as it is for a stand-alone Application Server. You can install and customize the base WebSphere Application Server for z/OS function more than once on a single machine as long as you keep the nodes in different directories.

If applicable, migrate applications, security settings, and the remaining configuration from WebSphere Application Server for z/OS Version 4.0.x and later. You can also choose to coexist with these older WebSphere Application Server for z/OS versions. See “Migrating and coexisting” on page 76 for further information.



The runtime servers use other z/OS functions, such as z/OS UNIX and TCP/IP. Part of installing WebSphere Application Server for z/OS includes configuring these functions for use by the runtime.

J2EE servers contain at least one Web container and one EJB container. The Web container manages Web applications (servlets and JavaServer Pages), while the EJB container manages enterprise beans.

The HTTP internal transport, which is part the J2EE application server, is a functional component that acts as an HTTP protocol catcher for Web applications. The HTTP internal transport is depicted in the server1 of the J2EE scalable application server.

The JMS server, which is also part of the J2EE scalable application server, hosts the JMS function of the WebSphere Application Server for z/OS. The JMS server is depicted in the server1 J2EE scalable application server.

Follow the steps below to configure your WebSphere Application Server for z/OS base Application Server environment.

1. If you plan to eventually end up with a Network Deployment environment, set up an overall naming scheme. Plan the scheme with the knowledge that the new cell structure will match that of the deployment manager. The following high level steps apply if your desired result is a deployment manager:
   a. Configure and create a base Application Server node.
   b. Configure and create a deployment manager node.
   c. Federate the base Application Server node into deployment manager cell.
   d. Repeat as needed.

   The steps above in conjunction with the planning knowledge in the bullets below provide adequate information for you to design your naming scheme.

   - Federating a base Application Server node into a deployment manager cell means the base Application Server node will become part of the deployment manager's cell. Consequently, after you federate your base Application Servers into a Network Deployment cell, the cell name associated with the original deployment manager -- not those of any of the original base Application Servers -- will become the new Network Deployment cell name for the newly federated node.

     This cell name assignment affects anything that has a cell-qualified name and was formerly used by Application Servers on that base application node, such as:

     – SAF profile names (for example, SERVER, CBIND, or EJBROLE).
     – Any ″ENV=″ parameter that start z/OS Application Servers (recall that the form is ENV=<cell short name>.<node short name>.<server short name>). If you use z/OS WLM static application environment definitions, you may need to use the WLM administrative application (IWMARIN0) to update the parameter string for the servant process JCL procedure.
     – The RM name used to register the base Application Server with RRS. If you ran transactions on the base Application Server prior to federation, you may find RRS log records left behind that belonged to the base Application Server when it was in its own cell. You may choose to delete these records.
   - The z/OS location service that is part of the deployment manager's cell will become the effective location service for the newly federated node.
2. Follow the steps in "Using the Customization Dialog" on page 13 to configure your base Application Server. From there, you will see that a command similar to the following starts the ISPF Customization Dialog:

   ```
   ex 'WAS500.WAS.SBBOCLIB(BBOWSTRT)'
   ```

   a. Select the option in the Customization Dialog to configure the base Application Server and follow the panels.

   Note: The cell name for a base Application Server node should be considered temporary if the end goal of the configuration is a Network Deployment environment. When the base Application Server node is federated into a deployment manager cell, the base Application Server node's cell name is no longer used. The deployment manager's cell name is what continues to be used.

3. Enable the appropriate level of security after the installation is complete. See Securing your environment after installation for more information.
4. Develop and unit test application components.

   Load existing application components and modules into your development environment and debug them. See Developing for more information.
5. Assemble code into a main application module or enterprise archive (EAR) file. See Assembling or packaging for more information.
6. Start all servers in the test environment. See Starting servers for more information.
7. Deploy your applications in the test environment. See Deploying for more information.
8. Test all applications thoroughly.

   See Testing for more information.

   Follow normal test procedures as you move the test environment into production. Review the information in the “Migrating” on page 70 topic to understand what you must look for. In particular, review the table at the end of the topic that links you to specific recommendations and practices.

   You must configure your migrating applications to verify that they migrate in the way that you want. See “Migrating and coexisting” on page 76 for more information.
9. Prepare and monitor the environment into which you deploy applications. See Administering for more information.
10. Adjust application code, configurations, and system settings to improve performance. See Tuning for more information.
11. Fix any known problems. See Troubleshooting or problem determination for more information.
12. Once you finish running through the Customization Dialog options and have run all the generated jobs, validate the success of your base Application Server configuration by bringing up the Administrative Console.

Use the Administrative Console (see Using the Administrative Console for more information) or other administrative tools (see Welcome to System Administration for more information) to validate, observe and control your new environment. The console provides a central location for configuring, monitoring, and controlling your Application Server.

Simplifying your installation

This article presents some tips for simplifying your installation. Refer back to this when you are running through the Customization Dialog, especially when setting up your first or practice run-time.

Naming servers, cells, and nodes for manageability

It is important to plan your system by visualizing the results desired before even installing the product. This is a part of good planning and leads to good management and effective use of your system. This effective kind of long term thinking is important down to the very details of the system, especially naming. Upon initial configuration of the first base Application Server, you will be asked to provide long names and short names for the servers, nodes, and cells. It is imperative that simple names are selected that can be easily tracked through the installation process.

In general these are the nomenclature ″rules″ with examples given to indicate best practices for naming:

- Short names are generally 8 or fewer alphanumeric characters, cannot have a numeric first character, are capitalized, and are used for MVS constructs. The @ and # national characters are allowed, but not $.

  5.1 + By default, server short names are a maximum of 7 characters in length. An ″S″ is automatically appended to indicate that it is the controller started task jobname. If you choose to designate an

8-character server short name, you must then explicitly set the JOBNAME for the corresponding servant process to the 8-character name, excluding the ″S″. For more information, see Converting a 7-character server short name to 8 characters.

Short name examples:

- 5.1+ Server short name: **V5SRV1**, **P5SVR1D**
- Node short name: **V5NODEA**
- Cell short name: **V5CELL**

- Long names are mixed case and can be as long as 60 characters, though shorter names are recommended to help ward off unmanageably long path names in the configuration HFS.

  Long name examples:

  - Server long name: **v5srv1**
  - Node long name: **v5nodea**
  - Cell long name: **v5cell**

Notice in the examples the same names are used for long name and short name for server, node, and cell. This is an easy way to remember the nomenclature appointed in your environment, while differentiating the long names by using mixed or lower case. When selecting server, cell, or node names, always avoid special non-alphameric characters because they are used as HFS directory names and are parsed in XML files that may have problems with special characters such as blanks, slashes, dashes, tildes, question marks, or underscores.

Simplifying proclib

WebSphere Application Server for z/OS V5.1 allows all the servers and location service daemons in a cell to be started with just three procs. This is radically different than in earlier releases of the Application Server, such as V4 where new procedures in proclib needed to be created for new servers. To utilize the same three procs without issue or complexity it is recommended that simple names, with understood meaning, be given, such as **V5ACR**, **V5ASR**, and **V5DMN**.

Simplifying WLM definitions

In the ISPF Customization Dialog, you are asked to specify a **Cluster Transition Name**. This **Cluster Transition Name** becomes the Workload Manager (WLM) Application Environment name, as well as the **server_generic_short_name**. Dynamic application environments are now supported, making it unnecessary to go into the WLM panels to define new application environments for each server. When selecting names in the Customization Dialog, it is suggested that simplification apply whenever possible to avoid overly confusing nomenclature later on. A simple example might be naming all the **Cluster Transition Names**(Application Environments) by appending **V5CL**, or some other such naming structure at the beginning of the name. This will simplify the RACF definitions for the CBIND and SERVER class profiles.

Simplifying RACF definitions

For an initial test configuration, it is unnecessary to set up complicated sets of RACF userids and profiles. It is easier to assign a single userid, such as **V5USR**, and a single groupid, such as **V5GRP**, for each user in a cell. Additionally, once you run the BBOCBRAJ job that produces the RACF definitions, the profiles can be tailored so they can be used for other servers without further RACF definitions. Here are some simple example RACF profiles for all servers with a short name beginning with **V5SRV** and application environments starting with **V5CL**:

- CBIND Class profiles
  - Access to generic servers:
    - **CB.BIND.<appl.env_name>** or, as an example **CB.BIND.V5CL***.

  - Access to objects in servers:
    - **CB.<appl.env_name>** or, as an example **CB.V5CL***.
- SERVER Class profiles: servant to access control regions
  - Static Application Environments:
    - *See example for dynamic application environments*.
  - Dynamic Application Environments:
    - **CB.<server_name>.<appl.env_name>.<cell_name>** or, as an example **CB.*.V5CL*.***.
- STARTED Class profiles
  - Assign user and group IDs to controller STCs (V5ACRn profile):
    - **<proc_name>.<job_name>**.
  - Assign user and group ID to servant STC jobnames (V5SR profile):
    - **<job_name>.<job_name>**.
- LOGSTRM Class profiles
  - Protect access to error log:
    - **log.stream.name**.

Organize your port usage

In order to understand the levels of communication going on with your Application Server, it is important to be familiar with port assignments by default. See the article on z/OS port assignments for reference.

Use the ISPF Customization Dialog Effectively

While the ISPF Customization Dialog coordinates the specification of customized variables, it is not a remedy for careful planning and understanding of the z/OS environment. Here are some suggestions:

- It is imperative to develop a good naming convention for server JCL procedures, userids, WLM application environments and other profiles.
- Chose unique UIDs and GIDs for the new userids defined in the Dialog. You may use the ISHELL User List and sort on UIDs using the File pull-down to get a list of all UIDs assigned in your system.
- Validate the customized variables with your communication specialist, and security administrator.
- The very first time you use the ISPF Dialog, you should go through the whole process of generating the job streams, using the IBM default values.
- Take advantage of the help screens (via PFKey 1). They contain a lot of useful information.
- Use the same ISPF application name -- APPL(xxxx), or load the saved variables when you move from option to option (e.g., from ″Configure base Application Server node″ to ″Configure deployment manager node″.)
- Examine the members in the target .CNTL and .DATA partitioned datasets and the .SAVECFG dataset where you saved them. Also examine the *.SDCFG dataset where you saved your security domain variables.
- Understand how your customized variables affect the generated jobs and data files by browsing each one using the customized instructions in the BBOSSINS member of the .CNTL dataset as a guide.

Note: A common mistake users make is to change any of the values on the Customization Dialog after generating the jobs and data and then attempting to make manual corrections to the target CNTL or DATA members. Don't attempt to do this, instead regenerate all the jobs and data. In the long run it will take far less time and energy.

Installing a practice runtime

Customizing a WebSphere Application Server for z/OS base Application Server the first time around can be a frustrating experience. Most users find that they do not have everything specified correctly in the Dialog, or may not understand the effects of the variable specifications before they start submitting the jobs. Because of this, we recommend that you install a practice runtime on a test system where you can easily back it out. This helps you understand the structure of the WebSphere Application Server for z/OS runtime when you dismantle it and start over. (The whole process takes less than an hour once you get the practice.)

The intention of this article is to help you learn the ropes with the Customization Dialog. Please use it in conjunction with the other articles in the section about the Dialog.

Some things to note:

- This is altogether a ″reading, spelling, and typing″ exercise, and a ″follow the directions″ drill. Most problems occur from making an innocent typing mistake, or from not following all the steps as specified in the instructions. So be careful when entering information--ensure you type everything in correctly. Note the suggestions along each step of the way--they can be very helpful especially your first time around.
- Pay attention to the ″User ID requirements″ in the generated instructions. This is a frequent cause for jobs not running successfully.
- Verify each job or step as you run it and check it off on your customized instructions so you do not skip any steps.
- Look carefully at the job’s output (not just the return code) to make sure everything is installed correctly. You should see no error messages in the job’s output except where specifically noted in the instructions.

1. Run through the Dialog and follow all the steps for configuring a security domain up until you view the generated instructions.
2. Follow the generated security domain instructions, which executes BBOSBRAJ and BBOSBRAK.
3. Issue the RACF commands necessary to define the users, groups, profiles, and SSL digital certificates that are effective for all WebSphere Application Server for z/OS runtime servers.
4. Save your security domain definition.
5. Run through the Dialog and follow all the steps for configuring a base Application Server up until you view the generated instructions.
6. Perform the manual configuration updates in the generated base Application Server instructions. These steps affect parts of your system that are usually tightly controlled. They produce changes that your systems programmer responsible for this z/OS system should review.

   Note: You could have used ″canned″ jobstreams to update WLM, SYS1.PARMLIB, and the TCP/IP configuration, but, for good reason, most installations would not allow this.
7. Create your logstreams and data sets. The first four jobs (BBOMSGC - BBOWCTR) are straightforward system programming tasks.
8. Update your server-specific security definitions. The next two jobs (BBOCBRAJ and BBOCBRAK) generate and then issue the RACF commands necessary to define the users, groups, profiles, and permissions for the WebSphere Application Server for z/OS runtime servers. The recommended approach is to run the jobs that generate the commands and simplify the profile names.

   Submit the BBOCBRAK job, or take it and BBOCBRAJ to your security administrator for approval. Then let your security administrator issue those commands or submit the supplied jobstreams. If your installation has different profiles structures, you may have to modify the RACF commands generated by these execs to suit your particular needs.

Note: Your installation must have ″list of groups″ on for these to work, because the servers must be connected to the WebSphere Application Server for z/OS administrator group.

After you get the runtime up and successfully run the IVT, you may want to replace the built-in administrators defined for WebSphere Application Server for z/OS, or at least change their passwords. If you want to prevent the user IDs assigned to started tasks from being used to sign on or submit jobs, you can make the user ID protected by adding the keywords NOPASSWORD and NOOIDCARD to the user's RACF profile.

If you have some other security product such as Top Secret or ACF2 instead of RACF, then you must put the RACF commands into the appropriate syntax.

Note: Ask your customer's security administrator to contact the vendor (CA) for the latest maintenance and guidance on WebSphere Application Server for z/OS customization.

9. Create and populate the configuration HFS. The next three jobs (BBOWCHFS - BBOMCFGU) and BBOWCPY2 run BPXBATCH shell scripts to define, customize, and load data into the configuration HFS and manipulate the ownership and permission attributes. For this reason these jobs must be submitted with job ownership of a user with a UID=0 (despite what the instructions say):
   - Submit by a user with a UID=0,
   - Submit by a user with surrogate access to a user ID with UID=0, or
   - Specify the user ID and password of a user ID with UID=0 on the JOB card.

   Review the output of these jobs so you can see where the configuration files were built and how the directories are structured. Also note that ownership is reassigned to the WebSphere administrator and permission bits are set so you must be connected to that group to even read them.

After you finish running through the instructions with these simplified instructions, you will have a practice runtime set up. Now you can uninstall it and reinstall WebSphere Application Server for z/OS for real.

Some common mistakes to watch out for:

- Pay close attention to the ″User ID requirements″ in the generated instructions. This often is the cause for jobs not running successfully.
- Navigating the configuration HFS with a UID of 0 can alter files or their ownership and permission attributes, thus making them inaccessible to the WebSphere Application Server for z/OS runtime servers and administrators. It is better to use the WebSphere Application Server for z/OS administrator's user ID.
- If you decide to change any of the customized variables after you have submitted any of these jobs, don't be tempted to make manual modifications to the generated jobstreams or data. Back off the installation and start over by regenerating all the jobstreams and start over from the BBOMCFG job.

Handling workload management and server failures

During operations, if your application fails repeatedly, causing the application servants to terminate, workload management may terminate the application environment for the application. WebSphere Application Server for z/OS issues the following message if it tries to use a failed application environment:

```
BBOO0075E Unable to schedule work. WLM application environment applenv has stopped.
```

You must fix the problem with your application, then restart the application environment with the RESUME option on the VARY WLM command.

Perform these steps to check and start the workload management application environment.

Note: The dynamic application environment commands apply only if you are running z/OS V1.2 or above with the WLM-DAE support PTF (APAR OW54622). See the assembling applications information in the InfoCenter for more information.

1. Display the application environment. To do so, issue

   ```
   d wlm,applenv=*
   ```

   for static application environments or

   ```
   d wlm,dynappl=*
   ```

   for dynamic application environments.
2. Start the application environment. To do so, issue

   ```
   v wlm,applenv=environment_name,resume
   ```

   for static application environments (where **environment_name** is the application environment name) or

   ```
   v wlm,dynappl=applenv,resume
   ```

   for dynamic application environments.

You know you are done when a re-display of the application environment shows it is available.

Setting up a high availability environment

- The high availability sysplex must include at least two LPARs. These LPARs should be on separate hardware instances to eliminate hardware SPOFs.
- There must be a network path redundancy leading up to the Web servers and Applications Servers in your sysplex.
- If you are using HTTP sessions, session state must be shared between the cluster members using DRS or session in DB2. If you are using stateful session EJBs (not a best practice), the stateful session persistent store must be configured on a shared HFS.

To set up a WebSphere Application Server for z/OS high availability environment:

1. Configure a WebSphere Application Server for z/OS V5.1 server instance (node) on each LPAR that is configured in the Network Deployment cell. The deployment manager Server, which is required, must be configured on its own node. It can be configured on either LPAR or on a separate LPAR.
2. Use the Administrative Console to make sure a location service daemon has been defined on each LPAR that has one or more nodes in the same cell. Modifying location service daemon settings describes how to define a location service daemon it one is not already defined.
3. Define a V5.1 Application Server on each node, and form them into a server cluster (see Creating clusters).
4. Define the following dynamic virtual IP addresses (DVIPAs) through the z/OS Operating System's Sysplex Distributor.
   - A dynamic virtual IP address as the daemon's IP name for the cell. This IP address enables WLM-balanced workload routing and fail over between the LPARs for IIOP requests.
   - A dynamic virtual IP address as the HTTP transport name for the cell. This IP address enables WLM-balanced routing and fail over between the LPARs for sessionless HTTP requests.

   z/OS Communications Server IP Configuration Guide Version 1 Release 4, SC31-8775-02, describes how to define IP addresses through the z/OS Sysplex Distributor. This publication is available at http://www.ibm.com/servers/eserver/zseries/zos/bkserv/v1r4books.html.
5. Define a static IP address for each node as an auxiliary HTTP transport name for the cell. This IP address enables directed HTTP routing for sessional HTTP requests.
6. Configuring Web server plug-ins in each of the Web servers and configure them to use the HTTP DVIPA for sessionless requests, and the static IP addresses for sessional requests.

After you have set up a high availability environment, you can control application rollout and workload routing.

High availability configuration

The objective of any high availability configuration is to eliminate all single points of failure (SPOFs).

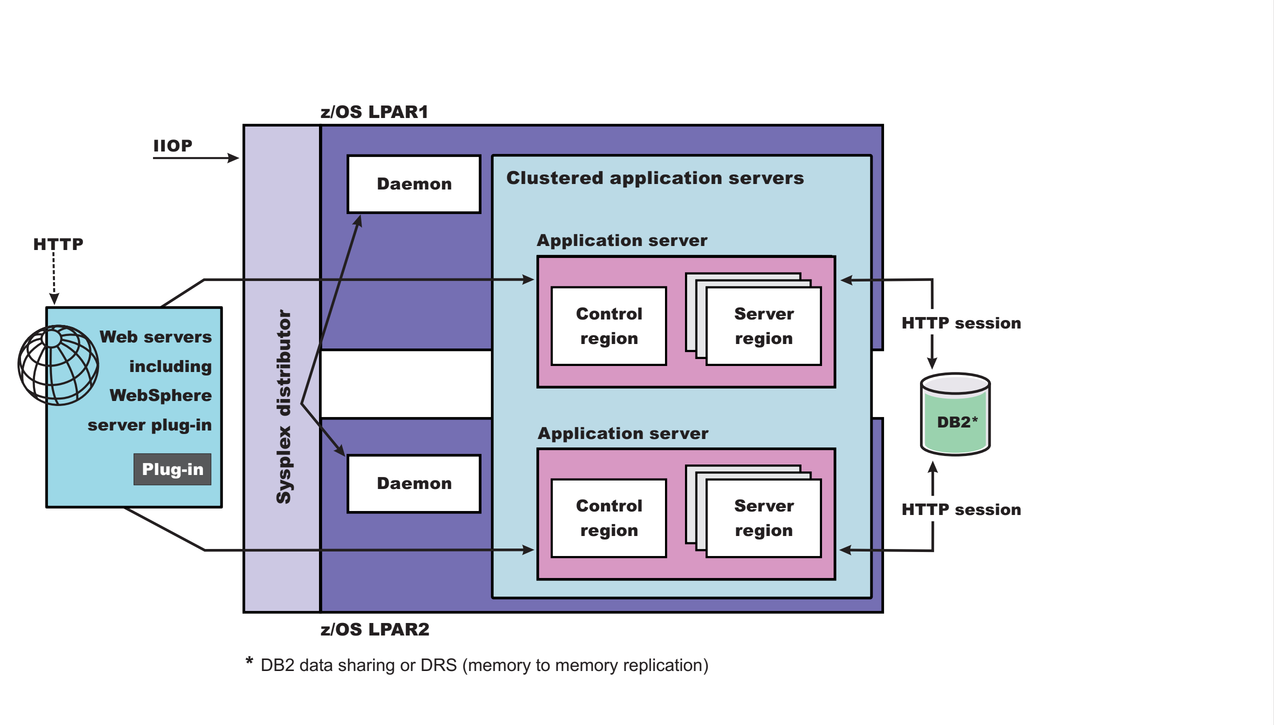


Figure 1. WebSphere Application Server for z/OS High Availability Configuration. This figure illustrates the recommended WebSphere Application Server for z/OS configuration for high availability. The key elements are described in the text that accompanies this figure.

Following are the key elements of a WebSphere Application Server for z/OS high availability configuration:

- Network path redundancy leading up to the Web servers and Applications Servers.
- Redundant Web servers. (There must be at least two LPARs in a high availability sysplex configuration.)
- A highly available sysplex configuration. These LPARs should be on separate hardware instances to eliminate hardware and software Single Points of Failures (SPOFs).
- A WebSphere Application Server for z/OS V5.1 configuration instance (node) on each LPAR that is configured into a Network Deployment cell. The deployment manager server (required, and configured on its own node) can be configured on either LPAR or on a separate LPAR. (The deployment manager server is not depicted in the preceding figure.) Also note, there is a daemon process (WebSphere CORBA Location Service) on each LPAR that has one or more nodes in the same cell.
- A V5.1 Application Server defined on each node, and formed into a server cluster.
- A dynamic virtual IP address (DVIPA) defined through the z/OS Sysplex Distributor as the daemon IP name for the cell. This IP address enables WLM-balanced routing and fail over between the LPARs for IIOP requests.
- A dynamic virtual IP address (DVIPA) defined through Sysplex Distributor as the HTTP transport name for the cell. This IP address enables WLM-balanced routing and fail over between the LPARs for sessionless HTTP requests.
- A static IP address is required for each node as an auxiliary HTTP transport name for the cell. This enables directed HTTP routing for sessional HTTP requests.
- A WebSphere plug-in for Web servers must be installed in each of the Web servers and configured to use the HTTP DVIPA for sessionless requests, and the static IP addresses for sessional requests.

- If using HTTP sessions, session state must be shared between the cluster members using DRS or session in DB2. If using stateful session EJBs (not a best practice), the stateful session persistent store must be configured on a shared HFS.

Controlling application rollout and workload routing in a high availability configuration

Prepare your environment for high availability Application Update.

Workload routing is controlled by stopping the Application Server on which the cluster member being updated resides. This results in a quiesce of that server. All existing requests already in the server are allowed to complete, but no new requests are accepted. Both the Sysplex Distributor and the WebSphere Application Server's Web server plug-in will route work away from the quiescing server. After all work has completed, start the application update process on this server:

1. Disable all forms of Automatic Synchronization, across all nodes in the Cell and save the changes. Perform one of the following processes to complete this step:
   - Using the Administrative Console:
     a. Select **System Administration** > **Node agents** > *node_agent_name* > **File Synchronization Service**.
     b. Unselect the **Automatic Synchronization** and **Startup Synchronization** checkboxes.
     c. Select the **Synchronize changes with nodes** checkbox.
     d. Click **Save**.
   - Use wsadmin scripting to specify the following commands and then restart all affected node agents:

     ```
     set node NODE
     set na_id [$AdminConfig getid /Node:$node/Server:nodeagent/]
     set syncServ [$AdminConfig list ConfigSynchronizationService $na_id]
     $AdminConfig modify $syncServ {{autoSynchEnabled false}}
     $AdminConfig modify $syncServ {{synchOnServerStartup false}}
     $AdminConfig save

     set nodeSync [$AdminControl completeObjectName type=NodeSync,node=$node,*]
     $AdminControl invoke $nodeSync sync
     ```

     Note: For a production environment, it is reasonable to always run the node agent with automatic synchronization disabled. However, it is advisable for startup synchronization to be enabled for the node agent so that it can acquire configuration updates that occur when the node agent is down. Startup Synchronization can be left enabled provided you can ensure that you will not restart the node agent manually, through automation, or through automatic restart manager during the application update process.
2. Update the application in the Master Configuration Repository on the deployment manager server. Perform one of the following processes to complete this step:
   - Using the Administrative Console:
     a. Select **Applications** > **Enterprise Applications**.
     b. Select the checkbox next to the application you want to update.
     c. Complete the application update process.
     d. Save your changes to the Master Configuration. **DO NOT** select the **Synchronize changes with nodes** checkbox .
   - Use wsadmin scripting to issue the following command:

     ```
     set app_loc /path/to/app
     set app_options {application options ie: -appname app}
     set options [list -update]  lappend options $app_options
     $AdminApp install $app_loc $options
     $AdminConfig save
     ```

Note: This procedure does not give you the nodeSync mbean and synchronize.

At this point, you have the updated the version of your application (App v2 in the following figure) in your Master Configuration. However, the original version of your application (App v1 in the following figure) is still running in the cluster that has Cluster members on LPAR1 and LPAR2.

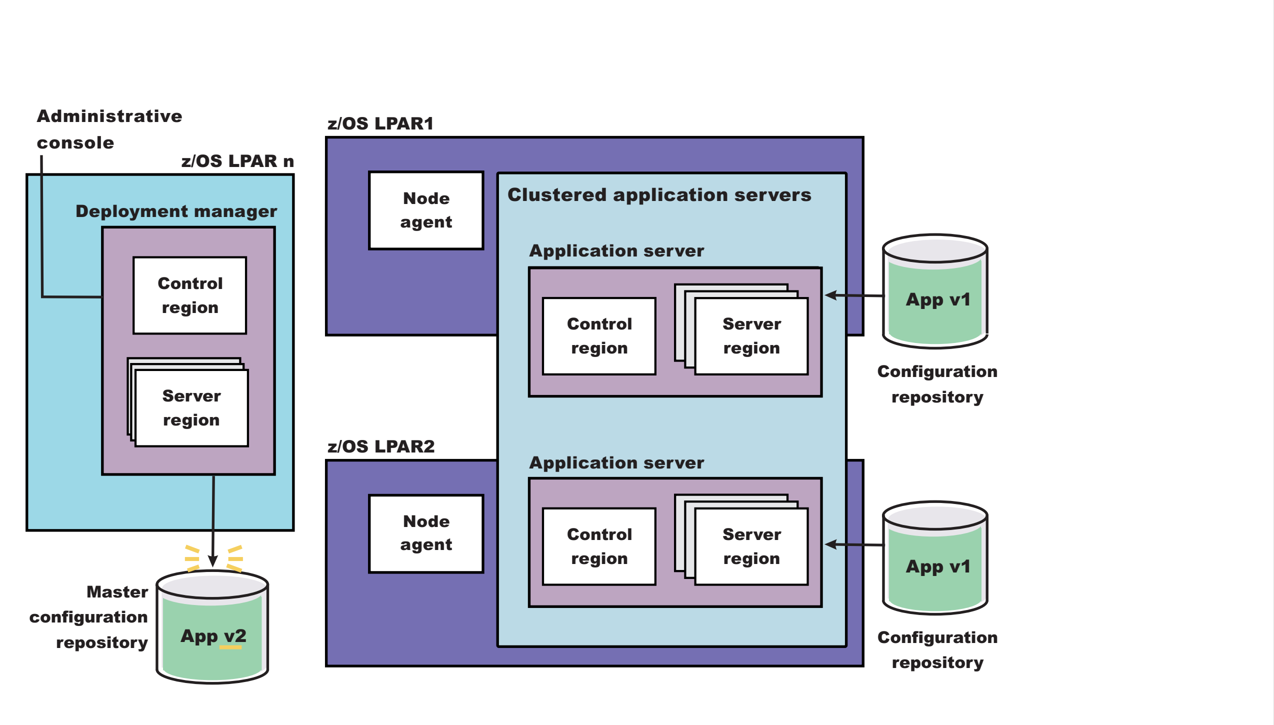


Figure 2. Install application update. This figure illustrates the first stage of an application update in a high availability environment.

3. Stop the Application Server on LPAR1 and manually synchronize the node to the updated version of the application. This step may take time to complete because the server must wait for all currently assigned work items to complete before shutting down.

   Perform one of the following processes to complete this step:

   - Using the Administrative Console:
     a. Select **Servers** > **Application Servers** .
     b. Select the checkbox next to the cluster member you want to stop and update. This cluster member should be on LPAR1.
     c. Select **STOP**. The Cluster Stop method should not be used, because it will stop all Servers within the cluster and the application will no longer be available.
   - Use wsadmin scripting: to issue the following commands:

     ```
     set node NODE
     set server SERVER
     $AdminControl stopServer $server $node
     ```
   - Issue the following command from the MVS Console:

     ```
      STOP short_server_name
     ```

     For example:

     ```
     STOP BBOS001
     ```

4. Synchronize the node. Perform one of the following processes to complete this step:

   - Using the Administrative Console:
     a. Select **System Administration** > **Nodes**.

   b. Select the checkbox next to the node you want to synchronize, and then select **Full Resynchronize**.

- Use wsadmin scripting to issue the following commands:

```
set node NODE
set nodeSync [$AdminControl completeObjectName type=NodeSync,node=$node,*]
$AdminControl invoke $nodeSync sync
```

As illustrated in the following figure, the updated version of the application (App v2) now resides in the node on LPAR1.

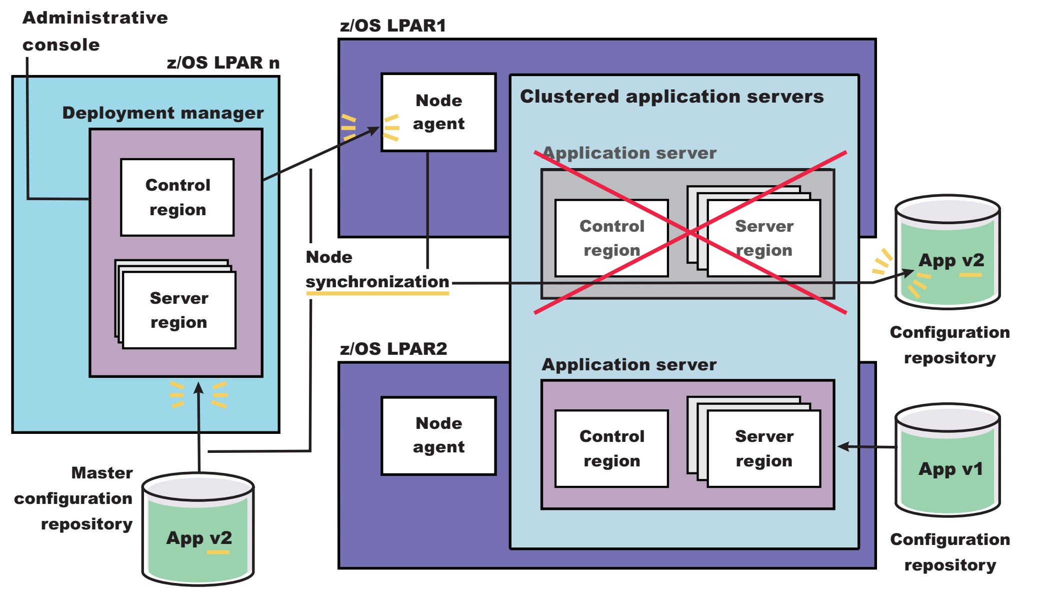


Figure 3. Update the node on LPAR1. This figure illustrates the first stage of an application update in a high availability environment with two LPARs.

5. Restart the server on LPAR1. Perform one of the following processes to complete this step:
   - Using the Administrative Console:
     a. Select **Servers**.
     b. Select the checkbox next to the server you want to start, and then select **START**.
   - Use wsadmin scripting to issue the following commands:

```
set node NODE
set server SERVER
$AdminControl startServer $server $node
```

   - Issue the following command from the MVS Console:

```
START procname,JOBNAME=server_short_name.ENV=cell_short_name.node_short_name.server_short_name
```

   For example:

```
START BBO5ACR,JOBNAME=BBOS001,ENV=PLEX1.SY1.BBOS001
```

   When this server comes back up, it will be running the new version of the application (App v2),

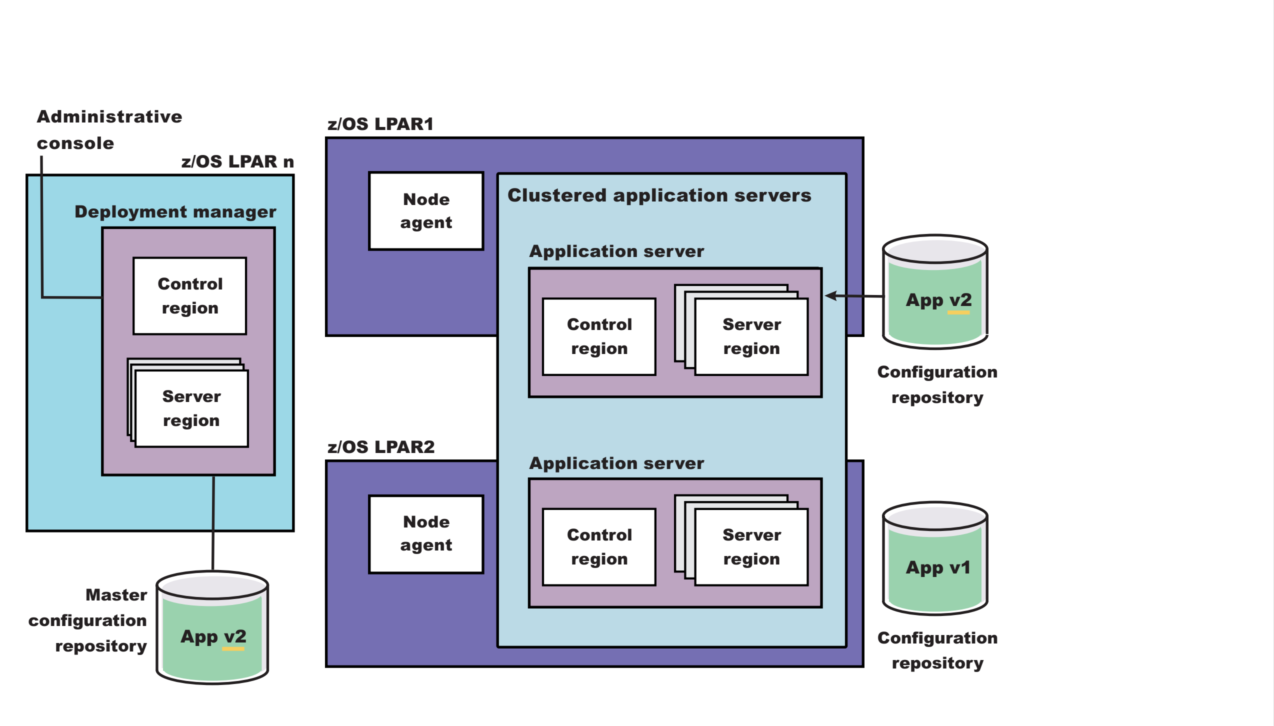


Figure 4. Restart the server on LPAR1. This figure illustrates the completion of the first stage of an application update in a high availability environment.

6. With the new version of the application running on LPAR1, repeat the preceding three steps on the other LPARs in the cluster to update them with the new version of the application. The following figure illustrates what your configuration will look like in a two LPAR cluster.

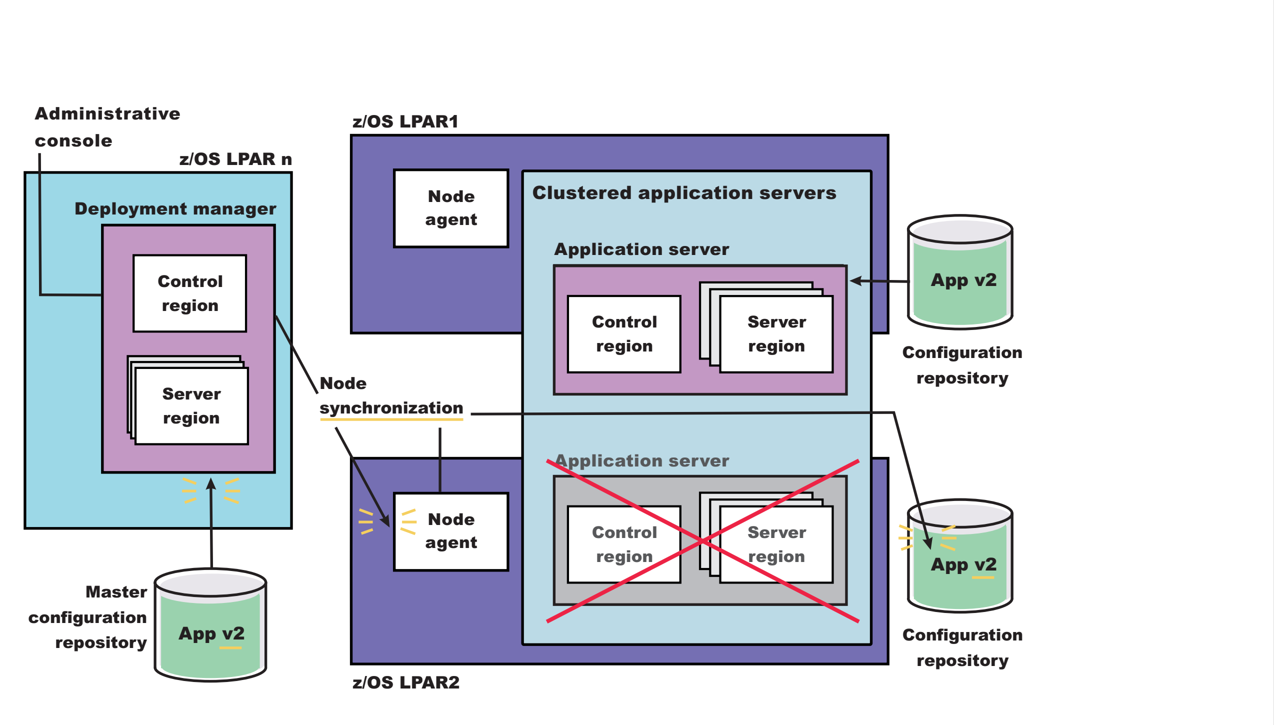


Figure 5. Update the node on LPAR2. This figure illustrates the second stage of an application update in a high availability environment.

The application update process is complete when the new version of the application is running on all of the LPARs in the cluster. The following figure illustrates what a two LPAR cluster will look like after you restart the server on LPAR2.

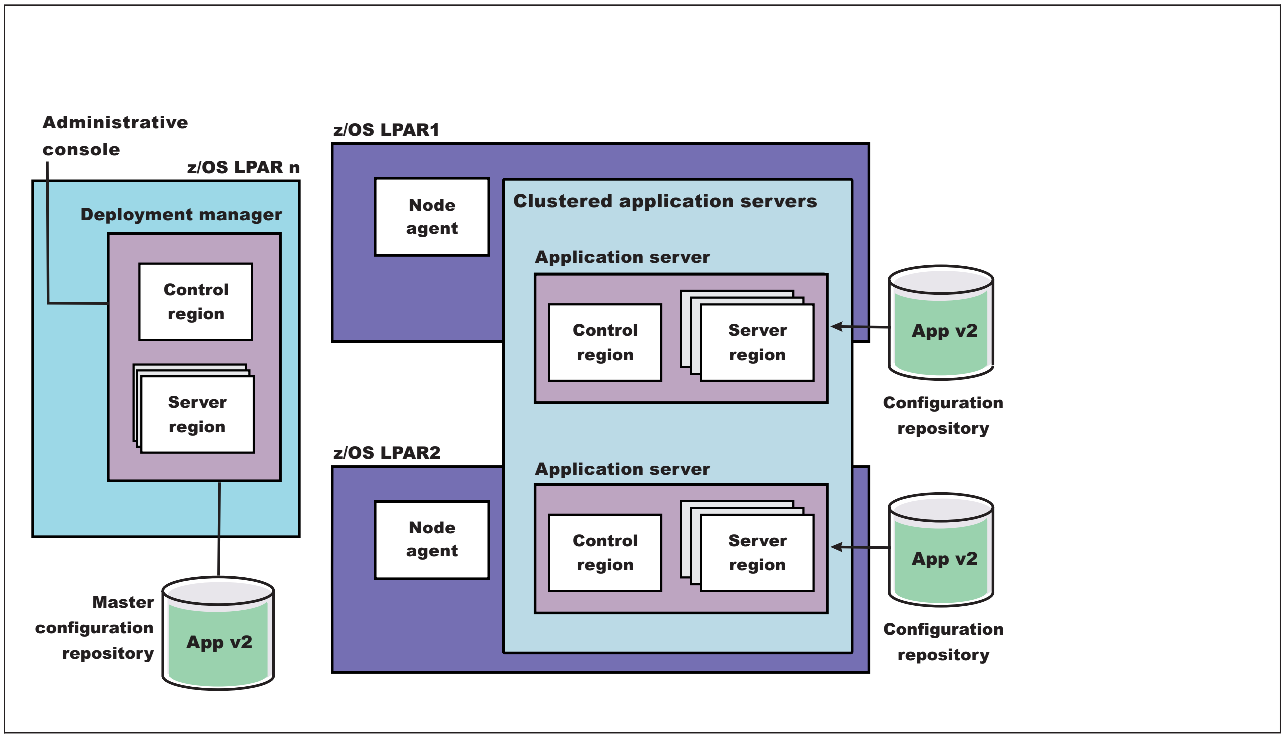


Figure 6. Restart server on LPAR2. This figure illustrates what a two LPAR cluster will look like after you restart the server on LPAR2.

Application update procedure in a high availability environment

Application update involves distributing new application binaries to each of the servers in a cluster during configuration synchronization.

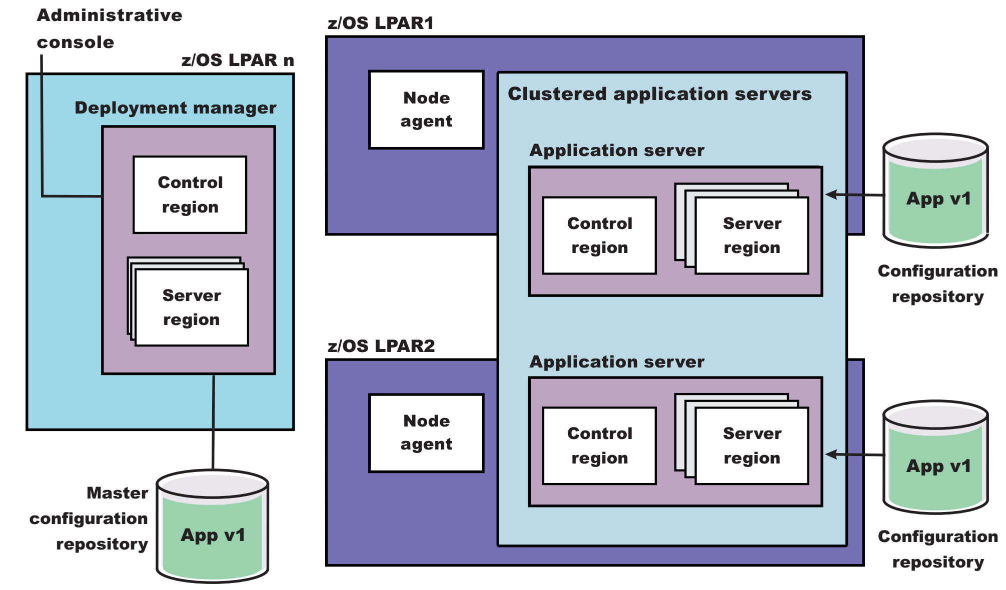


Figure 7. Application steady state - configuration view. This figure illustrates the configuration view of an application in steady state.

This distribution occurs via HTTP and happens even if the servers and the deployment manager are all located on the same LPAR. If the **synchronize with nodes** option is checked when the application update is saved, the synchronization request is sent to each node.

Normally, upon receiving the request, each node asynchronously orchestrates the synchronization process with the deployment manager. During this synchronization process, the application's binaries are downloaded from the deployment manager and stored on the node in the designated location (for example, `installedApps`).

The act of storing the new binaries triggers a configuration change event listener which then stops and restarts the application. Depending on such variables as dispatcher behavior, LPAR weighting, etc., a variance in the order and pace at which each server makes the new application available may be observed.

Because of the concurrent, asynchronous nature of node synchronization, continuous availability of the application being updated is not guaranteed. This is because there is no correspondence between the actual state of the application and the workload routing mechanisms. Client requests may be routed to a server even if that particular application is temporarily unavailable.

In a high availability environment, an application must remain available even during an update process. Therefore, application rollout to each cluster member, as well as workload routing to the cluster members must be carefully controlled to prevent workload from being routed to a cluster member that is undergoing the update process. When these two aspects are carefully controlled, an update can be installed on each cluster member without client requests arriving at a cluster member during the update process.

Setting up a Network Deployment environment

This article leads you through the background of a Network Deployment cell as well as the tasks involved in setting up a WebSphere Application Server for z/OS Network Deployment environment.

After you have your WebSphere Application Server for z/OS base Application Server cell up and running, there are many reasons why you should continue on and configure a Network Deployment cell. Overall, the Network Deployment cell provides a more robust operating environment. See “Choosing between a base or Network Deployment configuration” on page 22 for more information on the differences between the two environments.

Before you can set up a WebSphere Application Server for z/OS Network Deployment environment, you need to first install and customize the base WebSphere Application Server for z/OS function on each system to create an Application Server node on the system. See “Setting up a base Application Server cell” on page 37 for instructions and further information.

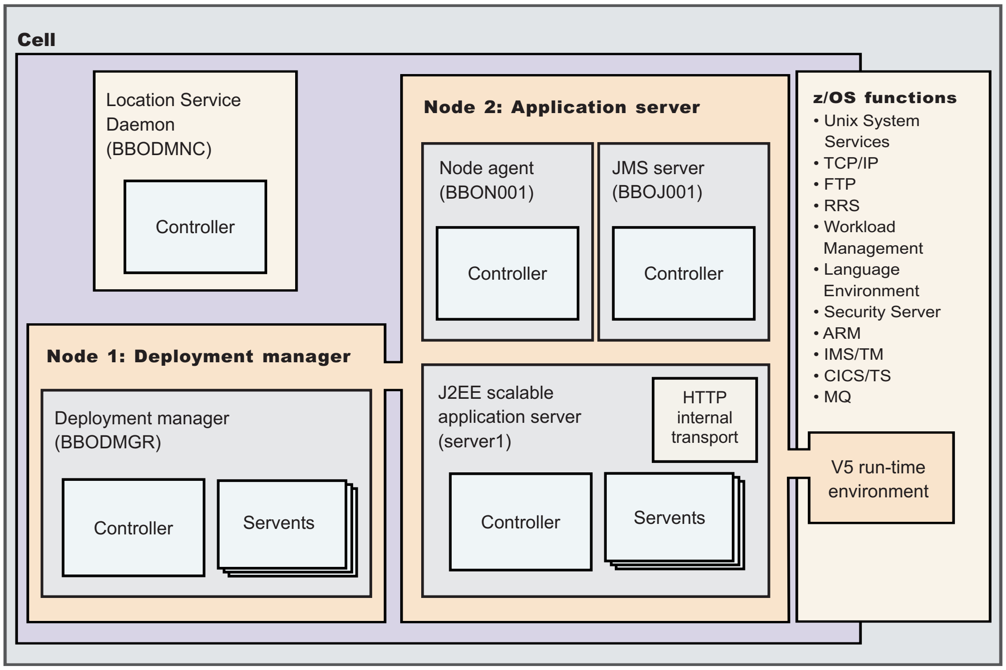
A Network Deployment cell is a group of WebSphere Application Server for z/OS nodes, which includes a single deployment manager node and one or more Application Server nodes. The deployment manager node contains only one server, the deployment manager server named ″dmgr,″ because the deployment manager node isn’t allowed to define Application Servers.

The dmgr server is the administrative server for the entire Network Deployment cell, which means that all administrative actions taken on Application Servers within that cell are performed through the dmgr server.

The dmgr server also hosts the administrative application, known as the ″Administrative Console,″ which provides a browser-based, graphical user interface for performing administrative operations on the Network Deployment cell. Similarly scripted administrative actions (through the wsadmin utility) are executed by the dmgr server. Remote scripted adminstrative actions are performed by the wsadmin utility through the dmgr JMX SOAP or RMI connector.

A Network Deployment cell has the unique ability to:
- Span multiple systems in a sysplex
- Fully utilize clustering
- Start and stop servers from the Administrative Console

z/OS system: WebSphere for z/OS Network Deployment run-time configuration



The picture above depicts a typical WebSphere Application Server for z/OS Network Deployment runtime. It includes a location service daemon (BBODMNC) and two nodes (both housed in a cell, which can span systems); one for the deployment manager and one for the Application Server. The deployment manager node includes a deployment manager (BBODMGR) with a controller and any number of servants. The Application Server node includes a node agent (BBON001), a JMS server (BBOJ001), and an Application Server (server1) with a controller and any number of servants.

Setting up a Network Deployment environment involves several steps performed on each of the systems that comprise the cell. Follow the steps below to configure your WebSphere Application Server for z/OS Network Deployment environment.

1. Configure your deployment manager node on one of the systems in the cell. The deployment manager is the central administrative manager for your Network Deployment configuration. Follow the steps in "Using the Customization Dialog" on page 13 and the information in "Deployment manager node configuration considerations" on page 54 to configure your deployment manager node.
2. Start the deployment manager process.

   There are two ways to start the deployment manager:
   - As a monitored process, which restarts automatically if a failure occurs.
   - As an unmonitored process, which is what the `START` *`dmgr_proc_name`* command does. See START dmgr_proc_name command for more information.

   Run the **startManager** command from the `/bin` directory of the installation root of the deployment manager.

   For production systems, running the deployment manager as a monitored process is recommended.

   Note: It is sometimes necessary to start the deployment manager on an MVS image different from the one on which it was originally configured. See "Starting a deployment manager on a different MVS image" on page 55 for information.
3. Federate desired base Application Server nodes into your new deployment manager cell. Federation is the act of taking the node structure of a base Application Server node and merging it into the cell structure of a deployment manager cell. This act of federation may be done within a given MVS image or across MVS images within the same sysplex. Follow the steps in "Using the Customization Dialog" on page 13 and the information in "Base Application Server node federation consideration" on page 55 to federate the nodes.

   In performing the federation process, you will run the `addNode.sh` script on every node that you plan to federate into the cell. The **addNode** command incorporates a base WebSphere Application Server product node into a deployment manager cell. You must run this tool on every base application system that you plan to make part of a Network Deployment cell. There are several parameters for the **addNode** command, but the most important are `includeapps`, the host name of the deployment manager node, the JMX connector type, and the JMX port of the deployment manager node.

   Note:
   - Because `addnode.sh` stops any running Application Servers on the base Application Server node that you are federating, stop the Application Servers manually before performing the actual federation step.
   - To federate a z/OS base Application Server node, use the Customization Dialog to configure your `addNode.sh` parameters.

   See "Applying or restoring off a service level" on page 58 and addNode command for more information. Alternatively, you can use the Administrative Console of the deployment manager to add running Application Server nodes to the cell. See Using the administrative console for more information.
4. Enable the appropriate level of security after the installation is complete. See Securing your environment after installation for more information.
5. Develop and unit test application components.

   Load existing application components and modules into your development environment and debug them. See Developing for more information.

6. Assemble code into a main application module or enterprise archive (EAR) file. See Assembling or packaging for more information.
7. Start all servers in the test environment. See Starting servers for more information.
8. Deploy your applications in the test environment. See Deploying for more information.
9. Test all applications thoroughly.

   See Testing for more information.

   Follow normal test procedures as you move the test environment into production. Review the information in the "Migrating" on page 70 topic to understand what you must look for. In particular, review the table at the end of the topic that links you to specific recommendations and practices.

   You must configure your migrating applications to verify that they migrate in the way that you want. See "Migrating and coexisting" on page 76 for more information.
10. Prepare and monitor the environment into which you deploy applications. See Administering for more information.
11. Adjust application code, configurations, and system settings to improve performance. See Tuning for more information.
12. Fix any known problems. See Troubleshooting or problem determination for more information.

You can create a working WebSphere Application Server for z/OS cell.

Use the Administrative Console or other administrative tools to validate, observe and control the incorporated nodes and the resources on these nodes. The console provides a central location for configuring, monitoring, and controlling all Application Servers on all nodes within the cell. See Welcome to System Administration for more information.

Deployment manager node configuration considerations

This article presents considerations for when you configure your deployment manager node.

- Ensure that you run the Dialog on the MVS image on which the server will operate.
- When loading customization variables for this option, use the file from your base Application Server customization. This provides consistency between the two configurations, which is important because you will eventually federate the base Application Server node into the deployment manager cell.
- When allocating target data sets for this option, it is possible, though not recommended, to use the same target data sets that you used for the base Application Server node. The job names for each configuration are very close to one another, and, if you use the same target data sets, you may find it difficult to keep the two sets of jobs separate. Therefore, it is better to create a new set of target data sets and keep the two sets of jobs separate from one another.
- Because you loaded the variables from the base Application Server node into this deployment manager configuration, the base Application Server node values appear when you go to define variables. You can leave many of these variables as-is, but you must change some of them. See "Define variables to configure deployment manager node - definitions" on page 139 for more information.
- If possible, set up your HFSs such that the root HFS is shared among all processors and the deployment manager's configuration is in a configuration HFS on a system-generic mount point.

  Note: This configuration scenario is the best for certain tasks, such as starting the deployment manager on another system, that you may wish to perform later on.
- Ensure you save the variables from your deployment manager configuration to a different file than the one you used for your base Application Server configuration.
- When you go to generate the customized jobs, code the job card information to specify a user that has UID=0 authority. This ensures that each customized job's JOB card will have this information, which is important because that authority is required for many of the jobs to run properly.

Note: Later, as the deployment manager federates base WebSphere Application Server for z/OS nodes, it expands the cell that it manages.

Base Application Server node federation consideration

This article presents considerations for when you federate your base Application Server node.

- Ensure that you run the Dialog on the MVS image on which the base Application Server you will federate resides.
- The original base Application Server node's cell structure is no longer used once federation completes--the active cell structure is the deployment manager's cell. The location service daemon server of the original base Application Server node is also no longer used. The original base Application Server node's HFS and its mount point remain, however, as well as the entire contents of the HFS, which are only copied over to the deployment manager's HFS. (This is in case you want to un-federate the node later.) When federation occurs, the contents of the base Application Server node's HFS is modified to reflect the new cell in which it resides. This means the symbolic links used during server startup are modified, and some of the XML files are updated to reflect the new cell name.
- When loading customization variables for this option, ensure that your ″WAS home directory″ value is correct. Select the ″Load customization variables″ option and specify the proper saved variables data set prior to defining your variables:
  - If you are running at a service level prior to W502000 (indicated by an unmodifiable ″WAS home directory″ field), load the saved variables from the base Application Server configuration. Then go through and manually modify the rest of the fields with the correct deployment manager and node agent values.
  - If you are running at the W502000 service level or above (indicated by a modifiable ″WAS home directory″ field), load the saved variables from the deployment manager configuration. Then, if necessary, modify the ″WAS home directory″ field and provide the correct base Application Server directory value.
- When allocating target data sets for this option, it is recommended that you use different target data sets than any used before.
- When defining variables for this option, you will need to modify some of the defaults depending on from which configuration you loaded the variables. See "Define variables for federate base Application Server node - definitions" on page 145 for more information.
- Ensure you save your variables to a different file than one you used for any other configuration.
- When you go to generate the customized jobs, code the job card information to specify the authority of the WebSphere administrator ID. This ensures that each customized job's JOB card will have this information, which is important because that authority is required for many of the jobs to run properly.
- Prior to running the BBOWADDN job according to the instructions from the Customization Dialog, stop your base Application Server manually by stopping its location service daemon. The federation process will attempt to stop the server automatically, but doing it manually ensures it is stopped correctly. Also make sure the files bbowaddn.out and bbowaddn.err don't already exist in your /tmp directory.

Starting a deployment manager on a different MVS image

This describes steps you must follow to start your deployment manager on an MVS image different from the one on which it was originally configured.

To start your deployment manager on a different system, your system must be running at the W502000 service level.

The ability to start your deployment manager on an MVS image different from the one on which it was originally configured is handy if your original system becomes unavailable, either through a planned outage or a system failure. This way, you can still start and stop applications, make configuration updates, utilize monitors that use the PMI interface, perform other control functions, and so on. Perform the following steps to start your deployment manager on a different MVS image and ensure that client requests will successfully find the deployment manager at its new location.

Note: This works only if the deployment manager on the original MVS image is down. WebSphere Application Server for z/OS allows only one copy of the deployment manager to run at one time for any given cell.

1. Ensure that the MVS image to which you are moving the deployment manager contains a node that is already part of the cell of the deployment manager you wish to move.
2. Ensure that the location service daemon on the MVS image to which you are moving the deployment manager is up and running before you move the deployment manager.
3. Set, in the Customization Dialog, your host names and ports appropriately:
   - Ensure that the host names and ports for the deployment manager are not specific to a particular system.
   - Ensure that you use a DVIPA generic host name, rather than a system-specific host name, for the ″Node host name″ value and an asterisk (″*″) for both the ″ORB Listener host name″ and ″HTTP Transport host name.″

     Note: After initial configuration, if necessary, you can change the ″Node host name,″ ″ORB Listener host name″ and ″HTTP Transport host name″ values in the Administrative Console.
   - Consider configuring a secondary DVIPA in case the system with the primary VIPA is down.
4. Ensure that Sysplex Distributor is enabled so that, regardless of where the DVIPA has moved, it automatically routes any inbound traffic to the deployment manager.
5. Ensure that access to the PROCLIB is the same for both the original MVS image and the MVS image to which you wish to move the deployment manager.
6. Start the deployment manager on the new system. There are three ways to accomplish this, depending on the configuration of your HFSs. Follow the scenario that matches your configuration.

   Note: If you use ″static″ WLM ApplEnvs instead of Dynamic WLM Application Environments, do not specify the symbolic ″&SYSNAME″ for the node component of the ENV= string (`<node_shortname>` in the steps below).

   Scenario 1: Root HFS is shared among all processors, deployment manager’s configuration is in a configuration HFS on a system-generic mount point.

   a. Issue the start command for the deployment manager on the system on which you wish it to reside:

   ```
   S <controller_procname>,JOBNAME=<jobname>,
   ENV=<cell_shortname>.<node_shortname>.<server_shortname>
   ```

   Scenario 2: Root HFS is shared among all processors, deployment manager’s configuration HFS is mounted under a system-specific directory.

   Note: This is an undesirable scenario that you should try to avoid from the start of your system configuration. If you find yourself with this setup, however, follow these steps for the workaround.

   a. Create a symbolic link at the equivalent system-specific location on the target MVS image. The contents of the symbolic link should point back to the actual mount point, which means you should not use $SYSNAME anywhere.

   b. Issue the start command for the deployment manager on the system on which you wish it to reside:

   ```
   S <controller_procname>,JOBNAME=<jobname>,
   ENV=<cell_shortname>.<node_shortname>.<server_shortname>
   ```

   Scenario 3: Root HFS is not shared among any processors, deployment manager’s configuration HFS is mounted and accessible to only one system at a time.

a. Unmount the configuration HFS from the original MVS image and remount it (at a mount point with the same name) on the new MVS image.

b. Issue the start command for the deployment manager on the system on which you wish it to reside:

```
S <controller_procname>,JOBNAME=<jobname>,
ENV=<cell_shortname>.<node_shortname>.<server_shortname>
```

You know you are done when your deployment manager is up and running on a different MVS image and you are able to use it to make configuration updates to your environment.

Performing WebSphere Application Server for z/OS post-installation tasks

Once you have WebSphere Application Server for z/OS running smoothly on your system, there are many things you can do to test and modify the product to suit your needs even further. This article introduces the various post-installation tasks available to you.

1. Install service, as described in “Installing service to WebSphere Application Server for z/OS.”

   This task helps you apply maintenance to WebSphere Application Server for z/OS.
2. Detect and recover from failures using “Automatic restart management” on page 61.
3. Install the embedded messaging feature for use as the integral JMS provider, as described in “Tips for installing the embedded messaging feature” on page 63.
4. Run the installation verification test (IVT), as described in “Using the installation verification test” on page 64. Although the IVT is a step you must perform when customizing the product with the Customization Dialog, you can also run it any time you choose using other methods.
5. Install a new release of WebSphere Application Server for z/OS, as described in “Installing new releases and maintenance levels of WebSphere Application Server for z/OS” on page 65.

   This task helps you maintain the availability of sysplex systems when installing a new release or maintenance level of WebSphere Application Server for z/OS.

Installing service to WebSphere Application Server for z/OS

Maintaining the base z/OS environment in addition to the WebSphere Application Server can require an in-depth knowledge of applying service and the product service stream. This is especially important in a high yield production environment where availability is of the utmost importance. Therefore planning service and knowing how to best apply service are essential skills.

Contact the IBM Software Support Center for information about preventive service planning (PSP) upgrades for WebSphere Application Server for z/OS. For more information about PSP upgrades, see *WebSphere Application Server for z/OS V5.0: Program Directory*. Although the *Program Directory* contains a list of required program temporary fixes (PTFs), the most current information is available from the IBM Software Support Center.

Make copies of the product data sets and HFS structure, and apply maintenance to the copies when applying service to WebSphere Application Server for z/OS. Use the following procedure to put the maintenance into production.

1. Stop the Application Servers and the WebSphere Application Server for z/OS location service daemon.
2. Stop CTRACE.
3. Switch to the newly-serviced WebSphere Application Server for z/OS product data sets. You can do this by:

   a. Renaming the new data sets to replace the old ones.

   b. Recataloging product data sets, if the names are identical, or change WebSphere Application Server for z/OS cataloged procedures to refer explicitly to the new data sets. If you are using new data set names, you need to also handle program controls, so update the program control list.

   Verify that the MVS link list and APF list refer to the newly-serviced data sets.

4. If the WebSphere Application Server for z/OS run-time is loaded into the link pack area, delete the old modules and load the new ones, or IPL the system to load the new modules into the LPA.
5. Verify that the newly-serviced HFS data sets are correctly mounted.
6. Perform any other migration actions as instructed in PTF or APAR cover letters.
7. Start the location service daemon and Application Servers.
8. Complete any necessary post-install tasks.

You can maintain service to clients when upgrading the host cluster of WebSphere Application Server for z/OS.

Applying or restoring off a service level

5.1+ When using SMP/E, there are additional actions that you need to perform in order to apply a new service level or restore off a service level. These new WebSphere Application Server for z/OS add-ons, which you perform on the WebSphere run-time home directory that is created in the HFS by the Customization Dialog, were put into place to reduce post-install service actions. Among other things, the new add-ons also allow you to update customized files (such as setupCmdLine.sh or server.xml, which contain unique content for each node) that exist in each WebSphere run-time home directory, and to see new files appear in the WebSphere run-time home directory after you add them to the SMP/E home directory.

Note that, in addition to WebSphere Application Server for z/OS, the post-install function now applies to WebSphere Business Integration Server Foundation for z/OS. After you apply service with SMP/E, and then restart the server, the post-install function applies the post-install actions to each of these products, as long as they are installed.

For WebSphere Application Server for z/OS, WebSphere Business Integration Server Foundation for z/OS and WebSphere Portal Server for z/OS, the post-installer:

- Automatically detects and applies post PTF service
- Backs out (uninstalls) service.

The post-install function runs on all three products automatically; you don't need to specify a product. However, when you back off service, you will need to specify the product and service level you want to back off by using the ″backoutPTF.sh script command.

Note: You may find it useful to set up a rule to combat performance problems related to post-install. See Workload management (WLM) tuning tips for z/OS for more information.

Post-install tasks for after SMP/E applying a new service level:

This topic describes post-install tasks you complete after applying a new service level.

The post-install functionality includes actions you perform the first time the server is restarted after you install service with SMP/E. This applies to WebSphere Application Server for z/OS as well as WebSphere Business Integration Server Foundation for z/OS and WebSphere Portal Server for z/OS. Depending on your system restrictions, you can choose to initialize post-install processing either automatically or manually.

Automatic mode, which is recommended, includes a new step that automatically launches the ″applyPTF.sh″ shell script in the server procedure. The applyPTF.sh script, which is located in your bin directory ($WAS_HOME/bin, where $WAS_HOME is the absolute path of the WebSphere run-time home directory), makes sure that any pending post-install actions are properly applied before starting the server.

Manual mode is necessary only if the automatic mode does not conform to your organization standards (for example, if multi-step procedures are not allowed), if there is something else in your configuration that prevents the applyPTF.sh shell script from running properly, or if you are applying a particular service release that requires manual intervention.

Since post-install processing executes under the controller proc, it executes with the identity assigned to that particular proc. While that identity has sufficient authority to perform MOST HFS file actions required on the WebSphere run-time home directory, you cannot assume it to have the appropriate authority for your applyPTF.sh shell script. Therefore, you cannot perform any post- install action that requires special authority by ″inline″ apply processing. You must manually perform any such apply processing in ″batch″ mode, running applyPTF.sh or the appropriate jobs from a user that has the needed authority to apply a particular action.

Note:

- When you launch the shell script from the server proc, it automatically runs in ″inline″ mode and can only apply service that qualifies as ″inline.″
- If batch service is delivered, you'll find the needed user authority noted in the service level documentation.

You can run the post installer either automatically or manually. Follow the set of steps below that applies to your circumstance.

1. Run the post installer automatically. It is recommended that you read the ++Hold for Action instructions to determine if you need to run the post-installer in batch mode.
   a. Classify BPXBATCH using the WLM Workload Classification Rules for OMVS work. Use the started task jobnames to classify this work into the appropriate service class. This facilitates more efficient execution in automatic mode.
2. OR, run the post installer manually
   a. Edit your server control process procedures (for example, BBO5ACR) to remove the BPXBATCH step that invokes the applyPTF.sh script.
   b. After rolling service to each system, run batch job BBOWAPLB for a base Application Server node or BBOWAPLD for a deployment manager node.

The post installer component enforces that certain actions are performed successfully and preconditions are met before applying service. Otherwise, warnings or error messages result. If you launched the applyPTF.sh script from the server proc, the output is appended to the file $WAS_HOME/properties/service/logs/ applyPTF.out. If you ran applyPTF.sh manually from the shell, the output goes right to stdout (on the shell from where the applyPTF.sh script was run) and WTO messages are issued. In the latter case, no output is appended to the applyPTF.log log file.

Whenever actions are performed on the run-time home directory, a log file is kept. When warnings or errors occur, the absolute path to the log file is displayed so that you can examine the details of the problem. The following is a list of common errors that might occur when the post installer (applyPTF.sh) applies service. The errors will most likely appear in the form of error codes on the console on which applyPTF.sh was issued.

The post installer ran and determined that the SMP/E home directory was SMP/E rolled back in service to an earlier version than that at which the WebSphere run-time home directory is running.
This occurs if the SMP/E restore off was performed without first running the backoutPTF.sh shell script. The server will not start if it detects this condition, and post-install action halts without applying service.

The post installer issued a warning message while installing service.
The WebSphere Application Server for z/OS multi-product post-installer detected warnings that were

issued during the application of post-install service for the product listed. The application of service was successful, but the warning messages should be examined. The warning messages are listed in the log file in the HFS file that is specified in the message.

The post installer incurred an error and stopped processing.

The WebSphere Application Server for z/OS multi-product post-installer encountered an error while installing service for the product indicated in the message. The details of the error are contained in the HFS log file that is specified in the message. When this happens, review the log and correct the error. The servers will not be permitted to start until the error is corrected.

The post-installer encountered warnings while applying service.

The WebSphere Application Server for z/OS multi-product PTF post-installer detected warnings that were issued during the application of post-install service. The warnings may have been issued while post-install service was being applied for WebSphere Application Server for z/OS or any of the extension products that are installed. A BBOO0250W message should have already been issued for each product that encountered warnings when the post-install service was applied.

Reply with `Continue` to continue starting the Application Server. Reply with `Cancel` to cancel starting the Application Server. Because service has already been completed, the multi-product PTF post-installer will not run again the next time the server is restarted..

Post-install tasks to complete before SMP/E restoring off a service level:

This topic describes post-install tasks you complete before SMP/E restoring off of a new service level.

When restoring off service, you need to ″back off″ any post-install actions that were applied during that service. Follow these steps to run the ″backoutPTF.sh″ shell script, which is located in your bin directory ($WAS_HOME/bin, where $WAS_HOME is the absolute path of the WebSphere run-time home directory) and will take care of this for you.

The following is a list of common errors that might occur when you use backoutPTF.sh to back out of service. The output goes right to stdout (on the shell from where the applyPTF.sh script was run).

A valid, currently installed PTF or APAR name and the target service level must be specified (case sensitive).

If a PTF or APAR isn't specified, then the ″usage″ of the command is printed along with a listing of the currently installed and pending service units.

1. Bring down those servers in the node for which you are doing backoff processing.
2. 5.1+ Launch the backoutPTF.sh script. You will need to specify the product (Websphere|WBI) and the committed service level to which you intend SMP/E to revert (″restore off″).

   Example: This is an example of the command used to run the backoutPTF.sh shell script, run from the shell by a user with the proper authority. It prepares the WebSphere Business Integrated Server Foundation for z/OS run-time home directory to run at service level ″W510001.″

   ```
   backoutPTF.sh Websphere W510001
   ```

   Note:

   - For WebSphere Application Server for z/OS only, the shell script uses service levels of the form ″Wvrmxyy″ (PTF), ″AQxxxxx″ (++APAR), or UQxxxx (SMP/E service).
   - The shell script is case sensitive.
3. After using the backoutPTF.sh shell script, follow the normal SMP/E procedures for restoring off a service level.

You know you are done when you are successfully back on the desired service level.

Automatic restart management

If you have an application that is critical for your business, you need facilities to manage failures. z/OS provides rich automation interfaces, such as automatic restart management, that you can use to detect and recover from failures. Automatic restart management handles the restarting of servers when failures occur.

WebSphere Application Server for z/OS uses the z/OS Automatic Restart Management (ARM) to recover Application Servers. Each Application Server running on a z/OS system (including servers you create for your business applications) are automatically registered with an ARM group. Each registration uses a special element type called SYSCB, which ARM treats as restart level 3, assuring that RRS restarts before any Application Server.

Tips:

- If you have automatic restart management (ARM) enabled on your system, you may wish to disable ARM for the WebSphere Application Server for z/OS address spaces before you install and customize WebSphere Application Server for z/OS. During customization, job errors may cause unnecessary restarts of the WebSphere Application Server for z/OS address spaces. After installation and customization, consider enabling ARM.
- If you are ARM-enabled and you cancel or stop a server, it will restart in place using the `armrestart` command.
- It is a good idea to set up an ARM policy for your deployment manager and node agents.
- If you start the location service daemon on a system that already has one, it will terminate.
- Every other server will come up on a dynamic port unless the configuration has a fixed port. Therefore, the fixed ports must be unique in a sysplex.
- If you issue STOP, CANCEL, or MODIFY commands against server instances, be aware of how automatic restart management behaves regarding WebSphere Application Server for z/OS server instances:

Table 4. Behavior of automatic restart management regarding WebSphere Application Server for z/OS server instances

If you issue . . .	**Automatic restart management will . . .**
STOP *address_space*	not restart the address space
CANCEL *address_space*	not restart the address space
CANCEL *address_space*, ARMRESTART	restart the address space
MODIFY *address_space*, CANCEL	not restart the address space
MODIFY *address_space*, CANCEL,ARMRESTART	restart the address space

Activating automatic restart management

You must have access to the couple data set format utility, IXCL1DSU, in SYS1.MIGLIB. If you plan to modify the automatic restart management policy, you must have access to the administrative data utility, IXCMIAPU, also in SYS1.MIGLIB, and have UPDATE authorization to the RACF FACILITY class MVSADMIN.XCF.ARM. To start a policy, you must have READ authorization to the RACF FACILITY class MVSADMIN.XCF.ARM.

Though servers automatically register with automatic restart management, you must activate the arm component itself, which means you must:

1. Allocate an ARM couple data set
2. Start the automatic restart management policy

If automatic restart management is not active, WebSphere Application Server for z/OS issues an error message to the hardcopy log.

You are not required to change the automatic restart management policy. However, you will have to modify this policy if you want to create custom restart groups. For instance, it is not required or recommended that you start the node agent or deployment manager servers on another system. These servers will never have any transactional recovery to perform. Therefore, they should only be set up for restart-in-place. For complete information about how to modify the policies, see *z/OS MVS Setting Up a Sysplex, SA22-7625*.

The following procedure is intended to give you enough information to get automatic restart management running. Defining automatic restart management policies would require the z/OS manual mentioned above.

1. If you have not already formatted a couple data set for policies, do so now. For details, see *z/OS MVS Setting Up a Sysplex*
2. Submit the job to format the ARM couple data set.
3. Optional: Modify the automatic restart management policy. To get started, you do not need to modify the policy. If you do want to modify the automatic restart management policy, go to *z/OS MVS Setting Up a Sysplex*, and follow the instructions in that manual.
4. Issue the following operator commands to start the automatic restart management policy:

   ```
   SETXCF COUPLE,TYPE=ARM,PCOUPLE=(dsname,vvvvvv)
   SETXCF START,POLICY,TYPE=ARM
   ```

   where

   dsname

   Is the data set name for the couple data set.

   vvvvvv

   Is the volume serial of the volume on which the couple data set resides.

You are done when the SETXCF commands complete successfully.

Changing automatic restart management policies

Because server instances register with the default restart group, when a system failure occurs, automatic restart management attempts to restart the entire default group on another system in the sysplex. If you want to create a restart group other than this default group, you must follow the following rules and restrictions that apply for z/OS automatic restart management policies. For more information about how to actually change these policies, see, *z/OS MVS Setting Up a Sysplex, SA22-7625*.

- To change the policy, you need to know the existing element names for the server instances and how to name new elements for additional instances. The element names for these server instances are formed by concatenating the cell short name and the servers specific short name.

  For example, if you have a cell named PLEX1 and server named BBOS001, the ARM element name would be PLEX1BBOS001.

  Since wildcard characters can be used in the ARM policy, it is possible to exclude an entire group of servers by using a common naming scheme within your cell.

  For example, the following section of the ARM policy will prevent any node agents from starting, assuming each node agent in your cell has a name that adheres to the form BBONxxx:

  ```
  RESTART_GROUP(WEBSPHERE)
  ELEMENT(PLEX1BBON*)
  RESTART_ATTEMPTS(0,150)
  RESTART_TIMEOUT(600)
  READY_TIMEOUT(1200)
  TERMTYPE(ALLTERM)
  RESTART_METHOD(BOTH,PERSIST)
  ```

  This ARM policy will also prevent the node agent from restarting in-place. This specification can be modified by changing the RESTART_METHOD and TERMTYPE parameters. See *z/OS MVS Setting Up a Sysplex*, SA22-7625 for more information.

- If you create a restart group, keep the following in the same restart group and set the restart order for the elements as indicated:
  1. RRS
  2. DB2 with IRLM, IMS, CICS, and other transaction or resource managers, if used by your Application Servers in the restart group
  3. Your server instances.

Either set up the location service daemon and node agent for restart-in-place or remove them from your ARM policy. Since WebSphere Application Server must be running on all systems that might be used to perform recovery, the Application Servers will use the location service daemon and node agent that are already running on the alternate system. If the location service daemon attempts to restart on the alternate system, it will fail. If the node agent restarts on the alternate system, it will have no recovery work to do.

Displaying the status of ARM-registered address spaces

Perform the following steps to use ARM to display the status of ARM registered address spaces (including the address spaces of server instances) in the WebSphere Application Server for z/OS environment:

1. Initialize all servers.
2. Display all registered address spaces (including the address spaces of server instances). Issue the following command:
   ```
   d xcf,armstatus,detail
   ```

Tips for installing the embedded messaging feature

When installing IBM WebSphere Application Server for z/OS, you can install the embedded messaging feature for use as the Java Message Service (JMS) provider, which is also frequently called the integral JMS provider. The integral JMS provider is installed using the Customization Dialog.

Installing the embedded messaging server feature if WebSphere MQ Version 5.3 is already installed

You have a choice if you already have WebSphere MQ Version 5.3 installed:

- You can install only the embedded messaging client feature on a machine that already has WebSphere MQ Version 5.3.

  To use WebSphere MQ Version 5.3 as the JMS provider, install the IBM WebSphere Application Server product with only the embedded messaging client feature. Installing and using the WebSphere Application Server embedded messaging client feature is recommended with either the server feature or the full WebSphere MQ Version 5.3 product.

  WebSphere Application Server messaging applications can use the WebSphere MQ Version 5.3 product as the JMS provider. Using the client feature, however, requires that you install the WebSphere MQ Version 5.3 Java messaging feature.
- You can install the embedded messaging server and client features on a machine that already has WebSphere MQ Version 5.3.

  To install the embedded messaging server feature when WebSphere MQ Version 5.3 is already installed, upgrade WebSphere MQ Version 5.3:
  - Apply the CSD01 update to the original WebSphere MQ Version 5.3 release, or move to the WebSphere MQ Version 5.3 refresh release (which includes CSD01).
  - Install the WebSphere MQ Version 5.3 features, server and Java messaging, which the WebSphere Application Server embedded messaging server feature requires.

If you install WebSphere MQ Version 5.3 without the required features, the installation of either IBM WebSphere Application Server embedded messaging feature is unsuccessful because of prerequisite check errors. The IBM WebSphere Business Integration Server Foundation package includes installation images of the WebSphere MQ Version 5.3 product and the WebSphere MQ Event Broker product, with restricted licensing. You can use the products to install the required WebSphere MQ Version 5.3 features or to install the refresh release of WebSphere MQ Version 5.3 for use with IBM WebSphere Business Integration Server Foundation.

Avoiding a coexistence problem between embedded messaging, IBM WebSphere Studio Application Developer Integration Edition, and IBM WebSphere Application Server

The IBM WebSphere Studio Application Developer Integration Edition and IBM WebSphere Application Server both include an option to install embedded messaging. The embedded messaging option in these two products is incompatible.

To avoid this problem, do not install embedded messaging for both products on the same machine.

For current information available from IBM Support on known problems and their resolution, see this customized IBM Support page.

IBM Support has documents that can save you time gathering information needed to resolve this problem. Before opening a PMR, see this customized IBM Support page.

Using the installation verification test

You initially run the installation verification test (IVT), which verifies that WebSphere Application Server is configured correctly for your system, during ISPF customization of each of your systems. If you want to run the IVT at a time other than during initial customization, however, there are two methods from which you can choose.

Note: These options are available only if the user is running a base Application Server configuration. Select either method to invoke the IVT:

- Use the BBOWIVT job, as described in “Running the Installation Verification Test with a job.”
- Run `ivt.sh` from a command line, as described in “Running the Installation Verification Test from a command line.”

Running the Installation Verification Test with a job

The Application Server must be running.

Follow these steps to run the Installation Verification Test using the BBOWIVT job.

1. Verify that the Application Server is running.

   Note: The IVT will end unsuccessfully if the server is not running.
2. Confirm that the ivtApp application is installed and started.
3. Submit the job BBOWIVT.

After initialization, the IVT runs its series of verification tests and reports pass or fail status for each in the messages generated by the BBOWIVT job. Once it finishes, check the results in the install_root/logs/ivt.log file.

Running the Installation Verification Test from a command line

The Application Server must be running.

Follow these steps to run the Installation Verification Test from a command line.

1. Verify that the Application Server is running.

   Note: The IVT will end unsuccessfully if the Application Server is not running.

2. Confirm that the ivtApp application is installed and started.
3. From a command line, navigate to the `/WebSphere/V5R0M0/AppServer/bin` directory.
4. Issue the following command:

   ```
   ivt.sh [-p port_number] [-host host_name]
   ```

   where
   - `-p port_number` is an optional argument that specifies your port number. If you do not specify a port number, the program will use the default port number value of 9080.
   - `-host host_name` is an optional argument that specifies your host name. If you do not specify a host name, the program will use the host name value that is set in your TCP/IP hosts file.

   Example:

   ```
   /WebSphere/V5R0M0/AppServer/bin> ivt.sh -p 9090 -host myhost
   ```

The IVT will run, producing a series of verification tests and reporting pass or fail status in the messages generated by the BBOWIVT job. It will also log results to the `install_root\logs\ivt.log` file.

Note: After command line initialization, the IVT runs its series of verification tests and reports pass or fail status for each. Once it finishes, check the results in the install_root/logs/ivt.log file.

Installing new releases and maintenance levels of WebSphere Application Server for z/OS

This task uses IBM functions and methods to migrate WebSphere Application Server for z/OS in a sysplex from one functional level to the next, with as little disruption as possible. This task does not apply to WebSphere Application Server for z/OS running in a monoplex, or on a single system in a sysplex. Installing a new code level in a monoplex requires you to shut down WebSphere Application Server for z/OS. You have no choice but to disrupt service to clients of the monoplex system. You can also ignore this topic if disrupting service to clients is not a problem when running WebSphere Application Server for z/OS in a sysplex.

IBM functions and methods for migrating WebSphere Application Server for z/OS include the following:
- Documenting types of migration methods.
- Providing a function to offload WebSphere Application Server for z/OS configuration data for reloading later into a new or existing configuration.
- Managing WebSphere Application Server variables in a central system-management database, so that there is no confusion about where to go for authoritative configuration data.
- Supporting differing functional levels of WebSphere Application Server for z/OS within the same network or within the same z/OS sysplex while you perform an orderly migration of the WebSphere Application Server for z/OS run time from one functional level to another. The assumption is that this migration happens over a relatively short period of time, perhaps a few weeks.

You can install new functional levels of WebSphere Application Server for z/OS without disrupting service to your clients, provided you have the proper HFS structure in a sysplex and you use what is defined here as a *rolling upgrade*. The rolling upgrade method upgrades the host cluster of WebSphere Application Server for z/OS by upgrading clustered host instances one at a time. This lets you keep available service to clients while you do the upgrade. Availability of service continues because only one system is removed from the host cluster at any time, while other clustered host instances remain running.

You have the option of using either of two methods for performing the rolling upgrade. The optional methods differ in the HFS structure they use. One method uses a version-specific HFS structure. The other uses an alternate HFS structure. Read through the procedure for using each method to determine which is better for you. The main determinant is whether to use a shared HFS.

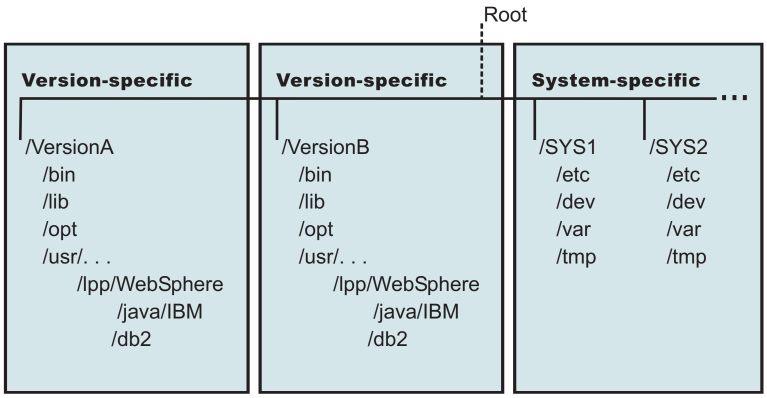
1. Optional: “Using a version-specific HFS structure to upgrade WebSphere Application Server for z/OS.”
2. Optional: “Using an alternate HFS structure to upgrade WebSphere Application Server for z/OS” on page 67.

You can maintain service to clients when upgrading the host cluster of WebSphere Application Server for z/OS.

Using a version-specific HFS structure to upgrade WebSphere Application Server for z/OS

This task describes one of the two methods for performing a *rolling upgrade* of WebSphere Application Server for z/OS in a sysplex that is introduced by “Installing new releases and maintenance levels of WebSphere Application Server for z/OS” on page 65. This task describes using a version-specific HFS structure to upgrade WebSphere Application Server for z/OS.

Using a version-specific HFS structure in a rolling upgrade means that two different versions of WebSphere Application Server for z/OS run in the sysplex at the same time, each with a separate HFS structure. A conventional sysplex environment has but one HFS structure that all WebSphere Application Server for z/OS systems share. To run a second, different version of WebSphere Application Server for z/OS at the same time requires a second HFS structure that is specifically for the second version as shown in the following illustration:



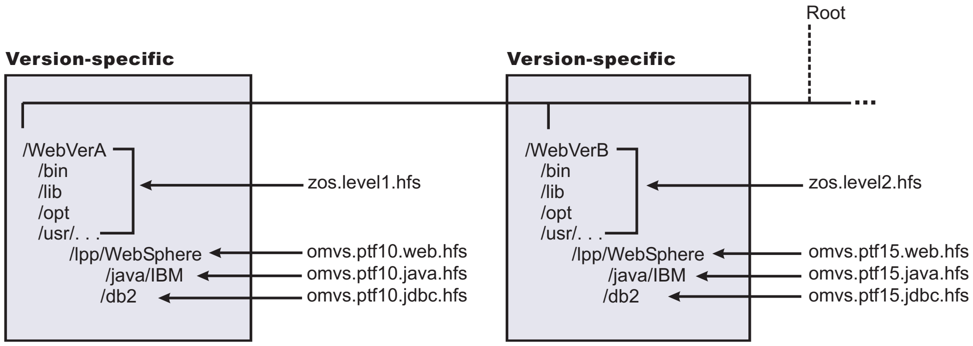
Each version-specific HFS structure requires its own `/usr` directory (containing a version-specific level of WebSphere Application Server for z/OS and Java) and `/db2` directory (containing a version-specific level of JDBC) because of code-level interdependencies between products.

With the dual HFS structure in place, you can mount one code-level version of WebSphere Application Server for z/OS, run the host cluster from that mount point, and upgrade a second mount point where you mounted the other version of WebSphere Application Server for z/OS.

Example: Assume you mount a version-specific HFS for one service level (PTF 10) at /VersionA and another version-specific HFS for another service level (PTF 15) at /VersionB:

```
mount omvs.ptf10.was.hfs  at /VersionA/usr/lpp/WebSphere
mount omvs.ptf10.java.hfs at /VersionA/usr/lpp/java/IBM
mount omvs.ptf15.was.hfs  at /VersionB/usr/lpp/WebSphere
mount omvs.ptf15.java.hfs at /VersionB/usr/lpp/java/IBM
mount omvs.ptf15.jdbc.hfs at /VersionB/usr/lpp/db2
```

The HFS structure appears in the following illustration:



You can use a symbolic link to determine which code version any system in the sysplex addresses:

```
/usr --> $VERSION/usr
```

You use the **SETOMVS** command to control how $VERSION resolves. In this example, set the initial value of $VERSION for each system in the sysplex to `VersionA`. All systems that use the symbolic link to refer to the `/usr` directory resolve to `/VersionA/usr`.

For any system in the sysplex to use the HFS structure associated with PTF15, change the value of $VERSION on that system (and only on that system) to `VersionB`. Accordingly, any references on that system to `/usr` resolve to `/VersionB/usr` through the symbolic link.

Use the following procedure to switch the code level for a clustered host instance:

1. Install the new code, copy it to a new data set, and mount the data set at the VersionB mount point.
2. Shut down all Application Servers and WebSphere Application Server for z/OS on that clustered host instance
3. Use the **SETOMVS** command to change $VERSION to VersionB
4. Using SET PROG, load the LPA modules from data sets associated with the new level
5. Change the start procedures to address the new code level load libraries
6. Restart WebSphere Application Server for z/OS and the Application Servers.
7. Repeat the prior steps of this process for each clustered host instance, one at a time, to upgrade the code level of WebSphere Application Server for z/OS throughout the sysplex without disrupting service to your clients.

Each code level of WebSphere Application Server for z/OS is designed to tolerate an older code level. Differing levels of WebSphere Application Server for z/OS can coexist compatibly within the sysplex during the upgrade process.

In cases when a new level of WebSphere Application Server for z/OS introduces new functions, all members of the host cluster run in compatibility mode during the upgrade process. When all clustered host instances are at the new code level, restart each instance, one by one, to enable the new function.

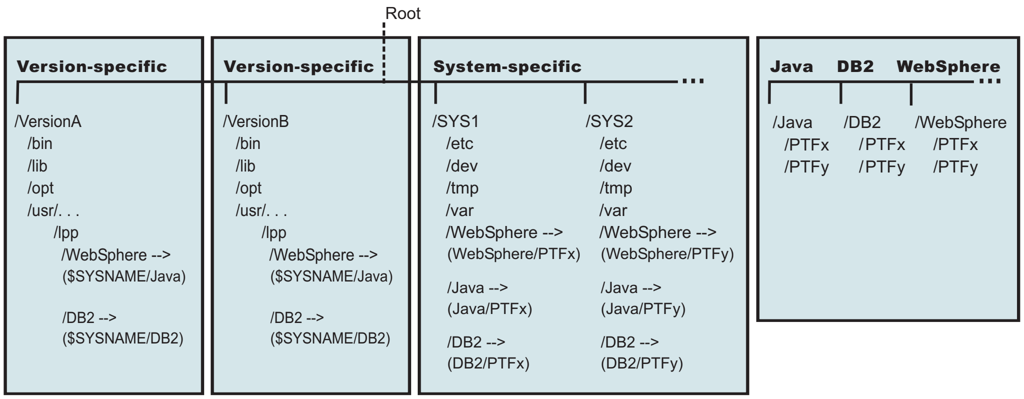
Read about “Using an alternate HFS structure to upgrade WebSphere Application Server for z/OS.”

Using an alternate HFS structure to upgrade WebSphere Application Server for z/OS

This task is the second of the two methods for performing a *rolling upgrade* of WebSphere Application Server for z/OS in a sysplex that is introduced by “Installing new releases and maintenance levels of WebSphere Application Server for z/OS” on page 65. “Using a version-specific HFS structure to upgrade

WebSphere Application Server for z/OS" on page 66 describes the first method. This task describes using an alternate HFS structure to upgrade WebSphere Application Server for z/OS.

The alternate HFS structure accomplishes the same objective as the version-specific HFS structure, but does not mount a product HFS directly off the version-specific subdirectories (referred to by the $VERSION symbolic link). Rather, version-specific subdirectories refer to system-specific subdirectories using symbolic links with the $SYSNAME symbol. In turn, system-specific subdirectories refer to program product subdirectories through symbolic links. The alternate HFS structure is depicted in the following illustration:



The alternate HFS structure has the following characteristics:

- **Version-specific subdirectories that allow systems in the sysplex to refer to differing versions of system code.**

  The WebSphere Application Server for z/OS, Java, and DB2 for OS/390 (JDBC) subdirectories do not contain product code but instead, contain symbolic links to system-specific subdirectories through the use of the $SYSNAME symbol. As far as WebSphere Application Server for z/OS is concerned, you need not change these symbolic links. However, you should plan for creating version-specific structures for future system upgrades.
- **System-specific subdirectories that contain symbolic links to WebSphere Application Server for z/OS, Java, and DB2 for OS/390 (JDBC) subdirectories in the sysplex root.**

  The symbolic links point to specific code levels (for example, `WebSphere/PTFx`). To change the code level that a system uses, you change these symbolic links.
- **Individual subdirectories for WebSphere Application Server for z/OS, Java, and DB2 for OS/390 (JDBC) components.**

  Each of these subdirectories can have one or more subdirectories for a specific code level.
- **Shared subdirectories, such as the `WebSphere390` subdirectory.**

With the alternate HFS structure in place, you can mount one or more code levels of WebSphere Application Server for z/OS, Java, or DB2 for OS/390 (JDBC) under their individual component subdirectories. Each system-specific subdirectory uses symbolic links to component code levels and can refer to new code levels by changing those symbolic links.

There are certain advantages to the alternate HFS structure:

- You can stage product upgrades and service in a sysplex environment with minimal impact to availability. You can stage product upgrades or service without applying the upgrade or service to all products at the same time.
- You can upgrade one product without duplicating another version-specific ($VERSION) structure by placing the level of control at the system-specific subdirectories and linking to those subdirectories through the $SYSNAME symbol. However, it is beneficial to plan for a second version-specific structure so you are prepared for future system upgrades.

- You can help performance by using version-specific subdirectories that are read-only. Changes are in the system-specific ($SYSNAME) subdirectory, which is read/write.
- You can save storage space because you need not duplicate version-specific HFS structures for program product upgrades.

Example: Assume you have an individual component directory for WebSphere Application Server for z/OS (/WebSphere), Java (/Java), and DB2 for OS/390 (JDBC) (/DB2). Assume that each directory contains two subdirectories, one for PTFx (/PTFx) and one for PTFy (PTFy). Also assume that the code for each component update is in its own HFS data set, such as OMVS.PTFX.WEB.HFS, OMVS.PTFX.JAVA.HFS, and OMVS.PTFX.JDBC.HFS, for example. The mount commands are:

```
MOUNT FILESYSTEM('OMVS.PTFX.WEB.HFS')   MOUNTPOINT('/WebSphere/PTFx') TYPE(HFS) MODE(RDWR)
MOUNT FILESYSTEM('OMVS.PTFX.JAVA.HFS')  MOUNTPOINT('/Java/PTFx')      TYPE(HFS) MODE(RDWR)
MOUNT FILESYSTEM('OMVS.PTFX.JDBC.HFS')  MOUNTPOINT('/DB2/PTFx')       TYPE(HFS) MODE(RDWR)

MOUNT FILESYSTEM('OMVS.PTFY.WEB.HFS')   MOUNTPOINT('/WebSphere/PTFy') TYPE(HFS) MODE(RDWR)
MOUNT FILESYSTEM('OMVS.PTFY.JAVA.HFS')  MOUNTPOINT('/Java/PTFy')      TYPE(HFS) MODE(RDWR)
MOUNT FILESYSTEM('OMVS.PTFY.JDBC.HFS')  MOUNTPOINT('/DB2/PTFy')       TYPE(HFS) MODE(RDWR)
```

System SYS1 refers to the PTFx levels of code through these symbolic links:

```
/WebSphere --> /WebSphere/PTFx
/Java      --> /Java/PTFx
/DB2       --> /DB2/PTFx
```

If you want system SYS1 in the sysplex to use the HFS structure associated with PTFy, change the symbolic links for /WebSphere, /Java, and /DB2:

```
/WebSphere --> /WebSphere/PTFy
/Java      --> /Java/PTFy
/DB2       --> /DB2/PTFy
```

Use the following procedure to switch the code level for the WebSphere Application Server for z/OS for the clustered host instance on SYS1.

1. Install the new code for WebSphere Application Server for z/OS, Java, and DB2 for OS/390 (JDBC), copy each component to its own data set, and mount the data under its component subdirectory. WebSphere Application Server for z/OS, Java, and DB2 for OS/390 (JDBC) code levels are usually interdependent, so keep the level of each component coordinated with the others.
2. Shut down all Application Servers and the WebSphere Application Server for z/OS clustered host instance on SYS1.
3. Change the symbolic links for the system-specific subdirectories for SYS1.
4. Load new run-time modules into LPA and update the link list. You can do this dynamically, but IBM recommends you re-IPL the system.
5. Change the start procedures to address the new code level load libraries.
6. Restart WebSphere Application Server for z/OS and the Application Servers.

By repeating this process for each clustered host instance, one at a time, you can upgrade the code level of WebSphere Application Server for z/OS throughout the sysplex without disrupting service to your clients.

Post-installation notes on the error log

After installation is complete, use the administrative console to change the log stream name or create new log stream names for servers or servants.

Note:

- A server’s error log stream setting overrides the general WebSphere Application Server for z/OS setting, and a servant setting overrides a server setting. Thus, you can set up general error logging, but direct error logging for servers or servants to specific log streams.

- If you create a new log stream name through the administrative console, you must configure a new log stream on z/OS and, if using the coupling facility, define a corresponding new coupling facility log stream.
- If you changed an existing log stream, or created a new one, you probably need to restart WebSphere Application Server for z/OS. When the name of a log stream is changed through the Administrative Console, in most cases a restart of WebSphere Application Server for z/OS is required before the change becomes effective. The only case when the change takes effect automatically is when the log stream name is changed for a server along with other changes that cause the server to be restarted.

If you want WebSphere Application Server for z/OS messages that occur during execution of a z/OS client to be recorded in an error log stream, code the client_ras_logstreamname WebSphere variable in its environment file, then initialize the client. For more information about client_ras_logstreamname and the related variable ras_log_logstreamName, see the WebSphere variables in the administrative console or the information center.

Our RACF samples BBOWBRAC and BBODBRAC give UPDATE authority to the run-time control and servant user IDs for the log stream you created (they require that you supply a log stream name). After installation and customization, if you want to grant access to the log stream:
- For each server identity that writes to the log stream (or client identity, if you allow clients to write to the error log stream), assign UPDATE access to the log stream.
- For each user who browses the error log stream, assign READ access.

Follow the sample RACF commands in BBOWBRAC or BBODBRAC.

Migrating

Migrating is an activity in which you take advantage of existing materials. Migration tasks and tools help you upgrade the product and its prerequisites, reuse existing application components when feasible, and transfer administrative configurations from your past version to a current one.

Migration of WebSphere Application Server products is about leveraging the existing environment and applications and changing them to be compatible with the current product version.

5.1 + Because of the changes in architecture/structure compared to prior WebSphere Application Server for z/OS releases, the IBM WebSphere Application Server, Version 5.1 migration strategy is one of coexistence with levels prior to V5.0. Upgrading of WebSphere Application Server Version 3.5SE or WebSphere Application Server for z/OS Version 4.0.1 to V5.1 is not supported. However, V5.1 can coexist with prior levels of WebSphere Application Server for z/OS on a z/OS image or Sysplex. Support for migrating V4.0.1 applications to V5.1 has been shipped in V4.0.1 service (W401502). This support has been provided as a new -v5mp option for the 390fy tool. WebSphere Application Server for z/OS V5.1 servers must be defined manually in preparation for the migration of the application.

5.1 + Starting with V5.1, additional migration utilities have been shipped to assist with the migration of configurations and applications from an existing V5.0 configuration to a newly installed V5.1 configuration. In addition, customers in a 5.0 network deployment configuration have the capability to migrate to V5.1 in a staged manner.

Migration Redbook

Migrating to WebSphere V5: An End-to-End Migration Guide is available from the Redbooks Web site at http://www.ibm.com/redbooks. To locate the Redbook, search for the document number SG24-6910-01. The Redbook provides a broader coverage than the information center articles, including more detailed planning information for application migration and WebSphere Studio Application Developer tooling and samples. *Migrating Applications from WebSphere for z/OS V4 and V3.5 to V5* is also available from the Redbooks Web site. Search on document number SG24-7044-00 to locate this book.

Version 5.0 migration 5.1+

Migration from Version 5.0 to Version 5.1 involves minimal change because both releases implement the Java 2 Platform, Enterprise Edition (J2EE) 1.3 specification. Version 5.0.x supports the Java 2 SDK Version 1.3.1. Version 5.1 supports the Java 2 SDK Version 1.4.1. V5.0 applications can run on either release without change. V5.1 applications can run on either release without change if you compile applications on a V5.1 node with the compatibility options. See the ″Changes for application developers″ section for more information.

A new migration tool is available for the migration of V5 configurations. This tool allows you to save an existing V5 configuration and restore it to a new installation of V5.1. See "V5.1 Migration utility" on page 79 for more information on this tool.

Version 4.0.1 migration 5.1+

Migration from Version 4.0.1 to Version 5.1 involves minimal change because both releases implement the Java 2 Platform, Enterprise Edition (J2EE) specification. (V4 implements the J2EE 1.2 specification. V5.1 implements the J2EE 1.3 specification.) Existing V4.0.1 EAR files can be moved to V5.1 after being processed through the new -v5mp option of the 390fy tool.

Version 3.5 SE migration: Moving to the J2EE model 5.1+

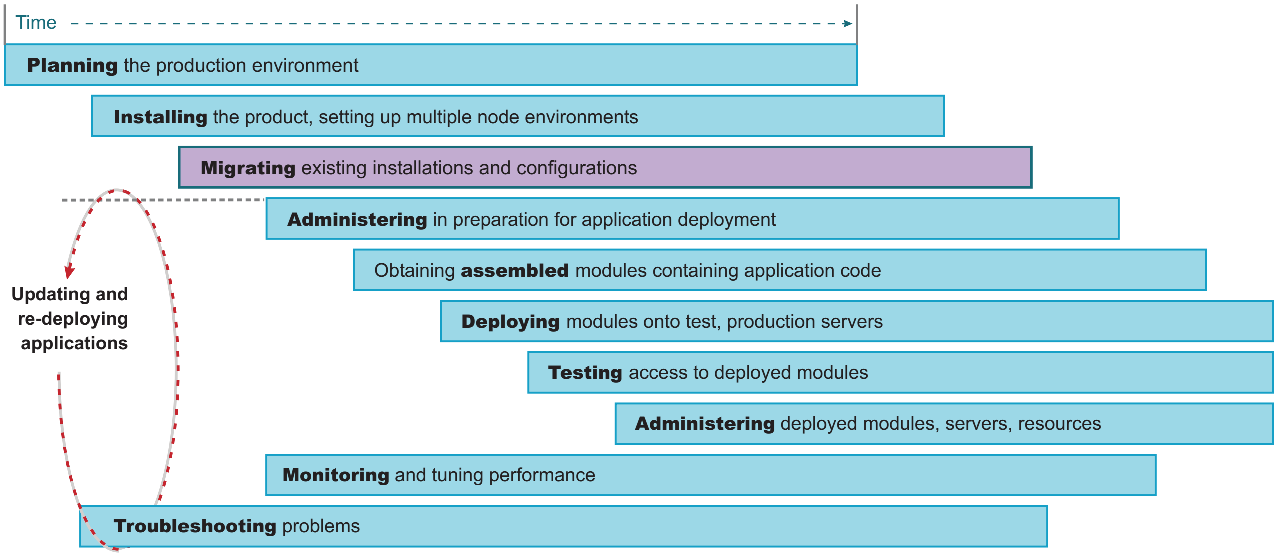
V3.5 SE users upgrading to V5.1 are moving to a platform that is fully compliant with J2EE specifications. J2EE technology clearly separates development and the creation of applications from application administration, deployment and management. Migration from V3.5SE involves changes in application structures, development, and deployment.

The J2EE model enables you to develop applications independently from their final deployment environment. This task separation simplifies the process of promoting an application from initial development through production, or moving an application from one server to another. The intent is to change only the application deployment parameters, and not the application code.

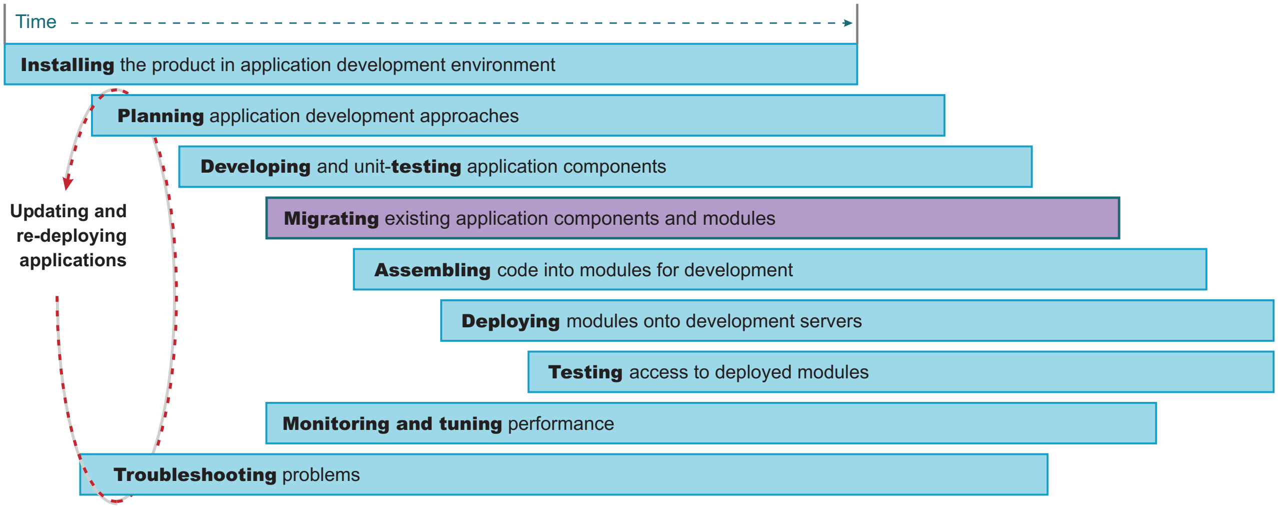
New assembly and deployment user roles and tools

5.1+ V3.5SE applications are file based and defined to the server through the WAS.CONF file. V5.1 application settings are defined in J2EE deployment descriptors, by *application assemblers* using the assembly toolkit, which is available on the IBM WebSphere Application Server Toolkit (ASTK) CD-ROM. A *deployer* follows the instructions in the deployment descriptor to install the application into a particular server environment. (See Deploying).

A simple timeline of activities for Planning, Installer and Administrator roles.


Time
Planning the production environment
Installing the product, setting up multiple node environments
Migrating existing installations and configurations
Administering in preparation for application deployment
Obtaining **assembled** modules containing application code
Deploying modules onto test, production servers
Testing access to deployed modules
Administering deployed modules, servers, resources
Monitoring and tuning performance
Troubleshooting problems
Updating and re-deploying applications

A simple timeline of activities for an Application Developer role.


Time
Installing the product in application development environment
Planning application development approaches
Developing and unit-**testing** application components
Migrating existing application components and modules
Assembling code into modules for development
Deploying modules onto development servers
Testing access to deployed modules
Monitoring and tuning performance
Troubleshooting problems
Updating and re-deploying applications

5.1 + Suppose you need to update an application that is already deployed to the server. Unless you must change information that affects the bindings of an installed application, you can edit and save the deployment descriptors in place. Redeploy such an application, by opening the assembly toolkit, which is available on the IBM WebSphere Application Server Toolkit (ASTK) CD-ROM directly on the `installedApps` folder that holds the application.

5.1 + You can also create or edit applications manually. For example, to add a JavaServer Page (JSP) file or to change a servlet class, place the new or changed file in the appropriate location in the `installedApps` folder. Redeploy an installed application that requires changes to binding, by exporting the application through the Administrative Console, edit the application in the assembly toolkit, which is available on the IBM WebSphere Application Server Toolkit (ASTK) CD-ROM, and reinstalling the application through the administrative console. Because existing binding information is saved during the export step, the only additional binding information needed is for the components or modules added during editing.

Version 5.1 administration 5.1+

Version 5.1 administrators can install a Version 5.1 application onto the server and bind it to an environment. This capability enables administration at the application and module level. Administrators no longer must manage individual servlets, JSP files, or enterprise beans.

The relationship between applications and Application Servers has changed in the move to a J2EE base. After creating a J2EE application, deploy it by installing the application onto Application Servers through the administrative console. Through the administrative console, you can view, start, or stop deployed modules by the application to which they belong or by the Application Server on which they are deployed.

Support for J2EE resources. In addition to JDBC drivers and data sources, several resource types are added as the product transitioned to a J2EE base. Administrators now can configure URLs, the Java Message Service (JMS), and the JavaMail API. In each case, you can create a resource provider (such as a JMS provider) and then create resource factories for each provider (such as JMS destinations and JMS connections). The default JavaMail API provider does not appear in the administrative console because it is not configurable. You cannot create additional JavaMail API providers.

More information about the J2EE standard is available from the links in “Installation: Resources for learning” on page 157.

Role based security. Version 5.1 security is consistent with J2EE role-based security specifications. Deployment descriptors specify roles for an application. As the administrator installs the application, these roles are bound to users or to groups.

In the administrative console, a Security Center lets you perform all security tasks from a single location. Such tasks include everything from changing the binding information for roles in an application to setting Secure Sockets Layer (SSL) properties to enabling security. Application-specific security tasks are also supported, through the property sheets for each application.

The ability to administer sensitive security functions, such as authorization tasks, can be configured as limited to z/OS System programmers using SAF facilities, if desired.

V3.5SE developers manually create, edit, and view application configurations.

5.1+ V5.1 developers use the assembly toolkit, which is available on the IBM WebSphere Application Server Toolkit (ASTK) CD-ROM to package, edit, and view J2EE applications.

Packaging J2EE applications includes:
- Copying appropriate files into the enterprise archive (EAR) file, including classes, JSP files, HTML files, image files, and so on
- Defining deployment descriptor files for modules and applications

5.1+ In Version 5.1, the assembly toolkit is available on the IBM WebSphere Application Server Toolkit (ASTK) CD-ROM. The assembly toolkit lets you copy files with appropriate relative paths into the archive and provides a GUI method for defining deployment descriptors. Also, developers can set environment-specific binding information through the assembly toolkit. These bindings are defaults for the administrator to use when installing the application through the administrative console.

You can define IBM extensions to the J2EE specification, for example supporting servlets to be served by class name. These extensions are saved in an XML file that is separate from the standard J2EE deployment descriptor to ensure portability to other Application Servers.

5.1 + If your existing applications use IBM extensions from earlier product versions, it might be necessary for you to perform mandatory or optional migration to use the same kinds of extensions in Version 5.1.

Version 5.1 supports the Java 2 SDK Version 1.4.1. There are several implications for migration and coexistence. The V5.1 Network Deployment cell is a mixed node environment, where some Application Server nodes might be V5.0.x running Java 2 SDK 1.3.1 and some nodes might be V5.1 nodes running Java 2 SDK 1.4.1.

Applications compiled on V5.0.x nodes can also run on V5.1 nodes. The IBM SDK 1.4.1 in V5.1 provides this backward compatibility.

There is also a cross compiler option for the IBM SDK 1.4.1 that supports compiling code against a bootstrap and extension classes of a previous IBM SDK version. By default, the javac compiler compiles against the bootstrap and extension classes on its platform: the IBM SDK 1.4.1 compiles for 1.4 by default. You can use the IBM SDK 1.4.1 to compile for IBM SDK 1.3.1 compatible output. For example, a developer can issue this command on a V5.1 node to compile against a 1.3 target:

```
/ javac -target 1.3 -bootclasspath jdk1.3\lib\classes.zip \ -extdirs "" OldCode.java
```

The compiled code runs on a 1.3 Java virtual machine (JVM). The -target 1.3 parameter generates class files that are compatible with 1.3 JVMs. If you do not specify the -target option, the -bootclasspath option and the -extdirs option, the compiled code runs on 1.4 JVMs only. You must specifically compile for 1.3 JVMs on a V5.1 node, to avoid compiling against a Java 2 Platform API that is not present on a 1.3 JVM. A 1.4 application fails at run time on a V5.0.x node.

Migrating APIs and specifications

5.1 + Migrating APIs and specifications means moving to the current Java component level and to other technologies that IBM WebSphere Application Server, Version 5.1 supports.

Migrating APIs and specifications also includes moving to the most contemporary open specification levels. If your existing applications currently support different specification levels than those supported by this product version, updating at least some aspects of the applications to comply with the new specifications is probable.

5.1 + From V3.5 to Version 5.1, main migration area concerns are with IBM extensions, uplifting Servlet/JSP to the appropriate J2EE 1.2 specification levels, and packaging as J2EE applications. In contrast, little migration is necessary to move from Version 4.0.1 to Version 5.1.

See “Specification level summary of WebSphere Application Server” on page 99 which shows the various releases of the WebSphere for z/OS family and the specification and functional differences between them.

The following table summarizes potential migration areas.

Functional area	Support in V3.5.SE or V4.0.1	Must migrate from V3.5.SE?	Must migrate from V4.0.1?	Must migrate from V5?	Details
Enterprise beans	EJB 1.1	Not applicable	No	No	Full support for EJB 1.1 is provided. See Migrating enterprise bean code to the supported specification.

Functional area	Support in V3.5.SE or V4.0.1	Must migrate from V3.5.SE?	Must migrate from V4.0.1?	Must migrate from V5?	Details
Java 2 Connectors	Java 2 Connectors	Not applicable	Yes	No	The preliminary Java 2 Connector support in V4.0.1 is completed in V5.1. Some changes might be necessary to take full advantage of this support. See J2EE Connector Architecture migration tips.
JDBC API	JDBC API	Yes	Not applicable	No	Many applications can run unchanged in V5.1 although some changes might be required or recommended.
JSP files	JSP 1.0 Specification	No	No	No	JSP 1.0 APIs are a pure subset of JSP 1.2. See Developing JavaServer Pages files with WebSphere extensions.
	JSP 1.1 Specification	No	No	No	JSP 1.1 APIs are a pure subset of JSP 1.2. See Developing JavaServer Pages files with WebSphere extensions.
Security	IBM security	Yes	Yes	No	Changes might be required due to J2EE security. See Migrating security configurations from WebSphere Application Server Version 4.0.1.
Servlets	Servlet 2.1 Specification and IBM extensions	Yes	Yes	No	Many Servlet 2.1 applications can run unchanged in V5.1 although changes might be required or recommended. See Developing servlets with WebSphere Application Server extensions.
Servlets	Servlet 2.2 Specification	No	No	No	Servlet 2.2 APIs are a pure subset of Servlet 2.3. See Developing servlets with WebSphere Application Server extensions.
Sessions	IBM sessions	Yes	Yes	No	Many applications can run unchanged in V5.1 although changes might be required or recommended. See Developing session management in servlets and Migrating HTTP sessions.
Transactions	IBM transactions	No	No	No	V3.5SE did not support application initiated transactions through JTA APIs. See Using the transaction service.
Web services	Apache SOAP 2.2	Not applicable	Yes	No	Many applications can run unchanged although changes to use new support are recommended.
XML parser	XML 2.0.x supported	Yes	Not applicable	Not applicable	Changes to move to the supported API XML4J V4.2.2 level are required. See “Migrating XML applications” on page 103.
XML parser	XML4J V3.1	Not applicable	Yes	Not applicable	Recompilation is required to convert to XML4J V4.2.2.
Administrative console	Administrative console	Not applicable	Yes	No	Use the JMX support provided by wsadmin.

Migrating and coexisting

This topic describes migrating, which is copying the configuration from a previous release of a WebSphere Application Server product into a new release. This topic also describes coexisting, which is running a new release of a WebSphere Application Server product on the same machine at the same time as you run an earlier release.

Determine if an existing version of WebSphere Application Server is installed on the machine where you plan to install your Version 5.1 product.

If you have a previous version, you must plan whether to migrate the configuration and applications of the previous version to the new version. Migration does not uninstall the previous version. The earlier release is still functional. If you run the earlier release at the same time as the WebSphere Application Server for z/OS V5 installation, the two versions are *coexisting*. The migration articles describe migration from WebSphere Application Server for z/OS V3.5SE, WebSphere Application Server for z/OS V4.0.1, and WebSphere Application Server for z/OS 5.0 levels and coexisting with these prior levels. Unless otherwise noted, information pertaining to V3.5SE migration is valid for customers moving from the V4.0.1 Alternate Configuration Option. See “Migration and coexistence overview” for more information.

You can coexist with, or migrate the applications and configuration from a previous version of WebSphere Application Server.

Migration from V5.0 to V5.1 should be minimal; new migration utilities have been provided to facilitate migration of configurations as well as applications (see “V5.1 Migration utility” on page 79). Migration from V4.0.1 to V5 does not require extensive tuning. Migration from V3.5SE to V5 does require that you examine the migrating applications. After planning for migration and coexistence, continue the installation.

For more information on coexistence among releases, see “Migration and coexistence overview.”

To migrate from V3.5SE to V5, see “Migrating from V3.5 Standard Edition (SE) to V5.1” on page 84.

To migrate from V4.0.1 to V5, see “Migrating from V4.0.1” on page 90.

5.1 + To migrate from V5 to V5.1, see “Migrating from V5.0” on page 98.

Migration and coexistence overview

5.1 + WebSphere Application Server for z/OS supports the migration paths to Version 5.1 from the following releases:

- V3.5
- V4.0.1
- V5.0

Each path is slightly different in the form of tools available and applicable migration scenarios. In addition to migration it is possible for the various version of the Application Server mentioned above to coexist with V5.1.

V3.5: The migration from V3.5 to V5 involves significant changes in application structures, development, and deployment. This migration is a manual effort.

If you have V3.5SE installed, see “Migrating from V3.5 Standard Edition (SE) to V5.1” on page 84 for more information.

V4.0.1: WebSphere Application Server provides support for migrating V4.0.1 applications to V5. This utility is available as V4.0.1 service (W401502) against 390fy [Direct Deployment Tool] and provides a new -v5mp option.

The migration tool performs a fairly routine migration from V4 to V5. For example, Java 2 Platform, Enterprise Edition (J2EE) 1.2 EAR files in V4 work in V5 of WebSphere Application Server, which also supports the J2EE 1.3 specification. Similarly, it is not necessary to redeploy EJB 1.1 JAR files when moving them from V4 to V5, which also supports EJB 2.0 JAR files. V5 server configurations must be defined manually. If you have V4.0.1 installed, see "Migrating from V4.0.1" on page 90 for more information.

5.1+ **V5.0:**A new set of migration tools has been made available in V5.1. To learn more about the migration tools and how they interact with the migration tools available for other platforms, see "Migration tools" on page 78. For detailed information on the new migration tools particular to z/OS, see "V5.1 Migration utility" on page 79.

Coexistence: The remaining migration articles assume that V5 is being installed in an environment where it must coexist with prior levels of WebSphere Application Server. Consider the following items while planning to enable coexistence:

- Update prerequisites to the levels required by V5. Prior levels of WebSphere Application Server for z/OS will generally continue to run at the higher prerequisite levels. For more information, see "Prerequisites needed for WebSphere Application Server for z/OS V3.5SE, V4.0.1, V5, and V5.1" on page 101.
- Set up V5 to eliminate potential LPA conflicts with a prior V4.0.1 install. V5 and V4.0.1 both require the placement of some code into LPA (SBBOLPA). In addition it is recommended that additional product code (SBBOLOAD) be placed into LPA for performance reasons. Because of naming conflicts, V5 and V4.0.1 product code can not be in LPA at the same time. To support coexistence:
  - Place the V5 SBBOLPA dataset in the STEPLIB of the V5 daemon
  - If a prior level of SBBOLOAD is in LPA, add a V5.1 STEPLIB for SBBOLOAD.

  In V5, put BBORTSS5 into LPA so ctrace will work. This does not cause any coexistence issues since the dll name is different than the V4.0.1 version.
- Only one set of PPT entries can be active at one time for a given program. WebSphere Application Server for z/OS Versions 4.0.1 and 5.0 both use the BBOCTL program as a control region. If you are running Version 4.0.1 and Version 5.0 on the same system, their BBOCTL programs share the same PPT entry. Prior to Version 4.0.1, service level W401610, the PPT attribute SYST was required for the BBOCTL program; it is not required after that service level. Thus, including the SYST keyword in the PPT entry for the Version 5.0 BBOCTL program will cause an informational message (IEF188I PROBLEM PROGRAM ATTRIBUTES ASSIGNED) to appear when starting a Version 5.0 server (Version 5.0 does not require the SYST attribute). This message does not affect the functionality of the WebSphere address spaces. If you do not want this message to appear, and your Version 4.0.1 service level is at least W401610, you can delete SYST from the SCHEDxx member to stop the message from being generated. If you are not at this service level, you must leave SYST in the PPT for BBOCTL in order to start the Version 4.0.1 server. IEF188I will be issued upon starting the Version 5.0 server as long as BBOCTL is defined as a system task.
- Utilize WLM DAE when configuring V5 to allow both the use of a specific server name by V5 and by a server on V4.0.1.
- Review the ports that have been defined to ensure that the V5 installation does not conflict. In particular, when installing to coexist with V4.01, note that the default daemon port definition for both V5.0 and V4.0.1 is the same. For more information, see "Coexistence port definitions" on page 103. See z/OS port assignments for more information on ports.

Consider these other issues in a migration or coexistence scenario:

- Conflicting context roots when attempting to share the same Web server.

  A procedure for configuring a Web server for sharing between WebSphere Application Server versions is described in the information center for the base WebSphere Application Server product.

- Installing WebSphere Application Server multiple times on the same z/OS image or sysplex.

Migrating configuration data

You can migrate administrative configurations with the migration tools, as this task describes.

If you use an earlier version of WebSphere Application Server, the system administrator might have fine-tuned various application and server settings for your environment. It is important to have a strategy for migrating these settings with maximum efficiency and minimal loss.

Before using the migration tools, consult the V5.1 Release Notes document, as well as the PSP bucket, to understand what fixes you must apply to earlier versions. Applying fixes to an earlier version might also apply fixes to files that have a role in the migration. Apply any fixes to ensure the most effective migration of configurations and applications possible.

The migration tools in V5.1 support migration from WebSphere Application Server V5.

IBM provides a set of migration tools for migrating administrative configurations from V5 to V5.1.

The overall migration process when you issue the commands to use the migration tools is:

1. Save the current configuration and necessary files with the migration tool.
2. Install the V5.1 product.
3. Restore the configuration from the earlier release with the migration tool.

See “V5.1 Migration utility” on page 79 for a detailed description of the V5.1 migration tools.

You can use the migration tools to migrate from one version of WebSphere Application Server to another.

Migration tools

This topic introduces the migration tools that WebSphere Application Server provides. All migration tools are shipped with the V5.1 product as shell scripts and JCL, which can be found in the <WAS_HOME/INSTALL_ROOT>/bin directory, the default is WebSphere/V5R1M0/AppServer/bin, after installation. The JCL defined below is the primary migration tool, but works in conjunction with the migration shell scripts, which are called as part of the JCL. Additionally, the clientUpgrade tool is shipped as part of the WebSphere Client Development Kit and is located under the <INSTALL_ROOT>/bin directory on the Windows-based client machine.

clientUpgrade.sh (and clientUpgrade.bat)
Upgrades the client application to a new release level.

BBOXMIG1
One of two JCL batch jobs. This particular job prepares the Version 5 environment for migration by modifying the application servers in the node to recovery mode.

BBOXMIG2
Two of two JCL batch jobs. This particular job performs the Version 5 post processing. The job is executed against the new Version 5.1 application servers in the node being migrated.

The clientUpgrade command:

Use the **clientUpgrade** command to migrate application clients.

Note: WebSphere Application Server Version 5.1 does not support the Windows NT platform.

The command file is located in the `bin` subdirectory of the *`WAS_install_root`*, or the *`ND_install_root`* directory. By default, the *WAS_install_root* for WebSphere Application Server and WebSphere Application Server Enterprise is:

- Windows NT or Windows 2000 operating platforms - *`drive`*`\WebSphere\AppServer` directory

- AIX or UNIX-based operating platforms - /usr/WebSphere/AppServer directory

By default, the *ND_install_root* for WebSphere Application Server Network Deployment is:

- Windows NT or Windows 2000 operating platforms - *drive*\WebSphere\DeploymentManager directory
- AIX or UNIX-based operating platforms - /usr/WebSphere/DeploymentManager directory

Linux and UNIX-based platforms:

```
clientUpgrade.sh EAR_file [-clientJar client_jar ]
                       [-traceString trace_spec [-traceFile file_name ]]
```

Windows platforms:

```
clientUpgrade.bat EAR_file [-clientJar client_jar ]
                       [-traceString trace_spec [-traceFile file_name ]]
```

Parameters

Supported arguments include the following:

EAR_file
Use this parameter to specify the fully qualified path to the EAR file that contains client JAR files to process.

-clientJar
Use this optional parameter to specify a JAR file for processing. If not specified, the program transforms all client JAR files in the EAR file.

-traceString -traceFile
Use these optional parameters to gather trace information for IBM Service personnel. Specify a trace_spec of "*=all=enabled" (with quotation marks) to gather all trace information.

-logFileLocation
Use this optional parameter to specify an alternate location to store the log output.

The following example demonstrates correct syntax:

```
clientUpgrade EAR_file -clientJar ejbJarFile
```

V5.1 Migration utility:

The V5.1 migration utility consists of two new JCL batch jobs that assist with the migration from V5.0 to V5.1. This model takes advantage of staging, rather than an abrupt migration, by creating side by side configurations that transfer applications and configurations from a V5.0 environment to the new V5.1 environment.

Preparing for migration to V5.1

Before migrating from V5.0 to V5.1, you need to:

- Gather the following information: the sysplex name, system name, cell names, node names, and server names used for your V5.0 configuration.
- Mount the V5.1 SMP/E image.
- Mount a suitable JVM.
- Prepare and mount a new HFS to hold the new V5.1 configuration root.

Also, if you saved the customization variables from your V5.0 installation, they will be useful in migrating to V5.1.

Configurations that mix V5.0 and V5.1 nodes are generally supported. However, you cannot mix V5.0 and V5.1 nodes in the same cell *on the same LPAR*. Therefore, if you have multiple V5.0 nodes in the same cell on the same LPAR, you must migrate all V5.0 nodes in the same cell *on that LPAR* to V5.1 at the same time.

Customizing the migration JCL

Two JCL jobs control the migration process. These jobs are in the **SBBOJCL** dataset under the names **BBOXMIG1** and **BBOXMIG2**. **BBOXMIG1** and **BBOXMIG2** should be submitted under the WebSphere Administrator ID. Copy these files to your own dataset and customize them as described below to work with your system. Further references to **BBOXMIG1** and **BBOXMIG2** in this article refer to your customized versions of these files, not the two templates shipped with WebSphere Application Server for z/OS V5.1.

Both **BBOXMIG1** and **BBOXMIG2** contain a set of symbolic parameters. These parameters need to be updated to match your configuration in the following ways:

- The DLGDSN and DLGVOL symbolics should point to the dataset and volume, respectively, where your customization dialog output can be found.
- The V50HFS symbolic should point to the HFS of the V5.0 node you are going to migrate
- **BBOXMIG1** contains an additional symbolic, MODE, which should be toggled as needed. When set to '0', the job will alter the V5.0 configuration to force PRR mode. When set to '1', the job will restore a V5.0 configuration that was previously altered.

Performing the migration

The migration process is relatively simple once everything has been prepared. The first step is to clear the transaction log in your V5.0 system:

1. Stop your V5.0 servers.
2. Submit **BBOXMIG1** with the MODE parameter set to '1'.
3. Start your V5.0 servers.
4. After starting, each server will stop automatically after flushing its transaction log.
5. Submit **BBOXMIG1** with the MODE parameter set to '0'.

Next, migrate each node, one server at a time, starting with the Deployment Manager node. The Deployment Manager must be the first server migrated in an ND configuration. Follow these steps:

1. Edit **BBOXMIG2**, changing the V50HFS symbolic to select the node to be migrated
2. Using the Dialog walkthrough below, configure your first server.
3. After you have generated the jobs for your server, submit **BBOXMIG2**.
4. If the Deployment Manager was migrated, you must start it now, before any other servers are migrated.
5. Repeat steps 2 through 4 until each server within the node has been migrated.
6. If you still have nodes remaining to migrate, return to step 1.

Dialog walkthrough

This section contains a walkthrough for using the customization dialog. The path through the dialog described below is the shortest path the user can take. Default values may be accepted for all settings except those which are explicitly named below.

Option 1: Configure a security domain

1. Load default values from the suggested dataset.
2. Define variables:

- The SYSPLEX NAME must be updated to match the SYSPLEX name of your existing installation.

3. Save security configuration.

Option 2: Configure a base application server node

1. Load security variables that were saved in the previous panel.
2. Allocate a target dataset.
3. Define variables:
   a. System locations:
      - The SYSTEM NAME must be updated to match the SYSTEM name of your existing installation.
      - The location of the HFS components must be updated; it is a good idea to choose pathnames that will not interfere with filesystems that may already by mounted for your existing V5.0 configuration.
      - The SMP/E HOME must point to where the V5.1 SMP/E HFS will be (or currently is) mounted.
      - The JMS CLIENT SMP/E HOME must point to where the V5.1 JMS Client SMP/E HFS will be (or currently is) mounted.
      - The JAVA HOME must point to where the 1.4 JRE you intend to use with your V5.1 configuration will be (or currently is) mounted.

      Note: You will not have to mount these filesystems yourself; if they are not already mounted, the migration jobs will mount them as needed.

   b. System environment:
      - The MOUNT POINT must be updated to point the where your V5.1 configuration HFS will be (or currently is) mounted.

   c. Server Customization:
      1) Accept defaults.
      2) Accept defaults again.
      3) Define variables:
         - The CELL NAME (short) and CELL NAME (long) should match those used in your V5.0 configuration.
         - The NODE NAME (short) and NODE NAME (long) should match those used in your V5.0 configuration.
         - The SERVER NAME (short) and SERVER NAME (long) should match those used in your V5.0 configuration.
         - The NODE HOST NAME requires a dummy value to satisfy the dialog; use N.N.N.N.
         - The IP NAME requires a dummy value to satisfy the dialog; use M.M.M.M.
      4) Generate customization jobs.
4. Save security configuration.

Option 4: Configure a Deployment Manager node (this option will only be needed when migrating Network Deployment systems.)

1. Load security variables that were saved in the previous panel.
2. Allocate a target dataset.
3. Define variables:
   a. System locations:
      - The SYSTEM NAME must be updated to match the SYSTEM name of your existing installation.
      - The location of the HFS components must be updated; it is a good idea to choose pathnames that will not interfere with filesystems that may already by mounted for your existing V5.0 configuration.

- The SMP/E HOME must point to where the V5.1 SMP/E HFS will be (or currently is) mounted.
- The JMS CLIENT SMP/E HOME must point to where the V5.1 JMS Client SMP/E HFS will be (or currently is) mounted.
- The JAVA HOME must point to where the 1.4 JRE you intend to use with your V5.1 configuration will be (or currently is) mounted.

Note: You will not have to mount these filesystems yourself; if they are not already mounted, the migration jobs will mount them as needed.

b. System environment:

- The MOUNT POINT must be updated to point the where your V5.1 configuration HFS will be (or currently is) mounted.

c. Server Customization:

1) Accept defaults.
2) Accept defaults again.
3) Define variables:
   - The CELL NAME (short) and CELL NAME (long) should match those used in your V5.0 configuration.
   - The NODE NAME (short) and NODE NAME (long) should match those used in your V5.0 configuration.
   - The SERVER NAME (short) and SERVER NAME (long) should match those used in your V5.0 configuration.
   - The NODE HOST NAME requires a dummy value to satisfy the dialog; use N.N.N.N.
   - The IP NAME requires a dummy value to satisfy the dialog; use M.M.M.M.
4) Generate customization jobs.

Migrating V5.0 JCL procedures to V5.1

If you are planning to use the same set of procedures in V5.1 that you are using to start your V5.0 configuration, then you will need to make some adjustments to ensure that your V5.1 configuration is started instead. The JCL procedures whose names *do not* end with the character ’Z’ contain a symbolic parameter named ROOT. If you are editing your existing JCL, then this parameter should be updated to point to your V5.1 configuration root.

For example, you might have a procedure from your V5.0 installation that contains a symbolic, such as this:

```
//  SET ROOT='/WebSphere/V5R0M0'
```

You would change this symbolic to point to your new configuration root (which, in this example, is /zWebSphere/V5R1M0) like this:

```
//  SET ROOT='/zWebSphere/V5R1M0'
```

Coexisting V5.0 and V5.1 procedures

Two sets of JCL procedures can coexist. If you are planning to use V5.0 anywhere in your sysplex after migration, then you must use two sets of JCL procedures, one for V5.0 and one for V5.1. If you are planning to do this, then you must use a different set of names for your V5.1 procedures. When you use different names for your V5.1 procedures, you must uncomment and edit the step in **BBOXMIG2** before you begin to migrate. This will update your V5.1 configuration to use your new procedure names.

To generate a set of V5.1 procedures, you can either make copies of the V5.0 procedures you are using now, or you can make copies of the procedures generated by the dialog. In either case, you must be sure that the symbolic ROOT parameter in the V5.1 procedures is pointing to your V5.1 configuration root, as discussed above.

Both sets of procedures should be modified to use STEPLIB statements to point to the modules rather than relying on the contents of the LPA. The JCL procedures that *do* have names that end with the character 'Z' contain some common environmental settings that are referenced by the ENV parameter of other JCL procedures. These environment files should be edited, and the STEPLIB parameters should be updated.

For example, given an installation where the V5.1 WebSphere Application Server for z/OS modules are in the IBM.WS51.* datasets, the environment files would be updated to include the following STEPLIB statement:

```
//STEPLIB  DD DSNAME=<HLQ>.SBBOLPA,DISP=SHR
//         DD DSNAME=<HLQ>.SBBOLOAD,DISP=SHR
//         DD DSNAME=<HLQ>.WS51.SBBOLD2,DISP=SHR
```

Post-migration issues

Only one version of WebSphere Application Server for z/OS modules can be loaded into the LPA. You must choose which, if any, version of WebSphere Application Server for z/OS modules will be loaded from the LPA. Any other version of WebSphere Application Server for z/OS that runs on the same LPAR must be configured using STEPLIB DD statements in the JCL procedures that are used to run the servers running that version of WebSphere Application Server for z/OS.

STEPLIBed datasets that contain the V5.1 WebSphere Application Server for z/OS modules should be APF-authorized. If they are not, you will encounter security errors when attempting to start your V5.1 server.

If you customized a `variables.xml` file in your V5.0 installation, you will need to inspect your migrated V5.1 configuration, reapplying these customizations as necessary. The following variables in particular are set appropriately by V5.1, and will need to be reset with your customizations as needed:

- WAS_SERVER_ONLY_default_jvm_options
- WAS_SERVER_ONLY_control_region_classpath
- WAS_SERVER_ONLY_server_region_classpath
- WAS_SERVER_ONLY_control_region_libpath
- WAS_SERVER_ONLY_server_region_libpath

Coexistence support

Coexistence, as it applies to WebSphere Application Server products, is the ability of multiple installations of WebSphere Application Server to run on the same z/OS image or sysplex at the same time. Multiple installations include multiple versions and multiple instances of one version. Coexistence also implies various combinations of Web server interaction.

Version 5.x WebSphere Application Server products can coexist with the following supported versions:

- IBM WebSphere Application Server for z/OS Version 3.5SE
- IBM WebSphere Application Server for z/OS V4.0.1
- IBM WebSphere Application Server for z/OS V5

In addition to coexistence with earlier versions of WebSphere Application Server for z/OS, WebSphere Application Server for z/OS also lets you install multiple times on one z/OS image or sysplex (multiple installation instances), or install once and have multiple configurations (multiple configuration instances).

Multiple Version 5 instances on one machine include:

- Multiple base Application Server instances from multiple installations of the Network Deployment product
- Multiple deployment manager instances from multiple installations of the Network Deployment product

Migrating from V3.5 Standard Edition (SE) to V5.1

Determine whether you have WebSphere Application Server for OS/390 V3.5 Standard Edition installed on the system where you plan to install your Version 5.1 product. If you are interested in utilizing the Simple Configuration Option for V4.0.1, then the content in this article will apply to you.

To migrate V3.5 Standard Edition (hereafter referred to as V3.5SE) applications and configurations to Version 5.1, follow this procedure:

1. Prepare to migrate or update product prerequisites and corequisites to supported versions. Significant functional dependencies on the z/OS infrastructure have been added to the new product over those that had been required for V3.5SE. See “Prerequisites needed for WebSphere Application Server for z/OS V3.5SE, V4.0.1, V5, and V5.1” on page 101 comparing V3.5SE, V4.0.1, V5, and V5.1 prerequisites. Review the PSP bucket for the latest PTF information. There is no dependency on a particular V3.5SE service level to allow coexistence with V5.1.

   Unlike V3.5SE, V5 and higher for the z/OS platform allows distributed platform Web servers for distributed platforms to communicate with the WebSphere Application Server for z/OS product. Refer to IBM WebSphere Application Server supported hardware, software, and APIs for supported distributed platform Web servers.
2. Install the new Application Server product.

   Note: V5, and higher, requires the use of SMP/E 3.1. Installation instructions have been provided with your ServerPac or PDO order.
3. Customize the product by utilizing the customization dialog, which is provided, and verify success by running the Installation Verification Test. The customization dialog will prompt you for various TCP/IP ports that the Application Server should listen on for requests. You should ensure that you configure the new application server not to make use of the same ports as an existing server on that same image.

   Note: V5, and higher, does not conflict with the mount points or directory structure of the existing V3.5SE release.
4. Run the samples provided with the new product. There have been significant changes to Admin, system structure, etc. and for this reason, it is recommended to run the samples. See Samples Gallery for more information.
5. Define the new servers for applications being migrated.

   There is no tooling support for migrating Server configurations because of the massive differences in application packaging/deployment, server structure/architecture, application administration compared to V3.5SE. This process will need to be done manually. Some key differences that should be noted:
   - The servlet engine provided by V3.5SE has been replaced by a full J2EE 1.4.1 WebContainer.
   - The WAS.CONF file has been eliminated. All configuration that was previously done through the WAS.CONF is now done through the J2EE tooling (WSAD, Application Assembly Tool, and the adminstrative console). See “Migrating your existing was.conf file settings” on page 87 for more information.
   - The configuration is no longer restricted to running as a GWAPI out of the IBM HTTP Server for z/OS and OS/390. V5, and higher includes HTTP Transport function as part of its server structure. A default internal transport is provided, but a specific HTTP transport can also be configured using the adminstrative console. HTTP listening ports must be selected so as to not conflict with existing V3.5SE configuration. Because of the HTTP Transport function, customers now have three configuration options:
     a. Allow direct HTTP access to the J2EE server via the HTTP transport

   b. Utilize IBM HTTP Server V5.3 for z/OS and OS/390 with the WebSphere HTTP Plug-in for z/OS to front end the HTTP transport. This option requires new ServerInit, Service, and ServerTerm directives to be added to the HTTP server `httpd.conf` file. See "The WebSphere Application Server for z/OS HTTP internal transport" on page 7 for more information.
   c. Utilize any V5 and higher supported distributed platform Web server, along with the appropriate WebSphere plug-in for Web servers, to front end the HTTP transport.
- When establishing Server configuration, care should be taken to establish a naming convention that lends itself to changes and extensions.
- Create WLM and ARM policies for the Servers as part of the server configuration.

6. Install resources into the new server.

   Note: Prerequisites levels needed for WebSphere Application Server for z/OS may have changed from V3.5SE. Review these prerequisites at "Prerequisites needed for WebSphere Application Server for z/OS V3.5SE, V4.0.1, V5, and V5.1" on page 101.

7. Create new DB2 tables if your application uses HTTP Session State Persistence. The format of the DB2 tables needed to support HTTP Session State persistence has changed in V5. Consequently HTTP sessions can not be shared between V3.5SE and V5 and higher. If your application uses HTTP Session State Persistence using DB2, you must create new tables. See Creating a DB2 table for session persistence for more information.
8. Review the existing security configuration for Web applications. The authentication and authorization policy for URLs within J2EE Web applications has the ability to be fully described by the web.xml deployment descriptor. The web.xml file is contained in the Web application archive (.war) file. The war file is deployed into the WebSphere Application Server for z/OS J2EE Server. The enforcement of the rules at runtime is carried out by the Web container in the J2EE Server in conjunction with the configured user registry, such as SAF or LDAP. Implementing a security policy for the installation consists of the deployer granting permission for users and groups to be in various ROLEs that are described in the deployment descriptor of the application. At runtime, the Web container will authenticate the user using the means in the deployment descriptor and will subsequently insure that either the authenticated user, or a group in which he is a member, has been granted access to the necessary J2EE Roles by the deployer. For more information on J2EE Security, please see Migrating security configurations from WebSphere Application Server Version 4.0.1.

   The V3.5SE server executes as a GWAPI plug-in within the address space of an IBM HTTP Server for z/OS. Security policy for URLs (including those that get serviced by the WASv3.5SE plugin) that are served from the Web server are configured using the security directives in the httpd.conf file of that Web server. At runtime, the Web server is able to authenticate the user to the configured security server such as RACF or LDAP and determine if the user or a group that the user is a member of has been granted access to the specified URL before allowing access.

   In order to migrate Web applications and their security policy from V3.5SE, as well as execute the server independent of the existing Web server, you need to insure that the deployment descriptor for the Web applications being migrated contains security constraints that match what currently exists within the httpd.conf. It is recommended to make use of the deployment descriptors within the Web application as a means for describing the security policy of the application.

   V3.5SE does provide extended capability to supplement the Web server authorization processing by performing SAF checks against resources profiles that are defined in the SOMDOBJS facility class. If your existing V3.5SE servers use this capability, you will need to insure that you understand the policy and configure the equivalent within the deployment descriptor.
9. Move applications to the new product level tooling. WSAD 4.0 and WSAD IE 4.1.1 generated applications are not supported for deployment into V5 and higher runtimes. This necessitates movement of the application to V5 and higher level tooling. You can also find information about migrating to the V5 tooling levels in the IBM Redbook, Migrating to WebSphere V5: An End-to-End Migration Guide, SG24-6910-00. There is a link to the book in the Resources for Learning topic.
10. Review applications to be migrated to determine if any changes are required. See Migrating Web application components for more information. Some of the areas to review are:

- **Update application to supported specification level.** V3.5SE provided a compatibility option allowing the use of Servlet 2.1/JSP 1.0 levels. Servlets/JSPs must be brought up to at least the minimal level that is supported by J2EE 1.4.1 (Servlet 2.2/JSP 1.1). The MigrateWC tool has been provided to assist in conversion of JSPs and Servlets coded at specification levels of previous WebSphere versions. The MigrateWC tool can be found on the WebSphere Developer Domain downloadable tools Web site at http://www.ibm.com/software/.
- **Repackage application as a J2EE application.** V3.5SE did not require J2EE packaging. In V5 and higher, Web applications must be packaged as WAR/EAR files for deployment into a V5, and higher, webcontainer. J2EE packaging includes the declaration of application properties through deployment descriptors that are contained in a separate xml file. WSAD/WSAD IE tooling provides this packaging capability.
- **Migrate Common Connector Framework (CCF) into a J2C connector application, if applicable.** The CCF function is a deprecated function, but is still supported in the V5 and V5.1 runtime only. The intent is to support users in a migration to J2C. The WebSphere Developer Domain has a series of 3 part articles that address the tooling, the Application Server, and the migration path from CCF to J2C, entitled CCF to J2C migration.
- **Review classloader options.** V3.5SE provided a single class loader and it was highly recommended that a web application only reference entities contained within the web application. Review the changes in the classloader options compared to V3.5SE and change as appropriate. See “Class loader options” on page 88 for more information.
- **Review servlet for change related to content type setting.** A change was made in the web container behavior (required by J2EE) that involves the setting of content type. If a servlet writer does not set the content type it is no longer defaulted by the web container and it is returned as ″null″. This can cause some browsers to incorrectly display the resulting tags sent by the web container. This problem can be eliminated by either modifying the servlet to set the content type, or by setting the autoResponseEncoding IBM extension to *true* for web modules.
- **Indicate external resources.** Web applications that use JDBC must include a deployment descriptor indicating that JDBC is an external resource that needs to be accessed.
- **Modify JNDI calls.** The com.ibm.ejs.ns.jndi.CNInitialContext Factory class can not be used for a V5, or higher, Web application to obtain an initial context factory for gaining access to the JNDI name space. Change the application to obtain JNDI access through the J2EE defined mechanism.
- **Remove any usage of Connection Manager APIs.** These APIs were deprecated in V3.5SE and are not supported.
- **Review application resource authentication for possible changes.**
  - Existing application components that specify a user ID and password on the getConnection method must be reassembled with a resource authentication property of *Application* to maintain the current behavior of the application. Otherwise, the J2EE server ignores the user ID passed on the method.
  - Application code dependent on the ACEE being placed on the thread of execution will need to change. An example of this is the CTG best practice recommendation to not provide a password/ID on the getConnection call for applications running in V3.5SE. This type of application must be changed.
- **Convert to J2EE Connector Architecture.** Customers should recode, reassemble, and reinstall existing applications to work with WebSphere for z/OS-supported connectors that are designed to implement the J2EE Connector Architecture.
- **Make code page related changes.** The JVM default code page has changed from EBCDIC to ASCII to be consistent with the WebSphere Application Server family. Application changes may be required. See “Code page changes” on page 89 for more information.

11. Install the application into the new server using the adminstrative console tool.
12. Choose your Web server. Existing V3.5SE configurations include a V3.5 SE plug-in running as a GWAPI out of the IBM HTTP Server for z/OS and OS/390. V5, or higher, does not require the use of the IBM HTTP Server for z/OS and OS/390. You can choose to either:
    - Use a distributed platform Web server, in which a V5, or higher WebSphere plug-in for Web servers has been installed, or
    - Run without a Web server fronting the HTTP transport provided as part of the new J2EE server

Either of these alternatives can coexist with the existing V3.5SE configuration. Alternatively, you may choose to continue to use an IBM HTTP Server for z/OS and OS/390, in which the V5 WebSphere HTTP Plug-in for z/OS has been installed. In this case, the IBM HTTP Server for z/OS and OS/390 you are using must be at a V5.3 level.

With this in mind, several migration configurations are possible:

- Install the WebSphere HTTP Plug-in for z/OS into the existing IBM HTTP Server V5.3 for z/OS and OS/390 (V3.5SE plug-in and V5 and higher WebSphere HTTP Plug-in for z/OS can coexist).
- Install the WebSphere HTTP Plug-in for z/OS into a separate IBM HTTP Server V5.3 for z/OS and OS/390.

13. Make necessary operational changes to support production.
    - Scripting capability that can be used for automation.
    - ARM policies should be updated based on the new topology/infrastructure.

    See WebSphere Application Server for z/OS V5 Operations and Administration for more information.

Migration from Version 4.x to Version 5 or higher, does not require extensive tuning.

Migration from Version 3.5.x to Version 5 or higher does require you to examine the migrating applications.

After planning for migration and coexistence, you are ready to continue the installation.

Migrating your existing was.conf file settings

The following table describes how to migrate your was.conf configuration file settings to your WebSphere Application Server V5 environment.

was.conf file setting	How to set in V5
appserver.compliance.mode	Not applicable.
appserver.version	Not applicable.
appserver.usesystemclasspath	Defined using administrative console J2EE server settings.
appserver.libpath	Defined using administrative console shared library settings.
appserver.classpath	Defined using administrative console runtime and application settings.
appserver.name	Defined using administrative console when application server is created.
appserver.jvmpropertiesfile	Not applicable. JVM properties are defined using the administrative console.
appserver.loglevel	Defined using administrative console trace and error control settings.
appserver.logdirectory	Not applicable. The name of the log directory is pre-defined.
appserver.jspbasehrefadd	Defined at the application level.
appserver.workingdirectory	Not applicable.
appserver.permisssions	Not applicable.
appserver.nodetach	Not applicable.
appserver.java.system.property	Not applicable.
appserver.java.extraparm	Not applicable.
appserver.configviewer	Not applicable.
appserver.initializeonwebappfailure	Not applicable.

session.enable	Not applicable.
session.invalidationtime	Defined using administrative console session management settings.
session.tableoverflowenable	Defined using administrative console session management settings.
session.tablesize	Not applicable.
session.dbenable	Defined using administrative console session management settings.
session.dbjdbcpoolname	Defined using administrative console data source settings.
session.dbtablename	Defined using administrative console Web container custom properties settings.
session.urlrewriting.enable	Defined using administrative console session tracking settings.
session.cookies.enable	Defined using administrative console session tracking settings.
session.protocolswitching.enable	Defined using administrative console session tracking settings.
session.cookie.name	Defined using administrative console session management cookie settings.
session.cookie.comment	Not applicable.
session.cookie.maxage	Defined using administrative console session management cookie settings.
session.cookie.path	Defined using administrative console session management cookie settings.
session.cookie.secure	Not applicable.
session.cookie.domain	Defined using administrative console session management cookie settings.
Connection pool properties	Not applicable.
Virtual host properties	Defined using administrative console virtual host settings.
Application definition (webapp properties)	Defined during the application assembly process.
Application deployment (deployedwebapp properties)	Defined using administrative console application install settings.
Java Virtual Machine (JVM) definition	Defined using administrative console Java Virtual Machine settings.
objectleveltrace.enabled	Not applicable.
objectleveltrace.host	Not applicable.
objectleveltrace.port	Not applicable.
inline.comment	Not applicable.

Class loader options

WebSphere Application Server V3.5 Standard Edition for z/OS and OS/390 maintains an Application Server class loader that is global to the JVM. In addition each Web application has its own class loader.

The Application Server class loader serves as the parent to all of the Web application class loaders. The default search mode used when locating a class to load is ″Parent First″. In Parent First mode, the servlet engine first attempts to locate the specified class in the parent class loader or load it from one of the libraries or directories supported by that class loader. In the case where a Web application is being loaded, the servlet engine will first search the application level class loader for the target class. If not found, the servlet engine will then proceed to locate the class in the Web application specific class loader or from one of the libraries or directories specified in the deployedwebapp.*web_app_name* classpath directive for that Web application.

The application server classpath is used to specify classes that are used by multiple Web applications such as utility functions. The Web application contains the artifacts that are to be referenced by a single Web application. Web Applications do not have access to artifacts that are packaged with another Web application.

In V3.5, any text based files (properties or resource bundles) that are loaded from an open directory by the Application Level class loader are required to be encoded in EBCDIC. This anomoly no longer exists in V5. All property files and resource bundles are required to be encoded as US ASCII with no exception.

The technique for specifying a class loader policy is more robust in V5. Migrating the class loader Module Visibility Mode describes how to specify these policies in V5.

Code page changes

WebSphere Application Server for z/OS V5 is much closer to the other WebSphere Application Server platforms in how it treats code page issues. This improvement should allow greater portability of applications from other platforms or from the development environment. Encoding changes should no longer be required as part of deployment.

If you are migrating from WebSphere Application Server Standard Edition V3.5, you should be aware of the following potential migration issues:

1. V3.5 supports JSP .91, 1.0, and 1.1 specification levels. JSPs that were configured as .91 or 1.0 levels required the JSP source files to be stored in EBCDIC encoding. JSP specification level 1.1 clarifies the requirements for content of JSPs. Minimally they must be stored in ISO-88590-1 (US ASCII) encoding or a proper derivative. To successfully migrate your level .91, and 1.0 JSPs to V5.0, you must convert the source files to ASCII encoding.
2. V3.5 requires property files and resources bundles that were loaded from an open directory by the application server class loader to be encoded as EBCDIC. In V5, any resource bundles or property files that are used by your application that had previously been converted to EBCDIC encoding must be changed to ASCII before deployment.

   Note: Properties or resource bundles that were contained in .jar files or that were loaded from a Web application class loader were not required to be in EBCDIC. Therefore they should already be in ASCII format.
3. V3.5 does not override the default native file encoding of the JVM. This implies that the default file encoding used by the JVM in V3.5 is assumed to be CP1047 - EBCDIC.

   Even though the Java language allows programmers to explicitly specify the encoding of a file that is to be opened and read by the JVM, many programmers do not take advantage of this capability and rely on the default encoding. V5 should improves portability of applications by starting the JVM with a default file encoding of ISO-8859-1 (US ASCII). This implies that applications that rely on JVM behavior instead of explicit specification of encoding will be able to execute unchanged on a WebSphere Application Server for z/OS V5 system. If you have applications which access EBCDIC encoded files, you must ensure that they explicitly specify this encoding in the application code, or you must change the encoding of the target file to match this new JVM setting.

   Note: If you have applications which access EBCDIC encoded files, you need to insure that they are explicitly specifying this encoding in the application code or change the encoding of the target file to match the new JVM setting.

Migrating from V4.0.1

Determine whether you have an existing version of WebSphere Application Server installed on the machine where you plan to install your Version 5.1 product.

To migrate from V4.0.1 applications and configurations to Version 5.1, follow this procedure:

1. Prepare to migrate or update product prerequisites and corequisites to supported versions. If your existing configuration makes use of a non-390 Web server and associated AE plugin, you should review the required Web server levels and update as appropriate. Refer to IBM WebSphere Application Server supported hardware, software, and APIs for current requirements.

   Functional dependencies on the 390 infrastructure have changed. See "Prerequisites needed for WebSphere Application Server for z/OS V3.5SE, V4.0.1, V5, and V5.1" on page 101 comparing V3.5SE, V4.0.1, V5.0, and V5.1 prerequisites.

   V4.0.1 must be at minimum service level W401501 to coexistence with V5.1.
2. Review existing V4.0.1 environment for possible coexistence issues related to port definitions and placement of WebSphere Application Server modules in LPA. See "Migrating and coexisting" on page 76 for more information.
3. Install the V5.1 product, which requires the use of SMP/E 3.1. Installation instructions have been provided with your ServerPac or PDO order.
4. Customize V5.1 utilizing the customization dialog provided with the product and verify success by running the Installation Verification Test. Note that the V5 default */install root* directory on z/OS is WebSphere/V5R0M0/AppServer, and it does not conflict with mount points or directory structure of the existing V4.0.1 release.
5. Run the V5.1 samples. With the introduction of V5, significant changes to Admin, system structure, etc. were introduced. For this reason, it is always recommended the new samples are run to experience the environment. See Samples Gallery for more information.
6. Define the new servers for applications being migrated.

   V5.1 requires the use of the HTTP transport, and HTTP listening ports must be selected so as to not conflict with the existing V4.0.1 configuration.
   - There is no tooling support for migrating Server configurations because of the massive change in the Administration console support between V4.0.1 and V5.1. This process will need to be done manually. Use the V4.0.1 administrative console capability to to dump existing V4.0.1 settings for review. See "Utilizing WebSphere administrative capabilities to support migration from V4.0.1" on page 93 for more information.
   - Establish a naming convention that lends itself to changes and extensions when establishing Server configurations.
   - Create copies of policies such as WLM and ARM and modify them for the V5.1 Server configuration.
7. Install resources into the V5.1 server. Prerequisites levels needed for WebSphere Application Server for z/OS may have changed from V4.0.1. Review these prerequisites at "Prerequisites needed for WebSphere Application Server for z/OS V3.5SE, V4.0.1, V5, and V5.1" on page 101.
8. Move applications to V5.1 level tooling. Most V4.0.1 applications will continue to run in the new Servers. However, applications generated in WSAD 4.0 and WSAD IE 4.1.1 are not formally supported for deployment into V5.1 runtimes. Formal support necessitates movement of the application to V5.1 level tooling. In the case of WSAD 5.1, this tool has the flexibility to generate both J2EE 1.2 and J2EE 1.3 applications allowing convergence to a single tool to support V4.0.1 and V5.1. In the case of WSAD IE 5.1, there are conflicts in tooling jars with WSAD IE 4.1.1. You must maintain both tooling levels until you have migrated everything to the latest levels. You can also find information about migrating to the V5 tooling levels in the IBM Redbook, Migrating to WebSphere V5.0: An End-to-End Migration Guide, SG24-6910-00. There is a link to the book in the Resources for Learning topic.
9. Review applications to be migrated to determine if any changes are required. The Class API checker tool can be used to analyze servlets and ejbs to determine if any APIs are unsupported or

deprecated. The CACT tool can be found at WebSphere Developer Domain downloadable tools. If required changes are found, it will be necessary to modify them and then install into the V5 servers. If changes are not required, then the migration utility can be utilized. Some of the key things to note that may cause a need for application changes are:

- The default code page has changed to be consistent with the V5 family. In V5.0, and above the JVM is started with a default file encoding of iso-8859-1 (US ASCII). This enables applications that rely on JVM behavior instead of explicit specification of encoding to execute unchanged on a WebSphere Application Server for z/OS V5 system. This should improves portability of applications across platforms.

  Even though the Java language allows programmers to explicitly specify the encoding of a file that is to be opened and read by the JVM, many programmers do not take advantage of this capability and rely on the default encoding. If you have applications which access EBCDIC encoded files, you must ensure that they explicitly specify this encoding in the application code, or you must change the encoding of the target file to match this new JVM setting.
- JCA connector applications developed using WSAD IE 4.1.1 to create an atomic JCA connector Web service that communicates to a backend resource manager should still be able to run under V5.1. However, to do so, one must first redeploy the application to V5.1. The redeploy can be done during the installation of the application. Although, these applications should run under V5.1 after redeploying, this should be viewed as a temporary method of operation. Customers should still eventually transition their WSAD IE 4.1.1 JCA connector applications to the WSAD IE 5.1 tooling.
- There are cases where Web Applications may need to be modified. See Migrating Web application components for more information.

10. Utilize the migration utility that was shipped in W401502 to process V4.0.1 EARs. This utility is being provided as an option on 390fy. In V4.0.1 it was required that EARs be processed through the 390 AAT or 390fy to condition them for deployment into V4.0.1. In V5, and higher, this conditioning is not required. This utility will remove the conditioning that was done for existing V4.0.1 EARs that are being migrated. It should be noted that if the application is being sent back through the WSAD 5.0, or later version tooling, that the resulting EAR is ready for installation into V5.1 and that the use of migration utility is not required. Support for migrating V4.0.1 applications to V5.1 has been shipped in V4.0.1 service (W401502). This support has been provided as a new -v5mp option for the 390fy tool. See “Utilizing WebSphere administrative capabilities to support migration from V4.0.1” on page 93 for more information.
11. Address differences between V4.0.1 and V5.1 by use of the AAT.
    - Review the changes in the classloader options in V5.1 compared to V4.0.1. Determine the new option that is appropriate and utilize the AAT to make the change. See Migrating the class loader Module Visibility Mode for more information.
    - A change was made in the web container behavior that was required by J2EE that involves the setting of content type. If a servlet writer does not set the content type it is no longer defaulted by the web container and it is returned as ″null″. This can cause some browsers to incorrectly display the resulting tags sent by the web container. This problem can be eliminated by using the AAT to set the autoResponseEncoding IBM extension to true for web modules while migrating enterprise applications.
12. Install the application into the V5.1 server.

    For Web applications being migrated from V4.0.1 the following should be noted. In V4.0.1, the classes generated from JSPs are in a package based on the directory structure of the WAR.

    Any JSP at the top of the context root is in the unnamed package, and JSPs in subdirectories of the root are in packages named after the subdirectories.

    In V5 and higher, the classes generated from JSPs are all in the package **org.apache.jsp**. Therefore, the classfiles are not compatible between versions.

    When migrating an enterprise application from V4.0.1 to V5.1, the JSPs need to be recompiled so that the classfiles are regenerated into the correct packages. Use the *-preCompileJSPs* option of wsadmin tool during the installation of the application to force this recompilation.
13. Migrate clients for the applications that have been migrated to the V5.1 Server.

Update clients that reference EJBs.

The naming structure changed in V5. As a result the EJB references that were valid in previous versions no longer work in V5 or higher. Existing V4.0.1 clients will need to be changed appropriately.

Alternatively, a name binding can be added to map the old name into the new name in V5. If you pick this alternative, you can find some helpful examples in the IBM Redbook, Migrating to WebSphere V5.0: An End-to-End Migration Guide, SG24-6910-00. There is a link to the book in the Resources for Learning topic.

Update data resources for client enterprise applications.

The client container supported by V4.0.1 utilized a WebSphere Configuration API model that is different from the one used by V5 and higher. As a result, data resources will need to be converted to run correctly with V5 or later versions.

14. Update your Webserver to the V5.1 prerequisite level and the appropriate plug-in. See Chapter 2, ″Deprecated Version 5.1 Features″ to determine if you should plan migration paths for any of your functions that might be deprecated. Most V4.0.1 configurations will include a Web server. This can be a 390 Web server or any Web server supported by V4.0.1 AE. If a Web server configuration is required, then the Web server to be used must be updated to the V5.1 prerequisite level and the appropriate plug-in must be installed. It should be noted that the V5.1 plugin can support both V4.0.1 and V5.1 application servers. With this in mind, several migration configurations are possible:
    - Install the V5.1 plugin into the existing Webserver (the V4.0.1 plugin and V5.1 plugin coexist).
    - Create a unique Webserver instance and install the V5.1 plugin.
    - Replace the V4.0.1 plugin with the V5.1 plugin and setup to drive both the V4.0.1 and V5.1 images.

    See “Setting up Version 4.0.x and Version 5.x coexistence” for more information.
15. Migrate the existing V4.0.1 security configuration for the applications that have been migrated. See Migrating security configurations from WebSphere Application Server Version 4.0.1 for more information.
16. Make the necessary operational changes to support production.
    - Scripting based on SM API is not valid in Version 5, or later levels. It must be converted. See Migrating from Systems Management Scripting API (SMAPI) V4 to wsadmin V5 for more information.
    - ARM policies should be updated based on the new topology/infrastructure.

    See the Operations and Administration Guide for more information.
17. Post migration activities.
    - Some interfaces have been deprecated at the V5.1 level. Although they are still supported at V5.1, they should be eliminated to facilitate migration to later WebSphere Application Server levels. See Migrating security configurations from WebSphere Application Server Version 4.0.1 for some areas that should be changed.
    - Migrate applications to the latest specification levels.

Migration from Version 4.0.1 to Version 5.1 does not require extensive tuning.

Migration from Version 3.5SE to Version 5.1 does require you to examine the migrating applications.

After planning for migration and coexistence, you are ready to continue the installation.

Setting up Version 4.0.x and Version 5.x coexistence

You must migrate prerequisite and corequisite programs to the levels required by WebSphere Application Server, Version 5.1. You must also identify ports in use in Version 4.0.1 before you begin the installation, to avoid possible conflicts during coexistence.

You can install WebSphere Application Server Version 4.0.1, with a minimum Service level of W401502, and Version 5.1 on the same node. IBM recommends that you keep the service level current.

By default, there are port conflicts between Version 4.0.1 and Version 5.1 that you must resolve. See “Migrating from V4.0.1” on page 90 for more information. The next step is to setup the Webserver to support V4.0.1 and V5.1 coexistence.
Associate a Web server with each WebSphere Application Server.

- Use a separate Web server for each WebSphere Application Server.
  1. Create a Web server instance using the Web server documentation.
  2. Install the appropriate Web Server plug-in feature, and identify the Web server configuration file location. (For example, identify the location of the `httpd.conf` file for IBM HTTP Server.)
- Use the same Web server for both WebSphere Application Server versions.

  To use the same Web server for both Application Server versions, you must first upgrade the Web server to the common level supported by both versions of the application server.

  Follow this procedure to use the same Web server for both WebSphere Application Server versions.

1. Install the appropriate Web Server plug-in feature, and identify the Web server configuration file location. (For example, identify the location of the `httpd.conf` file for the IBM HTTP Server that is associated with Version 4.0.1.)
2. Edit the Web server configuration file to remove entries for Version 4.0.1, as described in the *Migrating plug-ins, one machine at a time* (tins_websmig2) topic in the base Application Server information center. The WebSphere Application Server Version 5, or higher, plug-in acts as the routing agent to route requests to both versions.
3. Generate the plug-in configuration files for both versions of the Application Server.
4. Edit the Version 4.0.1 `plugin-cfg.xml` file and the Version 5.1 `plugin-cfg.xml` file, to merge their entries into a combined file of all Web context roots. Web context roots must be unique across Application Server versions. If there is a common context root across versions, requests are served by the plug-in instance that was last loaded.

   If you have both Version 4.0.1 Samples and Version 5.1 Samples in the file, you can access only the Version 5.1 Samples after merging the configuration files because there is one context root for Samples.

   This is described in the *Preparing to install and configure a Web server* (tins_webserver) topic in the base Application Server information center.

   Whenever you modify the `plugin-cfg.xml` file for either version, you must manually merge the files again, to make a master file.

   This is described in the *Preparing to install and configure a Web server* (tins_webserver) topic in the base Application Server information center.

   Whenever you modify the `plugin-cfg.xml` file for either version, you must manually merge the files again, to make a master file.
5. Replace the original `plugin-cfg.xml` file of the Version 5 installation with the master file.

Utilizing WebSphere administrative capabilities to support migration from V4.0.1

There are a few steps in utilizing WebSphere Application Server V5 for z/OS administrative capabilities to support migration from V4.0.1 to V5.1:

1. “Migrating your server from WebSphere Application Server for z/OS V4.0.1”
2. “Migrating your applications from WebSphere Application Server for z/OS V4.0.1 to V5.1” on page 94
3. Migrating from Systems Management Scripting API (SMAPI) V4 to wsadmin V5
4. “Mapping of V4.0.1 environment variables to V5 WebSphere console settings” on page 96

Migrating your server from WebSphere Application Server for z/OS V4.0.1:

Migrating your server from WebSphere Application Server for z/OS V4.0.1 to V5.1 is a manual process. The export server capability of the V4.0.1 adminstrative console allows the dumping of V4.0.1 server settings to an XML file for review prior to manually configuring the V5.1 server.

1. Use the ″Export server″ task within the admistrative console to specify <export_dir> as the Output directory within your HFS. You can take an ACTIVE conversation (one currently being used), and find the J2EE Server (for example, the Server name J2SERV1) that you want to EXPORT.

   Then, you can specify <export_dir> (for example, in the Output directory) for output and export the J2EE server by clicking on OK.

   Note: The directory ″/export_dir/J2SERV1″ **MUST** already exist in order for this export to succeed.
   - ″/export_dir/J2SERV1″ should contain the *.xml file for each J2EE Resource that is used within the J2EE server that has been exported.
   - ″/export_dir/J2SERV1″ should contain the server.xml file that contains the configuration settings of the exported J2EE server.
   - ″/export_dir/J2SERV1″ should contain the <UUID> .ear file for each application that has been installed into the J2EE server being exported.

     You can look into server.xml to find out exactly which <UUID> .ear file maps to which application name.
2. Use <export_dir>/<server_name>/server.xml and transfer (or dump) the configuration.
3. Configure the V5.1 setup.
   a. Change the current configuration settings using the administrative console.

      Note: You should review the settings when you did a dump of the server.
   b. Save your SM configuration by one of two ways:
      1) Use the following administrative console command to commit the SM configuration:

         ```
         wsadmin> $AdminConfig  save
         ```
      2) Press the ″Save″ link in the administrative console to save your SM configuration into the master configuration.

      Here's an example of where it may be saved:

      ```
      /WebSphere/V5R0M0/AppServer/config/cells/SY1/...*.xml
      ```

Migrating your applications from WebSphere Application Server for z/OS V4.0.1 to V5.1:

Utilize the migration utility that was shipped in W401502 to process V4.0.1 EARs. This utility is being provided as an option on 390fy. In V4.0.1 it was required that EARs be processed through the 390 AAT or 390fy to condition them for deployment into V4.0.1. In V5 and higher, this conditioning is not required. This utility will remove the conditioning that was done for existing V4.0.1 EARs that are being migrated to V5 or higher. It should be noted that if the application is being sent back through the WSAD 5.0 tooling, that the resulting EAR is ready for installation into V5 or higher and that the use of migration utility is not required. Support for migrating V4.0.1 applications to V5 and higher has been shipped in V4.0.1 service (W401502). This support has been provided as a new -v5mp option for the 390fy tool.

The V4 migration option on 390fy is enhanced via feature WS18354.08 which is shipped in V4.0.1 service (W401609) to detect whether one or more methods of an EJB have syncToOSThread specified in the extended deployment descriptor. If 390fy finds syncToOSThread specified on any method within an EJB, it will add the following new env-entry to the standard deployment descriptor of the output EJB jar file for each of the EJBs that contained at least one or more synchToOSThread methods:

```
<env-entry>
<description>SyncToOSThread  Setting</description>
<env-entry-name>com.ibm.websphere.security.SyncToOSThread</env-entry-name>
<env-entry-type>java.lang.Boolean</env-entry-type>
<env-entry-value>true</env-entry-value>
```

In WAS 4.0.1, syncToOSThread could be defined at the individual method level. With this new functionality, if any method within an EJB specifies syncToOSThread, then syncToOSThread will be defined for the entire EJB.

The following are the steps needed to migrate your V4.0.1 application to V5 and higher:

1. Use the ″Export server″ task to specify <export_dir> within your HFS. See “Migrating your server from WebSphere Application Server for z/OS V4.0.1” on page 93 for more information on how to do this.
2. Change the <UUID> .ear name format to the regular name.

   Note: This step is optional.
3. Run the 390fy -v5mp <export_dir> /<server_name> option to migrate all applications that are exported into the <export_dir> /<server_name> directory. Each of the exported ear files will be in <UUID> format unless they have been changed in optional step 2 above.

   Your J2EE server (for example, J2SERV1) might contain the following installed applications:
   - J2APPL1
   - BVTEC
   - PolicyIVP

   When you export the J2EE server (J2SERV1 in this example) into the <export_dir>/<server_dir> (for example, ″/export_dir/J2SERV1″) directory, the directory will look like the following:

   ```
   /export_dir/J2SERV1/> ls -1
   <UUID_for_J2SERV1>.ear
   <UUID_for_BVTEC>.ear
   <UUID_for_PolicyIVP>.ear
   <UUID_for_some_DataSource1>.xml
   <UUID_for_some_DataSource2>.xml
   server.xml
   ```

   When you run the 390fy command in this directory it will generate a new *_v5mp.ear for each EAR file processed by the 390fy v5mp option.

   Note: The last parameter of the ″390fy″ command for the ″-v5mp″ option is a directory. In this case, it is the directory of the <export_dir>/<server_name> where the *.ear has been exported. For example:

   ```
   /export_dir/J2SERV1/> /usr/lpp/WebSphere/bin/390fy -v5mp .
   ```

   where ″**.**″ is the target directory where the *.ear files can be found, or

   ```
   /export_dir/J2SERV1/> /usr/lpp/WebSphere/bin/390fy -v5mp /export_dir/J2SERV1/
   ```

   where ″**/export_dir/J2SERV1/**″ is the target directory where the *.ear files can be found.

   The following demonstrates the 390fy v5mp option for this example:

   ```
   /export_dir/J2SERV1/> /usr/lpp/WebSphere/bin/390fy -v5mp .
   /export_dir/J2SERV1/> ls -1
   <UUID_for_J2SERV1>.ear
   <UUID_for_BVTEC>.ear
   <UUID_for_PolicyIVP>.ear
   <UUID_for_J2SERV1>_v5mp.ear
   <UUID_for_BVTEC>_v5mp.ear
   <UUID_for_PolicyIVP>_v5mp.ear
   <UUID_for_some_DataSource1>.xml
   <UUID_for_some_DataSource2>.xml
   server.xml
   ```

The UUID for each application can be looked up in the server.xml file to find the original application name.

Mapping of V4.0.1 environment variables to V5 WebSphere console settings: The following table lists only V5 WebSphere variables related to diagnosis, along with their equivalent V4.0.1 environment variables. This information is provided only as an aid to IBM service personnel.

Warning: Do not use this information to manually modify the contents of the *was.env* file. The *was.env* file is managed by WebSphere, and its content is rewritten with each change made to the WebSphere configuration. Therefore, any hand-editing will be overwritten. To determine which WebSphere variable, custom property or administrative console field must be updated in order to change the value of a specific internal variable, see Changing the values of variables referenced in BBOM0001I messages.

Table 5. V4.0.1 environment variables and their equivalent V5 WebSphere variables

V4.0.1 environment variables	Equivalent V5 WebSphere variables
BBOC_HTTPALL_TCLASS_FILE	**http_transport_class_mapping_file**
BBOC_HTTP_INPUT_IDENTITY	**protocol_http_defaultIdentity**
BBOC_HTTP_INPUT_TIMEOUT	**protocol_http_timeout_input**
BBOC_HTTP_LISTEN_IP_ADDRESS	**protocol_http_listenIPAddress**
BBOC_HTTP_MAX_PERSIST_REQUESTS	**protocol_http_max_persist_requests**
BBOC_HTTP_OUTPUT_TIMEOUT	**protocol_http_timeout_output**
BBOC_HTTP_OUTPUT_TIMEOUT_RECOVERY	**protocol_http_timeout_output_recovery**
BBOC_HTTP_PERSISTENT_SESSION_TIMEOUT	**protocol_http_timeout_persistentSession**
BBOC_HTTP_PORT	**protocol_http_port**
BBOC_HTTP_SSL_IDENTITY	**protocol_https_default_identity**
BBOC_HTTP_SSL_INPUT_TIMEOUT	**protocol_https_timeout_input**
BBOC_HTTP_SSL_LISTEN_IP_ADDRESS	**protocol_https_listenIPAddress**
BBOC_HTTP_SSL_MAX_PERSIST_REQUESTS	**protocol_https_max_persist_requests**
BBOC_HTTP_SSL_OUTPUT_TIMEOUT	**protocol_https_timeout_output**
BBOC_HTTP_SSL_OUTPUT_TIMEOUT_RECOVERY	**protocol_https_timeout_output_recovery**
BBOC_HTTP_SSL_PERSISTENT_SESSION_TIMEOUT	**protocol_https_timeout_persistentSession**
BBOC_HTTP_SSL_PORT	**protocol_https_port**
BBOC_HTTP_SSL_TRANSACTION_CLASS	**protocol_https_transactionClass**
BBOC_HTTP_TRANSACTION_CLASS	**protocol_http_transactionClass**
BBOC_LOG_RESPONSE_FAILURE	**protocol_bboc_log_response_failure**
BBOC_LOG_RETURN_EXCEPTION	**protocol_bboc_log_return_exception**
BBODUMP	**ras_dumpoptions_dumptype**
BBODUMP_CEE3DMP_OPTIONS	**ras_dumpoptions_ledumpoptons**
BBOLANG	**nls_language**
BBOO_WORKLOAD_PROFILE	**server_region_workload_profile**
CBCONFIG	**config_root**
CLASSPATH	**server_region_classpath**
CLIENT_RESOLVE_IPNAME	Replaced by the corbaloc function in J2EE 1.3 CosNaming INS
CLIENT_RESOLVE_PORT	Replaced by the corbaloc function in J2EE 1.3 CosNaming INS

Table 5. V4.0.1 environment variables and their equivalent V5 WebSphere variables (continued)

V4.0.1 environment variables	Equivalent V5 WebSphere variables
CLIENT_TIMEOUT	**protocol_iiop_local_timeout**
CLIENTLOGSTREAM	**client_ras_logstreamname**
CONFIGURED_SYSTEM	**server_configured_system_name**
CTL_LIBPATH	**control_region_libpath**
DAEMON_IPNAME	**protocol_iiop_daemon_listenIPAddress**
DAEMON_PORT	**protocol_iiop_daemon_port**
DAEMON_SSL_PORT	**protocol_iiop_daemon_port_ssl**
DM_GENERIC_SERVER_NAME	**daemonName**
DM_SPECIFIC_SERVER_NAME	**daemonInstanceName**
ENABLE_TRUSTED_APPLICATIONS	**control_region_security_enable_trusted_applications**
GENERIC_SERVER_NAME	**server_generic_short_name**
GENERIC_UUID	**server_generic_uuid**
IIOP_SERVER_SESSION_KEEPALIVE	**protocol_iiop_server_session_keepalive**
IVB_DEBUG_ENABLED	**ras_debugEnabled**
JVM_ENABLE_VERBOSE_GC	No longer a variable. Specify as Java option: -verbose:gc
JVM_HEAPSIZE	No longer a variable. Specify as Java option: -Xmx<size>
JVM_MINHEAPSIZE	No longer a variable. Specify as Java option: -Xms<size>
JVM_LOCALREFS	**control_region_jvm_localrefs**
JVM_LOCALREFS	**server_region_jvm_localrefs**
JVM_LOGFILE	**control_region_jvm_logfileserver_region_jvm_logfile**
LIBPATH	**server_region_libpath**
LOGSTREAMNAME	**ras_log_logstreamName**
MAX_SRS	**wlm_maximumSRCount**
MIN_SRS	**wlm_minimumSRCount**
OTS_DEFAULT_TIMEOUT	**transaction_defaultTimeout**
OTS_MAXIMUM_TIMEOUT	**transaction_maximumTimeout**
RAS_MINORCODEDEFAULT	**ras_minorcode_action**
RECOVERY_TIMEOUT	**transaction_recoveryTimeout**
REM_PASSWORD	**client_protocol_password**
REM_USERID	**client_protocol_user**
SPECIFIC_SERVER_NAME	**server_specific_short_name**
SPECIFIC_UUID	**server_specific_uuid**
SRVIPADDR	**protocol_iiop_listenIPAddress**
SSLIIOP_SERVER_SESSION_KEEPALIVE	**protocol_iiop_server_session_keepalive_ssl**
SSL_KEYRING	**security_sslKeyring**
TRACE_EXCLUDE_SPECIFIC	**ras_trace_exclude_specific**
TRACEALL	**ras_trace_defaultTracingLevel**
TRACEBASIC	**ras_trace_basic**
TRACEBUFFCOUNT	**ras_trace_BufferCount**
TRACEBUFFLOC	**ras_trace_outputLocation**

Table 5. V4.0.1 environment variables and their equivalent V5 WebSphere variables (continued)

V4.0.1 environment variables	Equivalent V5 WebSphere variables
TRACEBUFFSIZE	**ras_trace_BufferSize**
TRACEDETAIL	**ras_trace_detail**
TRACEMINORCODE	**ras_trace_minorCodeTraceBacks**
TRACEPARM	**ras_trace_ctraceParms**
TRACESPECIFIC	**ras_trace_specific**
WAS_JAVA_OPTIONS	No longer a variable. Java options are specified in the JVM configuration of the servant process.

Migrating from V5.0

Determine whether you have an existing version of WebSphere Application Server installed on the machine where you plan to install your Version 5.1 product.

To migrate applications and configurations from V5 to V5.1, follow this procedure:

1. Prepare to migrate; update product prerequisites and corequisites to supported versions. If your existing configuration makes use of a Web server running on a platform other than zSeries and an associated WebSphere Application Server V5.0 (other than WebSphere Application Server for z/OS V5.0) plug-in, you should review Web server levels required by V5.1 and update as appropriate.

   Some key dependencies (SDK 1.4.1, z/OS 1.2) have changed in V5.1. See “Prerequisites needed for WebSphere Application Server for z/OS V3.5SE, V4.0.1, V5, and V5.1” on page 101 for additional details.

   There is no service level agreement for V5.0 to allow coexistence with V5.1. However, V5 must have Fix Pack 2 installed to support migration from V5 to V5.1. Migration tooling has been introduced in V5.1 to allow a staged migration for existing V5 Network Deployment configurations. See “V5.1 Migration utility” on page 79 for more information on migration tooling.
2. Review your existing V5 configuration for possible coexistence issues related to port definitions and placement of WebSphere Application Server modules in LPA. See “Migrating and coexisting” on page 76 for additional information.
3. Install the new product. Note that as of Version 5, the use of SMP/E 3.1 is a requirement. Installation instructions have been provided with your ServerPac or CBPDO order.
4. V5.1 provides tooling to simplify the customization process. See “V5.1 Migration utility” on page 79 for instructions on customizing the new product. Verify your success with the Installation Verification Test.
5. Optional: Run the new samples supplied with the product. See Samples Gallery for more information on accessing the samples.
6. Migrate your existing V5.0 configuration using the migration tools provided. The migration tools provided are used to create a V5.1 clone of an existing V5.0 configuration, while leaving the V5.0 configuration intact. Since the V5 and V5.1 configurations are identical, only one configuration can be active at a time.

   Note: While the migration tools are able to migrate all of your resources and applications, they do not migrate entities in your classes directory. Migration saves the following files in the backup directory:

   - classes
   - config
   - installableApps
   - installedApps
   - properties

See “Migration tools” on page 78 for additional details.

7. Install resources into the new server, if necessary. Although resources are migrated as a part of the migration process, prerequisite levels needed for WebSphere Application Server for z/OS may have changed from V5.0. Review the prerequisites at “Prerequisites needed for WebSphere Application Server for z/OS V3.5SE, V4.0.1, V5, and V5.1” on page 101.
8. Move applications to V5.1 level tooling. Most V5 applications that were generated with WebSphere Studio Application Developer V5 and WebSphere Studio Application Developer IE V5 and are running on WebSphere Application Server for z/OS V5 will continue to run on V5.1, and can be deployed using the V5.1 administrative tools. However, WebSphere Studio Application Developer V5, WebSphere Studio Application Developer V5.1, and WebSphere Studio Application Developer IE V5 are not supported for build or unit test of applications targeted for a WebSphere Application Server V5.1 runtime. This necessitates movement of the application to V5.1 level tooling (V5.1.1 for WebSphere Studio Application Developer). WebSphere Studio Application Developer V5.1.1 and WebSphere Studio Application Developer IE V5.1 have the flexibility to generate and deploy applications to both V5 and V5.1 runtimes, allowing convergence to a single tool to support V5 and V5.1 applications.
9. Review applications to be migrated to determine if any changes are required. In general, existing V5 applications are expected to run. However, JDK 1.4.1 has added interfaces to the Throwable class that may conflict with similar method calls that were provided by previous WebSphere Application Server releases. Applications using these interfaces may need to be changed.
10. Update your Web server to the V5.1 prerequisite level and the appropriate plug-in. Most V5 configurations will include a Web server, which could be a z/OS Web server or any other Web server from another platform that is supported by WebSphere Application Server V5. If a Web server configuration is required for V5.1, then the Web server to be used must be updated to the V5.1 prerequisite level, and the appropriate V5.1 plug-in must be installed. Note that the V5.1 plug-in can support both V5 and V5.1 of WebSphere Application Server. Keeping this in mind, several migration configurations are possible:
    - Install the V5.1 plug-in into the existing Web server. Coexistence with V5 and V5.1 plug-ins are supported.
    - Create a unique Web server instance and install the V5.1 plug-in.
    - Replace the V5 plug-in with the V5.1 plug-in and setup to drive both the V5 and V5.1 images.
11. Perform post-migration activities.
    - Some activities have been deprecated at the V5.1 level. Deprecated items are sometimes still supported at the current level with the understanding that migration plans should be in place to bring the items up to supported levels. See Chapter 2, ″Deprecated Version 5.1 Features″ to determine if you should plan migration paths for any of your functions that might be deprecated.
    - If you have not already done so with V5, migrate applications to the latest specification levels.

After planning for migration and coexistence, you are ready to continue the installation.

Specification level summary of WebSphere Application Server

This topic shows the various releases of the WebSphere Application Server for z/OS family and the specification and functional differences between them.

Table 6. Specification and functional differences between WebSphere Application Server for z/OS releases

Specifications	V3.5SE	V4.0	V4.0.1	V5	V5.1

Table 6. Specification and functional differences between WebSphere Application Server for z/OS releases (continued)

Java related	• Requires SDK 1.3 • Supports specification levels: – Servlet 2.1/2.2 – JSP .091/1.0/1.1 – JDBC 1.2	• Requires SDK 1.3 • Supports J2EE 1.2 levels: – Servlet 2.2 – JSP 1.1 – JDBC 2.0 – EJB 1.1 – JNDI 1.2 – JTA 1.0 – RMI/IIOP 1.0	• Requires SDK 1.3 • Supports J2EE 1.2 levels: – Servlet 2.2 – JSP 1.1 – JDBC 2.0 – EJB 1.1 – JNDI 1.2 – JTA 1.0 – RMI/IIOP 1.0 – JMS 1.1 – JavaMail 1.1 – JAP 1.0 – Client container	• Requires SDK 1.3 • Supports J2EE 1.3 levels: – Servlet 2.3 – JSP 1.2 – JDBC 2.0 – EJB 2.0 – part of J2SE 1.3 – JTA 1.0 – part of J2SE 1.3 – JMS 1.1 – JavaMail 1.2 – JAP 1.0 – Client container – Java Authentication and Authorization Service (JAAS) 1.0 – Java API for XML Parsing (JAXP) 1.1 – J2EE Connector Architecture 1.0	• Requires SDK 1.4.1 • Supports J2EE 1.3 levels: – Servlet 2.3 – JSP 1.2 – JDBC 2.0 – EJB 2.0 – part of J2SE 1.3 – JTA 1.0 – part of J2SE 1.3 – JMS 1.1 – JavaMail 1.2 – JAP 1.0 – Client container – Java Authentication and Authorization Service (JAAS) 1.0 – Java API for XML Parsing (JAXP) 1.1 – J2EE Connector Architecture 1.0
Web Services	N/A	N/A	Introduces Web Services equivalent to those provided by: • WebSphere AE V4.0 – HTTP to Stateless Session beans • Requires SDK 1.3 PTF • Supports specification levels: – SOAP 1.1 – Apache SOAP V2.2	• UDDI V2 • SOAP 2.3 • jaxrpc 1.0 GA • jsr109 V1.0 GA	• UDDI V2 • SOAP 2.3 • jaxrpc 1.0 GA • jsr109 V1.0 GA

- **WebSphere V3.5SE** adds support for Servlet/JSP specification levels required by the J2EE 1.2 specifications, allowing customers to take advantage of the latest levels and to start early migration of applications in preparation for movement to J2EE 1.2 servers.
- **WebSphere V4.0** introduces a J2EE server that provides support for enterprise applications consisting of Web applications and enterprise java beans (EJBs). Support for Servlets, JSPs, and EJBs is

compliant to the J2EE 1.2 architecture and provides all the benefits of this architecture. Tooling supports the J2EE 1.2 required packaging scheme. The J2EE server, unlike prior JAVA runtimes provided by WAS SE, provides container managed services for items such as transactions and security, simplifying the application programmers job. In addition the J2EE server utilizes the z/OS infrastructure to provide QOS expected on the 390 platform.

- **WebSphere V4.0.1** finishes delivery of all functional items required for certification. This includes delivery of JMS, JAVAMail, and client container support. In addition, WebSphere 4.0.1 provides the initial delivery of Web Services.
- **WebSphere V5** provides support for next generation technologies- J2EE 1.3 compatible with support for key Web services. V5 is Network Deployment compliant and continues to build towards WebSphere family consistency in architecture, administrave console, and programming APIs.
- **WebSphere V5.1** continues to support WebSphere brand consistency. Additionally, this product provides complete client and server support for SDK 1.4.1.

Prerequisites needed for WebSphere Application Server for z/OS V3.5SE, V4.0.1, V5, and V5.1

The following table describes the prerequisites needed for WebSphere Application Server for z/OS V3.5SE and V4.0.1 Alternate Configuration Option, V4.01, V5, and V5.1.

Table 7. Prerequisites needed for WebSphere Application Server for z/OS V3.5SE and V4.0.1 Alternate Configuration Option, V4.01, V5, and V5.1.

Pre-Reqs	V3.5 and V4.0.1 Alternate Configuration Option	V4.0.1	V5.0	V5.1
SDK (included with WebSphere Application Server for z/OS)	SDK 1.3	SDK 1.3	SDK 1.3	SDK 1.4.1
OS/390 or z/OS	R8 and above	R8 and above	R10 and above z/OS 1.2 is required for the Dynamic application environment	z/OS 1.2 or higher z/OS.e 1.3 or higher
HTTP Server	IBM HTTP Server for OS/390 ONLY	Options: • Direct browser to HTTP transport • IBM HTTP Server for OS/390 with no HTTP transport • IBM HTTP Server for OS/390 with HTTP transport • Any HTTP server that supports AEplugin 4.0.2 and above.	Must use HTTP transport Options for connecting to HTTP transport: • Direct Browser • IBM HTTP Server for OS/390 (R10) • Any HTTP Server supported by V5.0 family plugins	Must use HTTP transport Options for connecting to HTTP transport: • Direct Browser • IBM HTTP Server for z/OS 1.2 • Any HTTP Server supported by V5.1 family plugins
Sysplex	Not required	Required	Required	Required
OS/390 Communications server (TCP/IP)	Required	Required	Required	Required

Table 7. Prerequisites needed for WebSphere Application Server for z/OS V3.5SE and V4.0.1 Alternate Configuration Option, V4.01, V5, and V5.1. (continued)

OS/390 Unix System Services and Hierarchical file system (HFS)	Required	Required Shared HFS is required for a multiple image sysplex	Required Shared HFS is supported, but is no longer required for a multiple image sysplex	Required Shared HFS is supported, but is no longer required for a multiple image sysplex
SecureWay Security Server (RACF) or Equivalent	Required	Required	Required	Required
System Logger	Not required	Required	Required	Required
LightWeight Directory Access Protocol (LDAP) Server	Not required	Required	Not required	Not required
Workload Manager in Goal mode	Only required for scaleable mode	Required	Required Dynamic AE (programmatic setup of WLM application environment) requires PTF against z/OS 1.2	Required Dynamic AE (programmatic setup of WLM application environment) requires PTF against z/OS 1.2
Resource Recovery Services (RRS)	Not required	Required	Required	Required
FTP server	Not required	Required	Not required	Not required
System SSL Security	Required for SSL	Required for SSL	Required for SSL	Required for SSL
DB2	V5 or later if JDBC or Session State is used	V7 required for Admin DB	Not required or supported for SM function V7 required for user data (JDBC compliant driver)	Not required or supported for SM function V7 or higher required for user data (JDBC compliant driver)
WSMQ	Not required	Not required Optional support for point to point: MQSeries 5.2+PTF; SupportPac MA88 Optional support for pub/sub: MQ SI 2.1	Not required - Integrated JMS provided as part of WAS 5.0 If integrated provider is not required: • Can continue to use the same product stack as in WAS 4.0.1 **Note:** Functions such as XA support are only available with MQ5.3.1. • Can purchase and install full MQ5.3.1 product.	Not required - Integrated JMS provided as part of WAS 5.1 If integrated provider is not required: • Can continue to use the same product stack as in WAS 4.0.1 **Note:** Functions such as XA support are only available with MQ5.3.1. • Can purchase and install full MQ5.3.1 product.

Table 7. Prerequisites needed for WebSphere Application Server for z/OS V3.5SE and V4.0.1 Alternate Configuration Option, V4.01, V5, and V5.1. (continued)

IMS JCA	Not applicable	IMS Connect 1.2+PTFs (requires IMS V7.1) IMS Connect 2.1 (requires IMS V8.1)	IMS Connect 2.1 (requires IMS V8.1)	IMS Connect 2.1 (requires IMS V8.1)
CICS JCA	Not applicable	CICS Transaction Gateway 4.0.2, 5.0, 5.0.1 (requires CICS TS 1.3)	CICS Transaction Gateway 5.0.1 (requires CICS TS 1.3)	CICS Transaction Gateway 5.1 (requires CICS TS 1.3)

Coexistence port definitions

The following table defines port definitions for WebSphere Application Server for z/OS versions.

See “z/OS port assignments” on page 153 for more information on ports.

Table 8. Coexistence port definitions

WebSphere Application Server for z/OS V5 and higher	WebSphere Application Server for z/OS V4.0.1	WebSphere Application Server for z/OS V3.5SE
Daemon 5655 5656 (SSL)	Daemon 5555	
Application server	Application server	Web server
IIOP Dynamically chosen Equal to bootstrap port	IIOP Dynamically chosen	
Bootstrap 2809 if base server 9810 if part of ND cell	Application Server did not have a bootstrap port	
IIOPS Dynamically chosen	IIOPS Dynamically chosen	
HTTP 9080 (same as distributed)	HTTP No Default (8080 in examples)	HTTP 80
HTTPS 9443 (same as distributed)	HTTPS No Default (8080 in examples)	HTTPS 443
NodeAgent	SM	
Bootstrap 2089 (same as distributed)	Bootstrap 900	
IIOP 2089 (separate in distributed)		
IIOPS Dynamically chosen		
LDAP Not Applicable	LDAP 1439	
Naming/IR Not Applicable	Naming/IR Dynamically chosen	

Migrating XML applications

The following step is needed to migrate Version 3.5SE XML applications to supported XML APIs.
Migrate Version 3.5.x XML applications to supported XML APIs

If your XML applications use XML for Java API, Version 2.0.x or earlier, you must migrate them to API Version 3.1 or the equivalent open-source version. Although there are inherent performance improvements in later versions of the XML for Java API, you can gain additional performance by explicitly using nonvalidating parsers in application environments where you can trust the data.

The most significant change is that the TX-compatible APIs are no longer available. The Document API retains the XML manipulation APIs that were in TXDocument, but you must rewrite the following functionality:

- Creating and loading an XML parser: Use a Java API for XML Processing (JAXP) factory class.
- Writing out the Document Object Model (DOM) tree: Use a serializer. One drawback to the DOM Level 2 implementation in this level of the XML for Java API is that the grammar (DTD or schema) is no longer a node in the DOM tree, so you cannot write it out. As a result, only external grammars are recommended. You can query the system ID of the root element and use it to retrieve the name from the statement. After the tree is written to an XML file, you can read the file as text and insert a statement.

Note: In addition to the XML API changes, it is important to understand that J2EE Version 1.3 mandates the use of JAXP 1.1, DOM Version 2, and SAX Version 2. JAXP Version 1.2, DOM Version 3, and SAX Version 3 are not allowed in products that are compliant with the J2EE Version 1.3 specification. This prohibition exists because the newer versions were *experimental* at the time of the J2EE Version 1.3 specification. Because WebSphere Application Server is compliant with the J2EE Version 1.3 specification, WebSphere Application Server has support for JAXP Version 1.1, DOM Version 2 and SAX Version 2 only.

You must only recompile a Version 4.0.1 XML application to migrate it to the Version 5 level.

XML parser for Java code

On November 9, 1999, the Apache Software Foundation announced the creation of the xml.apache.org project for Open Source Extensible Markup Language (XML) solutions.

As part of that announcement, IBM announced that it was donating its XML4J, XML4C, and LotusXSL technologies to the xml.apache.org project. The parsing technologies have been renamed Xerces, and the LotusXSL technology has been renamed Xalan.

IBM is shifting its XML parsing development resources to work on the Xerces parsers. The objective is to use the Xerces code base as the foundation for XML4J and XML4C. This version of XML4J is based on the Apache Xerces2 code base.

What is the difference between XML4J and Xerces?

IBM has tested this version of XML4J in addition to the testing done by the xml.apache.org project. The names of the main Java archive (JAR) files have changed from `xml4j.jar` to `xmlParserAPIs.jar` and `xercesImpl.jar`, and from `xml4jSamples.jar` to `xercesSamples.jar`.

This version of XML4J contains one major API change and a few relatively minor API changes from the last major release. It’s major features are many bugfixes and performance improvements, as well as the addition of a few features.

XML4J 4.2.x ships with the official W3C DOM HTML Recommendation API . The major change to the API was the removal of setSelected from the HTMLOptionElement interface. To workaround this change from XML4J 4.1.x, use setAttribute and removeAttribute from the interface to modify the selected attribute.

Which part of the API is public, and which is subject to change?

To answer this question, you must understand the terms used for application program interface (API) status:

- **Public** - The typical client developer codes for this API. Any severe problem in this API is addressed. Also, the majority of this API is defined as ″public″ and has reached World Wide Web Consortium (W3C) Recommendation status or a similar status in the XML Core Working Group. Not much change in these interfaces is expected. A W3C Recommendation indicates that a specification is stable, contributes to Web interoperability, and has been reviewed by all W3C members, who are in favor of supporting its adoption by the industry.
- **Experimental** - These interfaces and classes reflect the latest W3C specifications and Simple API for XML (SAX) specifications from the XML Core Working Group. Because these specifications are not finalized, the interfaces are subject to change. As an experimental specification is adopted at the highest level, such as at the W3C Recommendation level, the specification is upgraded to the Public category.
- **Internal** - These classes are considered internal to Xerces, even though they might be public and have public methods. Developers with complex and specific needs, such as building an XML parser can use these classes. However, the architecture is subject to change.

The following table summarizes API status by interface content:

API Status	API contents (packages, interfaces, classes and methods)	Comments
Public	DOM Level 1 interfaces (found in org.w3c.dom, org.w3c.dom.html) and DOM Level 2 (DOM2) interfaces (found in org.w3c.dom and subpackages) The DOM2 interfaces are implemented in the same interfaces as DOM1 as new methods. SAX Level 1 interfaces (found in org.xml.sax. and subpackages) and SAX Level 2 (SAX2) interfaces (found in org.xml.sax. and subpackages)	DOM L1, DOM L2, SAX1, and SAX2 Interfaces are stable.
Experimental	DOM Level 3 (DOM3) and Core DOM Level 3 (DOM3) Abstract Schemas and Load and Save (found in org.apache.xerces.dom3 and subpackages).	DOM L3 is in working draft status. XML4J provides a subset of DOM L3 support.
Internal	All other packages are internal.	The internal Xerces architecture can change.

For more information see XML4J Information.

As an Apache Open Source project, the Xerces community is interested in your questions and feedback regarding the entire API, not just the part that is designated as public.

If you have specific questions, patches, or feedback regarding the Xerces API or code, visit the Apache Web site, and sign up for the mailing list. Send more basic questions or questions specific to XML4J to the AlphaWorks discussion forum.

Which APIs do I use for new development?

Use the org.apache.xerces.parsers.* classes for new development. The four compatibility parser classes are replaced by the following classes:

```
org.apache.xerces.parsers.SAXParser
org.apache.xerces.parsers.DOMParser
```

Validation control has become a feature of the parser, and no longer requires separate classes.

The future direction for the parser instantiation classes is a parser instantiation API that results from the W3C DOM Level 3 effort, which is just getting underway.

To make sure that your code is as stable as possible, use the interfaces that are specified in the Public section of the preceding table (for example DOM1 and SAX1). Updates are made to the table to reflect the new interfaces and the classes that are given Public status (for example, DOM2 and SAX2).

Is IBM making any additional support guarantees with XML4J?

IBM is not making any additional support guarantees for XML4J. In particular, IBM is not certifying that XML4J is Y2K compliant. XML4J makes no internal date calculations.

Migrating applications to use the XML4J 4.2.2 parser and the XSLT4J 2.5.4 transformer

The Java API for XML Processing (JAXP) specification defines a pluggability mechanism for a SAX or a DOM parser using the javax.xml.parsers APIs. Transformers are pluggable using the javax.xml.transform APIs.

The IBM SDK 1.4.1 bundles in Version 5.1 include an XML4J 4.2.2 parser and an XSLT4J 2.5.4 transformer. You can use a different implementation of JAXP in an application. Package the parser and transformer in the application.

You can change an application to remove its dependency on the API in a previous version of the parser or the transformer from an earlier version of WebSphere Application Server. Package the JAR files in the application.

In both cases, set the class loader delegation mode to PARENT_LAST on the application or Web module.

Recommendation: Have your applications use the JAXP API instead of using the parser or transformer implementation API directly.

You must recompile a V4.0.x XML application or a V5.0.x XML application to migrate it to the Version 5.1 level.

Customization Dialog help panels - worksheets and definitions

Follow the linked articles to find worksheets and definitions for all the variables you come across when going through options in the Customization Dialog.

Load customization variables - worksheet

Print out this worksheet for each option in the Customization Dialog and use it to record your values as you customize WebSphere Application Server for z/OS.

Table 9. Load Customization Variables

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Data set name Volume	(null)	

Specify the data set from which you will prime the variables.

Rules:

- The data set must be a sequential data set. Do not specify a member name.
- Place quotes around the data set name.

Load security domain variables - worksheet

Print out this worksheet for each option in the Customization Dialog and use it to record your values as you customize WebSphere Application Server for z/OS.

Table 10. Load Security Domain Variables

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Data set name Volume	(null)	

Specify the data set from which you will prime the variables.

Rules:

- The data set must be a sequential data set. Do not specify a member name.
- Place quotes around the data set name.

Allocate Target Data Sets - Worksheet

Print this worksheet for each option in the Customization Dialog (or copy the table into a spreadsheet) and use it to record your values as you customize WebSphere Application Server for z/OS.

Table 11. Allocate Target Data Sets

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
High Level Qualifier	(null)	

This panel asks you to specify the high-level qualifiers (HLQs) for the target data sets. Target data sets are those into which the Customization Dialog places the customized jobs and other data. The data sets are:

hlq.CNTL

A partitioned data set of fixed block, 80-byte records, that contains WebSphere Application Server for z/OS customization jobs.

hlq.DATA

A partitioned data set of variable length records that contains other data produced by the Customization Dialog.

Define variables for security domain configuration - worksheets

This article presents worksheets that will assist you when completing the WebSphere Application Server for z/OS Customization Dialog. See the related definitions article for definitions of each of the terms.

Note: Any instance of ″(cannot change)″ in the ″Your value″ column indicates that that particular value is displayed for reference purposes only and you can't change it in that particular panel. Either the value is static throughout the Dialog, or you need to go back to a previous task to make the change.

For more information about security, see related sections in the information center.

Define variables for Security Domain Configuration (1 of 2)

A security domain definition in z/OS provides WebSphere Application Server for z/OS with a set of cell-wide z/OS Security Server (RACF) security definitions. In this circumstance, an unfederated base Application Server is, for security purposes, considered a cell. A security domain definition includes the following:

- Common definitions for administrative users and groups
- X509 Digital Certificate Authorities (to ensure SSL communication)
- SAF groups to which servers belong
- Authorization profiles.

Note: Creating multiple security domains may isolate WebSphere Application Server for z/OS cells within a sysplex.

This panel asks you to supply some RACF groups and user IDs that are common throughout WebSphere Application Server for z/OS. The Dialog creates the RACF commands to define these new user IDs and groups for your security system. It also creates and saves a set of variables that you must load and use in subsequent Dialog options.

Note:

- If you anticipate the use of multiple security domains in your WebSphere Application Server for z/OS configuration, we recommend that you create a naming scheme for saved security domain definition data.
- To minimize the number of RACF definitions, RACF authorizations are at the group level rather than the user ID level. In a later panel, the Dialog asks for user IDs for the run-time servants and controllers. These user IDs will be connected to their proper RACF groups.
- For controllers, which run system authorized code, you can create a single group. Thus, you can safely use the RACF group that the Dialog created for all controllers on your system. This Dialog creates a RACF group for the initial base Application Server. Depending on the resources accessed, your installation will either reuse this group or define new ones to handle additional Application Servers.

Table 12. Define variables for Security Domain Configuration (1 of 2)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Use security domain identifier in RACF definitions	N	
Security domain identifier	(null)	
Sysplex name	MCLXCF01	
Generate default RACF realm name	N	
Default RACF realm name	MCLXCF01	
WebSphere Application Server Configuration Group Information		
Group	WSCFG1	

Table 12. Define variables for Security Domain Configuration (1 of 2) (continued)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
GID	2500	
WebSphere Application Server Administrator Information		
User ID	WSADMIN	
UID	2403	
Password	WSADMIN	
Unauthenticated User Definitions for Base Servers		
User ID	WSGUEST	
UID	2402	
Group	WSCLGP	
GID	2502	
WebSphere Application Server Asynchronous Administration Task		
User ID	WSADMSH	
UID	2504	
Configure for local OS security registry	Y	

Define variables for Security Domain Configuration (2 of 2)

Table 13. Define variables for Security Domain Configuration (2 of 2)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere Application Server Common Groups and User IDs		
Servant group for base servers	WSSR1	
Servant GID for base servers	2501	
SSL Customization		
Certificate authority keylabel	WebSphereCA	
Generate certificate authority (CA) certificate	Y	
Expiration date for CA authority	2010/12/31	
Default RACF keyring name	WASKeyring	
Enable SSL on location service daemon	N	
Additional z/OS Security Customization Options		
Use SAF EJBROLE profiles to enforce J2EE roles	Y	
Enable PassTickets for z/SAS authentication	N	
PassTicket KEYMASK value	(null)	

Table 13. Define variables for Security Domain Configuration (2 of 2) (continued)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Enable SAF authentication using LTPA or ICSF login tokens	Y	
Use APPL Profile to restrict access to WebSphere Application Server	N	

Define variables for security domain configuration - definitions

This article lists definitions for the terms you will come across in the WebSphere Application Server for z/OS Customization Dialog.

Note: Some term definitions--those that you cannot set in the related panel--are not displayed. To see the definition, you must to go to the Information Center article for the panel on which you **can** change that particular term.

Security Domain Configuration (1 of 2)

In some of the following, specifying ″Y″ (yes) tells the Dialog to define the profile or enable an option. Specifying ″N″ (no) tells the Dialog to not define the profile or enable the option.

Use security domain identifier in RACF definitions

Specify ″Y″ if you want to use a security domain identifier in your RACF definitions, which will cause the Customization Dialog jobs to include the security domain identifier in any CBIND, APPL, PASSTKT, or EJBROLE profiles generated by WebSphere Application Server for z/OS. The WebSphere Application Server for z/OS runtime also uses this information to indicate the profiles that require checking.

If you specify ″Y″, ensure you also specify a standard RACF 8-character type prefix for the security domain name. Selecting ″N″ will ignore any security domain identifier value specified.

Security domain identifier

If you are using a security domain identifier, specify its name here. It is generally a good idea to make the value the same as the short name of the WebSphere Application Server for z/OS cell that uses it.

Rule: The security domain identifier name must contain 8 or fewer characters.

Note: This value is also stored as a custom property in the Global Security definition in the Administrative Console.

Sysplex name

The sysplex name for the target z/OS system on which WebSphere Application Server for z/OS is installed.

Tip: If you are not sure what the system name (&&SYSNAME) and sysplex name (&&SYSPLEX) are, display them using the console command D SYMBOLS on the target z/OS system.

Generate default RACF realm name

Specify ″Y″ if you want to generate a default RACF realm name, which is a sysplex-wide SAF setting used to identify a particular RACF (or compliant) database. The CSIV2 protocol uses this value to identify the security realm for Local OS authentication.

Note:

- Only one value is in effect at any one time, so set this up only once per sysplex.
- The CSIV2 Local OS registry uses the location service daemon IP name as the security realm name if there is no value in RACF.

Default RACF realm name

If you are generating a default RACF realm name, specify its name here.

Rules: In this and the following sections, ensure you follow these rules:

- User IDs and groups must be unique names (1 to 8 characters).
- UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.
- Do not assign a UID of 0 (superuser) to any of these users.
- GIDs should be unique numbers between 1 and 2,147,483,647.

WebSphere Application Server Configuration Group Information:

Group

The default group name for the WebSphere Application Server for z/OS administrator and base server. This group allows you to more easily control who can and cannot make configuration changes to your servers.

Rule: These two users must have the same default group.

Note: If you are building base Application Server nodes that you might later federate into a deployment manager cell, make sure they all have the same group value.

GID The group identifier for the WebSphere Application Server for z/OS configuration group.

WebSphere Application Server Administrator Information:

User ID

The user ID you use to log onto Telnet to perform administrative actions against your server.

UID The user identifier for the WebSphere Application Server for z/OS user ID.

Password

The password for the WebSphere Application Server for z/OS user ID.

Unauthenticated User Definitions for Base Servers:

User ID

If you allow unauthenticated client requests, this is the default user ID under which those requests run.

UID The user identifier for the unauthenticated user.

Group

The group for unauthenticated users.

GID The group identifier for unauthenticated users.

WebSphere Application Server Asynchronous Administration Task:

User ID

Specifies the user ID under which the administration asynchronous operations procedure executes. Ensure that the userid has permission to all four of the administrative roles and exist in the same RACF group as all the Application Servers.

UID The user identifier for the WebSphere asynchronous administration task user ID.

Configure for local OS security registry

Specify ″Y″ if you plan to configure WebSphere Application Server for z/OS security to use the Local OS Security Registry (for example, RACF or other SAF-compliant security product) as the active registry for user authentication and identification. Specify ″N″ if you plan to configure WebSphere Application Server for z/OS security to use an LDAP or custom registry.

Security Domain Configuration (2 of 2)

In some of the following, specifying ″Y″ (yes) tells the Dialog to define the profile or enable an option. Specifying ″N″ (no) tells the Dialog to not define the profile or enable the option.

WebSphere Application Server Common Groups and User IDs:

Servant group for base servers

Specifies an additional group name to which the base server's user ID connects. This is used to control access to resources that are external to the Application Server (for example, DB2).

Servant GID for base servers
A group identifier that the Dialog uses for the WebSphere Application Server for z/OS base server's servant group.

SSL Customization:

Certificate authority keylabel
Name of the keylabel that identifies the WebSphere Application Server for z/OS certificate authority (CA) that is generated when you run the RACF jobs.

Generate certificate authority (CA) certificate
Select ″Y″ to generate a new CA certificate. Select ″N″ to have an existing CA certificate generate server certificates.

Expiration date for CA authority
The expiration date used for any X509 Certificate Authority certificates, as well as the expiration date for the personal certificates generated for WebSphere Application Server for z/OS servers. You must specify this even if you selected ″N″ for ″Generate Certificate Authority (CA) certificate.″

Default RACF keyring name
The default name given to the RACF key ring. The key ring names created for repertoires are all the same within a cell.

Enable SSL on location service daemon
Select ″Y″ if you wish to support secure communications using Inter-ORB Request Protocol (IIOP) to the location service daemon using SSL. If you specify ″Y″, a RACF key ring will generate for the location service daemon to use.

Additional z/OS Security Customization Options:

Use SAF EJBROLE profiles to enforce J2EE roles
Select ″Y″ to indicate the use of SAF EJBROLE profiles, rather than WebSphere Application Server for z/OS bindings created during application deployment, for authorization of J2EE and WebSphere Application Server for z/OS administrator roles. The value specified here is relevant when:

- WebSphere Application Server for z/OS security is enabled
- Local OS is the active registry.

When SAF EJBROLE profiles are used, it is the WebSphere administrator's responsibility to ensure that SAF EJBROLE profiles are defined, and a system administrator's responsibility to complete user-to-role mapping. The Customization Dialog generates sample EJBROLE profiles to restrict access to WebSphere Application Server for z/OS administration and naming services.

If you specify a security domain identifier, SAF EJBROLE profiles must prepend the security domain identifier to the profile name. For example, if your application role is defined as Teller and your security domain identifier is defined as CELL1, then WebSphere Application Server for z/OS checks if the caller has read access to the EJBROLE profile called ″CELL1.Teller″. If you don't use a security domain identifier is used, the access check is performed on the EJBROLE ″Teller″ profile.

Enable PassTickets for z/SAS authentication
Specify ″Y″ to enable PassTickets for z/SAS authentication, in which case KEYMASK is required.

PassTicket KEYMASK value
Specify any string of 16 hexadecimal characters as a secret KEYMASK for PassTickets.

Enable SAF authentication using LTPA or ICSF login tokens
Specify ″Y″ to enable the WebSphere Application Server for z/OS servant to authenticate users to the SAF registry without providing a password or SAF-specific authenticator. This is required when:

- WebSphere Application Server for z/OS security is enabled
- Local OS is the active registry

and either:

- ICSF or LTPA is the authentication mechanism **OR**
- A Trust Association Interceptor is in use.

Setting this value to ″Y″ permits the WebSphere Application Server for z/OS servant runtime (an unauthorized application) to log on to z/OS with a z/OS userid but no z/OS authenticator. This is required to establish a z/OS userid via the verification of a WebSphere Application Server for z/OS login token verification or Trust Association Intercepter.

Note: Easing some traditional z/OS system restrictions places additional responsibility on the WebSphere Application Server for z/OS administrator to ensure that installed applications do not contain malicious code. You can use Java 2 security to minimize this exposure.

Use APPL Profile to restrict access to WebSphere Application Server

If you activated the APPL class in your installation, specify ″Y″ to create a profile APPL class to represent WebSphere Application Server for z/OS for this security domain.

If you specified a security domain identifier, it is used as the APPL profile name. Otherwise, a profile name of ″CBS390″ is created. The Dialog gives universal access to this profile as well as permitting the WebSphere unauthenticated group for the security domain to this profile. If your installation does not require use of the APPL class, specify ″N″.

Define variables to configure base Application Server node - worksheets

This article presents worksheets that will assist you when completing the WebSphere Application Server for z/OS Customization Dialog. See the related definitions article for definitions of each of the terms.

Note: Any instance of ″(cannot change)″ in the ″Your value″ column indicates that that particular value is displayed for reference purposes only and you can't change it in that particular panel. Either the value is static throughout the Dialog, or you need to go back to a previous task to make the change.

1 System locations (directories, HLQs, etc)

System Locations (1 of 2)

This panel asks you for information about your base operating system and HFS-resident components.

Table 14. System Locations (1 of 2)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
System name	(System on which the Customization Dialog is running)	
Sysplex name	(Sysplex on which the cutsomization Dialog is running)	
PROCLIB	SYS1.PROCLIB	
PARMLIB	SYS1.PARMLIB	
SYSEXEC	(blank)	
SCEERUN	CEE.SCEERUN	
		In link list or LPA?
SCEERUN2	CEE.SCEERUN2	
		In link list or LPA?
SBBOLOAD	BBO.SBBOLOAD	
		In link list or LPA?

Table 14. System Locations (1 of 2) (continued)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
SBBOLD2	BBO.SBBOLD2	
		In link list or LPA?
SGSKLOAD	SYS1.GSK.SGSKLOAD	
		In link list or LPA?
SBBOMIG	BBO.SBBOMIG	
		In link list or LPA?
SBBOLPA	BBO.SBBOLPA	
		In link list or LPA?
SBBOEXEC	BBO.SBBOEXEC	
SBBOMSG	BBO.SBBOMSG	

System Locations (2 of 2)

Table 15. System Locations (2 of 2)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Locations of HFS Resident Components		
WebSphere Application Server SMP/E home directory	/usr/lpp/zWebSphere/V5R1M0	
WebSphere JMS Client Java Feature SMP/E home directory	/usr/lpp/mqm/V5R3M1	
Java home directory	/usr/lpp/java/IBM/J1.4	

2 System Environment Customization

5.1 + System Environment Customization (1 of 4)

Table 16. System Environment Customization (1 of 4)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere Application Server for z/OS HFS information		
Mount point	/WebSphere/V5R1M0	
Name	OMVS.WAS.CONFIG.HFS	
Volume, or ’*’ for SMS	*	
Primary allocation in cylinders	250	
Secondary allocation in cylinders	100	

5.1+ System Environment Customization (2 of 4)

Table 17. System Environment Customization (2 of 4)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere Error Log Stream Information		
Name	WAS.ERROR.LOG	
Data class	STANDARD	
Storage class	(null)	
HLQ for data sets	LOGGER	
Is log stream CF resident (Y|N)	Y	
If yes, specify structure name	WAS_STRUCT	
If no, specify:		
log stream size	3000	
staging size	3000	
RRS Log Stream Information		
Group name	(Cell on which the Customization Dialog is running)	
Data class	STANDARD	
Storage class	(null)	
HLQ for data sets	LOGGER	
Is log stream CF resident (Y|N)	Y	
Create RRS PROC (Y|N)	Y	

5.1+ System Environment Customization (3 of 4)

Table 18. System Environment Customization (3 of 4)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
CTRACE Writer Definitions		
Procedure name	BBOWTR	
User ID	STCRACF	
Group	SYS1	
Trace Data Set Information		
Name	SYS1. *systemname.* WAS390.CTRACE	
Volume, or “ *” for SMS	*	
Primary space in cylinders	10	
Secondary space in cylinders	0	

Table 18. System Environment Customization (3 of 4) (continued)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Trace Parmlib member suffix	50	

5.1+ System Environment Customization (4 of 4) 5.1+

Table 19. System Environment Customization (4 of 4)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Logging Details for Transaction XA Partner Log		
Use log stream (Y|N)	N	
Log Stream Information		
Name HLQ	HLQ	
Data class		
Storage class		
HLQ for data sets	IXGLOGR	
Is log stream CF resident (Y|N)	Y	
If yes, specify structure name	WAS_STRUCT2	
If no, specify: log stream size	256	
If no, specify: staging size	256	

3 Server Customization

Server Customization (1 of 4)

Table 20. Server Customization (1 of 4)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Application Server Definitions		
WebSphere Application Server home directory	/WebSphere/V5R1M0/AppServer	
Cell name (short)	*sysplex*	
Cell name (long)	*sysplex*	
Node name (short)	*system*	
Node name (long)	*system*	
Server name (short)	BBOS001	
Server name (long)	server1	
Cluster transition name	BBOC001	

Table 20. Server Customization (1 of 4) (continued)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Admin asynch operations procedure name	BBOW5SH	

Server Customization (2 of 4)

Table 21. Server Customization (2 of 4)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Application Server Definitions		
Controller Information		
Jobname	BBOS001	(cannot change)
Procedure name	BBO5ACR	
User ID	ASCR1	
UID	2431	
Servant Information		
Jobname	BBOS001S	(cannot change)
Procedure name	BBO5ASR	
User ID	ASSR1	
UID	2432	

Server Customization (3 of 4)

Table 22. Server Customization (3 of 4)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Application Server Definitions		
Node host name	(null)	
SOAP JMX Connector port	8880	
DRS Client Address port	7873	
ORB Listener host name	*	
ORB port	2809	
ORB SSL port	0	
HTTP transport host name	*	
HTTP port	9080	
HTTP SSL port	9443	

Server Customization (4 of 4)

Table 23. Server Customization (4 of 4)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Location Service Daemon Definitions		
Daemon home directory	/WebSphere/V5R1M0/Daemon	(cannot change)
Daemon job name	BBODMNB	
Procedure name	BBO5DMN	
User ID	WSDMNCR1	
UID	2411	
IP name	(null)	
Port	5655	
SSL Port	5656	
Register daemon with WLM DNS	N	

4 View Security Domain Configuration Panels

This panel displays values you previously set in the ″Configure security domain″ option--you can not change any of the values here. If you do wish to make changes, you must go back to the main Dialog panel and run through the ″Configure security domain″ option again.

Define variables to configure base Application Server node - definitions

This article lists definitions for the terms you will come across in the WebSphere Application Server for z/OS Customization Dialog.

Note: Some term definitions--those that you cannot set in the related panel--are not displayed. To see the definition, you must to go to the Information Center article for the panel on which you **can** change that particular term.

The WebSphere Application Server for z/OS run-time requires four base cell servers: Application Server, deployment manager, node agent, location service daemon. The panels corresponding to the following tables set up the names, network configuration, start procedures, and user IDs for a base server.

Recommendation: Use the IBM default names the first time you install WebSphere Application Server for z/OS to make the installation instructions easier to follow.

For identification, each controller and servant start procedure must have a user ID and will be defined in the STARTED class.

1 System locations (directories, HLQs, etc)

System Locations (1 of 2)

System name

The system name for the target z/OS system on which WebSphere Application Server for z/OS is installed.

Sysplex name

The sysplex name for the target z/OS system on which WebSphere Application Server for z/OS is installed.

Tip: If you are not sure what the system name (&SYSNAME) and sysplex name (&SYSPLEX) are, use the console command `D SYMBOLS` on the target z/OS system to display them.

For the following, specify the fully qualified data set names without quotes.

Rule: You can specify up to 42 characters for the data set names.

PROCLIB
An existing procedure library where the WebSphere Application Server for z/OS cataloged procedures are added.

PARMLIB
An existing parameter library for system definitions to support WebSphere Application Server for z/OS. This data set must be in the parmlib concatenation for the target z/OS system.

SYSEXEC
A variable-block (RECFM=VB, LRECL=255) data set into which the customization process places REXX EXECs to be called from TSO, such as the WebSphere Application Server for z/OS error log browser, BBORBLOG. You must allocate this data set and concatenate it as part of the SYSEXEC DD allocation in your installation-wide TSO logon PROC or allocation exec.

If your existing SYSEXEC DD data set concatenation consists of fixed-blocked (RECFM=FB) data sets, you must make a copy of the hlq.DATA data set (produced by the Customization Dialog) after the customization process is complete, and place the copy in the SYSEXEC concatenation.

If you do not specify a data set name, the customization process does not place any REXX EXECs in any data set.

Specify the following Language Environment and WebSphere Application Server for z/OS data sets and whether they are (″Y″) or are not (″N″) in the link list or the link pack area (LPA). ″N″ indicates the generated JCL will contain STEPLIB statements for these data sets. Refer to your SMP/E installation for the location of these data sets listed by their DD Name.

Note: If you run multiple versions of WebSphere Application Server for z/OS on your system, specify ″N″ for each of the following in order to generate the proper //STEPLIB DD cards for all the appropriate jobs and procs.

SCEERUN
A Language Environment run-time load module library. Access is required. If it is not in your link list, concatenate it in the STEPLIB DD statement. You need to stop and restart your queue manager to do this.

SCEERUN2
A Language Environment run-time load module library. Access is required. If it is not in your link list, concatenate it in the STEPLIB DD statement. You need to stop and restart your queue manager to do this.

SBBOLOAD
WebSphere Application Server for z/OS load module library that you installed through SMP/E. It has members that should go into the link list or LPA.

SBBOLD2
WebSphere Application Server for z/OS load module library that you installed through SMP/E. It has members that should go into the link list. **DO NOT** place them in LPA.

SBBOMIG
WebSphere Application Server for z/OS IPCS data set that you installed through SMP/E. It is not used during normal operations, but for dump formatting in IPCS only. **DO NOT** place them in LPA.

SBBOLPA
WebSphere Application Server for z/OS data set that you installed through SMP/E.

SGSKLOAD
DLLs that are stored in a partitioned dataset (PDS) when System SSL is SMP/E-installed. The DLLS are not automatically added to the link list or LPA, so contact your system programmer to determine whether or not the System SSL DLLs are in link list or LPA and, if they are not, the high level qualifier of the PDS that contains the System SSL DLLs.

Specify the following WebSphere Application Server for z/OS libraries so they can be accessed by the customized job streams the Dialog produces. These data sets must be cataloged.

SBBOEXEC
WebSphere Application Server for z/OS variable length file distribution PDS you installed through SMP/E.

SBBOMSG
SBBOMSG WebSphere Application Server for z/OS message skeletons for language translation you installed through SMP/E.

System Locations (2 of 2)

Locations of HFS Resident Components:

WebSphere Application Server SMP/E home directory
The name of the directory where WebSphere Application Server for z/OS files reside after SMP/E installation.

WebSphere JMS Client Java Feature SMP/E home directory
The name of the directory where the JMS client Java feature files reside after SMP/E installation.

Java home directory
The name of the directory where the Java SDK files reside after SMP/E installation.

2 System Environment Customization

System Environment Customization (1 of 4)

WebSphere Application Server Configuration HFS Information

Mount point
Read/write HFS directory mount point where application data and environment files are written. The customization process creates this mount point, if it did not already exist.

Name Hierarchical File System data set mounted at the above mount point.

Rule: You can specify up to 42 characters for the data set name.

Volume, or '*' for SMS
Specify either the DASD volume serial number containing the above data set or " *" to let SMS select a volume. Using " *" requires that SMS automatic class selection (ACS) routines be in place to select the volume. If you do not have SMS set up to handle data set allocation automatically, list the volume explicitly.

Primary allocation in cylinders
Initial size allocation in cylinders for the above data set.

Recommendation: The minimum suggested size is 250 cylinders (3390).

Secondary allocation in cylinders
Size of each secondary extent in cylinders.

Recommendation: The minimum suggested size is 100 cylinders.

System Environment Customization (2 of 4)

WebSphere Error Log Stream Information

Note: Having the error log in the log stream is optional. By default, the error log is written to JES SYSOUT.

Name Name of your WebSphere error log stream that is created.

Rules:
- The name must be 26 or fewer characters.
- Do NOT put quotes around the name.

Data class

An existing DFSMS data class for the log stream data set allocation. Applicable only if you use SMS.

Storage class

An existing DFSMS storage class for allocation of the DASD staging data set for this log stream. Applicable only if you use SMS.

HLQ for data sets

The high-level qualifier for your log stream data set name and staging data set name that is created.

Is log stream CF resident (Y|N)

If you want the log stream to be created on a coupling facility, specify “ Y”. If on DASD, specify “ N”.

If yes, specify structure name

If using the coupling facility, specify the coupling facility structure to be used for the log stream.

Rule: The name can be 1 to 16 characters, including alphanumeric characters, national characters, and an underscore, where the first character is uppercase alphabetic.

If no, specify: Logstream size

Specifies the size, in 4K blocks, of the log stream DASD data sets for the log stream being defined.

If no, specify: staging size

Specifies the size, in 4K blocks, of the DASD staging data set for the log stream being defined.

RRS Log Stream Information

If you do not have the RRS log streams set up, the Customization Dialog will create the jobs you can use to set up the log streams.

Group name

Specify the XCF group name.

Recommendation: Use your cell name.

Data class

Specify an existing DFSMS Data Class for the log stream data set allocation. Applicable only if you use SMS.

Storage class

An existing DFSMS storage class for allocation of the DASD staging data set for this log stream. Applicable only if you use SMS.

HLQ for data sets

The high-level qualifier for your log stream data set name and staging data set name.

Is log stream CF resident (Y|N)

If the log stream is to be created on a coupling facility, specify “ Y”. If on DASD, specify “ N”.

Create RRS PROC (Y|N)

If you answer “ Y”, the Dialog copies the ATRRRS cataloged procedure into SYS1.PROCLIB so that RRS can be started.

If you already have RRS set up, specify “ N”.

System Environment Customization (3 of 4)

WebSphere Application Server for z/OS uses component trace (CTRACE) to capture and to display trace data in trace data sets. WebSphere Application Server for z/OS identifies itself to CTRACE with the ″cell short name.″

CTRACE Writer Definitions

Procedure name

This is the CTRACE external writer start procedure to be created. It is identified in the WebSphere Application Server for z/OS CTRACE member (CTIBBOxx) in PARMLIB.

Rule: The name must be 1 to 7 characters.

User ID
RACF user ID to be created and associated with the CTRACE external writer start procedure.

Group RACF group name to be created and associated with this user.

Trace Data Set Information

Name Specify a fully qualified data set name, such as WAS390.CTRACE1, for the data set to be created. The default includes the system name of the system on which the Customization Dialog is running.

Rules:

- You can specify up to 42 characters for the data set name.
- Do not use quotes.

Volume, or "*" for SMS
Specify either the DASD volume serial number containing the above data set or " *" to let DFSMShsm select a volume. Using "*" requires SMS. Using "*" requires that SMS automatic class selection (ACS) routines be in place to select the volume. If you do not have SMS set up to handle data set allocation automatically, list the volume explicitly.

Primary space in cylinders
The primary space for the trace data set.

Secondary space in cylinders
The secondary space for the trace data set.

Trace Parmlib member suffix
Value that is appended to CTIBBO to form the member name for the Trace Parmlib member.

System Environment Customization (4 of 4)

Logging Details for Transaction XA Partner Log

Use Log stream (Y|N)
Indicates whether the XA partner log is located in the file system or in a logstream. Choose "N" to put the partner log in the file system.

Note: If you choose "Y" to use logger, you must continue with the rest of the variables on this panel. However, if you choose "N" to put the partner log in the file system, no further modification is required on this panel, and the XA logs will be written to the file system at the default location (/install root/AppServer/recoveryLogs), or at the value of $TRANLOG_ROOT.

Log Stream Information

Name HLQ
Name that becomes the first qualifier in the name of the log stream.

Note: No two servers may have the same log stream name. If you plan for your installation to have more than one server with the same short name, you must choose a different HLQ for each server.

The remainder of the logstream name consists of the server short name, followed by an M or D (short for Main and Delayed streams). For example, if the HLQ is `HLQ` and server name is `BBOS001`, the stream names would be `HLQ.BBOS001.M` and `HLQ.BBOS001.D`. As stated above, it is important that you select the HLQ carefully to ensure that two servers are not using the same stream. If you have two servers named `BBOS001` in two cells, the streams will overlap if they both use the same HLQ.**Rule:** The name can be 1 to 8 characters.

Data class
An existing DFSMS data class for the log stream data set allocation. Specify this if your installation requires you to allocate log streams in a particular data class. Applicable only if you use SMS.

Storage class

An existing DFSMS storage class for allocation of the DASD staging data set for this log stream. Specify this if your installation requires you to allocate log streams in a particular storage class. Applicable only if you use SMS.

HLQ for data sets

High-level qualifier that becomes both the HLQ for the DASD offload data sets produced when the log stream becomes full and the HLQ for the log stream data sets themselves, if they are using a DASD-only log stream.

Rule: The HLQ can be 1 to 8 characters.

Is log stream CF resident (Y|N)

If you want to create the log stream on a coupling facility, specify “ Y”. If on DASD, specify “ N”.

If yes, specify structure name

If using the coupling facility (CF), specify the coupling facility structure to be used for the log stream.

Rule: The name can be 1 to 16 characters, including alphanumeric characters, national characters, and an underscore, where the first character is uppercase alphabetic.

If no, specify: logstream size

Specifies the size, in 4K blocks, of the log stream DASD data sets for the log stream being defined. Specifying “1” indicates 4096 bytes of storage. It is recommended that you make the DASD logstream the same size as your RRS (Resource Recovery Services) main and delayed logstreams.

staging size

Specifies the size, in 4K blocks, of the DASD staging data set for the log stream being defined. Specifying “1” indicates 4096 bytes of storage.

3 Server Customization

Server Customization (1 of 4)

Application Server Definitions

Rule: In the following, unless specified otherwise, names must be 8 or fewer characters.

WebSphere Application Server home directory

Directory in which the Application Server configuration resides.

Note: You can alter only the portion after ″/WebSphere/V5R0M0.″

Cell name (short)

Parameter passed to the server’s start procedures that specifies the location of the cell’s configuration files and identifies the cell to certain WebSphere Application Server for z/OS-exploited z/OS facilities (for example, SAF).

Rules:

- Name must be 8 or fewer characters and all uppercase.
- Name must be unique among all other cells in the sysplex.

Cell name (long)

Primary external identification of this WebSphere Application Server for z/OS cell. This name identifies the cell as displayed through the administrative console and also appears as part of the directory path under the WebSphere Application Server for z/OS home’s configuration directory.

Note: If you intend to ever add this base server node to a deployment manager cell, ensure that the base server cell name is different than the deployment manager cell name.

Rules:

- Name must be 60 or fewer characters and can be of mixed case.
- Name must be unique among all other cells in the sysplex.

Node name (short)

Parameter passed to the server's start procedures that specifies the location of the node's configuration files and identifies the node to certain WebSphere Application Server for z/OS-exploited z/OS facilities (for example, SAF).

Rules:

- Name must be 8 or fewer characters and all uppercase.
- Name must be unique within the cell.

Node name (long)

Primary external identification of this WebSphere Application Server for z/OS node. This name identifies the node as displayed through the administrative console and also appears as part of the directory path under the WebSphere Application Server for z/OS home's configuration directory.

Note: If you intend to ever add this base server node to a deployment manager cell, ensure that all the nodes in the deployment manager cell are unique.

Rules:

- Name must be 60 or fewer characters and can be of mixed case.
- Name must be unique within the cell.
- The Application Server must be defined on its own node; no other server may exist on the same node as the Application Server.

Server name (short)

Name of the Application Server. This is the server's job name, as specified in the MVS START command JOBNAME parameter. This value is also passed as a parameter to the server's start procedures to specify the location of the server's configuration files and identify the server to certain WebSphere Application Server for z/OS- exploited z/OS facilities (for example, SAF).

Note: The server short name is used as the server JOBNAME.

Rule: Name must usually contain 7 or fewer all-uppercase characters. If you wish to instead specify 8 characters, you must follow the steps outlined in the Information Center article ″Converting a 7-character jobname to 8 characters.″

Server name (long)

Name of the Application Server and the primary external identification of this WebSphere Application Server for z/OS server. This name identifies the server as displayed through the administrative console and also appears as part of the directory path under the WebSphere Application Server for z/OS home's configuration directory.

Rule: Name must be 60 or fewer characters and can be of mixed case.

Cluster transition name

WLM APPLENV (WLM application environment) name for this server.

Note: If this server is converted into a clustered server, this name becomes the cluster short name. The cluster short name is the WLM APPLENV name for all servers that are part of the same cluster.

Rule: Name must be 8 or fewer characters and all uppercase.

Admin asynch operations procedure name

Specifies the JCL procedure name of a started task that is launched via the START command by node agents and/or Application Servers to perform certain asynchronous administrative operations (such as node synchronization) and add and remove a node.

Server Customization (2 of 4)

Application Server Definitions

Rule: In the following, unless specified otherwise, names must be 8 or fewer characters.

Note: If you wish to specify jobnames with 8 instead of 7 characters, you must follow the steps outlined in the Information Center article ″Converting a 7-character jobname to 8 characters.″

Controller Information

Jobname
The jobname, specified in the MVS START command JOBNAME parameter, associated with the Application Server controller.

Procedure name
Name of member in your procedure library to start the Application Server controller.

Rule: Name must be 7 or fewer characters.

User ID
The user ID associated with the Application Server controller.

Note: If you are using a non-IBM security system, the user ID may have to match the procedure name. Please refer to your security system's documentation.

UID The user identifier associated with this user ID.

Rule: UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

Servant Information

Jobname
The job name specified in the IWMSSNM parameter of the WLM Application Environment for the server.

Procedure name
Name of member in your procedure library to start the Application Server servant.

Rule: Name must be 7 or fewer characters.

User ID
The user ID associated with the Application Server servant.

Note: If you are using a non-IBM security system, the user ID may have to match the procedure name. Please refer to your security system's documentation.

UID The user identifier associated with this user ID.

Rule: UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

Server Customization (3 of 4)

Application Server Definitions

Rule: In the following, unless specified otherwise, names must be 8 or fewer characters.

Node host name
IP name or address of the system on which the server is configured. This value is used by other WebSphere Application Server for z/OS functions to connect to this server.

Note: The node host name must always resolve to an IP stack on the system where the Application Server runs. The node host name cannot be a DVIPA or a DNS name that, in any other way, causes the direction of requests to more than one system.

SOAP JMX Connector port
Port number for the JMX HTTP connection to this server based on the SOAP protocol. JMX is used for remote administrative functions, such as invoking scripts through wsadmin.sh.

Rule: Value cannot be 0.

DRS Client Address port
Port address for access to the server's data replication service. This is important for configurations that define replication groups.

Rule: Value cannot be 0.

ORB Listener host name
IP address on which the server's ORB listens for incoming IIOP requests. The default is ″*″, which instructs the ORB to listen on all available IP addresses.

ORB port
Port for IIOP requests which acts as the bootstrap port for this server and also as the port through which the ORB accepts IIOP requests.

Rule: Value cannot be 0.

ORB SSL port
Port for secure IIOP requests. The default is ″0″, which allows the system to choose this port.

HTTP transport host name
IP address on which the server's Web container should listen for incoming HTTP requests. The default is ″*″, which instructs the Web container to listen on all available IP addresses.

Note: The ″transport host name″ becomes the ″hostname″ in the virtualhosts.xml file, which makes setting a specific IP address here less than ideal because, if you do so, you are restricting yourself to that IP address until you go into the administrative console and add another virtual host.

HTTP port
Port for HTTP requests.

Rule: Value cannot be 0.

HTTP SSL port
Port for secure HTTP requests.

Rule: Value cannot be 0.

Server Customization (4 of 4)

Location Service Daemon Definitions

Rule: In the following, unless specified otherwise, names must be 8 or fewer characters.

The location service daemon is the initial point of client contact in WebSphere Application Server for z/OS. The server contains the CORBA-based location service agent, which places sessions in a cell. All RMI/IIOP IORs (for example, for enterprise beans) establish connections to the location service daemon first, then forward them to the target Application Server.

Daemon home directory
Directory in which the location service daemon resides.

Daemon job name
Specifies the jobname of the location service daemon, specified in the JOBNAME parameter of the MVS start command used to start the location service daemon.

Caution: When configuring a second cell, ensure you change the daemon job name from the default or value you used for the first cell.

Note: A server automatically starts the location service daemon if it is not already running.

Procedure name
Name of the member in your procedure library to start the location service daemon.

Rule: Name must be 7 or fewer characters.

User ID
The user ID associated with the location service daemon.

UID The user identifier associated with this user ID.

Rule: UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

IP Name
The fully qualified IP name, registered with the Domain Name Server (DNS), that the location service daemon uses. The default is your node host name.

Note: In a sysplex, you should consider using a virtual IP address (VIPA) for the location service daemon IP name.

Port The port number on which the location service daemon listens.

SSL Port
The port number on which the location service daemon listens for SSL connections.

Note: Select the IP name and port number for the location service daemon carefully. You can choose any name you want, but, once chosen, it is difficult to change, even in the middle of customization.

Register daemon with WLM DNS
If you use the WLM DNS (connection optimization), you must select ″Y″ to register your location service daemon with it. Otherwise, select ″N.″

4 View Security Domain Configuration Panels

This panel displays values you previously set in the ″Configure security domain″ option--you can not change any of the values here. If you do wish to make changes, you must go back to the main Dialog panel and run through the ″Configure security domain″ option again.

Define variables to configure integral JMS provider - worksheets

This article presents worksheets that will assist you when completing the WebSphere Application Server for z/OS Customization Dialog. See the related definitions article for definitions of each of the terms.

Note: Any instance of ″(cannot change)″ in the ″Your value″ column indicates that that particular value is displayed for reference purposes only and you cannot change it in that particular panel. Either the value is static throughout the Dialog, or you need to go back to a previous task to make the change.

1 System locations (directories, HLQs, etc)

System Locations (1 of 2)

This panel asks you for information about your base operating system, HFS-resident components, and MQ subsystem.

Table 24. System Locations (1 of 2)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
System name	(System on which the Customization Dialog is running)	
Sysplex name	(Sysplex on which the customization Dialog is running)	
Jes3 (Y/N)	N	
Full Names of Data Sets		
MACLIB	SYS1.MACLIB	
PROCLIB	SYS1.PROCLIB	
PARMLIB	SYS1.PARMLIB	
SCSQAUTH	CSQ531.SCSQAUTH	
		In link list or LPA?

Table 24. System Locations (1 of 2) (continued)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
SCSQANLx	CSQ531.SCSQANLE	
		In link list or LPA?
SCSQLINK	CSQ531.SCSQLINK	
		In link list or LPA?
SCSQMVR1	CSQ531.SCSQMVR1	
		In link list or LPA?
SCEERUN	CEE.SCEERUN	
		In link list or LPA?
SCEERUN2	CEE.SCEERUN2	
		In link list or LPA?
USER LOAD	CSQ531.CSQZPARM	
		In link list or LPA?
SCSQLOAD	CSQ531.SCSQLOAD	
SCSQMACS	CSQ531.SCSQMACS	
SCSQPROC	CSQ531.SCSQPROC	
SCSQSNLx	CSQ531.SCSQSNLE	

System Locations (2 of 2)

Table 25. System Locations (2 of 2)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
High Level Qualifier of IJP Operational Datasets		
CSQHLQ	CSQ531	
Integral Provider Output Volumes		
BSDS Vol1	(blank)	
BSDS Vol2	(blank)	
LogCopy 1 Vol1	(blank)	
LogCopy 1 Vol2	(blank)	
LogCopy 2 Vol1	(blank)	
LogCopy 2 Vol2	(blank)	
Integral Provider Page Volumes		
PageSet Vol0	(blank)	
PageSet Vol1	(blank)	
PageSet Vol2	(blank)	

Table 25. System Locations (2 of 2) (continued)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
PageSet Vol3	(blank)	
PageSet Vol4	(blank)	
PageSet Vol5	(blank)	
PageSet Vol6	(blank)	
PageSet Vol7	(blank)	
PageSet Vol8	(blank)	

2 Server Customization

These panels allow you to define values for your Integral JMS Provider. The values you initially set for your base Application Server and location service daemon are displayed for reference purposes only. If you wish to change them, you need to go back and reconfigure your base Application Server.

Recommendation: Use the IBM default names the first time you install WebSphere Application Server for z/OS to make the installation instructions easier to follow.

For identification, each controller and servant start procedure must have a user ID and will be defined in the STARTED class.

Server Customization (1 of 2)

Table 26. Server Customization (1 of 2)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere HFS Information		
Mount point	/WebSphere/V5R0M0	(cannot change)
Application Server Definitions		
WebSphere Application Server home directory	/WebSphere/V5R0M0/AppServer	(cannot change)
Cell name (short)	*sysplex*	(cannot change)
Cell name (long)	*sysplex*	(cannot change)
Node name (short)	*system*	(cannot change)
Node name (long)	*system*	(cannot change)
Server name (short)	BBOS001	(cannot change)
Server name (long)	server1	(cannot change)
Integral JMS Definitions		
JMS server name (short)	WMQX	
Command prefix	+	

Server Customization (2 of 2)

Table 27. Server Customization (2 of 2)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Integral JMS Server Parameters		
Archive retention (days)	9999	
Stat Interval Time (secs)	30	
JMS Server Queued Address port	5558	
JMS Server Direct Address port	5559	
JMS Server Security port	5557	

3 Security Customization

This panel allows you to specify authentication and authorization options for your run-time resources.

Security Customization (1 of 1)

Table 28. Security Customization (1 of 1)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Queue Manager Security		
Default user ID for commands	(blank)	
Resource auditing (Y/N)	Y	

Define variables to configure integral JMS provider - definitions

This article lists definitions for the terms you will come across in the WebSphere Application Server for z/OS Customization Dialog.

Note: Some term definitions--those that you cannot set in the related panel--are not displayed. To see the definition, you must to go to the Information Center article for the panel on which you **can** change that particular term.

1 System locations (directories, HLQs, etc)

System Locations (1 of 2)

System name

The system name, designated in your base cell customization, for the target z/OS system on which WebSphere Application Server for z/OS is installed.

Sysplex name

The sysplex name, designated in your base cell customization, for the target z/OS system on which WebSphere Application Server for z/OS is installed.

Tip: If you are not sure what the system name (&SYSNAME) and sysplex name (&SYSPLEX) are, use the console command `D SYMBOLS` on the target z/OS system to display them.

Jes3 (Y/N)

Indicate ″Y″ if you are using JES 3 or ″N″ if you are using JES 2.

Full Names of Data Sets

For the following, specify the fully qualified data set names without quotes.

Rule: You can specify up to 42 characters for the data set names.

MACLIB
The data set that contains the macro definitions for the system macro instructions that are provided by IBM.

PROCLIB
An existing procedure library where the WebSphere Application Server for z/OS cataloged procedures are added.

PARMLIB
An existing parameter library for system definitions to support WebSphere Application Server for z/OS. This data set must be in the parmlib concatenation for the target z/OS system.

Specify the following Language Environment and WebSphere Application Server for z/OS data sets and whether they are (″Y″) or are not (″N″) in the link list or the link pack area (LPA). ″N″ indicates the generated JCL will contain STEPLIB statements for these data sets. Refer to your SMP/E installation for the location of these data sets listed by their DD Name.

SCSQAUTH
The main repository for all MQSeries product load modules. It also contains the default parameter modules CSQZPARM and CSQXPARM. Ensure this library is APF-authorized.

Note: It is recommended that you place this library in LNKLST rather than the LPA because you can update LNKLST after maintenance by issuing the z/OS command `REFRESH LLA`. If you do decide to place it in the LPA, you need to manually copy any changed modules into the LPA whenever you apply maintenance.

SCSQANLx
Contains the load modules for various versions of MQSeries. The different versions, designated by letters in place of the ″x″, are U.S. English: mixed case (″E″), U.S. English: uppercase (″U″), Simplified Chinese (″C″), and Japanese (″K″).

Note: It is recommended that you place this library in LNKLST rather than the LPA because you can update LNKLST after maintenance by issuing the z/OS command `REFRESH LLA`. If you do decide to place it in the LPA, you need to manually copy any changed modules into the LPA whenever you apply maintenance.

SCSQLINK
The early code library. Contains the load modules that must reside in the link list because they are loaded at system initial program load (IPL). Ensure this library is APF-authorized and in the link list.

Note: It is recommended that you place this library in the LPA.

SCSQMVR1
Contains the load modules for distributed queuing when using LU 6.2 or TCP/IP with either the OpenEdition sockets or IUCV interface. Ensure this library is APF-authorized.

Note: It is recommended that you place this library in LNKLST rather than the LPA because you can update LNKLST after maintenance by issuing the z/OS command `REFRESH LLA`. If you do decide to place it in the LPA, you need to manually copy any changed modules into the LPA whenever you apply maintenance.

SCEERUN
A Language Environment run-time load module library. Access is required. If it is not in your link list, concatenate it in the STEPLIB DD statement. You need to stop and restart your queue manager to do this.

SCEERUN2
A Language Environment run-time load module library. Access is required. If it is not in your link list, concatenate it in the STEPLIB DD statement. You need to stop and restart your queue manager to do this.

USER LOAD
System parameter module that controls the logging, archiving, tracing, and connection environments that MQSeries uses in its operation. You need to create this linkedit target data set from scratch, using another data set as a template.

SCSQLOAD
The load library. Contains load modules for non-APF code, user exits, utilities, samples, installation verification programs, and adapter stubs. The library does not need APF-authorization nor must it reside in the link list.

SCSQMACS
Contains Assembler macros including sample macros, product macros, and system parameter macros.

SCSQPROC
Contains sample JCL and default system initialization data sets.

SCSQSNLx
Contains the load modules for various versions of the MQSeries modules that are required for special purpose function (for example, the early code). The different versions, designated by letters in place of the ″x″, are U.S. English: mixed case (″E″), U.S. English: uppercase (″U″), Simplified Chinese (″C″), and Japanese (″K″).

Note: It is recommended that you place this library in the LPA.

System Locations (2 of 2)

Each MQSeries subsystem in WebSphere Application Server for z/OS contains two bootstrap data sets (BSDSs), two sets of log data sets, and nine page data sets. Along with the volume 0 page data set, which is fixed and untailorable, there are eight other page data sets that you can organize to suit your needs.

Note: You must define all the BSDS, LogCopy, and PageSet volumes, but you can choose to make the values different or all the same. You might specify the same value for all the volumes if, for example, you run only a test system, but the system performs better if they are spread out.

High Level Qualifier of IJP Operational Data sets:

CSQHLQ
The high level qualifier that you create for your IJP operational data sets.

Integral Provider Output Volumes

BSDS Vol1
The bootstrap data set volume 1.

BSDS Vol2
The bootstrap data set volume 2.

LogCopy 1 Vol1
Volume 1 of copy 1 of the log data set.

LogCopy 1 Vol2
Volume 2 of copy 1 of the log data set.

LogCopy 2 Vol1
Volume 1 of copy 2 of the log data set.

LogCopy 2 Vol2
Volume 2 of copy 2 of the log data set.

Integral Provider Page Volumes

PageSet Vol0
The volume 0 page data set, which is fixed and untailorable.

PageSet Vol1 - PageSet Vol8
Eight volumes of page data sets. You must define them all, but you can choose to make the values different or all the same.

2 Server Customization

Server Customization (1 of 2)

Note: The Application Server information is provided for reference purposes only, so the definitions will not appear here.

WebSphere HFS Information

Mount point
Read/write HFS directory mount point where application data and environment files are written. The customization process creates this mount point, if it did not already exist.

Application Server Definitions

Rule: In the following, unless specified otherwise, names must be 8 or fewer characters.

WebSphere Application Server home directory
Directory in which the Application Server configuration resides.

Note: You can alter only the portion after ″/WebSphere/V5R0M0.″

Cell name (short)
Parameter passed to the server’s start procedures that specifies the location of the cell’s configuration files and identifies the cell to certain WebSphere Application Server for z/OS-exploited z/OS facilities (for example, SAF).

Rules:

- Name must be 8 or fewer characters and all uppercase.
- Name must be unique among all other cells in the sysplex.

Cell name (long)
Primary external identification of this WebSphere Application Server for z/OS cell. This name identifies the cell as displayed through the administrative console and also appears as part of the directory path under the WebSphere Application Server for z/OS home’s configuration directory.

Note: If you intend to ever add this base server node to a deployment manager cell, ensure that the base server cell name is different than the deployment manager cell name.

Rules:

- Name must be 60 or fewer characters and can be of mixed case.
- Name must be unique among all other cells in the sysplex.

Node name (short)
Parameter passed to the server’s start procedures that specifies the location of the node’s configuration files and identifies the node to certain WebSphere Application Server for z/OS-exploited z/OS facilities (for example, SAF).

Rules:

- Name must be 8 or fewer characters and all uppercase.
- Name must be unique within the cell.

Node name (long)
Primary external identification of this WebSphere Application Server for z/OS node. This name identifies the node as displayed through the administrative console and also appears as part of the directory path under the WebSphere Application Server for z/OS home’s configuration directory.

Note: If you intend to ever add this base server node to a deployment manager cell, ensure that all the nodes in the deployment manager cell are unique.

Rules:

- Name must be 60 or fewer characters and can be of mixed case.
- Name must be unique within the cell.
- The Application Server must be defined on its own node; no other server may exist on the same node as the Application Server.

Server name (short)

Name of the Application Server. This is the server's job name, as specified in the MVS START command JOBNAME parameter. This value is also passed as a parameter to the server's start procedures to specify the location of the server's configuration files and identify the server to certain WebSphere Application Server for z/OS- exploited z/OS facilities (for example, SAF).

Note: The server short name is used as the server JOBNAME.

Rule: Name must usually contain 7 or fewer all-uppercase characters. If you wish to instead specify 8 characters, you must follow the steps outlined in the Information Center article ″Converting a 7-character jobname to 8 characters.″

Server name (long)

Name of the Application Server and the primary external identification of this WebSphere Application Server for z/OS server. This name identifies the server as displayed through the administrative console and also appears as part of the directory path under the WebSphere Application Server for z/OS home's configuration directory.

Rule: Name must be 60 or fewer characters and can be of mixed case.

Integral JMS Definitions

JMS server name (short)

MQSeries subsystem base server name. This value defines the Integral JMS Queue Manager as a z/OS subsystem and forms the name of the Integral JMS Broker.

Rule: The JMS Server name must contain only 4 characters.

Command prefix

Character string value that WebSphere System Manager uses, in conjunction with the JMS Server short name, to form system commands that direct both the operation and administration of the Integral JMS Queue Manager.

Server Customization (2 of 2)

Integral JMS Server Parameters

Archive retention (days)

Specifies the retention period, in days, set when you create the archive log data set. The default is 9999.

Rule: The parameter must be in the range 0 through 9999.

Stat Interval Time (secs)

The interval, in seconds, between listener restart attempts. The default is 60.

JMS Server Queued Address port

Specifies the host and port number used to configure the WebSphere JMS Provider topic connection factory. The JMS Server Queued Address port is the listener port used for full-function JMS-compliant, publish/subscribe support. The default Queued Address port number is 5558.

Since the queue manager and queue broker for the WebSphere JMS Provider are configured outside the administrative console, changes to this endpoint require corresponding configuration changes to the queue manager and queue broker.

JMS Server Direct Address port

Specifies the host and port number used to configure the WebSphere JMS Provider topic connection factory. The JMS Server Direct Address port is the listener port used for direct TCP/IP connection (non-transactional, nonpersistent, and non-durable subscriptions only) for publish/subscribe support. The default Direct Address port number is 5559.

Since the queue manager and queue broker for the WebSphere JMS Provider are configured outside the administrative console, changes to this endpoint require corresponding configuration changes to the queue manager and queue broker.

JMS Server Security port

3 Security Customization

For more information about security, see related security sections.

Security Customization (1 of 1)

Queue Manager Security

Default user ID for commands

Specifies the default user ID used for command security checks. Ensure this user ID is defined to the ESM.

Rule: The name must be between 1 and 8 alphanumeric characters and start with a letter.

In the following, specifying "Y" (yes) tells the Dialog to define the profile or enable an option in RACF. Specifying " N" (no) tells the Dialog to not define the profile or enable the option.

Resource auditing (Y/N)

If you want to enable the creation of an audit trail, specify ″Y″. Otherwise, specify ″N″. Creating an audit trail may impact performance.

Define variables to configure deployment manager node - worksheets

This article presents worksheets that will assist you when completing the WebSphere Application Server for z/OS Customization Dialog. See the related definitions article for definitions of each of the terms.

Note: Any instance of ″(cannot change)″ in the ″Your value″ column indicates that that particular value is displayed for reference purposes only and you cannot change it in that particular panel. Either the value is static throughout the Dialog, or you need to go back to a previous task to make the change.

1 System locations (directories, HLQs, etc)

This panel asks you for information about your base operating system and HFS-resident components.

System Locations (1 of 2)

Table 29. System Locations (1 of 2)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
System name	(System on which the Customization Dialog is running)	
Sysplex name	(Sysplex on which the Customization Dialog is running)	
Full Names of Data Sets		
PROCLIB	SYS1.PROCLIB	
PARMLIB	SYS1.PARMLIB	
SYSEXEC	(blank)	

Table 29. System Locations (1 of 2) (continued)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
SCEERUN	CEE.SCEERUN	
		In link list or LPA?
SCEERUN2	CEE.SCEERUN2	
		In link list or LPA?
SBBOLOAD	BBO.SBBOLOAD	
		In link list or LPA?
SBBOLD2	BBO.SBBOLD2	
		In link list or LPA?
SBBOMIG	BBO.SBBOMIG	
		In link list or LPA?
SBBOLPA	BBO.SBBOLPA	
		In link list or LPA?
SGSKLOAD	SYS1.GSK.SGSKLOAD	
		In link list or LPA?
SBBOEXEC	BBO.SBBOEXEC	
SBBOMSG	BBO.SBBOMSG	

System Locations (2 of 2)

Table 30. System Locations (2 of 2)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Locations of HFS Resident Components		
WebSphere Application Server SMP/E home directory	/usr/lpp/zWebSphere/V5R1M0	
WebSphere Application Server JMS Client Java Feature SMP/E home directory	/usr/lpp/mqm/V5R3M1	
Java home directory	/usr/lpp/java/IBM/J1.4	

2 System Environment Customization

System Environment Customization (1 of 1)

Table 31. System Environment Customization (1 of 1)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere HFS Information		

Table 31. System Environment Customization (1 of 1) (continued)

Item	**Value in the Dialog after you load IBM defaults**	**Your value (Fill in the blanks)**
Mount point	/WebSphere/V5R1M0	
Name	OMVS.WAS.CONFIG.HFS	
Volume, or '*' for SMS	*	
Primary allocation in cylinders	250	
Secondary allocation in cylinders	100	

3 Server Customization

Server Customization (1 of 4)

Table 32. Server Customization (1 of 4)

Item	**Value in the Dialog after you load IBM defaults**	**Your value (Fill in the blanks)**
Deployment Manager Definitions		
WebSphere Application Server home directory	/WebSphere/V5R1M0/DeploymentManager	
Cell name (short)	*sysplex*	
Cell name (long)	*sysplex*	
Node name (short)	*system*	
Node name (long)	*system*	
Server name (short)	BBODMGR	
Server name (long)	dmgr	(cannot change)
Cluster transition name	BBODMGR	

Server Customization (2 of 4)

Table 33. Server Customization (2 of 4)

Item	**Value in the Dialog after you load IBM defaults**	**Your value (Fill in the blanks)**
Deployment Manager Definitions		
Controller Information		
Jobname	BBODMGR	(cannot change)
Procedure name	BBO5DCR	
User ID	DMCR1	
UID	2421	
Servant Information		
Jobname	BBODMGRS	(cannot change)

Table 33. Server Customization (2 of 4) (continued)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Procedure name	BBO5DSR	
User ID	DMSR1	
UID	2422	

Server Customization (3 of 4)

Table 34. Server Customization (3 of 4)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Deployment Manager Definitions		
Node host name	(null)	
SOAP JMX connector port	8879	
Cell Discovery Address port	7277	
DRS Client Address port	7989	
ORB Listener host name	*	
ORB port	9809	
ORB SSL port	0	
HTTP transport host name	*	
HTTP port	9090	
HTTP SSL port	9043	

Server Customization (4 of 4)

Table 35. Server Customization (4 of 4)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Location Service Daemon Definitions		
Daemon home directory	/WebSphere/V5R0M0/Daemon	(cannot change)
Daemon job name	BBODMNC	
Procedure name	BBO5DMN	
User ID	WSDMNCR1	
UID	2411	
IP name	(null)	
Port	5755	
SSL port	5756	
Register daemon with WLM DNS	N	

4 View Security Domain Configuration Panels

This panel displays values you previously set in the ″Configure security domain″ option--you can not change any of the values here. If you do wish to make changes, you must go back to the main Dialog panel and run through the ″Configure security domain″ option again.

Define variables to configure deployment manager node - definitions

This article lists definitions for the terms you will come across in the WebSphere Application Server for z/OS Customization Dialog.

Note: Some term definitions--those that you cannot set in the related panel--are not displayed. To see the definition, you must to go to the Information Center article for the panel on which you **can** change that particular term.

1 System locations (directories, HLQs, etc)

These definitions are the same as those for the Application Server, so see that article for help.

2 System Environment Customization

System Environment Customization (1 of 1)

WebSphere HFS Information

Mount point
Read/write HFS directory mount point where application data and environment files are written. The customization process creates this mount point, if it didn't already exist. Specify a different mount point and HFS file system name if you wish your deployment manager to reside in an HFS different from that of your base Application Server.

Name Hierarchical File System data set mounted at the above mount point.

Rule: You can specify up to 42 characters for the data set name.

Volume, or '*' for SMS
Specify either the DASD volume serial number containing the above data set or " *" to let SMS select a volume. Using " *" requires that SMS automatic class selection (ACS) routines be in place to select the volume. If you do not have SMS set up to handle data set allocation automatically, list the volume explicitly.

Primary allocation in cylinders
Initial size allocation in cylinders for the above data set.

Recommendation: The minimum suggested size is 250 cylinders (3390).

Secondary allocation in cylinders
Size of each secondary extent in cylinders.

Recommendation: The minimum suggested size is 100 cylinders.

3 Server Customization

The WebSphere Application Server for z/OS runtime requires four base cell servers: Application Server, deployment manager, node agent, location service daemon. The panels corresponding to the following tables set up the names, network configuration, start procedures, and user IDs for a deployment manager server.

Recommendation: Use the IBM default names the first time you install WebSphere Application Server for z/OS to make the installation instructions easier to follow.

For identification, each controller and servant start procedure must have a user ID and will be defined in the STARTED class.

Servers are started by the MVS START command using the controller start procedure.

Server Customization (1 of 4)

Deployment Manager Definitions

Rule: In the following, unless specified otherwise, names must be 8 or fewer characters.

Note: You need to change the cell and node values so they properly apply to your deployment manager. They must be different from the values used for the base Application Server node.

WebSphere Application Server home directory

Directory in which the deployment manager resides.

Note: You can alter only the portion after ″/WebSphere/V5R0M0.″

Cell name (short)

Parameter passed to the server's start procedures that specifies the location of the cell's configuration files and identifies the cell to certain WebSphere Application Server for z/OS-exploited z/OS facilities (for example, SAF).

Rules:

- Name must be 8 or fewer characters and all uppercase.
- Name must be unique among all other cells in the sysplex.

Note: This value, imported from the base Application Server node configuration, is a temporary value because the base Application Server node's cell goes away when the node is federated into the deployment manager's cell. So, you need to change the cell name to the one you want in the final configuration.

Cell name (long)

Primary external identification of this WebSphere Application Server for z/OS cell. This name identifies the cell as displayed through the administrative console and also appears as part of the directory path under the WebSphere Application Server for z/OS home's configuration directory.

Rules:

- Name must be 60 or fewer characters and can be of mixed case.
- Name must be unique among all other cells in the sysplex.

Note: This value, imported from the base Application Server node configuration, is a temporary value because the base Application Server node's cell goes away when the node is federated into the deployment manager's cell. So, you need to change the cell name to the one you want in the final configuration.

Node name (short)

Parameter passed to the server's start procedures that specifies the location of the node's configuration files and identifies the node to certain WebSphere Application Server for z/OS-exploited z/OS facilities (for example, SAF).

Rules:

- Name must be 8 or fewer characters and all uppercase.
- Name must be unique within the cell.
- The deployment manager must be defined on its own node; no other server may exist on the same node as the deployment manager.

Node name (long)

Primary external identification of this WebSphere Application Server for z/OS node. This name identifies the node as displayed through the administrative console and also appears as part of the directory path under the WebSphere Application Server for z/OS home's configuration directory.

Note: Ensure that the node name is unique (different from those in other deployment manager or base server configurations).

Rules:

- Name must be 60 or fewer characters and can be of mixed case.
- Name must be unique within the cell.
- The deployment manager must be defined on its own node; no other server may exist on the same node as the deployment manager.

Server name (short)

Name of the deployment manager server. This is the server's job name, as specified in the MVS START command JOBNAME parameter. This value is also passed as a parameter to the server's start procedures to specify the location of the server's configuration files and identify the server to certain WebSphere Application Server for z/OS- exploited z/OS facilities (for example, SAF).

Rule: Name must usually contain 7 or fewer all-uppercase characters. If you wish to instead specify 8 characters, you must follow the steps outlined in the Information Center article ″Converting a 7-character jobname to 8 characters.″

Server name (long)

Name of the deployment manager server and the primary external identification of this WebSphere Application Server for z/OS server. This name identifies the server as displayed through the administrative console and also appears as part of the directory path under the WebSphere Application Server for z/OS home's configuration directory.

Rule: Name must be 60 or fewer characters and can be of mixed case.

Cluster transition name

WLM APPLENV name for this server.

Note: The deployment manager is not clusterable, so this value never actually becomes the cluster short name of this server's cluster. However, like an Application Server, the deployment manager still needs an APPLENV, so the cluster transition name is used for this purpose.

Rule: Name must be 8 or fewer characters and all uppercase.

Server Customization (2 of 4)

Rule: In the following, unless specified otherwise, names must be 8 or fewer characters.

Note: The RACF profiles you created during the construction of the base Application Server node will also work for the deployment manager. Also, the STARTED profiles are generic enough to cover any MVS jobname starting with AZ*.

Deployment Manager Definitions

Controller Information

Jobname

The jobname, specified in the MVS START command JOBNAME parameter, associated with the Application Server controller.

Procedure name

Name of member in your procedure library to start the deployment manager controller.

Rule: Name must be 7 or fewer characters.

User ID

The user ID associated with the deployment manager controller. You can use the same user ID values throughout all your configurations.

Note: If you are using a non-IBM security system, the user ID may have to match the procedure name. Please refer to your security system's documentation.

UID The user identifier associated with this user ID.

Rule: UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

Servant Information

Jobname
The job name specified in the IWMSSNM parameter of the WLM Application Environment for the server.

Procedure name
Name of member in your procedure library to start the deployment manager servant.

Rule: Name must be 7 or fewer characters.

User ID
The user ID associated with the deployment manager servant. You can use the same user ID values throughout all your configurations.

Note: If you are using a non-IBM security system, the user ID may have to match the procedure name. Please refer to your security system's documentation.

UID The user identifier associated with this user ID.

Rule: UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

Server Customization (3 of 4)

Deployment Manager Definitions

Rule: In the following, unless specified otherwise, names must be 8 or fewer characters.

Note: You must change all the port values from those used by the base Application Server.

Node host name
IP name or address of the system on which the server is configured. This value is used by other WebSphere Application Server for z/OS functions to connect to this server.

Note: The node host name must always resolve to an IP stack on the system where the Application Server runs. (You can either have multiple IP stacks on a given MVS image and have the deployment manager and base Application Server tied to separate host names, or you can associate them with the same node host name.) The node host name cannot be a DVIPA or a DNS name that, in any other way, causes the direction of requests to more than one system.

SOAP JMX Connector port
Port number for the JMX HTTP connection to this server based on the SOAP protocol. JMX is used for remote administrative functions, such as invoking scripts through wsadmin.sh.

Rule: Value cannot be 0.

Cell Discovery Address port
Port number used by node agents to connect to this deployment manager server.

DRS Client Address port
Port address for access to the server's data replication service. This is important for configurations that define replication groups.

Rule: Value cannot be 0.

ORB Listener host name
IP address on which the server's ORB listens for incoming IIOP requests. The default is ″*″, which instructs the ORB to listen on all available IP addresses.

ORB port
Port for IIOP requests which acts as the bootstrap port for this server and also as the port through which the ORB accepts IIOP requests.

Rule: Value cannot be 0.

ORB SSL port
Port for secure IIOP requests. The default is ″0″, which allows the system to choose this port.

HTTP transport host name
IP address on which the server's Web container should listen for incoming HTTP requests. The default is ″*″, which instructs the Web container to listen on all available IP addresses.

Note: The ″transport host name″ becomes the ″hostname″ in the virtualhosts.xml file, which makes setting a specific IP address here less than ideal because, if you do so, you are restricting yourself to that IP address until you go into the administrative console and add another virtual host.

HTTP port
Port for HTTP requests.

Rule: Value cannot be 0.

HTTP SSL port
Port for secure HTTP requests.

Rule: Value cannot be 0.

Server Customization (4 of 4)

Location Service Daemon Definitions

Rule: In the following, unless specified otherwise, names must be 8 or fewer characters.

The location service daemon is the initial point of client contact in WebSphere Application Server for z/OS. The server contains the CORBA-based location service agent, which places sessions in a cell. All RMI/IIOP IORs (for example, for enterprise beans) establish connections to the location service daemon first, then forward them to the target Application Server.

Daemon home directory
Directory in which the location service daemon resides.

Daemon job name
Specifies the job name of the location service daemon, specified in the JOBNAME parameter of the MVS start command used to start the location service daemon. This value, imported from the base Application Server node configuration, is a temporary value because the original base Application Server location service daemon is no longer used when the node is federated into the deployment manager's cell. So, you need to change the daemon job name to the one you want in the final configuration.

Note: This is also the name that will be used for all post-federation location service daemons on other MVS images. Therefore, this name is generic and has no system indicator. When the deployment manager's controller started, its location service daemon was started automatically using this particular daemon job name value.

Caution: When configuring a second cell, ensure you change the daemon job name from the default or value you used for the first cell.

Note: A server automatically starts the location service daemon if it is not already running.

Procedure name
Name of the member in your procedure library to start the location service daemon.

Rule: Name must be 7 or fewer characters.

User ID
The user ID associated with the location service daemon.

UID The user identifier associated with this user ID.

Rule: UIDs must be unique numbers, between 1 and 2,147,483,647, within the system.

IP Name

The fully qualified IP name, registered with the Domain Name Server (DNS), that the location service daemon uses.

Adhere to the following regarding your location service daemon IP name:

- It must be unique in the sysplex.
- It must not have the same value as any one system's node host name.
- It should be a name that can be used in conjunction with a routing service that distributes requests among nodes in the cell (i.e. systems in the sysplex).
- It should be a virtual IP address (VIPA) if you are operating in a sysplex.

Note: IBM recommends you use z/OS Sysplex Distributor, via a ″Distribute Virtual IP address″ (DVIPA), and dynamic virtual IP addresses. See related sections in the information center for more information.

Port The port number on which the location service daemon listens.

SSL port

The port number on which the location service daemon listens for SSL connections.

Note: Select the IP name and port number for the location service daemon carefully. You can choose any name you want, but, once chosen, it is difficult to change, even in the middle of customization.

Register daemon with WLM DNS

If you use the WLM DNS (connection optimization), you must select ″Y″ to register your location service daemon with it. Otherwise, select ″N.″

4 View Security Domain Configuration Panels

This panel displays values you previously set in the ″Configure security domain″ option--you can not change any of the values here. If you do wish to make changes, you must go back to the main Dialog panel and run through the ″Configure security domain″ option again.

Define variables for federate base Application Server node - worksheets

This article presents worksheets that will assist you when completing the WebSphere Application Server for z/OS Customization Dialog. See the related definitions article for definitions of each of the terms.

Note: If ″(cannot change)″ is displayed in the ″Your value″ column, that particular value is shown for reference purposes only and you cannot change it in that particular panel. Either the value is static throughout the Dialog, or you need to go back to a previous task to make the change.

Define Variables for Federate Base Application Server Node (1 of 1)

This panel sets up the names, network configuration, start procedures, and user IDs for an additional Application Server node in the cell.

Note:

- For identification, each controller and servant start procedure must have a user ID and will be defined in the STARTED class.
- Servers are started by the MVS START command using the controller start procedure.

Table 36. Define variables for Federate base Application Server node (1 of 1)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
WebSphere Application Server home directory	/WebSphere/V5R1M0/AppServer	

Table 36. Define variables for Federate base Application Server node (1 of 1) (continued)

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Deployment Manager Access		
Node host name	(null)	
JMX SOAP port	8879	
Deployment manager security is enabled	N	
User ID	WSADMIN	(cannot change)
Include Apps	Y	
Base server ORB port	9810	
Node Agent Definitions		
Server name (short)		
Server name (long)		(cannot change)
JMX SOAP Connector port	8878	
DRS Client Address port	7888	
Node Discovery port	7272	
Node Multicast Discovery port	5000	
ORB listener host name	*	
ORB port	2809	
ORB SSL port	0	

Define variables for federate base Application Server node - definitions

This article lists definitions for the terms you will come across in the WebSphere Application Server for z/OS Customization Dialog.

Note: Some term definitions--those that you cannot set in the related panel--are not displayed. To see the definition, you must to go to the Information Center article for the panel on which you **can** change that particular term.

Rules:

- In the following, unless specified otherwise, names must be 8 or fewer characters.
- In some of the following, specifying ″Y″ (yes) tells the Dialog to define the profile or enable an option. Specifying ″N″ (no) tells the Dialog to not define the profile or enable the option.

WebSphere Application Server home directory

Directory in which the base Application Server configuration resides.

Note: Depending on your service level and the variables you loaded in the ″Load customization variables″ step, you may need to modify this value to reflect the correct directory.

Deployment Manager Access

Node host name

IP name or address of the system on which the deployment manager server is configured. This value, equivalent to ″cell host″ in addNode.sh, is used by other WebSphere Application Server for z/OS functions to connect to this server in order to federate the designated node into the deployment manager cell.

Note: The node host name must always resolve to an IP stack on the system where the Application Server runs. The node host name cannot be a DVIPA or a DNS name that, in any other way, causes the direction of requests to more than one system.

JMX SOAP port

JMX (Java Management Extensions) SOAP (Simple Object Access Protocol) connector port that the add node request uses to connect to the deployment manager. It provides the federation process with knowledge of which deployment manager is the target of the federation.

Deployment manager security is enabled

Specify ″Y″ to enable deployment manager security.

Include Apps

Specify ″Y″ if you want to include applications with your deployment manager node. Enabling this option instructs the addNode program to include applications from the node, as it would remove them prior to federation otherwise. If the application already exists in the cell, a warning is printed and the application is not installed into the cell.

Note: You must use this option to migrate all the applications to the new cell. Federating the node to a cell using the addNode command does not merge any cell-level configuration information, including that from virtualHost.

Base server ORB port

Port for IIOP requests which acts as the bootstrap port for this server and also as the port through which the ORB accepts IIOP requests. Port value cannot be 0.

Note: The add node operation creates the nodeagent administrative server with a default ORB port equivalent to the INS CosNaming default bootstrap port. Because this same port was previously used by the node's initial base server, the initial base server's ORB port must change to a new port value. The default value to which the Application Server's ORB port is set is 9810. If you configure multiple cells that intersect the same systems, use of the default value will cause a port conflict between these cells. This option helps you set the port number in case port 9810 was previously assigned.

Node Agent Definitions

Server name (short)

Name of the node agent server. This is the server's job name, as specified in the MVS START command JOBNAME parameter. This value is also passed as a parameter to the server's start procedures to specify the location of the server's configuration files and identify the server to certain WebSphere Application Server for z/OS-exploited z/OS facilities (for example, SAF).

Rule: Name must usually contain 7 or fewer all-uppercase characters. If you wish to instead specify 8 characters, you must follow the steps outlined in the Information Center article ″Converting a 7-character jobname to 8 characters.″

Server name (long)

Name of the node agent and the primary external identification of the nodeagent server. This name identifies the server as displayed through the administrative console and also appears as part of the directory path under the WebSphere Application Server for z/OS home's configuration directory.

Rule: Name must be 60 or fewer characters and can be of mixed case.

JMX SOAP Connector port

Port number for the JMX HTTP connection to this server based on the SOAP protocol. JMX is used for remote administrative functions and is invoked through scripts such as wsadmin.sh.

Rule: Value cannot be 0.

DRS Client Address port

Port used to access the server's data replication service. This is important for configurations that define replication groups.

Rule: Value cannot be 0.

Node Discovery port

Defines the TCP/IP port to which the node agents listens for discovery requests that originate from the deployment manager.

Node Multicast Discovery port

Defines the multicast TCP/IP port to which the node agent listens for discovery requests that originate from its managed servers.

ORB listener host name

IP address on which the server's ORB listens for incoming IIOP requests. The default is ″*″, which instructs the ORB to listen on all available IP addresses.

ORB port

Port for IIOP requests which acts as the bootstrap port for this server and also as the port through which the ORB accepts IIOP requests.

Rule: Value cannot be 0.

ORB SSL port

Port for secure IIOP requests. The default is ″0″, which allows the system to choose this port.

Save customization variables - worksheet

Print out this worksheet for each option in the Customization Dialog and use it to record your values as you customize WebSphere Application Server for z/OS.

Table 37. Save Customization Variables

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Data set name	(null)	

Specify the name of the data set into which you want to save the customization variables.

Rules:

- The data set must be a sequential data set. Do not specify a member name.
- Place quotes around the data set name.

Save security domain variables - worksheet

Print out this worksheet for each option in the Customization Dialog and use it to record your values as you customize WebSphere Application Server for z/OS.

Table 38. Save Security Domain Variables

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)
Data set name	(null)	

Specify the name of the data set into which you want to save the security domain variables.

Rules:

- The data set must be a sequential data set. Do not specify a member name.
- Place quotes around the data set name.

Generate customization jobs - worksheet

Print out this worksheet for each option in the Customization Dialog and use it to record your values as you customize WebSphere Application Server for z/OS.

Table 39. Generate Customization Jobs

Item	Value in the Dialog after you load IBM defaults	Your value (Fill in the blanks)

Job card information	`//jobname JOB (ACCTNO,ROOM),`*userid*`,CLASS=A,REGION=0M`
	`//*`
	`//*`
	`//*`	

Specify the job card according to your installation requirements. For each job, the Dialog generates a job name and the ″JOB″ keyword to match the member name of the PDS, but you specify the rest.

Note: If you need to run these jobs on a particular system in the sysplex (for example, JES2 MAS or JES3 complex), you should specify the necessary Scheduling Environment (SCHENV), JES2 JOBPARM, or JES3 //*MAIN statement at this time.

Example of a job card entry:

```
//jobname JOB 1234,USER1,NOTIFY=????,MSGCLASS=O,REGION=0M
//*             USER=SYSADM1,PASSWORD=SYSADM1
/*JOBPARM SYSAFF=SYSB
```

Note: This example is useful for jobs that require a user ID other than that of the logged-on TSO user (typically a user ID with UID=0). In that case, you can just put a comma at the end of the first line, put in the correct user ID on the second line, then un-comment that second line.

Sample Customization Dialog instructions

Follow the linked articles to find instructions for following the instructions produced by each option in the Customization Dialog.

Following the customized Security Domain Configuration instructions

This article gives an overview of the contents of the Customization Dialog instruction file for this option. You need to follow the actual file in the Dialog to configure your system.

The major stages of the security domain configuration process are:

Table 40. Following the customized security domain configuration instructions

Stage	Description
1	Make a variety of configuration changes to your z/OS system configuration. The customized instructions provide details and pointers to relevant documentation.
2	Builds the RACF commands for the WebSphere Application Server for z/OS security domain with job BBOSBRAJ.
3	Invoke RACF commands that instantiate the security rules set up in the previous job with job BBOSBRAK.

Following the customized Application Server node instructions

This article gives an overview of the contents of the Customization Dialog instruction file for this option. You need to follow the actual file in the Dialog to configure your system.

The major stages of the Application Server node customization process are:

Table 41. Following the customized Application Server node instructions

Stage	Description
1	Make a variety of configuration changes to your z/OS system configuration (PARMLIB, TCP/IP, workload management, and so forth). The customized instructions provide details and pointers to relevant documentation.
2	If you require message translation, set up MMS to translate messages for WebSphere Application Server for z/OS with job BBOMSGC. **Note:** If you are not using NLS (National Language Support), you can skip this job.
3	Define log streams used by WebSphere Application Server for z/OS and RRS through jobs BBOERRLG and BBORRSLS. You do not need to run BBORRSLS if RRS is already running on your target MVS system.
4	Allocate the CTRACE data set through job BBOWCTR.
5	Create a customized set of RACF commands for initial WebSphere Application Server for z/OS security setup through job BBOCBRAJ. The RACF commands are saved in member BBOWBRAK of the *hlq*. DATA data set. Job BBOWBRAK executes these RACF commands. Later, you can use the RACF commands saved in BBOWBRAK to help in defining security for additional servers or users.
6	Check the authorization level of all your user IDs and modify them where necessary.
7	Allocate and mount the WebSphere Application Server for z/OS runtime HFS through job BBOWCHFS. If your root HFS is mounted read-only, you may need to create one or more directories manually. See the instructions for details. Job BBOMCFG creates subdirectories in the WebSphere Application Server for z/OS runtime HFS and job BBOMCFGU, which is optional, configures the runtime HFS to include the directory for UDDIReg.
8	Copy customized PARMLIB and PROCLIB members into their proper locations using job BBOWCPY1.
9	Copy customized HFS files into their proper locations using job BBOWCPY2.
10	Create the was.env file for the location service daemon and the Application Servers through job BBOWC2N.
11	Install the administrative console and the IVT application through job BBOWIAPP.
12	(Optional) Copy digital certificates from RACF to the keystores in the HFS with job BBOCR2FA.
13	Complete the HFS initialization with job BBOMCFG2.
14	Activate Resource Recovery Services (RRS), if it is not already active, with the MVS command `START ATRRRS,SUB=MSTR`.
15	Start the CTRACE writer used by WebSphere Application Server for z/OS with the MVS command `TRACE CT,WTRSTART= BBOWTR`.

Table 41. Following the customized Application Server node instructions (continued)

Stage	Description
16	Start the Application Server with the following MVS command: `S <controller_procname, JOBNAME=server_shortname, ENV=<cell_shortname.node_shortname.server_shortname` This process causes information to be stored in the HFS. Once the process is complete, you have a working WebSphere Application Server for z/OS runtime that you will use to run the Installation Verification Test (using job BBOWIVT) and your own applications.
17	For reference purposes only, configure samples with jobs BBODEFR, BBOINST, and BBOUNIN. Ensure you reset all sample EAR files to their original state and delete the samples when you are finished with them.
18	Use the BBOWBRAC sample exec to help define security controls for clusters. It defines all the user IDs and groups that are necessary and sufficient for installing WebSphere Application Server for z/OS, among other things.

Following the customized integral JMS provider instructions

This article gives an overview of the contents of the Customization Dialog instruction file for this option. You need to follow the actual file in the Dialog to configure your system.

The major stages of the integral JMS provider customization process are:

Table 42. Following the customized integral JMS provider instructions

Stage	Description
1	Make a variety of configuration changes to your z/OS system configuration. The customized instructions provide details and pointers to relevant documentation.
2	Set the authorization bits for involved MQ files.
3	Establish security identity for the MQ started tasks.
4	Creates dual bootstrap data sets for usage by the Integral JMS Provider WebSphere MQ Queue Manager with job BBOJBSDS.
5	Create page data sets for usage by the Integral JMS Provider WebSphere MQ Queue Manager with job BBOJPAGR.
6	Assemble your parameters with jobs BBOJXPRR and BBOJZPRR. BBOJXPRR assembles the X parameters for usage by the Integral JMS Provider WebSphere MQ Channel Initiator, and BBOJZPRR assembles the Z parameters for usage by the Integral JMS Provider WebSphere MQ Queue Manager.
7	Copy the tailored start procedures and parameters to the run-time libraries with jobs BBOWCPYJ and BBOWCPYZ.
8	Stop the Application Server with the MVS command `STOP server_shortname`.
9	Run job BBOWCPJ2 to write z/OS UNIX resident files into the HFS for use by the application server.
10	Initialize the Integral JMS Provider run-time environment in the HFS for the application server with job BBOWJWC.

Table 42. Following the customized integral JMS provider instructions (continued)

Stage	Description
11	Start the Application Server with the following MVS command: `S <controller_procname, JOBNAME=server_shortname, ENV=<cell_shortname.node_shortname.server_shortname` This process causes information to be stored in the HFS. Once the process is complete, you have a working WebSphere Application Server for z/OS runtime that you will use to run the Installation Verification Test (using job BBOWIVT) and your own applications.
12	For reference purposes only, configure samples with jobs BBODEFRM, BBOINSTM, and BBOUNINM. Ensure you reset all sample EAR files to their original state and delete the samples when you are finished with them.

Following the customized deployment manager node instructions

This article gives an overview of the contents of the Customization Dialog instruction file for this option. You need to follow the actual file in the Dialog to configure your system.

The major stages of the deployment manager node customization process are:

Table 43. Following the customized deployment manager node instructions

Stage	Description
1	Make a variety of configuration changes to your z/OS system configuration (PARMLIB, TCP/IP, workload management, and so forth). The customized instructions provide details and pointers to relevant documentation.
2	Create a customized set of RACF commands for initial WebSphere Application Server for z/OS security setup through job BBODBRAJ. The RACF commands are saved in member BBODBRAK of the *hlq*. DATA data set. Job BBODBRAK executes these RACF commands. Later, you can use the RACF commands saved in BBODBRAK to help in defining security for additional servers or users.
3	Allocate and mount the WebSphere Application Server for z/OS runtime HFS through job BBODCHFS. If your root HFS is mounted read-only, you may need to create one or more directories manually. See the instructions for details. Job BBODCFG creates subdirectories in the WebSphere Application Server for z/OS runtime HFS.
4	Configure the run-time HFS to include the directory for UDDIReg with the job BBOMCFGU, if desired.
5	Copy customized PROCLIB members into their proper locations using job BBODCPY1.
6	Copy customized HFS files into their proper locations using job BBODCPY2.
7	Create the was.env file for the location service daemon and the Application Servers through job BBODC2N.
8	Install the administrative console and the filetransfer application through job BBODIAPP.
9	Complete the HFS initialization with job BBODCFG2.
10	Activate Resource Recovery Services (RRS), if it is not already active, with the MVS command `START ATRRRS,SUB=MSTR`.

Table 43. Following the customized deployment manager node instructions (continued)

Stage	Description
11	Start the deployment manager (and, hence, the location service daemon) with the MVS command `START BBO5DCR,JOBNAME=BBODMGR`. This process stores information in the HFS. Once the process is complete, you have a working WebSphere Application Server for z/OS Network Deployment runtime that you will use to run your own applications.
12	Use the BBODBRAC sample exec to help define security controls. It defines all the user IDs and groups that are necessary and sufficient for installing WebSphere Application Server for z/OS, among other things.

Following the customized Federate base Application Server node instructions

This article gives an overview of the contents of the Customization Dialog instruction file for this option. You need to follow the actual file in the Dialog to configure your system.

The major stages of the base Application Server node federation process are:

Table 44. Following the customized Federate base Application Server node instructions

Stage	Description
1	Make a variety of configuration changes to your z/OS system configuration. The customized instructions provide details and pointers to relevant documentation.
2	Add the Application Server(s) associated with the base Application Server node to the deployment manager's cell with job BBOWADDN.
3	Update the application server's WLM application environment.

4	(Optional) Start the node agent server with the MVS command:
	`START controller_procname,`
	`JOBNAME=server_shortname,`
	`ENV=cell_shortname.node_shortname.server_shortname`

5	Start the JMS server with the MVS command:
	`START controller_procname,`
	`JOBNAME=server_shortname,`
	`ENV=cell_shortname.node_shortname.server_shortname`

6	(Optional) Start the Application Server with the following MVS command:
	`S <controller_procname,`
	`JOBNAME=server_shortname,`
	`ENV=<cell_shortname.node_shortname.server_shortname`

Following the customized Web services gateway instructions

This article gives an overview of the contents of the Customization Dialog instruction file for this option. You need to follow the actual file in the Dialog to configure your system.

The major stages of the Web services gateway process are:

Table 45. Following the customized Web services gateway instructions

Stage	Description
1	Install the ″development kit″ onto your workstation.

z/OS port assignments

Purpose

The following table lists the default server values for WebSphere Application Server for z/OS V5.

z/OS port assignments

The following table lists the z/OS port assignments.

Port	Base location service daemon	Base Application Server	ND location service daemon	ND Application Server	Node Agent	JMS Server	Deploy. Manager
HTTP		9080		9080			9090
HTTP/S		9443		9443			9043
Bootstrap		2809		9810	2809	2810	9809
ORB	5655	2809	5755	9810	2809	2810	9809
ORB SSL	5656	0	5756	0	0	0	0
SOAP/JMX		8880		8880	8878	8876	8879
DRS		7873		7873	7888		7989
JMS Queued		5558				5558	
JMS Direct		5559				5559	
Node Discovery					7272		
Node Multi-cast Discovery					5000		
Cell Discovery							7277

Ensuring problem avoidance

To implement WebSphere Application Server for z/OS, you must implement the necessary features, subsystems, and resources required for the run-time environment. This section provides checklists for tasks you should verify before running your WebSphere Application Server for z/OS system in order to prevent the most common errors encountered during the installation.

Before you begin: Perform the following steps to ensure problem avoidance, checking off each item as you complete it:

1. Plan to prepare your z/OS environment:

Table 46.

Check off	Item
	Check that all the maintenance suggested in the PSP bucket WASAS500 subset WAS500 has been applied.

Table 46. (continued)

Check off	Item
	Make certain your address space is large enough. Some WebSphere Application Server for z/OS servers must be able to get a 1GB virtual region to run any workload. Make sure that your installation exits (IEFUSI) do not limit the virtual region size. We recommend that you specify REGION=0M so as not to limit their size.
	Add another local page data set, two if your system does any paging of the WebSphere Application Server for z/OS server address spaces.

2. Plan to prepare your DB2 subsystem (if you will use DB2):

Table 47.

Check off	Item
	Increase the MAX USERS (CTHREAD) and MAX BATCH CONNECT (IDBACK) in your DB2 environment settings. Use the sample job in DSN710.SDSNSAMP(DSNTEJ6Z) to display the ″ZPARMS″ settings of the running system. (An alternative is to use the DB2 Control Center to display these parameters.)
	Define at least 200 buffers to the DB2 BP32K buffer pool. Use this command to display the current bufferpool allocations: `-dis bpool(active)detail`. Verify JDBC 2.0 functionality. The JDBC IVT sample01 JAVA application does not exercise JDBC 2.0 drivers nor the RRS attach facility. A modified version that tests these functions can be found in the DB2 Conundrum whitepaper on Techdocs at http://www.ibm.com/support/techdocs/atsmastr.nsf/PubAllNum/WP100217 (This will also verify that the DSNJDBC plan is bound correctly and that it matches the .ser file.)
	Verify the level of DB2 code running on your system with the DSNTEJ6U sample job or run the DSNUTILB utility with the `DIAGNOSE DISPLAY MEPL` command. The module names, dates, and PTF number on the right of the report are in EBCDIC.
	Make sure that any updates to the DB2 ERLY code are installed, and that you have IPLed your system to activate them.

	Check the JDBC service installed on your system. Use the following Java program to display the service level:
	`export LIBPATH=/usr/lpp/db2/db2710/lib:$LIBPATH>`
	`java -cp /usr/lpp/db2/db2710/classes/db2j2classes.zip`
	`COM.ibm.db2os390.sqlj.util.DB2DriverInfo`
	The typical output message looks like this:
	`DB2 for OS/390 SQLJ/JDBC Driver build version is:DB2 7.1 PQ54756`

3. Plan to verify your USS/HFS configuration:

Table 48.

Check off	Item
	Specify enough threads, files, and processes in your BPXPRMxx member of parmlib. Here is a starting list if you don't have it set up yet: • MAXTHREADS:10000 • MAXTHREADTASKS:5000 • MAXFILEPROC:10000 • MAXSOCKETS in the AF_INET domain:12000

Table 48. (continued)

Check off	Item
	If you have an exit that checks for valid accounting codes, you may need to specify an accounting value for spawned address spaces. Use the _BPX_ACCT_DATA= variable in the was.env file.
	Ensure that the userid associated with running the installation jobs that run the BPXBATCH shell scripts has an OMVS segment that directs PROGRAM('/bin/sh') to use the z/OS shell rather than the tcsh (C) shell (at '/bin/tcsh'). These particular shell scripts won't run from the tcsh shell.

4. Plan your SMP/E tasks:

Table 49.

Check off	Item
	You can install WebSphere Application Server for z/OS into an SMP/E environment (SMP/E 3.1 or later) separate from the one you use for z/OS. This includes target and distribution zones, as well as HFS data sets. We recommend you use a separate environment, but you should enable the cross-zone checking so that any prerequisite service requirement can be checked between the WebSphere Application Server for z/OS and z/OS SMP/E zones.
	Verify that the DDDEF for the LTS data set describes a PDSE format data set. This will avoid LINK-EDIT errors during the SMP/E processing.
	We strongly recommend that you carefully read*WebSphere Application Server for z/OS V5.0: Program Directory*. This is a very large product and you have to make sure that there is sufficient space in all target and temporary data sets for receive and apply processing.

5. Plan your ISPF Dialog consideration:

Table 50.

Check off	Item
	Do not use ISPF Dialog in Split Screen mode. It may happen that you don't see everything on one screen. Keep in mind that there might be parameter values outside of the visible portion of your screen and that this may cause some of the WebSphere Application Server for z/OS installation options to be set to default values, which may not be what you want.
	Turn off PFSHOW on the ISPF Dialog screen. As with the split screen problem, there might be parameter values outside the visible portion of your screen, which may cause some of the WebSphere Application Server for z/OS installation options to be set to default values, which may not be what you want.
	You need to use a screen size of at least 32 lines to be able to invoke ISPF Dialog.
	You might want to consider starting with new CNTL and DATA data sets as output of the ″Generate the Jobs Stream″ function. During this task, you get a large number of confirmation messages for every successful generated job. It is very easy to overlook an error message during this process. Unfortunately this sometimes causes the job generation process to stop without recreating all JCL. When you run the jobs you may run ″old″ JCL and this will result in unpredictable problems.

6. Plan to verify your WLM environment:

Table 51.

Check off	Item
	Verify that WLM is running in goal mode with the `D WLM,SYSTEMS` command.

7. Plan to check your TCP/IP configuration:

Table 52.

Check off	Item

	Telnet into UNIX Systems Services and issue these commands to verify that you can find your host name by IP address or IP host-name:
	• Get the local host name: `hostname`
	– You will get a response such as: sc49.itso.ibm.com
	– Use the output from the hostname command for the following `nslookup` command.
	• Get host address by name: `nslookup sc49.itso.ibm.com`
	– You will get a response such as this:
	`Server:sc49.itso.ibm.com`
	`Address:9.12.6.15`
	`Name:sc49.itso.ibm.com`
	`Addresses:9.12.6.15`
	– Use the dotted IP address from this display for the following command.
	• Get host name by address: `nslookup 9.12.6.15`
	– You will get a response such as in the previous nslookup display.
	There is also a small Java program, InetInfo.java, which you can run to verify the same TCP/IP configuration. See techdocs for the program at http://www.ibm.com/support/techdocs/atsmastr.nsf/PubAllNum/TD100609.
	Example: This example shows you how to run the InetInfo Java code.
	`JAVA4 @SC42:/u/java4>export PATH=/usr/lpp/java/IBM/J1.3/bin`
	`JAVA4 @SC42:/u/java4>java InetInfo`
	`get Local Host`
	`IP Address:9.12.6.27`
	`get Host Name By Address using 9.12.6.27`
	`Host Name:wtsc42oe.itso.ibm.com`
	`get Host Address By Name using wtsc42oe.itso.ibm.com`
	`Host Address:9.12.6.27`

	Issue the `hometest` command from TSO. It should show the correct TCP Host name, corresponding IP address(es), and HOME IP addresses. If it doesn't produce the correct results, then TCP/IP is not configured correctly.
	If the fully qualified TCP/IP HostName is greater than 24 characters, then a DNS will be required. Otherwise, the /etc/hosts file can provide the naming lookup.
	Verify that the DNS name you are using is definitive (authoritative) in your installation.

8. Plan to verify that security is in place:

Table 53.

Check off	Item
	Check that the location service daemon has access to parmlib concatenation to retrieve CTRACE settings in the CTIBBOxx member.

Table 53. (continued)

Check off	Item
	Verify that all WebSphere Application Server for z/OS servers must have READ access to any data sets or files in their JCL procedures.
	Verify that your installation has the RACF list-of-groups turned on. (SETROPTS LIST will show you if turned on or off.) Without this list of groups turned on, an ID cannot belong to more than one group and ASSR1 associates with only WSSR1 instead of both WSSR1 and WSCFG1. Use the command `SETROPTS GRPLIST` to turn on the list of groups.
	Define the profile BPX.SAFFASTPATH in the FACILITY class to enable SAF fastpath support.
	If you do not load SBBOLOAD into LPA, you must add it to the program control list in the RACF PROGRAM class. (With z/OS V1.2, you can use the FACILITY class profile BPX.DAEMON.HFSCTL class. This will cause only HFS files to be checked for program control.)
	Verify that the /usr/lpp/java/IBM/J1.3/lib HFS file permission bits are set up correctly to allow the read capability to other bits (644) and that /usr/lpp/java/IBM/J1.3/bin and /usr/lpp/java/IBM/J1.3/bin/classic with the execute permission bit on (755) and APF authorized.
	Verify that the authorization bits for the WebSphere Application Server for z/OS HFS (default name is /usr/lpp/zWebSphere/V5R0M0) file are correctly set up for the WebSphere Application Server for z/OS configuration group.

9. Plan your installation phases:

Table 54.

Check off	Item
	After any maintenance has been applied, verify that the code loaded in LPALIB or LNKLST is synchronized with the code in the HFS. Check the location service daemon job log to verify that the correct maintenance level is in use.

You are done when you have checked all the applicable items.

Installation: Resources for learning

Use the following links to find relevant supplemental information about installation and customization. The information resides on IBM and non-IBM Internet sites, whose sponsors control the technical accuracy of the information.

These links are provided for convenience. Often, the information is not specific to the IBM WebSphere Application Server product, but is useful all or in part for understanding the product. When possible, links are provided to technical papers and Redbooks that supplement the broad coverage of the release documentation with in-depth examinations of particular product areas.

View links to additional information about:
- "Planning, business scenarios, and IT architecture" on page 158
- "Programming model and decisions" on page 160
- "Programming instructions and examples" on page 160
- "Programming specifications" on page 160
- "Administration" on page 160
- "Support" on page 160

Planning, business scenarios, and IT architecture

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

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

  The home of technical information for developers working with WebSphere products. You can download WebSphere software, take a fast path to developerWorks zones, such as VisualAge Java or WebSphere Application Server, learn about WebSphere products through a newcomers page, tutorials, technology previews, training, and Redbooks, get answers to questions about WebSphere products, and join the WebSphere community, where you can keep up with the latest developments and technical papers.
- IBM WebSphere Application Server library and information centers Web site

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

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

  The IBM WebSphere software platform home page introduces WebSphere products and describes how companies can easily transform to an e-business, with software that can grow as fast as the business it supports.
- Migrating to WebSphere V5.0: An End-to-End Migration Guide, SG24-6910-00

  This IBM Redbook is the definitive migration guide for migrating earlier versions of WebSphere Application Server to Version 5. The Redbook adds a broader scope, including planning for application migration and WebSphere Studio Application Developer tooling and samples. Read this book to formulate an optimal migration strategy.
- The power of Edge of Network technology in IBM WebSphere Application Server Version 5

  A description of WebSphere Application Server Edge Components Version 5.
- WebSphere Application Server - Express, V5

  A description of WebSphere Application Server Express, Version 5.
- WebSphere Application Server, Version 5

  A description of the base product, WebSphere Application Server, Version 5.
- IBM WebSphere Business Integration Server Foundation, Version 5

  A description of IBM WebSphere Business Integration Server Foundation, Version 5.
- IBM WebSphere Application Server for z/OS, Version 5

  A description of WebSphere Application Server for z/OS, Version 5.
- Information center for WebSphere Application Server Edge components

  The information center for WebSphere Application Server Edge components contains complete documentation for the Caching Proxy and the Load Balancer in these PDF online books, *WebSphere Application Server Concepts, Planning, and Installation for Edge Components*, the *WebSphere Application Server Caching Proxy Administration Guide*, and the *WebSphere Application Server Programming Guide for Edge Components*.
- Linux on IBM eserver zSeries and S/390: ISP/ASP Solutions

  This IBM Redbook describes how Linux can be combined with z/VM on zSeries and S/390 hardware. This combination of hardware and operating systems enables Internet Service Providers (ISP) and Application Service providers (ASP) to more efficiently provide services.

  The book assumes a broad definition of ASP that includes production enterprise solutions as simple as file serving. In a world of discrete servers, when a new resource is required, workload can either be added to an existing server or a new one can be purchased. Often a new server is installed and the server farm grows in the enterprise.

S/390 and zSeries hardware, microcode and software allow physical resources to be made virtual among Linux systems. Virtual physical resources allow many hundreds of Linux systems to exist on a single server. Running multiple Linux images as guests of VM/ESA or z/VM is a smart choice.

- Performance Analysis for Java Web sites

  A guide to developing applications that do not have performance and scalability problems when they get deployed to production. The *Performance Analysis for Java Web Sites* book walks you through planning, designing, and testing to verify that you are not surprised when you release your new site. All of the examples use WebSphere Application Server products and applications.

  Stacy Joines, Ruth Willenborg, and Ken Hygh have helped many customers with performance-related issues, and have authored this book to capture their experiences, and to help you test and tune your applications.

  The book is available in book stores and on Amazon.
- developerWorks: Patterns for e-business: Redbooks listing

  This Web page lists links to pattern resources under these categories:
  - Current patterns Redbooks
  - Superseded patterns Redbooks (valid for back-level product versions)
  - Independent analyst reports
  - Patterns CD order offer
  - Back-level version patterns Web site (zip downloads and old Flash tutorial)
  - Customer references
  - White papers
  - Multimedia presentations and screen cams
  - Webcasts
  - Patterns development kit
  - WebSphere technical exchange presentations
- developerWorks: IBM Patterns for e-business

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

  This White paper is from the IBM High Volume Web Sites team. The White paper describes component selection and management techniques you can use to make your Web site ready to adapt to increasing traffic. These techniques are the product of IBM experiences while working with customers seeking to improve the performance and availability of some of the largest Web sites in the world.

  Abstract: Optimizing for scalability remains a significant challenge for e-businesses as they balance the demands for availability, reliability, security, and high performance. Vendors are responding with infrastructure options and supporting hardware and software platforms that address these requirements. This update identifies current products and emerging trends that are most likely to improve the scalability of your e-business infrastructure.
- IBM WebSphere Application Server V5.0 System Management and Configuration: WebSphere Handbook Series

  This IBM Redbook provides the knowledge needed to implement WebSphere Application Server V5.0, Network Deployment runtime environment, to package and deploy Web applications, and to perform ongoing management of the WebSphere environment. It is one in a series of handbooks, designed to give you in-depth information on the entire range of WebSphere Application Server products.

  This book provides an overview of the architecture, topology options, and new features of WebSphere Application Server Version 5 and WebSphere Application Server Network Deployment Version 5. It takes you through the installation steps needed to install each topology. Platform-specific chapters are included for installation on Windows, AIX, and Solaris.

  The book then takes you through the process of configuring WebSphere Application Server. It is organized in the same manner as the WebSphere Application Server Administrative Console. It also includes information on packaging and deploying applications. It concludes with information on troubleshooting runtime problems.
- IBM WebSphere V4.0 Advanced Edition Handbook

This Redbook describes base application topologies and product mappings for WebSphere Application Server. Refer to the IBM Redbooks Web site for the latest update.

- The User centered design (UCD) for different project types, part 1

  This Web page is the first of two articles posted to IBM developerWorks that describes useful application design activities for different types of projects.
- The User centered design (UCD) for different project types, part 2

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

Programming model and decisions

- Designing e-business Solutions for Performance

  This White paper describes how the design or implementation of an e-business application can affect performance.
- Managing Web Site Performance

  This White paper contains tips and techniques for developers building applications that use session persistence. It also helps administrators to tune the WebSphere Application Server product appropriately for these applications.

Programming instructions and examples

- IBM developerWorks

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

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

Programming specifications

- J2EE information

  For more information about J2EE specifications, visit the Sun site.
- sun.net.inetaddr.ttl property

  The following Java 2 SDK, Standard Edition 1.4 Web site describes the private sun.net.inetaddr.ttl property, which works in both Java 2 SDK, Standard Edition 1.3 (WebSphere Application Server V5.0.0, V5.0.1, and V5.0.2) and Java 2 SDK, Standard Edition 1.4(WebSphere Application Server V5.1).
- java.net.URLConnection class

  The *Networking* section of this Java 2 SDK, Standard Edition 1.4 Web site describes a change in the behavior of the java.net.URLConnection class.

Administration

- Best Practices Zone on WSDD

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

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

Support

- AIX Fix Distribution Service Web site

A Web facility for downloading AIX Version 4 and AIX Version 3 fixes, with a limited search engine designed with the assumption that you know what fix you need. If you do not know what fix you need, there is a pointer at the Web site to the APAR Database Facility. You can also contact your authorized IBM business partner or IBM Support Center.

- Ten Steps to Getting Support for WebSphere Application Server

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

- WebSphere Application Server Support page

  Take advantage of the Web-based Support and Service resources from IBM to quickly find answers to your technical questions. You can easily access this extensive Web-based support through the IBM Software Support portal at URL `http://www.ibm.com/software/support/` and search by product category, or by product name. For example, if you are experiencing problems specific to WebSphere Application Server, click **WebSphere Application Server** in the product list. The WebSphere Application Server Support page appears.

Chapter 2. Deprecated Version 5.1 features

This document summarizes deprecated features in Version 5.1. As they become available, links to additional information will be provided to help you migrate away from deprecated features.

IBM Deprecation policy

A deprecated class or interface will be supported for at least two full product releases, or three full years (whichever is longer) before being removed from the product entirely.

Deprecation list

The following tables summarize what is deprecated, by release and version. Use these tables to calculate the earliest release in which the item might be removed, according to the deprecation policy.

For more information on deprecated APIs, refer to the ″Deprecated″ section at IBM WebSphere Application Server, Release 5 API Specification.

Each table represents what release/version the deprecation took effect. Each table describes what is being deprecated, for example; features, APIs, scripting interfaces, tools, wizards, publicly exposed configuration data, naming identifiers, constants, etc., and, where possible, the recommended migration action IBM suggests.

This article consists of the following versions/releases:

- Table 55 Deprecated features in V5.1
- Table 56 on page 167 Deprecated features in V5.0.2
- Table 57 on page 168 Deprecated features in V5.0.1
- Table 58 on page 169 Deprecated features in V5.0

Table 55. **Deprecated features in V5.1**

Installation and migration tools
The Application Assembly Tool (AAT) used for developing J2EE applications is being replaced by the Assembly Tool (ATk) component of the Application Server Toolkit (ASTk). **Recommended migration action:** Instead of running the Application Assembly Tool, users will install and run the Assembly Toolkit component of the Application Server Toolkit. ASTk is based on the eclipse framework. Upon starting the ASTk, the J2EE function is found by opening the J2EE Perspective.
JDOM (a Java representation of an XML document which provides an API for efficient reading, manipulating and writing documentation). The currently packaged version of JDOM in WebSphere Application Server will not be packaged in future releases of WebSphere. **Recommended migration action:** Go to JDOM and get the latest copy of JDOM and bundle it inside your application. **Note:** Customers running WebSphere Studio Application Developer Integration Edition Version 4.1 applications will need to migrate them to WebSphere Studio Application Developer Integration Edition Version 5.0.

Table 55. **Deprecated features in V5.1** *(continued)*

In future releases, IBM intends to remove the C++ Object Request Broker (ORB), the C++ library for IDL valuetypes and the WebSphere Application Server C++ security client. IBM will no longer ship or support the Common Object Request Broker Architecture (CORBA) C++ Developer Kit. The CORBA technology is a bridge for migration to a Java 2 Platform Enterprise Edition (J2EE) and WebSphere Application Server environment. In addition to the preceding information, the CORBA C++ client feature will be removed from the Application Clients installation image in future releases. **Recommended migration action:** It is recommended that customers migrate to the Object Request Broker (ORB) service for Java technology that ships with WebSphere Application Server. However, there is no equivalent J2EE functionality for the C++ security client or the C++ Valuetype library. Customers that require such functionality must provide or develop their own. For information on the ORB service for Java technology, see Managing Object Request Brokers. The deprecation of the CORBA C++ Developer Kit does not affect support for CORBA interoperability with vendor software for CORBA services. View the following links for additional information about interoperability: • CORBA Interoperability Samples documentation • IBM WebSphere Application Servers CORBA Interoperability white paper
IBM Cloudscape 5.1.x. **Recommended migration action:** No action is required at this time. However, changes will be required as of WebSphere Application Server Version 6.1. At that time, you will need to redeploy your applications, changing any Cloudscape-specific calls.
Servers and clustering features
IBM HTTP Server (IHS) 1.3.x. **Recommended migration action:** If you are using IHS 1.3.x with modules: • that are shipped as part of IHS 1.3.x packages, you do not need to take any action to migrate those modules. • supplied by a third party (including other IBM products), you need to obtain IHS/Apache 2 versions of these modules from the third party. • that have been customized or are in-house, you need to port these modules to the new IHS/Apache 2 API.
Application programming model and container support features
Bean Scripting Framework (BSF). JSP execution and debugging functionality is being deprecated in WebSphere Application Server version 5.1. **Recommended migration action:** If using the Javascript, Tcl, and Python languages, the functionality will need to be rearchitected. If using BSF scripting in your own custom applications, they will be unaffected. Custom scripts written for the WebSphere Application Server admin console will also be unaffected. This functionality will continue to exist in WebSphere Application Server app server release 5.1, and succeeding releases, until version 6.0. If debugging JSPs you may have to restart the app server during Javascript debugging sessions.

Table 55. **Deprecated features in V5.1** *(continued)*

Data access programming interfaces in com.ibm.websphere.rsadapter.

Relational resource adapter interface:

(com.ibm.websphere.rsadapter)

Methods have been deprecated in these types:

```
com.ibm.websphere.rsadapter.OracleDataStoreHelper
    public void doSpecialBLobWork(ResultSet rset,
                InputStream[] data, String[] blobColumnNames)
    public String assembleSqlString(String[] blobColumnNames,
                  StringBuffer whereClause, String[] varValues, String tableName)
```

Recommended migration action:

These relational resource adapter deprecated methods do not impact the application.
Note: You will not need to implement these deprecated methods in their subclasses if you have the subclass of OracleDataStoreHelper class. Those deprecated methods will not be called by the WebSphere Application Server runtime.

Webcontainer API modifications:
Note: There are no declared deprecations. The only changes are caused because of a Java API that changed between 1.3 and 1.4.

- Old method signature

  ```
  public String getStackTrace();
  // returns a String representation of the
  exception stack
  ```

- New method signature (JDK 1.4, WebSphere Application Server 5.1)

  ```
  public StackTraceElement[] getStackTrace();
  // returns an array of stack trace
  elements
  ```

- Replacement method (WebSphere Application Server 5.1) (a replacement method, which carries on the old functionality has been provided):

  ```
  public String getStackTraceAsString();
  // returns a String representation
  of the Exception Stack
  ```

Application services features

Data access binaries -- Common Connector Framework:

The following jar files will be deprecated in V5.1:

- ccf.jar
- ccf2.jar
- recjava.jar
- eablib.jar

Recommended migration action:

The J2EE Connector Architecture solution should be used instead of the Common Connector Framework.

For more information on the usage (tools and runtime) of the J2EE Connector Architecture see Data access from J2EE Connector Architecture applications.

Table 55. **Deprecated features in V5.1** *(continued)*

Setting the XA partner log directory via the 'TRANLOG_ROOT' variable is deprecated in V5.1. **Recommended migration action:** The setting currently stored in the TRANLOG_ROOT variable (if any) will need to be added to the Transaction Service panel for any servers who wish to use the XA partner log. If the default location is to be used, then no action is required. The Transaction Service panel can be found on the Administrative Console by selecting Application Servers on the left, choosing the application server to be modified, and selecting Transaction Service on the panel that is displayed. The directory currently in TRANLOG_ROOT should be entered in the Logging Directory box on the panel.
J2EE resource features
No deprecated features.
Security features

Security programming interfaces:
• The API is being deprecated for
`com.ibm.websphere.security.auth.WSPrincipal.getCredential()`
Recommended migration action:
Instead of getting the WSCredential from the principal, you should now use one of the following methods to get the Subject which contains the WSCredential:
– The RunAs Subject is the Subject used for outbound requests.
– The Caller subject is the Subject that represents the authenticated caller for the current request.
– The methods to use to get the runAs and caller subjects are
`com.ibm.websphere.security.auth.WSSubject.getRunAsSubject()`
and
`com.ibm.websphere.security.auth.WSSubject.getCallerSubject()`
respectively.
• The interface is being deprecated in
`com.ibm.websphere.security.auth.WSSecurityContext`
Recommended migration action:
Use JAAS for any authentication related functionality.
• The exception is being deprecated in
`com.ibm.websphere.security.auth.WSSecurityContextException`
Recommended migration action:
Use JAAS for any authentication related functionality.
• The class is being deprecated in
`com.ibm.websphere.security.auth.WSSecurityContextResult`
Recommended migration action:
Use JAAS for any authentication related functionality.

The Integrated Cryptographic Services Facility (ICSF) authentication mechanism is deprecated in V5.1 and will be removed in V6.1. **Recommended migration action:** Use the Lightweight Third Party Authentication (LTPA) instead.
Environment features
No deprecated features.
System administration features

Table 55. **Deprecated features in V5.1** *(continued)*

The following class is deprecated: `com.ibm.websphere.rsadapter.DB2390DataStoreHelper` **Recommended migration action:** If you currently use the DB2390DataStoreHelper class for the DB2 Legacy CLI-based provider when you are accessing data, you should now use the DB2DataStoreHelper class. If you currently use the DB2390DataStoreHelper class for the DB2 Universal JDBC provider driver when you are accessing data, you should now use the DB2UniversalDataStoreHelper class.
Performance features
No deprecated features.
Problem determination features
No deprecated features.

Table 56. **Deprecated features in V5.0.2**

Installation and migration tools
No deprecated features.
Servers and clustering features
No deprecated features.
Application programming model and container support features
Apache SOAP channel in Web services gateway. **Recommended migration action:** Gateway services should be deployed to the SOAP HTTP channel instead of the Apache SOAP channel. The Endpoint (URL) of the service will be different for this channel and therefore client programs that are talking to the gateway will need to use the new service Endpoint.
Apache SOAP, WEBSJAVA.SOAP: • soap.jar, • wssoap.jar **Recommended migration action:** See Migrating Apache SOAP Web services to Web Services for J2EE for more information.
Application services features

Table 56. **Deprecated features in V5.0.2** *(continued)*

Data access programming interfaces in com.ibm.websphere.rsadapter.

Relational resource adapter interface:

(com.ibm.websphere.rsadapter)

Methods have been deprecated in these types:

- com.ibm.websphere.rsadapter.DataStoreHelper

```
public int processSQL(java.lang.String.sqlString, int isolevel,
         boolean addForUpdate, boolean addextendedForUpdateSyntax);
public DataStoreAdatperException mapException(DataStoreAdapterException e);
```

- com.ibm.websphere.rsadapter.GenericDataStoreHelper

```
public int processSQL(java.lang.String.sqlString, int isolevel,
         boolean addForUpdate, boolean addextendedForUpdateSyntax);
public DataStoreAdatperException mapException(DataStoreAdapterException e);
```

- com.ibm.websphere.rsadapter.WSCallHelper

```
public static DataStoreHelper createDataStoreHelper(String dsClassName)
```

Recommended migration action:

These relational resource adapter deprecated methods do not impact the application.
Note: You will not need to implement these deprecated methods in their subclasses if you have the subclass of GenericDataStoreHelper. Those deprecated methods will not be called by the WebSphere Application Server runtime.

For com.ibm.websphere.rsadapter.WSCallHelper, please use the getDataStoreHelper(datasource) method to get a DataStoreHelper object.

J2EE resource features

No deprecated features.

Security features

No deprecated features.

Environment features

No deprecated features.

System administration features

The DB2390DataStoreHelper and the DB2390LocalDataStoreHelper classes.

Recommended migration action:

The DB2DataStoreHelper class now gives all the required helper information needed for the providers that currently use the DB2390DataStoreHelper and the DB2390LocalDataStoreHelper classes.

The DB2 390 Local JDBC Provider (RRS).

Recommended migration action:

This provider is replaced by the DB2 zOS Local JDBC Provider (RRS).

Performance features

No deprecated features.

Problem determination features

No deprecated features.

Table 57. **Deprecated features in V5.0.1**

Installation and migration tools

No deprecated features.

Servers and clustering features

Table 57. **Deprecated features in V5.0.1** *(continued)*

No deprecated features.
Application programming model and container support features
No deprecated features.
Application services features
Data access programming interfaces in com.ibm.websphere.rsadapter. Relational resource adapter interface: (com.ibm.websphere.rsadapter) Methods have been deprecated in these types: • `com.ibm.websphere.rsadapter.DataStoreHelper` `public int processSQL(java.lang.String sqlString, int isolevel);` • `com.ibm.websphere.rsadapter.GenericDataStoreHelper` `public int processSQL(java.lang.String sqlString, int isolevel);` • `com.ibm.websphere.rsadapter.DB2390DataStoreHelper` `public int processSQL(java.lang.String sqlString, int isolevel);` **Recommended migration action:** These relational resource adapter deprecated methods do not impact the application. **Note:** You will not need to implement these deprecated methods in their subclasses if you have the subclass of com.ibm.websphere.rsadapter.GenericDataStoreHelper. Those deprecated methods will not be called by the WebSphere Application Server runtime.
J2EE resource features
No deprecated features.
Security features
No deprecated features.
Environment features
No deprecated features.
System administration features
No deprecated features.
Performance features
No deprecated features.
Problem determination features
No deprecated features.

Table 58. **Deprecated features in V5.0**

Installation and migration tools
No deprecated features.
Servers and clustering features
No deprecated features.
Application programming model and container support features
No deprecated features.
Application services features

Table 58. **Deprecated features in V5.0** *(continued)*

The following three methods from com.ibm.websphere.appprofile.accessintent.AccessIntent are deprecated:
`public boolean getPessimisticUpdateHintWeakestLockAtLoad();`
`public boolean getPessimisticUpdateHintNoCollision();`
`public boolean getPessimisticUpdateHintExclusive();`
This is a base api.
Recommended migration action:
Rather than using the three deprecated methods on the AccessIntent interface, developers should use the following method from the same interface:
`public int getPessimisticUpdateLockHint();`
The possible return values are defined on the AccessIntent interface:
`public final static int PESSIMISTIC_UPDATE_LOCK_HINT_NOCOLLISION = 1;`
`public final static int PESSIMISTIC_UPDATE_LOCK_HINT_WEAKEST_LOCK_AT_LOAD = 2;`
`public final static int PESSIMISTIC_UPDATE_LOCK_HINT_NONE = 3;`
`public final static intPESSIMISTIC_UPDATE_LOCK_HINT_EXCLUSIVE = 4;`

Web application programming interfaces -- Various Version 5 methods in com.ibm.websphere.ServletErrorReport
J2EE resource features
No deprecated features.
Security features
No deprecated features.
Environment features
No deprecated features.
System administration features
No deprecated features.
Performance features
Performance Monitoring Infrastructure -- Various Version 5 public methods in: • com.ibm.websphere.pmi.stat.StatsUtil **Recommended migration action:** There is no replacement for StatsUtil. • com.ibm.websphere.pmi.PmiJmxTest **Recommended migration action:** Use PmiClient.findConfig(). • com.ibm.websphere.pmi.client.PmiClient **Recommended migration action:** The getNLSValue (String key) is replaced by getNLSValue (String key, String moduleID).
Problem determination features
No deprecated features.

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, New York 10594 USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
522 South Road
Poughkeepsie, NY 12601-5400
USA
Attention: Information Requests

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks and service marks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

- AIX
- AS/400
- CICS
- Cloudscape
- DB2
- DFSMS
- Domino
- Everyplace
- iSeries
- IBM
- IMS
- Informix
- iSeries
- Language Environment
- Lotus
- MQSeries
- MVS
- OS/390
- OS/400
- RACF
- Rational
- Redbooks
- RMF
- SecureWay
- SupportPac
- Tivoli
- ViaVoice
- VisualAge
- VM/ESA
- VTAM
- WebSphere
- z/OS
- zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.