

WebSphere Application Server V4.0.1 for z/OS and
OS/390:



System Management User Interface

WebSphere Application Server V4.0.1 for z/OS and
OS/390:



System Management User Interface

Note

Before using this information and the product it supports, be sure to read the general information under Appendix D, "Notices", on page 161.

Sixth Edition (June 2003)

This edition applies to WebSphere Application Server V4.0.1 for z/OS and OS/390 (5655-F31), and to all subsequent releases and modifications until otherwise indicated in new editions.

The most current versions of the WebSphere Application Server V4.0.1 for z/OS and OS/390 publications are at this Web site: http://www.ibm.com/software/webservers/appserv/zos_os390

© Copyright International Business Machines Corporation 2001, 2002, 2003. All rights reserved.

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

Contents

About this book.	vii
Highlighting conventions.	vii
Related information	vii
How to send your comments	viii

Summary of changes. xi

Part 1. Introduction 1

Chapter 1. Steps to install the system management user interface 5

Establish communication to the host	5
Host names.	5
HOSTS file	5
Additional information	6
Install the system management user interface on your workstation.	6
Define workstation environment variables for login options	6
BBONPARM	6
BBONDEBUG	7
Example.	7
Start the Administration or Operations application.	8
Login dialog	8
Login options	9
User profile	10
Define administrator identities	10
Deinstall the system management user interface from your workstation.	11

Part 2. Administration application 13

Chapter 2. Administration user interface 15

Main window	15
Tree for administration	15
Expand or collapse objects in the tree.	17
Icons in the tree	17
Properties form	18

Chapter 3. Administration tasks 19

Change a configuration — overview	19
Icons for conversations	20
States of a conversation	20
Change a configuration - refinement	22
Create a model	22
Add an object to a model.	23
Delete an object from a model	23
Modify an object in the model	24
Verify a model's validity	24
Commit a model	24
Display instructions	25
Complete the instructions	25

Activate an image	25
Example of modeling a configuration to add an application	26
Define the security class of a server	27
Import an application	31
Deploy J2EE applications.	32
Introduction	32
Step-by-step instructions	37
References.	45
Add a J2EE resource type.	45
Prepare performance recording	45
Migrate a test server to a production system	47
Prepare for cold start	48
Modify the IP-address.	49
Modify environment variables	49
Server instance run-time environment variables	50

Chapter 4. Administration objects 53

Location of Objects in the Tree	53
Application	54
Location in the tree:	55
Properties:.	55
Actions:	55
Application family	55
Location in the tree:	55
Properties:.	56
Actions:	56
Class	56
Location in the tree:	56
Properties:.	56
Actions:	57
Client interface	57
Location in the tree:	57
Properties:.	57
Actions:	58
Container	58
Location in the tree:	58
Properties:.	58
Guidelines for Container Properties	61
Actions:	61
Conversation	61
Location in the tree:	62
Properties:.	62
Actions:	62
DLL	63
Location in the tree:	63
Properties:.	63
Actions:	64
Home	64
Location in the tree:	64
Properties:.	64
Actions:	65
J2EE application.	66
Location in the tree:	66
Properties:.	66

Actions:	66
J2EE component:	66
Location in the tree:	67
Properties:	67
Actions:	67
J2EE module:	67
Location in the tree:	68
Properties:	68
Actions:	68
J2EE resource:	68
Location in the tree:	69
Properties:	69
Actions:	69
J2EE resource connection:	70
Location in the tree:	70
Properties:	70
J2EE resource instance:	70
Location in the tree:	71
Properties:	71
Actions:	71
J2EE server:	71
Location in the tree:	72
Properties:	72
Actions:	80
Logical resource mapping:	80
Location in the tree:	81
Properties:	81
Actions:	82
Logical resource mapping connection:	82
Location in the tree:	83
Properties:	83
Actions:	83
Logical resource mapping instance:	83
Location in the tree:	84
Properties:	84
Actions:	87
Server (Managed Object Framework):	87
Location in the tree:	87
Properties:	88
Actions:	95
Server instance:	96
Location in the tree:	96
Properties:	96
Actions:	98
Sysplex:	98
Location in the tree:	99
Properties:	99
Actions:	100
System:	101
Location in the tree:	101
Properties:	101
Actions:	101

Chapter 5. Instructions for z/OS tasks 103

Instructions overview	103
Instructions completion summary	103
Instructions task detail	104
Save the instructions	104

Part 3. Operations application . . . 105

Chapter 6. Operation user interface 107

Main window	107
Filter the operations window	107
Icons for operations	108
Properties form.	108
Work request list (not yet supported)	108

Chapter 7. Operation tasks. 111

Start a server or server instance	111
Stop a server or server instance	111
Perform a warm start for a server or server instance	111

Chapter 8. Operation objects. 113

J2EE server	113
Properties:	113
Actions:	118
Server (MOFW).	118
Properties:	119
Actions:	123
Server instance	123
Properties:	124
Actions:	125

Part 4. Messages and diagnosis 127

Chapter 9. Message log 129

Filter the message log	130
Print the message log.	131

Chapter 10. Messages. 133

Chapter 11. Trace and debug facilities 135

Traces	135
Enable tracing	135
View traces	135
Communications trace	136
Enable communications tracing	136
Debug facility	136
Enable debugging	136
View debug information.	136

Part 5. Appendixes 137

Appendix A. User Interface 139

Menu bar.	139
Menu bar actions for the Administration application	139
Menu bar actions for the Operations application	144
Menu bar actions for the message log	145
Tool bar	148
Pop-up menus	149
Customize the user interface	149
Get help	149
Use the mouse and keyboard	150
Select objects and actions	150
Expand an object's branch in the tree	151
Clear an input field	151

Use an active screen reader to have messages in the status bar read	151
Shortcut keys	151
Select an object or area	151
Display actions	152
Copy, cut and paste information	152
Search instructions or the message log	152
Refresh the window	153

Appendix B. Commands for operations. 155

Controlling server instances	155
Start a server instance	155
Cancel a server instance	156
Stop a server	156
Controlling application environments	156

Display application environments	156
Restart application environments	157

Appendix C. DDL keyword naming conventions 159

Appendix D. Notices 161

Examples in this book	162
Trademarks	163
Programming interface information	164

Glossary 165

Index 167

About this book

This book contains the online help information for the WebSphere for z/OS Administration and Operations applications, in slightly different form. The information is for those users who prefer the printed page. It describes the applications and explains how to use them to perform administration and operations tasks.

The product name is *WebSphere Application Server V4.0.1 for z/OS and OS/390*, called either the *Application Server* or *WebSphere for z/OS*.

When we speak of z/OS, we mean both, z/OS and OS/390, unless stated otherwise.

Highlighting conventions

Throughout this book,

- *Italics* is used for
 - book titles,
 - emphasis,
 - term definitions,
 - options / variables / parameters
- **Boldface** is used for
 - check box labels,
 - choices in menus,
 - column headings,
 - entry fields,
 - field names in windows,
 - menu-bar choices,
 - menu names,
 - radio button names, and
 - spin button names
- Monospace is used for
 - coding examples,
 - commands and subcommands,
 - entered data,
 - file names,
 - group and user IDs,
 - message text, and
 - path names.
- Underlined settings are
 - default values

Related information

Other books in the WebSphere for z/OS library are:

- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Program Directory*, GI10-0680, describes the elements of and the installation instructions for WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Licensed Information*, LA22-7855, describes the license information for WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, GA22-7834, describes the planning, installation, and customization tasks and guidelines for WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis*, GA22-7837, provides diagnosis information and describes the messages and codes associated with WebSphere for z/OS.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration*, SA22-7835, describes system operations and administration tasks.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*, SA22-7836, describes how to develop, assemble, and install J2EE applications in a WebSphere for z/OS J2EE server. It also includes information about migrating applications from previous releases of WebSphere Application Server for OS/390, or from other WebSphere family platforms.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling CORBA Applications*, SA22-7848, describes how to develop, assemble, and deploy CORBA applications in a WebSphere for z/OS (MOFW) server.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: System Management Scripting API*, SA22-7839, describes the functionality of the WebSphere for z/OS Systems Management Scripting API product.
- *WebSphere Application Server V4.0.1 for z/OS and OS/390: Migration*, GA22-7860, describes migration procedures for WebSphere for z/OS.

The documentation for the Administration and Operations applications also includes these files:

- A readme file (`Readme.txt`) contains late technical updates.
- Release notes (`Relnotes.htm`) contains more detailed late technical updates related to installing and using the applications, including details about establishing communication between the workstation and the host.

You might also need to refer to information about other products in the WebSphere Family, or to information that is common to all WebSphere Family products. All of this information is available through links at the following Internet locations (z/OS or OS/390):

<http://www.ibm.com/servers/eserver/zseries/zos/>
<http://www.ibm.com/servers/s390/os390/>

Another books that you might find particularly helpful is *Building Business Solutions with WebSphere*, SC09-4432

How to send your comments

Your feedback is important in helping to provide the most accurate and highest quality information. You can e-mail your comments to:

`wasdoc@us.ibm.com`.

Be sure to include the document name and number, the WebSphere Application Server version, 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.

Summary of changes

Summary of changes for SA22-7838-06 WebSphere for z/OS

This revision is due to minor editorial changes.

Summary of changes for SA22-7838-05 WebSphere for z/OS

This update includes the following information:

- Documentation updates change the following properties in “J2EE server” on page 71:
 - Production J2EE Server
 - Debugger allowed
 - Isolation policy
- Documentation updates change the following properties in “Server (Managed Object Framework)” on page 87:
 - Production Server
 - Debugger allowed
 - Isolation policy
- The code synchronizes the default JNDI name that is used on the administrative console with 390fy command options. Documentation updates reflect these changes. For more information, see “Step-by-step instructions” on page 37.

Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of changes for SA22-7838-04 WebSphere for z/OS

This update includes the following information:

- Previously, you used the z/OS Application Assembly Tool to assemble Enterprise JavaBeans™ (EJBs) and Web applications into Java™ 2 platform, Enterprise Edition (J2EE) applications. Now you can assemble applications with the z/OS Application Assembly Tool, the Application Assembly tool delivered on the workstation, or WebSphere Studio Application Developer. Documentation updates reflect the new options for assembling applications.
- New documentation explains the IIOP Firewall port property and the SSL Firewall port property. For more information, see “Server instance” on page 96 and “Server instance” on page 123.

Summary of changes for SA22-7838-03 WebSphere for z/OS

This update includes the following information:

- The description of the new sysplex-level configuration extension for connection management that appears in the WebSphere for z/OS Administration application, previously contained in WebSphere Application Server V4.0.1 for z/OS and OS/390: WebSphere for z/OS-Supported Connectors, which provided the documentation for APAR PQ55873, PTF UQ99329, Service Level W401030.
- A description of the new ALT+z function that is used to enable messages in the status area to be read by an activate screen reader that was made available with PTF PQ55873.
- A description of an alternate method for specifying the CLASSPATH for a JAR file (APAR PQ57023).
- Additional information about editing environment variables to reflect code changes shipped in PTFs PQ58665 and PQ54774.

**Summary of changes
for SA22-7838-00
WebSphere for z/OS**

This book contains information previously presented in SC33-6587-01, which supports Component Broker Version 3.02.

The following is a summary of changes to this information:

- Component Broker has been renamed to *WebSphere Application Server V4.0.1 for z/OS and OS/390*, which we will shortly call *WebSphere for z/OS* (although it still also applies to OS/390) or — more familiar — *Application Server*.
- The function *Install J2EE application* has been added to the Administration application. Thus, the chapter “Deploy J2EE applications” on page 32 has been added to the book.
- New objects have been defined to the Administration tree. They are described in the chapters
 - “J2EE server” on page 71,
 - “J2EE application” on page 66,
 - “J2EE module” on page 67,
 - “J2EE component” on page 66,
 - “J2EE resource connection” on page 70,
 - “J2EE resource” on page 68, and
 - “J2EE resource instance” on page 70.
- The object “J2EE server” on page 113 has been added to the Operations application.

The *datasources* of the Component Broker V4.0 Early Availability Version have been renamed to *J2EE resources*.

The following APAR required changes to this book:

APAR MD09319

Changes have been made to the following topics:

- “Icons for conversations” on page 20
- “Activate an image” on page 25

Part 1. Introduction

This manual refers to the WebSphere for z/OS Administration and Operations applications. These applications, which run on your Windows workstation and communicate with the System Management server, are often called the "system management user interface" or "system management enhanced user interface (SMEUI).

This guide describes how to use the Administration and Operations applications. It includes the parts:

- Introduction,
which contains a short introduction on the system management user interface, the steps to prepare a workstation for the installation, and, finally, the steps to install the Administration and Operations applications
- Administration,
which contains a description of the Administration application
- Operation,
which contains a description of the Operations application
- Messages and diagnosis,
which contains a description of the message and diagnosis functions of both the Administration and Operations applications.

Note that, to manage WebSphere for z/OS clients, you use the System Manager of the Application Server for NT.

The Application Server Administration application and the Operations application are separated:

Administration application

The *Administration application* is delivered to manage administration tasks. It allows you to display and modify the Application Server configuration, that is, the WebSphere for z/OS applications and the environment in which they run.

The configuration is displayed in the form of a tree of the objects you manage. The tree shows the relationship of the various objects. The objects include those that you create and modify, such as sysplexes, systems, J2EE servers, and server instances, as well as objects that you bring into the model when you install an application, such as J2EE application or application.

The highest level object in the tree is called a *conversation*. It contains an Application Server configuration.

The following picture shows an example of an administration window:

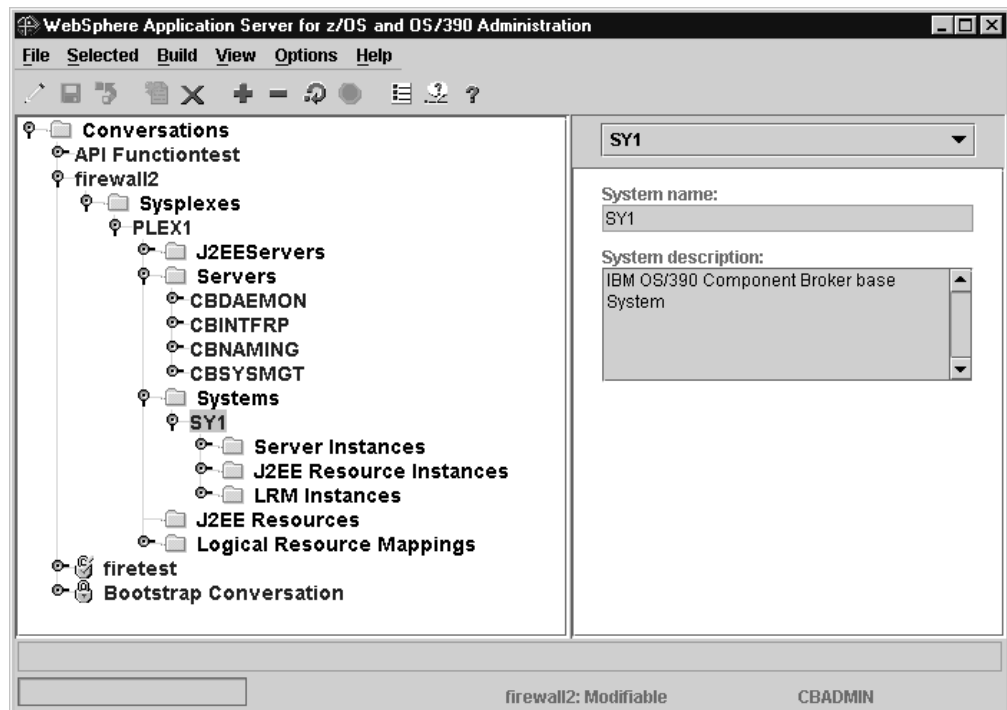


Figure 1. Example for an Administration Window

Operations application

The *Operations application* manages operating tasks. It lets you manage the WebSphere for z/OS servers and server instances.

Each server and server instance in the currently active configuration is represented by an icon. You can display the status of server instances and the properties of each object. You can start and stop servers and server instances.

Figure 2 shows an example of an operations window:

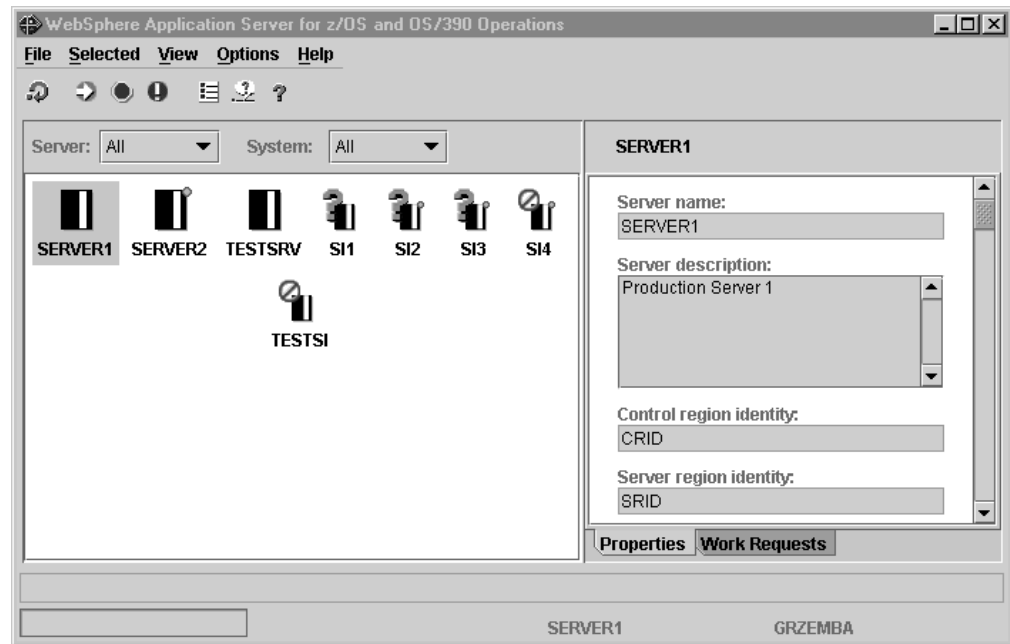


Figure 2. Example of an Operations Window

Chapter 1. Steps to install the system management user interface

This chapter describes the steps you have to take to install and run the system management user interface on your workstation:

1. Establish communication between the workstation and the host via TCP/IP
2. Install the system management user interface on your workstation
3. Define environment variables for login options
4. Start the Administration and Operations applications
5. Define administrator identities to control access to the Administration and Operations applications

and finally it describes how to

- Deinstall the Administration and Operations applications from your workstation

Establish communication to the host

The Administration and Operations applications both use TCP/IP to communicate between your workstation and the host. You may need to update your HOSTS file with the appropriate host names on your workstation to establish this communication:

Host names

All IP names used to communicate with the host must be defined either to your domain name server (DNS) or in your workstation HOSTS file. The IP names are:

- The *bootstrap server* IP name

The name associated with the initial connection to the host. It is defined by the RESOLVE_IPNAME parameter of the z/OS sysplex environment file.

- The *naming server* IP name

A generic name associated with your naming server and defined by the DAEMON_IPNAME parameter of the z/OS sysplex environment file. If you have more than one name server (federated name space) you must ensure that all the name servers' host names needed by the workstation can be resolved.

- The *host name of each system* in the sysplex in which WebSphere for z/OS runs.

HOSTS file

You use the HOSTS file to define the host names to your workstation:

Your workstation may have a HOSTS file that is used to associate TCP/IP host names with TCP/IP addresses. Ordinarily, TCP/IP addresses are associated with host names by the domain name server (DNS) for your system. If your system cannot resolve a host name using your domain name server, then it uses the HOSTS file.

The HOSTS file is generally found in

- c:\winnt\system32\drivers\etc if you are running Windows 2000 or Windows NT
- c:\windows if you are running Windows 95 or Windows 98.

If you do not have a HOSTS file, you can create one using any text editor and placing it in the appropriate directory. You may have a sample HOSTS file, LMHOSTS.SAM, that you can use to model your new HOSTS file.

Each entry in the HOSTS file consists of an IP address and a corresponding IP name. Each entry is surrounded by blanks and is on a single line. To make an association between a TCP/IP host name and address, you add an entry to the file. For example, to add an entry for a host called test at IP address 9.1.1.1 defined in domain acme.com, add the following lines:

```
9.1.1.1 test
9.1.1.1 test.acme.com
```

After you have updated your HOSTS file, save it. You can test your changes by opening a command window and issuing the ping command with the name you just added. For example, ping test.

Additional information

Setting up TCP/IP for WebSphere for z/OS is described in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*. Configuration of TCP/IP with z/OS is described in *OS/390 eNetwork Communication Server IP Configuration*. For the latest information, refer to the Release Notes (Relnotes.htm in the WebSphere for z/OS folder on your workstation).

Install the system management user interface on your workstation

After the Application Server is installed on z/OS, you can download the applications via FTP as a *binary* file. The file bboninst.exe is located on the host in the path /usr/lpp/WebSphere/bin.

Execute bboninst.exe. Installation instructions are displayed automatically.

For more information, see *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration*.

Define workstation environment variables for login options

BBONPARM

The BBONPARM workstation environment variable is used to pre-assign various login options for the Administration and Operations applications. These values may be overwritten when you start the Administration or Operations application (see "Start the Administration or Operations application" on page 8).

See the help for your operating system for information on how to specify environment variables.

Login options are in the format *-option* followed by a value, if any. For example, to specify that port 900 is to connect the naming server with this login option:
-nameport 900

The login options are:

-bootstrapserv <nameserver>
Names the default IP name for the naming server

-bootstrapport <port>

Names the port used to connect to the naming server

-loginuser <user ID>

Names the default user ID for the login

-loginpassword <password>

Names the default password for the login

-commtrace

When specified, indicates that the communications trace should be started.

Refer to Part 4, "Messages and diagnosis", on page 127 for further information.

-trace When specified, indicates that the internal trace should be started.

Refer to Part 4, "Messages and diagnosis", on page 127 for further information.

-debug

When specified, indicates that debug mode is active.

Refer to Part 4, "Messages and diagnosis", on page 127 for further information.

-newprofile

When specified, indicates that a new profile should be created to save user preferences

When duplicate options are assigned to the BBONPARM environment variable, the last one specified is used.

BBONDEBUG

The BBONDEBUG variable is used to assign various diagnostic levels, and should be used under the direction of IBM service personnel. BBONDEBUG is assigned a numeric value which indicates the diagnostic level. The levels are internally defined, but the most common is level 4. This level causes a console window to be active that displays trace records as various events occur.

Example

As an example of setting the default options for a bootstrap server called test and a default user ID of CBADMIN when running under Windows NT, perform the following steps:

1. Open **Start->Settings->Control Panel->System**
2. Click the Environment tab
3. Define a new variable BBONPARM
4. Assign the values **-bootstrapserver test -loginuser CBADMIN**
5. Click **Set** to set the new variable
6. Close the dialog

The default values will be primed in the login panel the next time the application is started.

Start the Administration or Operations application

To start the Administration or Operations application from your workstation,

- either
 1. Click **Start** on the Windows taskbar.
 2. Point to **Programs**.
 3. Point to **IBM WebSphere for z/OS**.
 4. Click **Administration** or **Operations**. The **Login** dialog appears.
- or, double click on the **Administration**



or on the **Operations**



icon on your desktop.

The **Login** dialog appears.

A screenshot of a Windows-style dialog box titled "Login". The dialog has a title bar with a globe icon on the left and a close button (X) on the right. The main area contains four input fields: "Bootstrap server IP name" (a text box), "Port" (a spin box with "900" displayed), "Userid" (a text box), and "Password" (a text box). At the bottom, there are four buttons: "OK", "Options...", "Cancel", and "Help".

Login dialog

The **Login** dialog box prompts for information that is needed to connect to your z/OS Systems Management server: the IP name of your bootstrap server, the port number, user ID and password.

1. Enter the IP name.

The IP name is the name of your bootstrap server. If the name is not known to your TCP/IP domain name server, you can add an entry for it to the HOSTS file for your workstation. The procedure for doing this is described in "HOSTS file" on page 5.
2. Enter the port number.

The port number for the bootstrap host is generally 900. Check with your systems programmer if the default has been changed.

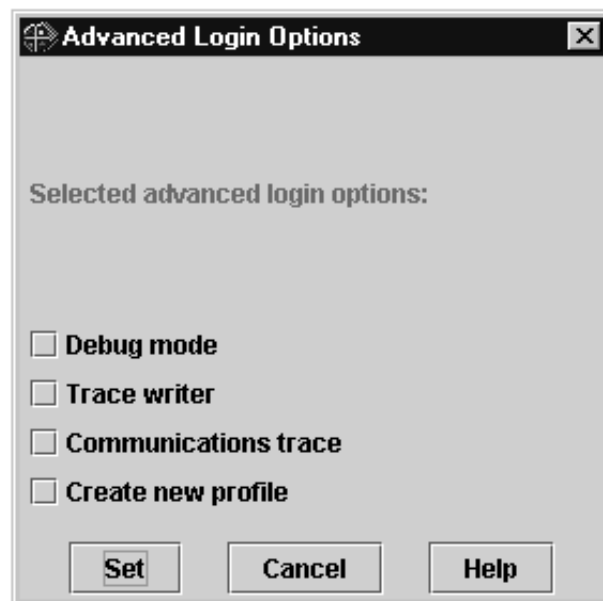
3. Enter your user ID and password.

The user ID and password must be valid for your z/OS system. In addition, the user ID must have been previously defined as an administrator to WebSphere for z/OS . See the *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization* for details on how to define an administrator user ID. If this is the first login to WebSphere for z/OS after installation, the user ID must be CBADMIN. Once you have established a session, you can define additional administrator user IDs for use.

4. If you want to set options, e.g. start tracing, click **Options**. Refer to “Login options” for details.
5. Click **OK** to login.

Login options

The login dialog allows to set various login options. These options are predefined by the workstation environment variable for login options, BBONPARM (refer to “Define workstation environment variables for login options” on page 6), but they can be changed for this session using the **Options** button.



Debug mode

When the **Debug mode** option is specified, the debug mode is active.

Trace writer

The **Trace writer** option causes trace entries to be collected.

Communications trace

The **Communication trace** option starts tracing of communication between the application and the Systems Management Server on z/OS.

Refer to Chapter 11, “Trace and debug facilities”, on page 135 to learn to know how to view debugging information or traces.

Create new profile

The **Create new profile** option will cause a refresh of your saved profile with a new profile containing the application defaults (refer to “User profile” on page 10).

User profile

The Administration and Operations applications each save information about your preferences in a profile. The profile for each application is stored by the Systems Management Server on z/OS and used each time you access the application. The profiles may contain things you have set with the **Options** menu bar choice, such as color choices and confirmation. Because the profile is stored on z/OS, your preferences are in effect regardless of which workstation you use to access the Administration and Operations applications.

If you want to discard the current profile for an application, for example, because it has been corrupted, you can use the **Create a New Profile** option when logging in to the Administration or Operations application. This will refresh your saved profile with a new profile containing the application defaults.

Define administrator identities

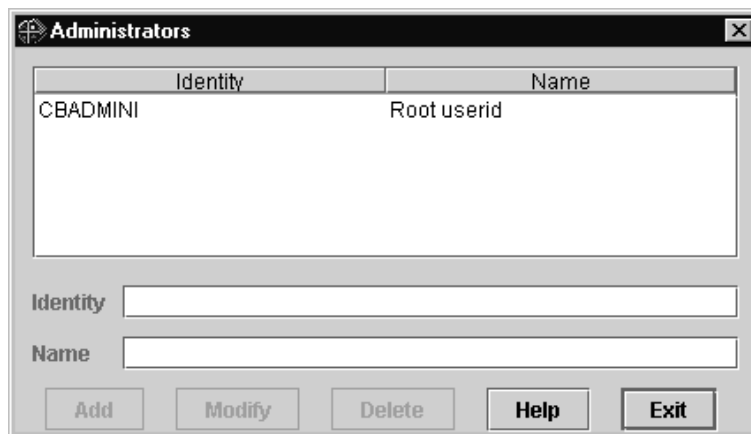
To access the Administration and Operations applications you need an administrator identity, which is an MVS user ID that has been defined as an administrator to WebSphere for z/OS, using the Administrators dialog.

The MVS user ID can be any valid MVS user ID. Give the new administrator user ID the same RACF authorizations as the original administrator, CBADMIN. A default administrator can be defined with the SM_DEFAULT_ADMIN environment variable. See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration* for more information.

To add an administrator identity:

1. From the Administration application, select the **Administrators** action of the **File** menu bar choice.

The Administrators dialog appears.



2. Type the administrator identity, which is an existing MVS user ID, in the Identity field.
3. Type a name in the Name field, if desired. It may be up to 256 characters.
4. Click **Add** .

Note: To use the new administrator identity, you must exit the application and then login with the new identity.

To modify or delete an administrator identity:

1. From the Administration application, select the **Administrators** action of the **File** menu bar choice.
The Administrators dialog appears (see above).
2. Click the identity in the list. The identity and name are displayed in the input fields below the list.
3. Modify the identity and name as desired.
4. Click **Modify** or **Delete**.

Notes:

- When an administrator identity is deleted, the MVS user ID should also be deleted if it is no longer needed.
- When you delete an administrator identity, all conversations belonging to that administrator identity are reassigned to the default administrator. It is recommended that before you delete an administrator, all the conversations assigned to the administrator have the status replaced or deleted.
- You cannot delete the default administrator. The default administrator is defined with the SM_DEFAULT_ADMIN environment variable. See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration* for more information.

Deinstall the system management user interface from your workstation

To deinstall the product from your workstation, perform the following steps:

1. If the Administration or Operations application is running, close it.
2. Click **Start->Settings->Control Panel->Add/Remove Programs**.
3. Find IBM WebSphere for z/OS in the list and click **Remove**.
4. Reply to the confirmation prompt to proceed with the deinstallation.

You should always use the "Add/Remove Programs" option to delete the Administration and Operations applications. Using this option not only deletes all the files and directories, but also removes entries placed in the system registry. Please note that only the files that we originally installed on your workstation will be removed during deinstallation. Files that you generated by the application during operation (e.g. trace files) will not be removed.

Part 2. Administration application

This part describes the Administration application for WebSphere for z/OS. It describes:

- The user interface of the Administration application,
- the administration tasks,
- the objects, their properties and actions that are available,
- the instructions for tasks that have to be performed on z/OS to accomplish the administration tasks.

Chapter 2. Administration user interface

This chapter describes the graphical user interface and the objects that are displayed in the Administration application. These are:

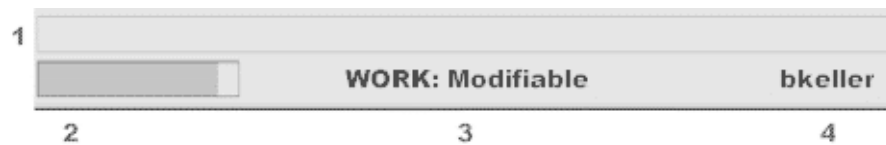
- The main window with the tree of objects and their properties
- The tree of objects in the left-hand frame of the window
- Actions on objects in the tree (expand or collapse)
- Icons in the tree
- The properties form in the right-hand frame of the administration window

General user interface topics are described in Appendix A, "User Interface", on page 139.

Main window

The main window for administration consists of two frames (see Figure 1 on page 2). The left frame shows WebSphere for z/OS objects as a tree. The right frame shows details about the selected object. It is also used to display instructions describing z/OS tasks.

At the bottom of the window is a variety of information, shown in the example below. A *message area* (1) is followed by an *informational line* that includes a *progress bar* (2), which indicates activity of a currently running process. The name and status of the selected conversation (3) is followed by the user ID (4) of the user.



The *Message Log* (refer to Chapter 9, "Message log", on page 129) uses a single frame to show messages issued during the session.

Tree for administration

The Administration application uses a *tree* to graph the Application Server configuration. The tree appears in the left-hand frame of the main administration window. It shows the hierarchy of conversations and the associated objects, such as sysplex, servers, and applications. Objects are grouped by type under a *label* that identifies the type.

Labels have folder icons: .

Example:

The following tree consists of four conversations: the active conversation named "firetest", the working models named "API Functiontest" and "firewall2" and replaced conversation "Bootstrap Conversation".

"firewall2" shows a label for sysplexes, followed by the sysplex named "PLEX1", a label for J2EE servers, a label for servers, followed by four servers, a label for systems, followed by the system named "SY1" with

labels for server instances, J2EE resource instances and LRM instances, a label for J2EE resources, and a label for logical resource mappings.

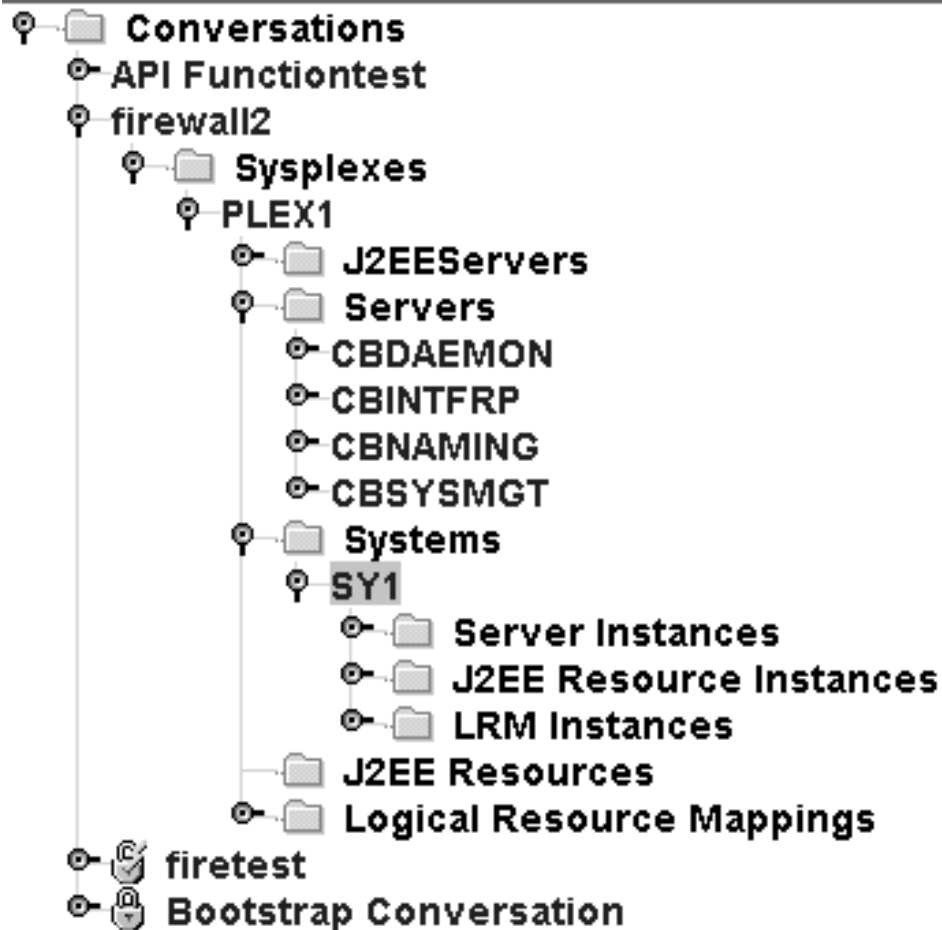


Figure 3. Example for an administration tree

You can scroll the tree up and down, and expand and collapse the tree or branches of the tree.

Icons in the tree also indicate the state of the objects. Some icons, such as those for a locked state and instructions, appear only at the level of the label for the conversation. Others icons, such as the one to indicate a deletion, appear at lower levels in the tree.

The selected object in the tree is highlighted.


The text in the tree changes to indicate activity. For example, when you expand the node next to a label for the first time, the word **Retrieving** is inserted before the label, to indicate that the application is retrieving the objects in that branch from the Systems Management Server on z/OS. Or, when you delete an object, the word **Deleting** is inserted before its name.

You can customize the tree, for example by setting colors or fonts. Refer to "Options" on page 142 for details.

Expand or collapse objects in the tree


You can expand or collapse the tree to show more or fewer levels in the hierarchy:

To expand a branch of the tree and show objects at the next level:



Click on the node button  to the left of the object or double click on the object.

When a branch is expanded, the node points down.

To expand a branch of the tree and all branches below it:


1. Select the object in the tree.
2. Click  on the tool bar or select the **Expand** action of the **View** choice on the menu bar.

To collapse a branch of the tree:

Click  next to the object in the tree, or select the object and either click  on the tool bar or select the **Collapse** action of the **View** choice on the menu bar.

To expand or collapse the entire tree

1. Select the **View** choice on the menu bar.
2. Select the **Expand tree** or **Collapse tree** action.

Note: Expanding the entire tree may take some time to complete. You can stop the expansion at any time by clicking  on the toolbar. The expansion is stopped at the next logical point, after the expansion of the currently expanding branch is complete.

A progress bar underneath the left frame indicates that expansion is in process. In addition, the word **Expanding** is inserted at the appropriate label in the tree.

Icons in the tree


Each object in the tree is represented by an *icon* which indicates its state.


Icons for conversations in their different states are:

 ,  ,  ,  ,  , and  .

Refer to “Icons for conversations” on page 20 for an explanation of these icons.

Icons common to all objects in the tree are:


 Deleted (label level)

 Deleted (object level)

Icons that identify parts of the tree are:

 Node for a branch that is expanded.

 Node for a branch that is collapsed.

 Label for a type of object; objects of that type appear under the label.

Properties form

The *properties form* displays the values for the selected Application Server object. It appears in the right frame of the window.

Details for the fields on the properties form for each object are described in Chapter 4, “Administration objects”, on page 53.

You can modify the properties to change their values.

To display the properties:

1. Select the object in the left frame by clicking it once with the left mouse button.
2. The properties form is displayed in the right frame, in browse mode.

If the instructions for the conversation are being displayed, you can switch back to displaying properties:

1. Position the mouse pointer on the instructions.
2. Press the right mouse button to display actions in a pop-up.
3. Select the **Restore properties** action.

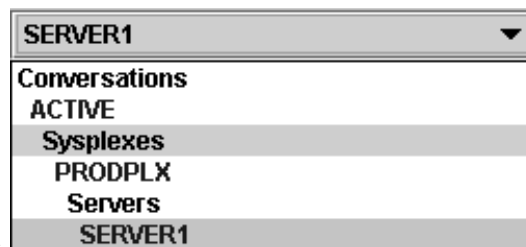
Note that returning to the instructions requires the application to obtain them from the Systems Management Server on z/OS.

To add or change values:

You can only change the values of an object that is part of a working model (refer to “Change a configuration — overview” on page 19). When you add a new object to the model, a properties form is opened in edit mode. When you open a properties form for an object that is already defined, the form is in browse mode. Refer to “Modify an object in the model” on page 24 to learn how to modify properties of an object.

To display the hierarchy of the object:

The first field in the properties form shows the name of the object. You can locate the object in the hierarchy of objects by clicking ▼ to the right of the name. This displays an abbreviated form of the tree, with only those labels and objects that pertain to the selected object.



For example:

You can select any object in this list. This causes the object to be selected in the tree and its properties to be displayed.

Chapter 3. Administration tasks

This topic describes how to use the Administration application to administer your Application Server. It describes how to modify a conversation and how to perform the following administration tasks:

- Change a configuration
- Define the security class of a server
- Import an application
- Prepare performance recording
- Migrate a test server to a production server
- Prepare for cold start
- Modify the IP address
- Modify environment variables

Change a configuration — overview

The configuration is displayed in the form of a tree of objects. The highest level object in the tree is called a *conversation*. It represents a WebSphere for z/OS configuration. The tree shows the relationship of the various objects which make up the configuration. The objects include those that you create and modify, such as sysplex, system, J2EE server or server, server instance, as well as objects that you bring into the model when you import an application, such as J2EE application or application.

The current configuration is called the *active image*. You can display the active image, but you cannot make changes to it.

To change the configuration, you create a new conversation, modify it, and then make this one the active image:

1. Create a new conversation which is called a *model*.
2. Modify the model:

You can add, delete or modify the objects in the model. To add a J2EE application to the model, you install the J2EE application on a J2EE server. Installing the J2EE application causes objects to be created in the tree. These objects cannot be modified.

3. Verify that the model is still valid.
4. Commit the model.

Commit locks the model so that no further changes can be made. A conversation that has been committed is called an *image*.

5. Complete the instructions for the image.

When you commit a model, you receive a set of instructions to help you complete additional z/OS tasks that are required to define how your sysplex will handle this new work.

6. Activate the image to put it into operation.

It replaces the previous active image.

Note: Work successively as it is described above. If you or you and another administrator created two models from the same active image, only one of them


would have the chance to replace it. Then, only a copy of this new active image can be determined to be the next active image.


The following figure shows the stages of a conversation as it goes from model to active image.




Icons for conversations


Each conversation in the tree is represented by an icon which indicates its state.


 (silver lock). The conversation is in the state "locked and replaced". You cannot change the properties of this object, but you can delete it.

 This conversation is the current active image, its state is "locked and active". You cannot change the properties of this object nor delete it.

 (golden lock). The conversation is in the state "replace pending". The activation process failed. You cannot delete this object.

 The conversation is an image in the state "committed". The instructions must be completed. You can delete this object.

 The conversation is an image in the state "ready for activate" or "activate in progress". In the state "activate in progress" the conversation is active, even if the activation process failed. The instructions are completed. You cannot delete this object.



 (blue lock). The active conversation has been saved to the host. WebSphere for z/OS has been stopped. No further changes can be made to any conversation until WebSphere for z/OS has been restarted (cold start) and the Administration application has been restarted. (Refer to "Prepare for cold start" on page 48 for more information.)









States of a conversation

After we know the steps how to create a new active image, let us have a detailed look at the different states of a conversation, because the state of a conversation changes dependent on the actions that are performed by the user.

We now see what happens to the former active image **a**, when a new image **b** is created.

Case 1: Successful activation













Conversation a	action	Conversation b
 "Active Image", State: "locked and active"		
	<i>add</i> conversation	
 "Active Image", State: "locked and active"		"Model", State: "working"
	<i>modify</i> b	

Conversation a	action		Conversation b
 "Active Image", State: "locked and active"			"Model", State: "working"
	<i>validate b</i>		
 "Active Image", State: "locked and active"			"Model", State: "working"
	<i>commit b</i>		
 "Active Image", State: "locked and active"			"Image", State: "committed"
	<i>complete instructions for b</i>		
 "Active Image", State: "locked and active"			"Image", State: "ready for activate"
	<i>activate b (successful)</i>		
 State: "locked and replaced"			"Active Image", State: "locked and active"

a has been replaced by **b**. Note that the new active image always is created from the former active image. Now, as **b** is the active image, no other modified copy of **a** ever has the chance to replace **b**.

Case 2: Unsuccessful activation

If image **b** does not really match the configuration - for example, you enabled DCE security for a server, but there is no DCE installed - then you either have to install DCE or you have to modify **b**, that is, create a new conversation (which really is a copy of **b**), let us call it **c**, then modify, verify, and commit it, complete the instructions for **c**, and finally activate it:

Conversation a	Conversation b	Conversation c
 "Active Image", State: "locked and active"	 "Image", State: "ready for activate"	
<i>activate b (not successful)</i>		
 State: "replace pending" or "locked and replaced"	 "Active Image"! State: "activate in progress"	
<i>add conversation</i>		
 State: "replace pending" or "locked and replaced"	 "Active Image", State: "activate in progress"	"Model", State: "working"
<i>modify, validate and commit c; complete instructions for c</i>		
 State: "replace pending" or "locked and replaced"	 "Active Image", State: "activate in progress"	 "Image", State: "ready for activate"
<i>activate c (successful)</i>		
 State: "locked and replaced"	 State: "locked and replaced"	 "Active Image", State: "locked and active"

If c hadn't been correct, you would have created another conversation and thus a chain of conversations in the state "replace pending", all marked with a golden lock. As soon as the activate action of the latest image is successful, each golden lock is converted into a silver lock, and the conversations turn their state into "locked and replaced".



Note: If two administrators work on a copy of the same conversation, be aware that only one of you can replace it. The other copy then will be obsolete. Thus, work successively!

Change a configuration - refinement




The following outline refines the steps you take to change a configuration and add a new application to it:

1. Plan the configuration. You will find planning information related to such things as performance and security in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, as well as in the description of the WebSphere for z/OS objects.
2. Create a model configuration. The model begins as a copy of the active image.
3. Create, delete or modify objects in the model, to define where the application will run and how the application will access data, such as with DB2, CICS or IMS. For example, you might:
 - Create any necessary J2EE servers or servers (MOFW) and server instances, where the application will run. Most of the properties, including those that control security, are defined at the server level. They are then inherited by the associated server instances. Server instances have an association with a particular system.
 - Create the J2EE resource for the required subsystem, and create the associated J2EE resource instance. These describe how the J2EE application will access data. J2EE resource instances have an association with a particular system.
4. Verify that the model is still valid. It's good practice to do this each time you make a change.
5. Import the WebSphere for z/OS application onto a server in the model. The application will run on all the server instances defined to the server after the conversation is activated.
6. Commit the model.
7. Display the instructions that are made available when the commit has finished. Complete the z/OS tasks they describe, and mark them complete.
8. Activate the image.

Create a model

1. Select the label **Conversations** in the tree.
2. Click  on the tool bar or choose the **Add** action of the **Selected** menu bar choice. The properties form appears in the right pane.
3. Complete the properties form by typing the name and description of the conversation. Be sure the name is unique.
4. Click  on the tool bar or choose the **Save** action of the **Selected** menu bar choice. A model is added to the tree. This may take some time. The new model is a copy of the active image, but you can modify it to suit your needs.

Add an object to a model


1. Select the label for the type of object in the tree. For example, to create a server, select the label  **Server** under the sysplex in which you want to place the server.
2. Click  on the tool bar or choose the **Add** action of the **Selected** menu bar choice. The properties form appears in the right frame.
3. Fill in the properties form.
4. Click  on the tool bar or choose the **Save** action of the **Selected** menu bar choice.
5. Once the object is added to the tree, click the node next to the object to expand the branch of the tree. This will show labels for any subordinate objects that may need to be created. Define those objects as needed.

Hints and tips:


- For information about the valid values for the property fields, see the online help for the object or Chapter 4, “Administration objects”, on page 53.
- Some objects appear in two places in the tree. For example, server instances appear both under servers and under systems. Before you add such an object to the tree, you must have defined both of the higher level objects. See the online help or the description for each object in this book for details.
- Adding an object to the model may cause WebSphere for z/OS to add associated objects. These objects are owned and required by WebSphere for z/OS. The names of these objects begin with the letters CB. To avoid confusion, you may want to avoid a leading CB in the name of objects you create.
- When WebSphere for z/OS adds its own objects to the model, the branch of the tree that contains the objects is collapsed. For example, if you add a new system to a sysplex, the 'Servers' branch of the tree for that sysplex is collapsed, because WebSphere for z/OS has created new server instances for the new system.

Delete an object from a model

Use caution when deleting objects. Delete cannot be undone.

1. Select the object to be deleted in the tree.
2. Click  on the tool bar, or choose the **Delete** action of the **Selected** menu bar choice.




Hints and tips:

- When you delete an object, all objects below it in that branch of the tree are also deleted. For example, when you delete a J2EE server, you delete all the server instances and J2EE applications under that server. Associated objects may also be deleted. For example, deleting a J2EE resource deletes associated J2EE resource connections.
- Most objects are not actually removed from the tree until the model is activated, but the deletion is made in the Systems Management Server database. In the tree, the object is marked as deleted with this icon.  (Conversations and LRM connections are the exceptions—they are immediately removed from the tree.)
- Be careful not to inadvertently delete objects that are owned and required by WebSphere for z/OS. The names of these objects begin with CB.
- You can delete a J2EE application that has been installed on a J2EE server, which will delete all the subordinate objects in that branch of the tree, but you cannot

delete any other type of object created by the install function individually. That is, you cannot delete a specific J2EE module or J2EE component.

Modify an object in the model

You can modify most of the fields on the properties form of most objects.


1. Select the object in the tree. Its properties form appears in the right frame.
2. Click  on the tool bar or choose the **Modify** action of the **Selected** menu bar choice.
3. Click on the field that you want to change and type the new value.
4. Click  on the tool bar or choose the **Save** action of the **Selected** menu bar choice. To cancel your changes without saving them, click  on the tool bar or choose the **Cancel** action of the **Selected** menu bar choice.

Notes:

- Fields that cannot be modified have a gray background even when the properties form is being edited.
- Be careful not to inadvertently modify objects that are owned and required by WebSphere for z/OS. The names of these objects begin with CB.
- You cannot modify the properties of objects that are imported into WebSphere for z/OS, that is, a J2EE application and any objects that appear beneath it in a branch of the tree — J2EE module, J2EE component or J2EE resource connection —or an application family with application, home, client interface, class or DLL.

Verify a model's validity

It is good practice to check the validity of the model you are working with any time you make significant changes to it.


1. Select any object in the conversation. Do not select a label (identified by )
2. Choose the **Validate** action of the **Build** menu bar choice, or press the F11 key. The Systems Management Server checks the model for syntax and issues a message when the validation is complete. This may take some time.

Commit a model

The commit process validates the model and locks it, preventing further changes.

Note: You cannot back out of the commit process once you start.


In the main window of the Administration application:

1. Select the conversation for the model in the tree.
2. Select the **Commit** action of the **Build** menu bar choice.
3. Click **Yes** on the confirmation window to continue. (Click **No** to cancel the commit process.) The Systems Management Server verifies the model for syntax. This may take some time. When the verification is complete, a message is issued and the instructions for additional z/OS tasks are provided. This is indicated by the  icon next to the conversation name. The tree is collapsed.


When you have committed the model, it is an image in the state "committed". Now you have to complete the instructions that describe the z/OS tasks.

Display instructions

To display the instructions that describe z/OS tasks, you:

1. Select the conversation in the tree. The conversation must have been committed, and should have the  icon next to it.
2. Select the **Instructions** action of the **Build** menu bar choice. The instructions are retrieved from the Systems Management Server and then displayed in the right frame of the window. This may take some time.


Complete the instructions

1. Perform the tasks described in the instructions. Refer to Chapter 5, "Instructions for z/OS tasks", on page 103 for a more detailed description of these tasks.
2. Indicate which tasks are complete by selecting the **Complete** action of the **Build** menu bar choice, then selecting the appropriate action from the list that is displayed. For example, select **Build - Complete - Security tasks** if all the tasks described under the Security Tasks heading in the instructions are complete.
3. To see an indication that tasks are complete, refresh the instructions:
 - a. Position the mouse pointer anywhere on the instructions.
 - b. Press the right mouse button. A pop-up menu is displayed.
 - c. Select the **Refresh** action.
4. Repeat this process until all the tasks are complete and marked complete. When all tasks are complete, the  icon will appear next to the conversation name in the tree.

You are now ready to activate the image.

Activate an image

After completing the tasks described in the instructions, you activate the image to make it the active image. In the main window of the Administration application:

1. Select an image in the tree, which is in the state "ready for activate" (marked with the icon ).
2. Select **Activate** from the **Build** menu bar choice.
3. Click **Yes** on the confirmation window to continue. (Click **No** to cancel the activate process.)
 - a. The Systems Management server builds the new image and activates it on the WebSphere for z/OS-configured sysplex.

Note: Even if the activation process fails, the new image might be the active conversation! Open the message log and look whether you find the message "Conversation *conversation-name* is now the active conversation".

- b. For each server instance the environment variables (the environment variables specified for this server instance as well as the ones that are inherited by the appropriate server and sysplex) are saved in the environment file, a shared HFS-file on the host. The name of the environment file is:

```
CBCONFIG /controlinfo/envfile/Sysplex_name/ServerInstance_name  
/current.env
```


where


- *CBCONFIG* is the path that you specified in the environment variable *CBCONFIG*. The default is */WebSphere390/CB390*.
- *Sysplex_name* is the name of the sysplex the server instance belongs to.
- *ServerInstance_name* is the name of the server instance.

Warning: If you have modified an environment file directly and not through the Administration application, any changes are overwritten when you activate a conversation.

Refer to “Modify environment variables” on page 49 or to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization* for more information on environment variables.


Note: Even if the activation process fails, the new environment variables might be written to HFS.

- c. The tree is rebuilt with the conversations collapsed. This may take some time. You will receive a message when the activate process is complete, and the active image will be marked with the  icon.

Note: If the activation process fails, the previous active image changes its status. The new status is “replace pending” or “locked and replaced”. The previous active image is marked with the  icon (golden lock). The added image changes the status to “activate in progress”. Re-attempt the activation process. (See “States of a conversation” on page 20 for more information.)

The progress bar at the bottom left of the window shows you that a transaction was started and the application is waiting for response from the server.

You cannot back out of the activate process once you have started.

The previous active image remains in the tree, but cannot be modified or re-activated. It can be deleted. It is marked with the  icon (silver lock).

For best results, you perform the activate during a period of low system usage.

When the server instances are started, some further processing will be done.

Example of modeling a configuration to add an application

Scenario: A new Component Broker application has been developed and needs to be added to your configuration. Object Builder has been used to create the necessary output files, including a DDL file to be imported. In addition, Object Builder has created an association between the application’s home and a container, which has been named `containerb`. This is a container that does not exist in your configuration yet.

The new application will use the IMS OTMA procedural application adaptor (PAA). This is a subsystem type that is not currently represented by a logical resource manager (LRM) in your configuration.

In preparation for importing the file, you need to modify your configuration for this new application. These are the steps you take to modify the configuration:

1. Create a new conversation, which you name `configb`. This new conversation contains a model configuration that matches your active image.

2. Expand configb and the appropriate sysplex below it to show the label for servers.
3. Create a new server, which you name serverb.
4. Expand serverb to show the labels for containers, server instances and application families.
5. Under serverb, create the new container, containerb, that is required by the application's home. Assume that the name of the container has been communicated to you by the developer of the new application.
6. Create a new server instance under serverb, selecting systema as the system on which the server instance will run.
7. Scroll down in the tree to the LRMs. Create a new LRM, which you name imspaalrm, selecting **IMS_OTMA_PAA** for the LRM subsystem type property.
8. Expand imspaalrm to show the label for LRM instances.
9. Create a new LRM instance, selecting systema as the system on which the LRM instance will run.
10. Scroll up in the tree to containerb, the container that you created. Expand the container to show LRM connections.
11. Create a new LRM connection, selecting the name imspaalrm, which is the name of the LRM that you created for the **IMS OTMA PAA** subsystem type.
12. Verify the model's validity. To do this, select the name of the model conversation in the tree, configb, then select the **Validate** action of the **Build** menu bar choice.

When the **Validate** action shows that the model is good, you're now ready to import the application. To do this,

1. Select serverb in the tree.
2. Select the **Import application...** action of the **Selected** menu bar choice.
3. In the **Import application** dialog box, type the name of the z/OS dataset or the fully qualified HFS-file that contains the DDL created by Object Builder. Click **OK** to start the import.
4. When the import completes, check the tree for the new objects.
5. Verify the model's validity.

When the Validate is successful, you are ready to commit and activate the model. After activation, configb will be the active image.

Define the security class of a server

To prevent unauthorized client access to resources and applications managed by the Application Server, determine a *security class* for each server that meets the individual requirements of your configuration:

Non-authenticated clients

Clients that have not been authenticated may connect to this server. When the server acts on behalf of a non authenticated server, it makes use of the "local identity" or "remote identity". The default value for these identities might be set as environment variable DEFAULT_UNAUTH_CLIENT_ID on the sysplex object. If it is not set, "CBGUEST" is used.

To indicate this security class,

- specify the associated security class property of the server object:
 - Allow non-authenticated clients

- assign values to the server properties
 - Local identity
 - Remote identity

Userid and password

Clients may connect to this server, with the MVS user ID and password being used for security. This approach is suitable for secure networks since the password flows "in the clear". Use the password when the clients may be distributed, that is, on platforms other than z/OS or on z/OS systems outside of the sysplex in which the server is running.

To indicate this security class, specify the associated security class property of the server object:

- Userid password allowed

Passticket

The *passticket* is a tamper-resistant encoded credential that must be used within ten minutes. It is a one-time-only, system-generated password. The *passticket* is recommended when the client and the server are on z/OS systems in the same sysplex.

To indicate this security class, specify the associated security class property of the server object:

- Userid passticket allowed

DCE

When a network is not secure, it may be useful to employ the *DCE (Distributed Computing Environment) Security Server* to deliver an authenticated DCE ticket to clients and servers. Using this ticket, a client signs a message before sending it to the server. This sends the encoded *extended privilege attribute certificate (or EPAC)* along to the server. The server is able to verify the signature against the ticket it received. The reverse process occurs on the outbound reply. In this way, the client and server must contact the DCE security server, acting as a third party, before communications can occur.

DCE is recommended when the clients may be distributed, that is, on platforms other than z/OS or on z/OS systems outside of the sysplex in which the server is running.

To indicate this security class, specify the associated security class property of the server object:

- DCE allowed, and
 - DCE quality of protection
 - DCE keytab file

SSL type 1 (basic authentication)

SSL Type 1 (or often called *SSL Basic Authentication*) is a security mechanism that authenticates the server using its digital certificate and encrypts messages flowing across the client/server connection. The server authentication entails ensuring that the server's certificate was granted by a certificate authority known to the client. The client's identity is established by user ID and password. This prevents unauthorized client access to WebSphere for z/OS resources.

For a better understanding of SSL Basic Authentication, read the definition of some terms:

Secure Sockets Layer (SSL)

is a communications protocol that provides secure communications over an open communications network (for example, the Internet). The SSL protocol is a layered protocol that is intended to be used on top of a reliable transport, such as Transmission Control Protocol (TCP/IP). SSL provides data privacy and integrity as well as server and client authentication based on public key certificates. Once a SSL connection is established between a client and server, data communications between client and server is transparent to the encryption and integrity added by the SSL protocol.

SSL connections

make use of public/private key mechanisms for authenticating the SSL session (server and optionally client) and agreeing on encryption keys to be used for the SSL session. Public/private key pairs must be generated, received and managed.

X.509 certificates

contain public keys. These certificates are managed in the *RACF Keyring*.

Certificate Authorities (CAs)

generate certificates. They must be connected to the *RACF Keyring*.

SSL Basic Authentication means that the client authenticates the server using SSL, the client is authenticated by a user ID and password:

1. The SSL protocol begins with a "handshake." During the handshake, the client authenticates the server, and the client and server agree on how to encrypt and decrypt information.

X.509 certificates are used by both the client and server when securing communications using System SSL. The client must verify the server's certificate based on the certificate of the Certificate Authority (CA) that signed the certificate or based on a self-signed certificate from the server.

2. The client and the server then use the session keys and begin encrypted communications.
3. The client sends an MVS user ID and password over the SSL encrypted session. It will be used to authenticate the client.

Thus clients can be sure that servers are trustworthy.

The server's certificate must be defined as the default certificate in the specified *RACF keyring*. This certificate identity will be authenticated in both inbound and outbound requests.

For more information about SSL, refer to *OS/390 Cryptographic Services System Secure Sockets Layer Programming Guide and Reference, Document Number SC24-5877*.

To indicate this security class, specify the associated security class property of the server object:

- SSL Basic Authentication (Type 1) allowed, and
 - SSL *RACF Keyring*
 - SSL V2 timeout
 - SSL V3 timeout
 - Security preference list

SSL client certificates

SSL Client Certificates ensure that the client authenticates the server and the server authenticates the client. Both client and server authentication mechanisms are done by SSL, each side presents an X509 certificate:

1. The SSL protocol begins with a "handshake." During the handshake, the client authenticates the server, the server authenticates the client, and the client and server agree on how to encrypt and decrypt information.

X.509 certificates are used by both the client and server when securing communications using System SSL. The client must verify the server's certificate based on the certificate of the Certificate Authority (CA) that signed the certificate or based on a self-signed certificate from the server. The server must verify the client's certificate using the certificate of the CA that signed the client's certificate.

2. The client and the server then use the session keys and begin encrypted communications.

This aspect of authentication guarantees that servers can trust their clients. (These credentials allow the server to initiate SSL client connections in server-as-client mode with SSL Asserted Identities.) The credentials allow to decide what a user is or is not allowed to perform in an object.

If you are using client authentication, then the client certificate must be specified in the RACF keyring. In addition, all certificate authorities for servers you need to access, must have certificates defined to RACF and be connected to the client's keyring.

To indicate this security class, specify the associated security class property of the server object:

- SSL Client Certificates allowed, and
 - SSL RACF Keyring
 - SSL V2 timeout
 - SSL V3 timeout
 - Security preference list

SSL kerberos

The Secure Socket Layer with Kerberos Client Authentication provides

- mutual authentication (both requester and supplier are authenticated),
- message protection (the messages which are exchanged between the requester and the supplier are protected), and
- delegation (if the supplier has to access another supplier on behalf of the original requester, the identity of the originator accompanies the request).

The requester sends the supplier a Kerberos *Generic Security Service Application Program Interface* (GSS_API) token which is sent over an SSL session. This token will then be used by the supplier to authenticate the identity of the requester. The SSL connection provides for message protection and authentication of the server to the client. The token that was provided by the requester will be used for delegation purposes.

For more information about Kerberos, refer to *OS/390 SecureWay Security Server Network Authentication and Privacy Service Administration*, SC24-5896, or *OS/390 SecureWay Security Server Network Authentication and Privacy Service Programming*, SC24-5697.

SSL asserted identities

The purpose of an asserted identity request is for an intermediate server to send an already verified identity to a target server. The client's identity must have been previously authenticated. The target server authorizes the intermediate server for identity assertion. The Application Server will use the CB.BIND *servername* RACF profile defined in the CBIND class for this purpose. If your server accepts SSL Asserted Identities, be sure that the CBIND class of RACF is activated on the target system and that all servers that you wish to authorize to send Asserted Identities to this server are permitted to have CONTROL access to the CB.BIND profile for your server.

SSL use confidentiality only

System SSL contains a set of cipher suites that it used to encrypt and authenticate data. The particular cipher suite used is determined by system SSL and its peer during the "handshake" process. The peer SSL systems will use the highest level of security they have in common. Confidentiality is defined as both encryption and authentication. A peer system may not have all cipher suites available that the host system does. Setting the server property **SSL Use Confidentiality Only** box requires system SSL to use only the confidentiality cipher suites. If the peer system does not have these suites available, the SSL connection will fail.

Where to find more information:

- Refer to page 74 for more information about the security properties of a J2EE server.
- Refer to page 90 for more information about the security properties of a (MOFW) server.
- For more information about the security concept of your configuration, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*.

Import an application

For each application, Object Builder has created one or two DDL files. Make sure that you know the name of the z/OS dataset or the name of the fully qualified HFS-file that contains the DDL file.

To import an application,

1. In the tree, select the (MOFW) server onto which you will import the application.
2. Choose **Import application...** from the **Selected** menu bar choice to display the **Import** dialog box.
3. For the input file, type the name of the dataset or of the fully qualified HFS-file that contains the DDL file. It should be accessible on the shared HFS. The name of the DDL dataset will be treated as a fully qualified dataset name. You do not need to enclose it in quotation marks.

4. For the output file, type the name of the dataset or of the fully qualified HFS-file that you want to contain the listing from the import process. The dataset need not have been allocated. You do not need to enclose it in quotation marks.
5. Click **OK**. For DDL file *name*, the Systems Management Server imports the file and builds a branch for it in the model. The Administration application refreshes the tree with the new branch. This may take some time.

Hints and tips:

- Before importing the DDL file, you might review it for container names, and then verify that the required containers are present in the tree.
- You can delete an application family that has been imported into the tree, which will delete all the subordinate objects in that branch of the tree, but you cannot delete any other type of object created by the import function individually. That is, you cannot delete a specific class, DLL or home.
- If you deleted an application from a server and you want to re-import it, you need to delete the application in one conversation, activate it, and create another conversation to re-import the application, then activate it.

Deploy J2EE applications

WebSphere Application Server V4.0.1 for z/OS and OS/390 allows Enterprise JavaBeans and Web applications consisting of servlets and JSPs to be fully formed on the workstation using according construction tools, such as VisualAge for Java, WebSphere Studio, or WebSphere Studio Application Developer. These EJBs and Web applications can be packaged into J2EE applications using WebSphere Studio Application Developer, the Application Assembly Tool delivered on the workstation, or the Application Assembly Tool delivered together with WebSphere for z/OS. A J2EE application can be *deployed* quickly and directly to an WebSphere Application Server V4.0.1 for z/OS and OS/390 server using the Administration application.

- The chapter “Introduction” introduces to the deployment of J2EE applications. Some often used terms are explained and their relationship is shown. For a complete introduction, read the explanation of these terms one after the other. If you only want some dedicated information, refer to the index of this book.
- The chapter “Step-by-step instructions” on page 37 explains how to deploy J2EE applications with the Administration application. Specific terms are provided with a reference to their explanation in the foregoing chapter.
- The chapter “References” on page 45 leads you to further references of this topic if you want to learn more about J2EE applications, their development and their deployment.

Introduction

Let’s clarify some terms before we get into the step-by-step instructions:

Java 2 platform, Enterprise Edition (J2EE)

The *Java 2 platform, Enterprise Edition (J2EE) specification* defines a standard architecture to reduce the cost and complexity of developing multi-tier services, resulting in services that can be rapidly deployed and easily enhanced:

The J2EE application model partitions the work needed to implement a multi-tier service into two parts:

Business and presentation logic

The business and presentation logic is to be implemented by the *developer*.

In the J2EE platform, middle-tier business functions are implemented as *Enterprise JavaBeans components* (see below), servlets or JSPs.

Standard system services

The standard system services — like enforcing an application's security roles, implementing its transaction semantics, and linking its components to the resources and other components they require — are provided by the *J2EE platform*.

WebSphere Application Server V4.0.1 for z/OS and OS/390 supports the J2EE 1.2 specification.

Enterprise JavaBeans (EJB)

The *Enterprise JavaBeans (EJB) specification* provides the server-side *component architecture* for J2EE. The Enterprise JavaBeans specification defines a standard architecture for implementing the business logic of multi-tier applications as reusable *components* for the J2EE platform. Enterprise JavaBeans allow service developers to concentrate on the business logic and let the *EJB server* handle the complexities of delivering a reliable, scalable service. The Enterprise JavaBeans architecture is based on the Java programming language.

WebSphere for z/OS V4.0.1 supports the EJB 1.1 specification and so far tolerates EJBs following the EJB 1.0 specification by mostly converting them to 1.1 EJBs.

In addition to Enterprise JavaBeans components, the architecture defines containers, servers, and clients:

Enterprise JavaBeans component

An EJB component is an *object* infused with additional critical capabilities that allow it to be properly managed in an industrial-strength business application environment. These capabilities include:

- Properly creating and destroying an object, and providing it with a location in a distributed network
- Establishing a system identity for an object so that it can be found again
- Mapping the state of an object to a persistent store
- Controlling access to an object in a secure fashion, and so forth

An EJB component is often called an *EJB*, an *enterprise bean*, or simply a *bean* in this context.

EJBs are composed into *J2EE applications*. In WebSphere for z/OS V4.0.1, J2EE applications organize EJBs into J2EE modules, and J2EE modules into J2EE components.

Enterprise JavaBeans container

An EJB container contains EJB components. EJB containers are designed to handle details of component life-cycle, transaction, and security management. By interceding between clients and components at the method call level, containers can manage transactions that propagate across calls and components, and even across containers running on different servers and different machines. This mechanism simplifies development of both component and clients. Component developers are free to focus on business logic, since containers provide services automatically by interceding in component method calls.

Enterprise JavaBeans server

EJB servers are designed to provide fundamental services for containers and the components they contain, such as transactions, security, life-cycle, threading, and persistence.

In WebSphere for z/OS, an EJB server is called a *J2EE server*.

Enterprise JavaBeans client view

The EJB client view of an EJB component is provided through two interfaces: the *home interface* and the *remote interface*. These interfaces are provided by classes constructed by the container when an EJB is deployed (as part of a J2EE application), based on information provided by the EJB.

Home interface

The home interface provides methods for creating, locating and accessing instances of the EJB component.

Remote interface

The remote interface provides the business logic methods for the EJB component, that can be called by client programs.

There are several *types* of EJBs, *session beans* and *entity beans*, representing different types of business logic abstractions.

Session beans

Session beans represent behaviors of EJB components associated with client sessions. They are generally implemented to perform a sequence of tasks within the context of a transaction. A session bean is a logical extension of the client program, running processes on the client's behalf remotely on the server. It may have a state, but it is not shareable among clients.

Stateful session beans

A stateful session bean has persistent fields. These fields are managed by the container, in a container-provided backing store.

Stateless session beans

A stateless session bean has no persistent fields.

Entity beans

Entity beans represent specific data or collections of data, such as a row in a relational database. Entity bean methods provide

operations for acting on the data represented by the bean. An entity bean is persistent; it survives as long as its data remains in the database.

The *Java Naming and Directory Interface* provides basic access to the EJB components.

Java Naming and Directory Interface (JNDI)

The *Java Naming and Directory Interface (JNDI)* is an *application programming interface (API)* that provides Java applications with a unified interface to multiple naming and directory services in the enterprise. JNDI works together with other J2EE technologies to organize and locate components in a distributed computing environment. It is defined to be independent of any specific directory service implementation. Thus a variety of directories can be accessed in a common way. The directory services can be based on different service providers, e.g. LDAP, CORBA, RMI.

The JNDI API uses *JNDI names* to reference objects and their attributes.

Deployment Descriptor

The *deployment descriptor* is part of an application package. It provides XML-based declarations which enable application deployers to modify the behavior of an application without having to modify any of the components themselves.

The deployment descriptor contains declarations of enterprise bean's references to

- EJBs

They are needed to locate the home interface of another enterprise bean. The container needs information about these *EJB references* that is supplied during deployment.

- external resources

They consist of an optional description, the *resource reference* name, the indication of the J2EE resource type expected by the enterprise bean code, and the type of authentication (bean or container). To access external managed resources on behalf of a bean, the container needs information that is supplied during deployment.

EJB references

EJB references allow bean providers to write code which does not make explicit references to other beans. All the EJB references are declared in the deployment descriptor.

Internal references are references to other beans in the same application. They are also called *EJB links*. All internal references are resolved automatically during install time. The *external references* — which are JNDI names — will be resolved by the user during deployment.

J2EE resource,

Resource,

Resource factory,

Resource manager connection factory

A *resource manager connection factory* (we also call it *J2EE resource, resource, or resource factory*) enables a container which manages EJBs, servlets and JSPs to connect to managed resources, e.g. a JDBC API data source.

J2EE resource connection

A *J2EE resource connection* connects a specific *J2EE component* (see below) to a J2EE resource.

J2EE resource references,

Resource references,

Resource manager connection factory references

Resource manager connection factory references (we call them shortly: *resource references* or *J2EE resource references* in the Administration application) allow beanproviders to write code which does not make explicit references to resources as for example DB2 datasources. All resource references are declared in the deployment descriptor.

The deployer binds the resource references to the actual J2EE resources that are configured in the container. Because these J2EE resources allow the container to affect resource management, the connections acquired through the resource manager connection factory references are called *managed resources*.

Archive files

The EJB 1.0 specification defined *JAR files* as the Java ARchives for Enterprise JavaBeans. J2EE defines three new archives - *WAR*, *application client jars* and *EAR*. WebSphere for z/OS V4.0.1 will support EAR files that contain JARs and WARs.

Assembly is the process of creating JAR, WAR and EAR files from class files, html files, gifs, etc. This process involves selecting all the files that go into a module and creating a XML deployment descriptor for it. One or more of these modules can then be combined with an XML descriptor to form an EAR file. An EAR file is an archive of a *J2EE application*.

J2EE component

The development life cycle of a J2EE application begins with the creation of discrete *J2EE components*. Generally, a J2EE component is an EJB, servlet or JSP.

The J2EE components are packaged with a component level deployment descriptor to create a J2EE module. J2EE modules have to be assembled with a J2EE application deployment descriptor and deployed as a J2EE application.

J2EE module

A *J2EE module* is a collection of one or more J2EE components of the same container type with one component deployment descriptor of that type. In WebSphere for z/OS V4.0.1, a J2EE module is a collection of EJB components or Web applications consisting of servlets or JSPs.

An *EJB module* is the smallest deployable and usable unit of enterprise beans. An EJB module is packaged and deployed as an EJB JAR file, a JAR file with a .jar extension. An *EJB JAR file* differs from a standard JAR file in one key aspect: it is augmented with a deployment descriptor that contains meta-information about one or more enterprise beans.

An EJB module contains:

- Java class files for the enterprise beans and their remote and home interfaces. If the bean is an entity bean, its primary key class must also be present in the EJB module.

- Java class files for any classes and interfaces that the enterprise bean code depends on that are not included with the J2EE platform. This may include superclasses and superinterfaces and the classes and interfaces used as method parameters, results, and exceptions.
- A EJB deployment descriptor that provides both the structural and application assembly information for the enterprise beans in the EJB module. The application assembly information is optional and is typically included only with assembled applications.

J2EE application

J2EE applications are composed of EJB and Web modules and one J2EE application deployment descriptor.

Packaging a J2EE application

A J2EE application is packaged as an *Enterprise ARchive (EAR) file* — a standard Java ARchive (JAR) file with an .ear extension — for deployment to any J2EE platform-compliant system. The goal of this file format is to provide an application deployment unit that is assured of being portable.

A minimal J2EE application package will only contain J2EE modules and the application deployment descriptor. A J2EE application package may also include libraries referenced by J2EE modules, help files and documentation to aid the deployer, etc.

J2EE server

A WebSphere for z/OS server will host either J2EE components or Managed Object FrameWork (MOFW) components; never both at once. A J2EE server is a server that hosts J2EE applications; in WebSphere for z/OS V4.0.1 meaning it hosts EJBs, servlets or JSPs. J2EE components' runtime execution is managed within a server by a *container*.

In WebSphere for z/OS V4.0.1, only two application containers per server region will be supported (one Web container and one EJB container). The server region initialization for a J2EE server will automatically initialize the Web and the EJB container.

Home registration will be done automatically when the server is initialized (as part of the control region startup).

Deploying a J2EE application

The deployment of a J2EE application proceeds in several steps:

1. Gathering of the J2EE application deployment descriptor from the application .EAR file (META-INF/application.xml).
2. Opening of each of the J2EE modules listed in the J2EE application deployment descriptor and gathering of the J2EE module deployment descriptor from the package.
3. Resolution of the EJB and resource references by the user.
J2EE resource connections are established.
4. Installation of the components described by each module deployment descriptor into the appropriate container according to the deployment requirements of the respective J2EE component specification.

Step-by-step instructions

Before you install a J2EE application, make sure that

1. a J2EE server with server instances is defined

2. the referenced J2EE resources and J2EE resource instances are defined
3. the J2EE application is assembled by your application assembly tool and resides on your local workstation.

Note: The container implicitly assigns the NotSupported transaction attribute to each container-managed transaction bean method to which no transaction attribute has been assigned. If you do not want the container to assign the NotSupported transaction attribute, transaction attributes can be assigned in an application assembly tool prior to installing the J2EE application.

To deploy a J2EE application with the Administration application,

1. In the tree, select the J2EE server (see “J2EE server” on page 37) to which you want to deploy the J2EE application (see “J2EE application” on page 37).
2. Choose **Install J2EE application...** from pop-up menu or from the **Selected** menu bar choice to display the **Install J2EE Application on Server: Servername** dialog box.
3. In the **Install J2EE application on Server: Servername** dialog box:

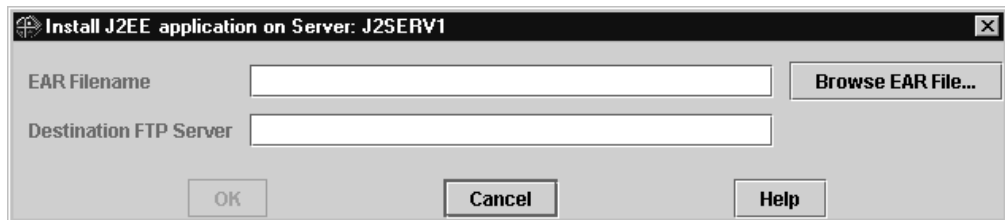


Figure 4. Install J2EE application Dialog: Install J2EE application on Server Dialog Box

enter the following values:

- a. In the **EAR Filename** entry field, enter the name of the EAR file which contains the J2EE application. It must reside on your local workstation. Use the **Browse EAR File...** button to find the location of the file.
Only regularly deployed WebSphere EAR files are accepted.
- b. In the **Destination FTP Server** entry field, enter the name of the FTP server for the sysplex to which you want to deploy the J2EE application. Usually, this is your bootstrap server IP name (see “Login dialog” on page 8) to which you have connected the Administration application. To login to the FTP server your login user ID and password will be used.

Note: Since the EAR file is transferred to the z/OS Sysplex using the FTP protocol, the FTP server has to be set up on the z/OS Sysplex to be able to complete the processing of the EJBs successfully. For details on how to set up the FTP server refer to *WebSphere for z/OS: Installation and Customization*.

- c. If you want to allow the Administration application to establish an FTP connection to the Destination FTP server despite a firewall, check the **passive FTP** box. Otherwise, a firewall could prevent the Administration application from establishing a connection.
- d. Click **OK**.

The **Reference and Resource Resolution** window appears.

4. In the **Reference and Resource Resolution** window,

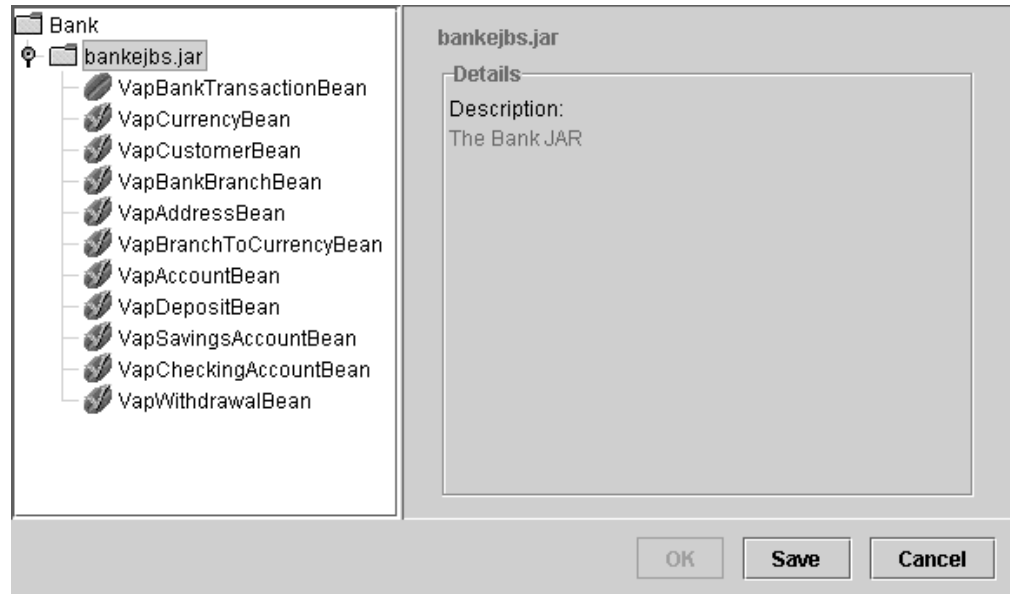


Figure 5. Install J2EE application Dialog: Reference and Resource Resolution Window

on the left-hand side of the dialog, you see a tree which represents the application/ module/ component hierarchy contained in the EAR file (see “J2EE application” on page 37 and “Packaging a J2EE application” on page 37).

For each bean, a JNDI name has to be conferred, and the EJB references and resource references have to be resolved (see “JNDI” on page 35, “EJB references” on page 35, “Resource references” on page 36 and “Deploying a J2EE application” on page 37). Up to this time, the state of the application is indicated by and the state of each bean without a JNDI name or with unresolved references is indicated by .

For each EJB which has got a JNDI name and whose references and resources are resolved, a check is displayed next to the bean’s name. When all beans are checked, the application is ready to be transferred to the host, which is indicated by a check next to the application name. Then, the **OK** button is active.

Click on an object in the list to see its details on the right-hand-side of the window.

5. For each bean in the list, on the right-hand side of the window, three different tabs are available: EJB, Reference, and Resource.

Click on each bean in the list that has not been checked yet, one after the other, and perform the following steps:

- a. Click on the EJB tab to confer a JNDI name for the EJB.

The *JNDI name* is employed to reference the EJB from outside (see “JNDI” on page 35). A JNDI name for an EJB is constructed from two parts, the path and the actual name.

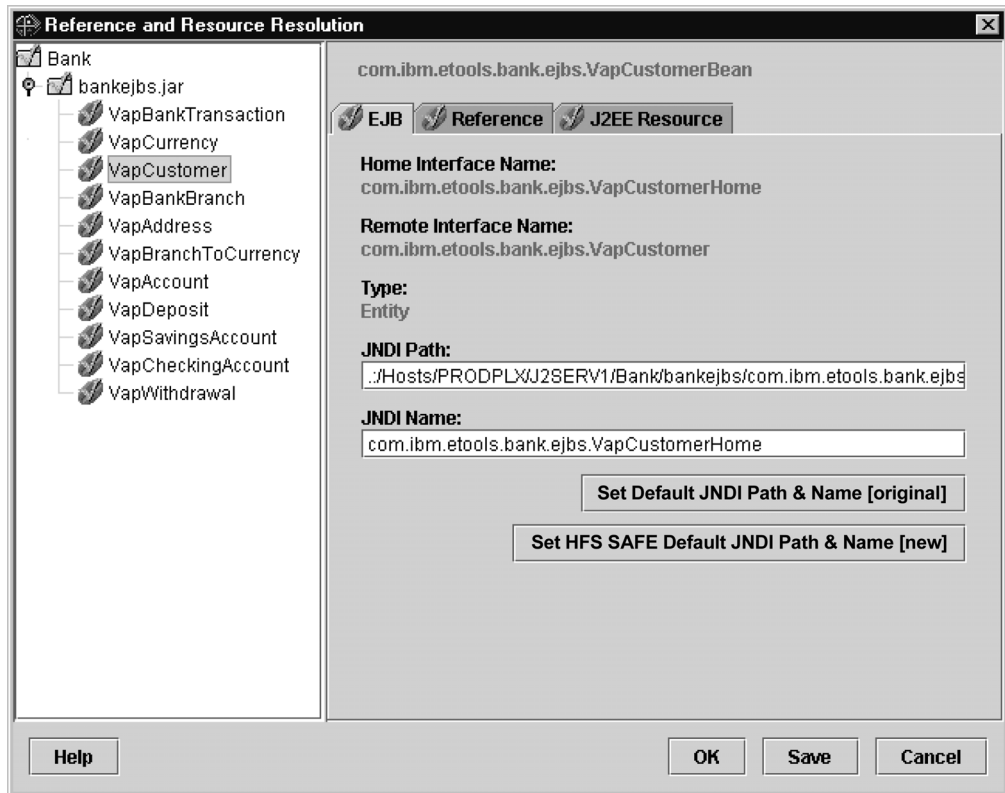


Figure 6. Install J2EE application Dialog: Reference and Resource Resolution: EJB Tab

Home Interface Name

Displays the home interface name (see “Home interface” on page 34) of the selected EJB.

Remote Interface Name

Displays the remote interface name (see “Remote interface” on page 34) of the selected EJB.

Type Displays the type (see “EJB types” on page 34) of the selected EJB.

Use the **Set Default JNDI Path & Name [original]** button or the **Set HFS SAFE Default JNDI Path & Name [new]** button to apply an automatically generated JNDI path to the **JNDI Path** entry field and a JNDI name to the **JNDI Name** entry field for the selected EJB. This automatically generated path and name is unequivocal, consisting of host, sysplex, server, application, module, component, and home name.

Note: The **Set Default JNDI Path & Name [original]** button is equivalent to the new **390fy -JNDIejbpb** option. The **Set HFS SAFE Default JNDI Path & Name [new]** button is equivalent to the existing **390fy -JNDIejbp** option.

- b. Click on the Reference tab.

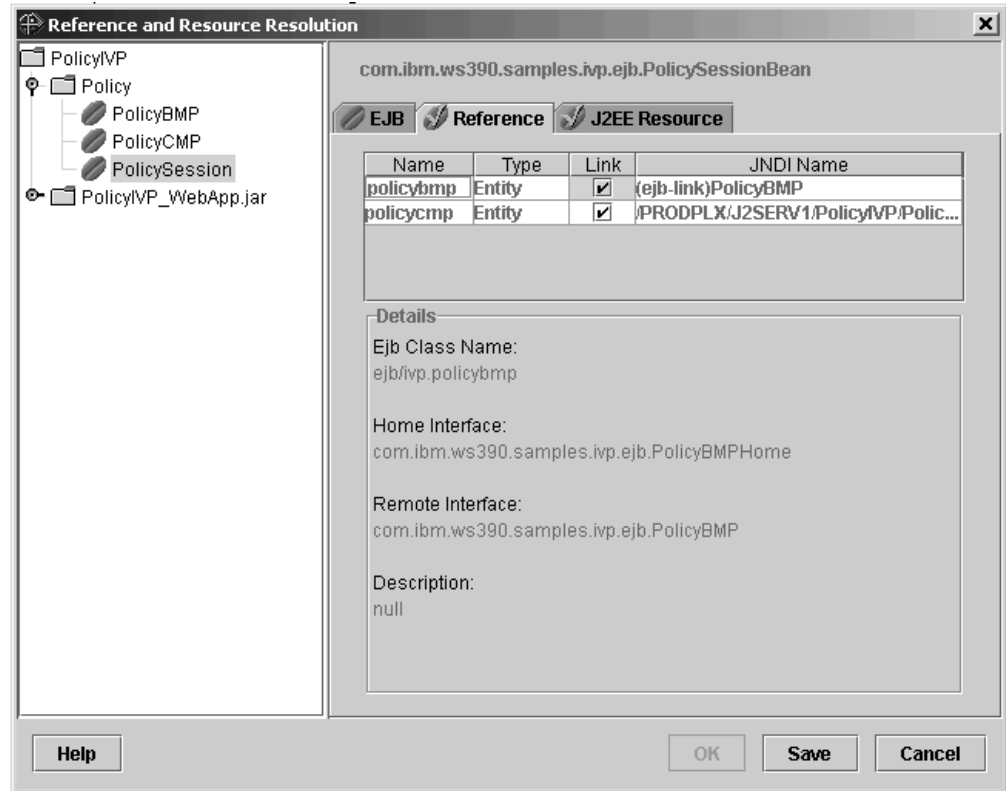


Figure 7. Install J2EE application Dialog: Reference and Resource Resolution: Reference Tab

A list of unresolved EJB references appears (see “EJB references” on page 35 and “Deploying a J2EE application” on page 37).

For each EJB name in the list, a JNDI name must be provided.

Name The referenced EJBs are listed by name, type and JNDI name.

Type Type of the referenced bean: *Stateful session*, *Stateless session*, or *Entity* (see “EJB types” on page 34)

Link This column contains checkmarks for all those bean references that have been statically bound to another bean within the same application using the `<ejb-link>` tag as defined in the EJB specification. It is now also possible to generate such links not only during application assembly time but also during bean deployment by resolving a reference to a bean that resides within the same application. While `<ejb-links>` that have been created by the application assembly tool cannot be changed to something different, the `<ejb-links>` that have been created in the SM GUI during deploy time can be changed back to ordinary references.

JNDI Name

Displays the JNDI name for the EJB.

Ejb Class Name

Displays the full name of the referenced EJB.

Home Interface

Displays the home interface name (see “Home interface” on page 34) of the referenced EJB.

Remote Interface

Displays the remote interface name (see “Remote interface” on page 34) of the referenced EJB.

Description

Displays the description of the referenced EJB.

To retrieve the **Jndi Name** to resolve the reference, click into the appropriate entry field. A list of names of EJBs of the requested type, the same home interface, and the same remote interface will appear. Select an EJB name to be inserted into the list. The JNDI name is then added to the field. This field cannot be edited manually.

For internal references, you will see an entry of the form “<EJB_link>name”. This field is not editable and will be automatically resolved as soon as the JNDI name is assigned for the referenced bean.

- c. Click on the J2EE Resource tab.

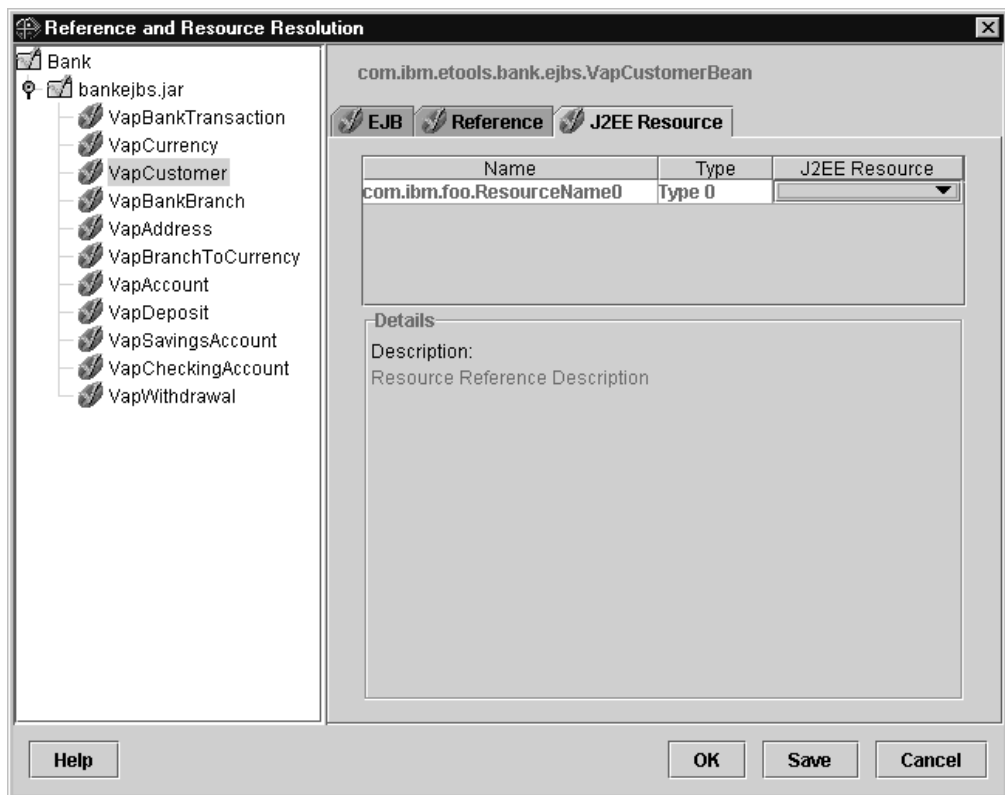


Figure 8. Install J2EE application Dialog: Reference and Resource Resolution: Resource Tab

A list of J2EE references appears (see “Resource references” on page 36 and “Deploying a J2EE application” on page 37).

Name The symbolic name of the reference used by the EJB.

Type Type of the referenced J2EE resource. The type is specified by the Java interface (or class) expected to be implemented by the J2EE resource. Note that the indicated type is the Java type of the resource manager connection factory, not the Java type of the

resource. E.g. the bean uses the javax.sql.DataSource resource manager connection factory type for obtaining JDBC API connections.

J2EE Resource

For each referenced J2EE resource in the list, a name of a J2EE resource must be provided for one of the currently defined J2EE resources compatible with the referenced type (see “JNDI” on page 35).

Description

Displays the description for the selected resource.

Click on the **J2EE Resource** entry field to resolve the reference. A list of J2EE resources will appear which have the same type as the referenced J2EE resource.

Pay attention to the fact that a referenced J2EE resource must have J2EE resource instances for all the systems that have server instances of the J2EE server to which the J2EE application is deployed.

6. To exit the dialog,
 - click **OK** to save your changes and continue with the deployment process. The **OK** button will close the dialog; it will not be enabled until all the entries are made;
 - click **Save** to create a new copy of the EAR file containing all the changes you have made so far. Use this button if you want to perform other changes (like creating J2EE resources) before you leave the dialog in order to continue later with the deployment.
 - click **Cancel** to cancel your changes, the **Install J2EE application..** dialog, and the whole deployment process.
7. The **Processing Install J2EE Application** window appears which indicates the progress of the EAR file being transferred to the z/OS Sysplex for further processing. To see the protocol of the following transactions, see the message log (refer to Chapter 9, “Message log”, on page 129).
8. Once the file has been transferred, the file is being processed by the System Management server, indicated by the word “deploying” in front of the J2EE server in the tree. This process may take some time.

Note: Please check whether your conversation is still valid: Each system with a server instance of a J2EE server must also have a J2EE resource instance of the J2EE resource it is connected to.

Hints and tips:

- If you deleted a J2EE application from a J2EE server and you want to re-install it, you need to delete the J2EE application in one conversation, activate it, and create another conversation to re-install the J2EE application, then activate it.

Alternate steps for insuring each EJB has a JNDI name, and references and resources for each EJB are resolved

Alternately you might want to use default values only for the JNDI names and automatically resolve all unambiguous references and resources. To do this:

1. Click on the application object in the tree to see the three default buttons on the right-hand-side of the window.
2. Use the **Set Default JNDI Path & Names for all Beans [original]** button or the **Set HFS SAFE Default JNDI Path & Names for all Beans [new]** button to

automatically apply generated JNDI path and names to all beans in this application .ear file. These automatically generated names are unequivocal, consisting of host, sysplex, server, application, module, component, and home name.

Note: The **Set Default JNDI Path & Name for all Beans [original]** button is equivalent to the new **390fy -JNDIejbpb** option. The **Set HFS SAFE Default JNDI Path & Name for all Beans [new]** button is equivalent to the existing **390fy -JNDIejbp** option.

3. Use the Resolve all unambiguous References button to automatically resolve all references where only one compatible, unique target reference exists.
4. Use the Resolve all unambiguous Resources button to automatically resolve all resources where only one compatible, unique target resource exists.
5. Unfold the tree and check whether all beans have been successfully processed. These beans are checked: Solved references.

For all unchecked beans proceed with the instructions above.

Steps for adding classes in addition to the ones specified in the module archive file

An application can require classes in addition to the ones included in the module (EJB jar file). Follow the following steps to specify these additional classes on the module Class-Path field:

1. Click on a module in the list to see the Class-Path field on the right-hand side of the window.
2. In the Class-Path field, specify the directory path and, where appropriate, the file for each set of classes.
 - Make the path relative to the root of the EAR file by using a period and a forward slash (./). The class loader ignores absolute paths.
 - Separate each path and file combination with a space
 - For .class files, only specify the path.
3. Repeat the preceding steps for each module.

Example: The EAR file, myapp.ear, contains the following files:

```
myejb.jar
class1.jar
class2.zip
xyz.class
```

The myejb.jar file is an EJB module. Files class1.jar and class2.zip contain additional classes and reside in the addclass subdirectory under the EAR file root. The xyz.class file is not packaged in a jar file, but is in the root of the EAR file.

Specify the following in the Class-Path field:

```
myejb.jar addclass/class1.jar addclass/class2.zip /
```

The xyz.class file is specified by ./.

Note: If the module is not a wrapper bean, the classpath is read from the manifest file that resides in the module (EJB jar file). If you modify the Class-Path field, the updated classpath is written to the MANIFEST file in the module. If the module is a wrapper bean, you also have a corresponding WAR file in your application (EAR file). However, you cannot see the WAR file in the SM EU. You can only see the module (EJB JAR file), which contains the wrapping bean for this WAR file. The classpath is read from the MANIFEST

file in the WAR file. If you modify the Class-Path field, the updated classpath is written to the MANIFESTfile in the WAR file and to the MANIFEST file in the module (EJB jar file).


References

- *Java 2 Platform, Enterprise Edition Specification Version 1.2*. Copyright 1999, Sun Microsystems, Inc. Available at <http://java.sun.com/j2ee/docs.html>.
- *Enterprise JavaBeans Specification, Version 1.1 (EJB specification)*. Copyright 1998, 1999, Sun Microsystems, Inc. Available at <http://java.sun.com/products/ejb>.
- *Java Naming and Directory Interface 1.2 Specification (JNDI specification)*. Copyright 1998, 1999, Sun Microsystems, Inc. Available at <http://java.sun.com/products/jndi>.
- *The Java 2 Platform, Enterprise Edition Application Programming Model*, Copyright 1999, Sun Microsystems, Inc. Available at <http://java.sun.com/j2ee/apm>.

Add a J2EE resource type

WebSphere for z/OS provides the J2EE resource type "DB2datasource" which allows J2EE applications to access DB2 resources. To provide access to other datasources, WebSphere for z/OS allows to dynamically add J2EE resource types. Although this interface has not yet been published, perhaps you will receive some files for a new type from IBM or some other vendor. You can easily include this new J2EE resource type into the Administration application by following these instructions.

To add a new J2EE resource type,

1. Copy the xml file which defines the J2EE Resource Type to the path `CBCONFIG/Sysplex-name/resources/templates`, where
 - `CBCONFIG` is the path that was specified in the environment variable `CBCONFIG`. The default is `/WebSphere390/CB390`
 - `Sysplex-name` is the name of the sysplex where the J2EE resource type is added.
2. Click  in the tool bar to refresh the connection to the Administration application
3. Update the classpath with accompanying jar-file.

Prepare performance recording

To decide if you can go into production with an application, it might be important for you to gather some information about it concerning

- capacity planning (how many transactions have run, what is the average and mean completion time for transactions running per server, how many clients are attached to each server instance),
- application profiling (information about the components of an application and their timing), and
- error reporting (a mechanism to detect and record soft failures).

The *System Management Facilities (SMF)* collect and record system and job related information on the z/OS system. For example, this information can be used to bill users, report reliability, analyze the configuration, schedule work, or profile system resource usage.

WebSphere for z/OS is able to produce the appropriate SMF records that will allow installations to perform these functions. These records will be used by existing products to provide performance reports for the Application Server environment.

Two *classes of data* can be gathered:

Server activity

Activities that run inside a z/OS application server. There is a single record for each activity running inside a server instance.

Container activity

Activities that run inside a container located inside a z/OS transaction server. There is a single record for each container that is part of an activity.

Two *types of records* can be produced:

Activity records

These records gather each transaction within a server after it is completed.

Interval records

The data of this record type are gathered at intervals specified in the server property **SMF Interval Length**.

To prepare SMF recording,

1. In SMF on z/OS,

Specify record type 120 in the parmlib member SMFPRMxx to enable WebSphere for z/OS performance monitoring.

For information on setting up SMF, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*, or *OS/390 MVS System Management Facilities*.

2. In the Administration application,

Specify the SMF recording properties of the appropriate server object:

Write Server Activity SMF Records

Use these records to perform basic charge back accounting as well as profiling of own applications to determine in detail what is happening inside the transaction server.

Write Container Activity SMF Records

Use these records to perform basic charge back accounting, application profiling, problem determination, and capacity planning.

Write Server Interval SMF Records

Use these records to profile your own applications.

Write Container Interval SMF Records

Use these records to perform application profiling, problem determination, and capacity planning.

SMF Interval Length

Specify the interval length, if you want to gather interval records (Server Interval SMF Records or Container Interval SMF Records).

Refer to page 94 for more information about the server properties.

**To access SMF records,
on the MVS console**

1. Enter `i smf` to switch the SMF datasets.
2. Run the SMF Dump Program (IFASMFDP) to create a sequential dataset from the raw dump (a sample JCL is shown in the SMF documentation).
3. View the dataset by a program that is able to display record type 120.
For more information refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Operations and Administration*, or *OS/390 MVS System Management Facilities*, GC28-1783.

Migrate a test server to a production system

If you have both a test system and a production system, you might want to migrate a server from the test system to the production system, after you have tested the applications on this server in the test environment.

A *server* in this context is either a MOFW server (simply called "server" in the Administration and Operations applications) or a J2EE server.

You can do this by the help of the actions **Export server...** and **Import server...** The action **Export server...** creates HFS-files for the server on the host. You then move these files to the host of the production system, and finally import the server (that is, the server and its subtree with almost all its properties, even referenced but not defined logical resource mappings and J2EE resources) to the production system with the action **Import server...**:

1. In the tree:
 - a. Select the test server (either MOFW or J2EE) within the active image.
 - b. Select the **Export server...** action of the **Selected** menu bar choice to display the **Export server** dialog box.
 - c. For **Output directory**, enter the HFS path on the host of the test system, which you want to use to export servers.
The *export directory* which will be used to store the server data is a subdirectory of the output directory with the name of the exported server: *output-directory-name/servername*.
 - d. Click **OK**.
 - e. A message shows you the name of the export directory.
2. Copy the subdirectory *servername* with all its files to your *input directory* on the host of the production system.
The input directory is an arbitrary directory which you use to store the data of imported servers.
The *import directory* *input-directory-name/servername* contains the data of the server to be imported.
3. In the tree:
 - a. Add a conversation, if necessary.
 - b. Select the **Servers** or **J2EEServers** folder.
 - c. Select the **Import server...** action of the **Selected** menu bar choice to display the **Import server** dialog box.
 - d. For **New server name**, enter the new name of the server that shall be created.
 - e. Enter the import directory, which consists of the input-directory-name and the old servername.

- 1) For **Input directory**, enter the name of the input directory which you use to store the data of imported servers on your production system (see above).
The data of the import directory will be used to import the server.
 - 2) For **Old server name**, enter the name of the server of the test system that was exported.
This name is the name of the subdirectory that has been stored on the host of the production system.
- f. Click **OK**.
 - g. Control and perhaps modify the properties of the server, especially **Control region proc name** and **Debugger allowed**.
 - h. If the logical resource mappings are not yet defined to the system, they are now defined automatically. Else, control and perhaps modify the correctness of the definitions of the logical resource mappings.
 - i. Define the logical resource mapping (LRM) instances.
 - j. Define the server instances.

Prepare for cold start

If you plan a cold start of the Application Server, you might want to save the configuration data of the active image on the host. WebSphere for z/OS will use these data during cold start to restore the active image.

The action **Prepare for cold start** on the active image saves it on the host for further use during a cold start of WebSphere for z/OS. After you have performed the cold start of the host and a restart of the Administration application, you will be able to work on the previous active image. All other conversations will be deleted. The definitions of the WebSphere for z/OS administrators will be saved.

1. Select the active image in the tree.
2. Select **Prepare for cold start** from the **Build** menu bar choice or from the pop-up menu.
3. In the **Confirm Prepare For Cold Start** dialog box, you can select the choice **Change Daemon IP Name** to enter a different IP name for your host.
4. Select **Yes** to confirm preparation for cold start.
5. WebSphere for z/OS will
 - save the definitions for the WebSphere for z/OS administrators
 - stop all application servers on the sysplex
 - carry over the environment file for each server instance by the data that are found in the properties for the appropriate sysplex, server, and server instance

Warning: If you have modified an environment file directly and not through the Administration application, any changes are overwritten when you prepare for cold start, because the information is retrieved from the System Management database.

Refer to “Modify environment variables” on page 49 or to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization* for more information on environment variables.

- save the configuration data of the active image on the host in the file `CBCONFIG/Sysplex-name/conversations/DdateTtime/configuration.xml` and
- save the deployed EAR files of each server instance of a J2EE server on the host in the directory

CBCONFIG/Sysplex-name/conversations/DdateTtime

- create a symbolic link from *CBCONFIG/Sysplex-name/current/configuration.xml* to *CBCONFIG/Sysplex-name/conversations/DdateTtime/configuration.xml* (that means a reference to the former filename leads to the latter file) where
 - *CBCONFIG* is the path that was specified in the environment variable *CBCONFIG*. The default is */WebSphere390/CB390*
 - *Sysplex-name* is the name of the sysplex
 - *date* is the current date
 - *time* is the current time.
- save all J2EE resource instances.

The directory structure on the host to save files to prepare for cold start is depicted in the following graph. It might be useful in case of errors.

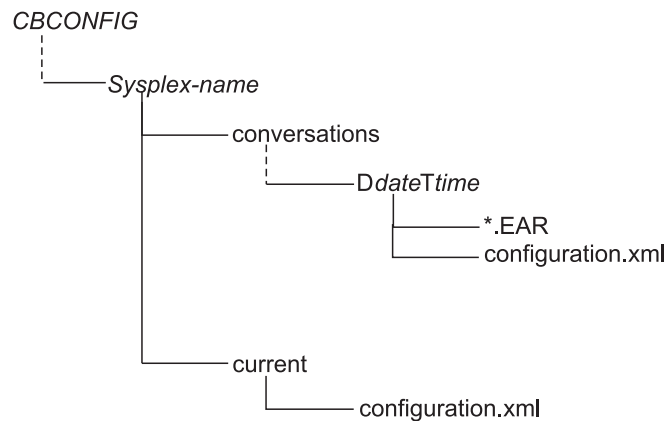



Figure 9. Directory Structure on the Host to Save Files to Prepare for Cold Start

6. Confirm the message that tells you that WebSphere for z/OS has been prepared for cold start. No further modifications of any conversation will be allowed. This state is indicated by the "blue lock" in front of the active image. 
7. Accomplish the cold start of WebSphere for z/OS . For more information about the *cold start*, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*
8. Restart the Administration application.
9. The only conversation that will be available is the active image that has been saved.

Modify the IP-address

If you have to modify the IP address of your host, you can declare the new IP address to WebSphere for z/OS as a side effect of the action **prepare for cold start** on the active image. Refer to "Prepare for cold start" on page 48 for more details.

Modify environment variables

The term *environment variable* is used in different contexts:

Server instance run-time environment variables

Server instance run-time environment variables are *Application Server* environment variables for each server instance object in a configuration.

The following chapter explains the concept of these environment variables.

Environment variables for z/OS clients

The Administration application does not manage environment variables for z/OS clients. You must create and manage z/OS client environment files and point to them from client programs. For more information, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*.

Workstation environment variables

Workstation environment variables are *Windows/NT* environment variables, and some of them can be set for the system management user interface.


For information about the workstation environment variables for the Administration and Operations applications refer to “Define workstation environment variables for login options” on page 6.

Server instance run-time environment variables

After the bootstrap process during installation and customization, the Application Server manages environment data through the Administration application and writes the environmental data into the system management database. To add or change environment variable data, you must enter an environment variable name and its value on the sysplex, server or server instance level. When you activate a conversation or prepare for cold start, the environment variable data are written to HFS files. The Application Server determines which values are the most specific for an environment file. For instance, a setting for a server instance takes precedence over the setting for the same variable for its server, and a setting for a server takes precedence over the setting for the same variable for its sysplex.

If you modify an environment file directly and not through the Administration application, any changes are overwritten when you activate a conversation or prepare for a cold start.

To modify an environment variable,

1. Decide whether this value should be valid
 - for all server instances belonging to each server of the sysplex,
 - for all server instances of the server or
 - for a single server instance.
2. Select the appropriate sysplex, server or server instance to open its properties form.
3. Click  on the tool bar or choose the **Modify** action of the **Selected** menu bar choice.
4. Select the Environment variable list in the properties form.

The Environment variable list contains three columns:

Environment variable list:

	Level	Name	Value
1	SPX	PATH	/usr/bin:/usr/l...
2	SPX	OWNPATH	/bin:/mybindir
3	SPX	ANOTHERONE	/usr/lib/demo/
4			
5			
6			
7			
8			
9			

Figure 10. An Environment Variable List

Level The **Level** field tells you at which level this variable is defined: SPX (Sysplex),SRV (Server), or SI (Server instance).

Name The name of the environment variable.

Value The value of the environment variable as it is defined at the level which is specified by its **Level**.

5. Double-click on the environment variable you want to edit.

If you want to add a new environment variable, click on an arbitrary row in the list.

The **Environment Editing** dialog appears.

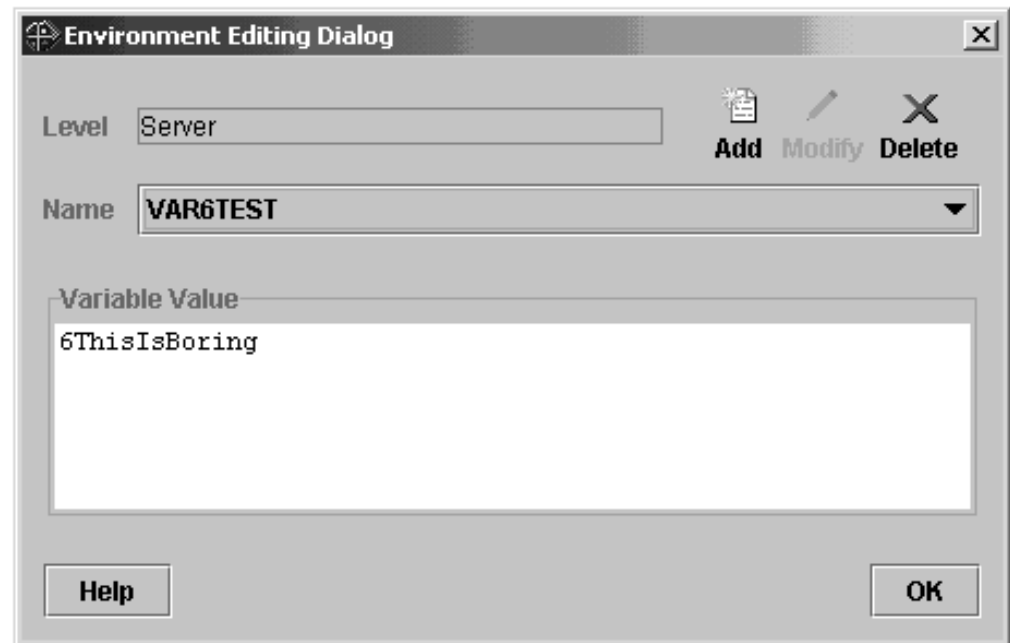


Figure 11. The Environment Editing Dialog

Name The **Name** field contains the name of an environment variable. To select a different environment variable, click on the ▾ button to expand the list of environment variables.

Level The **Level** field tells you at which level this variable is defined: Sysplex, Server, or Server instance. If you want to change the value of an environment variable on a more dedicated level, choose the **Environment Editing** dialog on this level and modify the environment

variable value. Afterwards, the list of names will contain a single entry for this environment variable on this level.

Add To add a new environment variable, click **Add**.

Modify

To modify an environment variable that is defined on a higher level, click **Modify**. You will change the value of the environment variable for this level. Afterwards, the list of names will contain a single entry for this environment variable on this level



Delete To delete an existing environment variable, click **Delete**. Note that only the definition on this level is deleted. If the environment variable is also defined on a higher level, the higher-level definition will still be valid.

Variable value

Enter the value of the environment variable in the **Variable value** field.

You might want to use the cut and paste function.

OK The **OK** button closes the dialog.

6. After you have closed the **Edit Variable** dialog, click  on the tool bar or choose the **Save** action of the **Selected** menu bar choice. To cancel your changes without saving them, click  on the tool bar or choose the **Cancel** action of the **Selected** menu bar choice.

Chapter 4. Administration objects

This chapter describes the objects that are depicted in the tree for administration. For each object, its properties and its appropriate actions are discussed. For general actions on these objects, refer to “Menu bar actions for the Administration application” on page 139.

The objects that make up your Application Server configuration include

- familiar z/OS entities, such as sysplex and system. They can be created, modified or controlled using the Administration and Operations applications, and
- objects that are specific to WebSphere for z/OS, such as application or J2EE application. They are created for you when you install an application into WebSphere for z/OS. You cannot modify these objects with the Administration application.

The objects that you define, modify or control using the Application Server Administration application are:

- Conversations. Each conversation represents a WebSphere for z/OS configuration.
- Sysplex and systems. These represent the z/OS sysplex and systems.
- Servers (both MOFW and J2EE servers) and server instances. These are the logical entities on which WebSphere for z/OS applications run. Most of the properties are defined at the server level. They include such things as whether it is a production server, and security properties that determine which clients can access the server.
- Containers. These hold the policies for a group of classes. These are associated with the homes of applications.
- Logical Resource Mappings, Logical Resource Mapping Instances, Logical Resource Mapping Connections, J2EE Resources and J2EE Resource Instances. These work together to give WebSphere for z/OS applications access to the resources (for example, DB2) that they need to run.

The objects that are created for you when you import an application into an Application Server configuration are: Application Family, Application, Home, Client Interface, Class, DLL, J2EE application, J2EE module, J2EE component, J2EE resource connection.

In addition, WebSphere for z/OS creates some objects for you when you create an object. The objects that are created for you by WebSphere for z/OS are owned by and required by WebSphere for z/OS. The names of these objects begin with CB. To avoid confusion, do not begin the names of objects you create with CB.

Location of Objects in the Tree

The following outline shows the location of the objects in the tree.

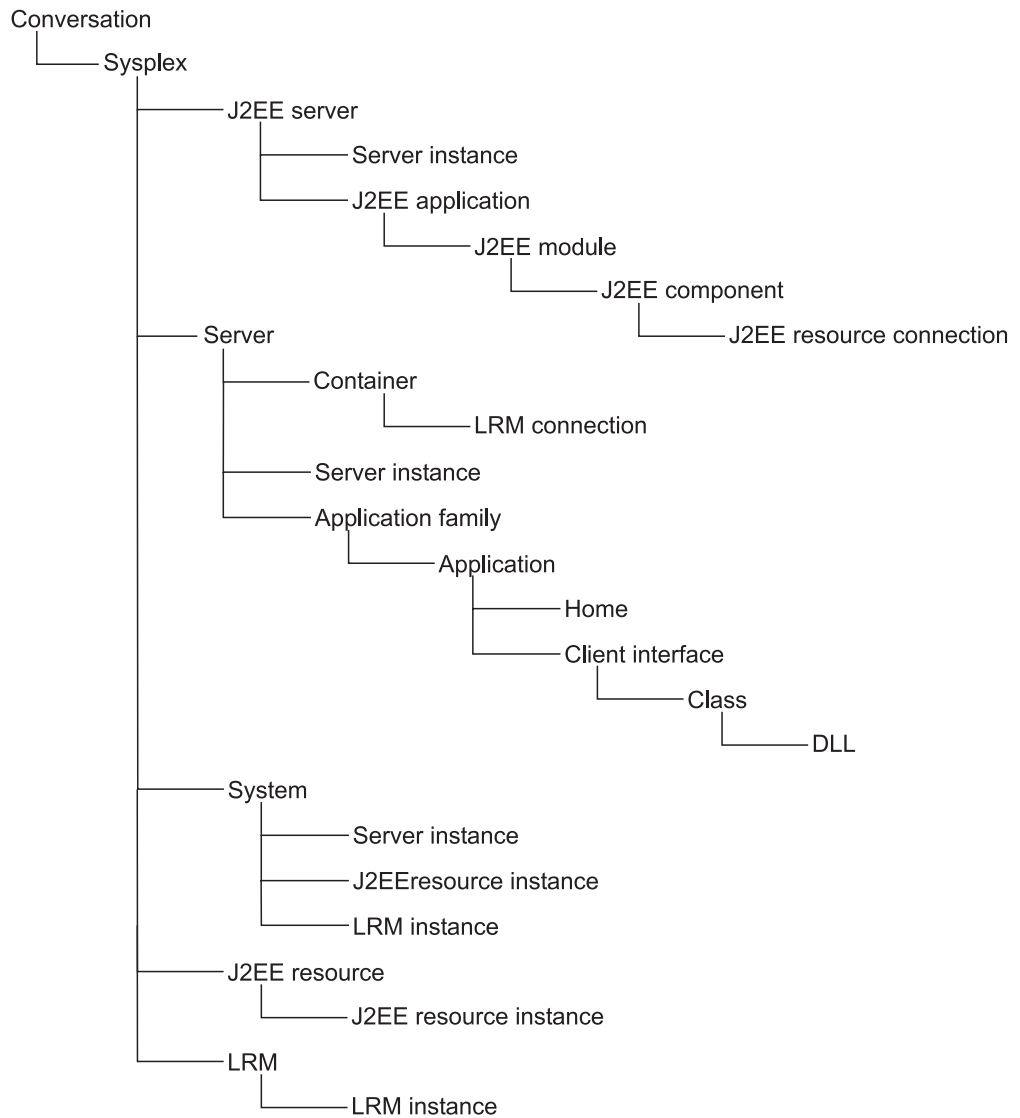


Figure 12. Location of Objects in the Tree

Application

An application is a program that runs as part of a business application package. The application contains DLLs, classes, client interfaces and homes. It is packaged as part of an application family.

Location in the tree:

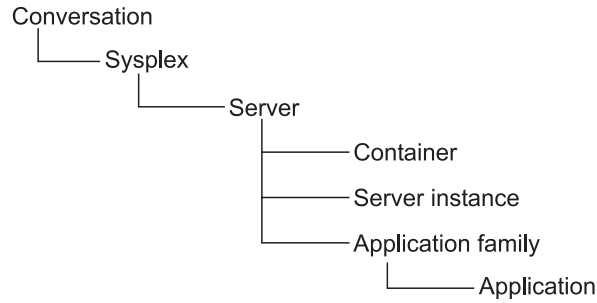


Figure 13. Location of a Application in the Tree

Properties:

The properties of an application are described below. They cannot be modified through the Administration application.

Application Name

Used to reference the resources that are associated with the application

Application Description

Descriptive text

Actions:

The actions that can be performed against an application are described below.

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the application belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Application family

An application family is a group of applications that will coexist in the same server. They are typically built and packaged in Object Builder, and imported to WebSphere for z/OS.

Location in the tree:

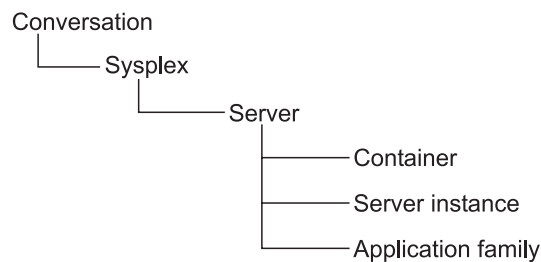


Figure 14. Location of a Application Family in the Tree

Properties:

The properties of an Application Family are described below. They cannot be modified through the Administration application.

Application Family Name

- The name of the application family
- Must be unique for the server

Application Family Description

Descriptive text

Actions:

The actions that can be performed against an Application Family are described below.

Delete Marks the application family and all of its subordinate objects as deleted

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the application family belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Class

A class is the type of a set of objects. Packaged within an application, it is imported into a model with the Administration application.

Location in the tree:

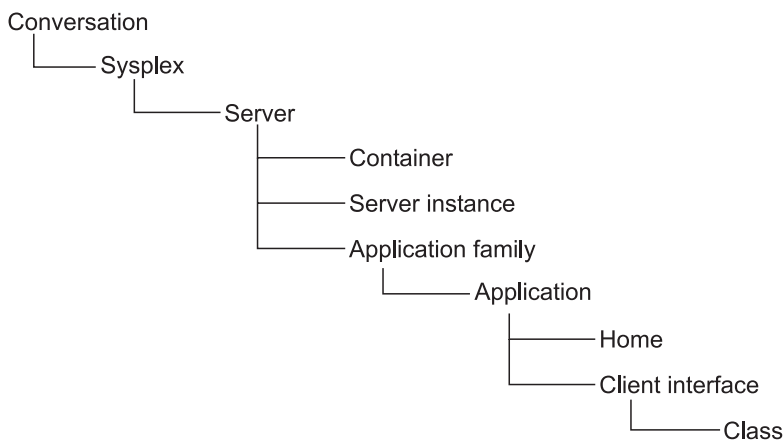


Figure 15. Location of a Class in the Tree

Properties:

The properties of a class are described below. They cannot be modified through the Administration application.

Class Name

- Used to reference the resources that are associated with the class
- Must be unique for the server

Class Description

Descriptive text

Class Major Version

Indicator of the level of the class

Class Minor Version

Indicator of the level of the class

Class Type

The type of the class

Create Function

The function that is used to create objects in the class

Actions:

The actions that you can perform against a class are described below.

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the class belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Client interface

A client interface consists of information defined for each of the clients on which an application runs. It is used by the application to reference the resources that are associated with the class. Packaged within the application, it is imported to WebSphere for z/OS using the Administration application.

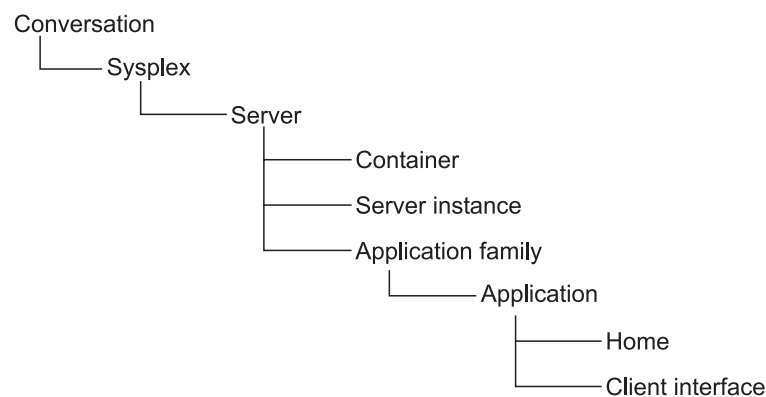
Location in the tree:

Figure 16. Location of a Client Interface in the Tree

Properties:

The properties of a Client Interface are described below. They cannot be modified through the Administration application.

Client Interface Name

- Used to reference the resources that are associated with the class
- Must be unique for the server

Client Interface Description

A description of the Client Interface

Actions:

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the client interface belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Container

Containers hold the policies for a group of classes. They provide object services for instances of managed objects, which are organized in homes. Each container can contain one or more homes. Containers provide translation, termination, and memory management policies, as well as caching mechanisms, for the managed objects within those homes, and maintain a list of their instances. A container is associated with an LRM through an LRM connection. Each server has a container, and each home must be connected to a container.

The association between a container and a home is established with Object Builder.

Location in the tree:

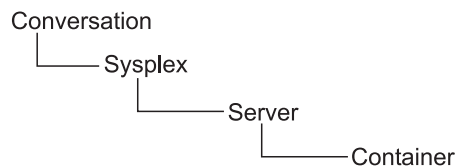


Figure 17. Location of a Container in the Tree

Properties:

The properties for containers are described below. For information about valid values refer to “Guidelines for Container Properties” on page 61.

Container name

- A handle that is used to reference the resources associated with the container
- 1 to 234 characters. The character may be letters, numbers or /, _ ,: , #, \$, or @

Container description

Up to 4096 characters of descriptive text

Method level access checks

- Defines that for security purposes, access is checked at the method level
- Select if required
- Default is that method level access checks are not required

Activation isolation policy

- Controls when, and how many copies of, an object are loaded into memory:

Transaction level

objects are loaded when a transaction touches them. This is the default. It is recommended for persistent objects

Container level

objects are loaded when a container touches them. This is recommended for transient objects.

- Change the value when editing the properties by clicking ▼ to display a menu of values and then selecting a value from the list

Passivation constraints

- Specifies whether objects can be passivated, that is, taken out of memory and returned to data storage:

Pinned

objects cannot be passivated. This is recommended for transient objects

Pinned for transaction life

objects can be passivated when a transaction completes

Not pinned

objects can be passivated at any time. This is the default. This is recommended for persistent objects.

- Change the value when editing the properties by clicking ▼ to display a menu of values and then selecting a value from the list

Managed object refresh policy

- Controls when a managed object is refreshed
- The default is At activation. This is recommended for persistent objects
- Change the value when editing the properties by clicking ▼ to display a menu of values and then selecting a value from the list

Transaction policy

- Defines the transactional support you determine for objects or Enterprise beans in this container.
- Used to effect transactional policy on the boundaries of business logic execution for a given object deployed in that container.
- Default is Required:

- The container either uses the client application's global transaction, or begins a global transaction on behalf of the client.

In a *global transaction*, the server application's processing is coordinated and treated as an atomic operation, that is, the application's updates to distributed resources are either all made (committed) or not made (rolled back).

- Use Required, unless you can guarantee that your server application abides by the rules for one of the other policies, as described below. These other policies improve portability and performance only for server applications with specific characteristics and processing.
- To provide transactional behaviors that simulate the absence of a global transaction in object space, use a *Hybrid-Global transaction*:

The hybrid-global transaction policies improve the portability and performance of only specific types of server applications, because these policies simulate the absence of a global transactional environment. In other words, the container does not represent and manage the

transaction for an object; instead, the container allows RRS and other z/OS resource managers to manage transactional context. To safely use these three alternative policies, you must thoroughly understand your server application's processing, and must abide by the rules for each policy. Otherwise, your application might not behave as you want or expect it to behave.

- Hybrid-global transactions are provided in several flavors:

Hybrid Global

The Application Server disregards the client transaction, if any, and allows RRS and other involved resource managers to manage the global transaction.

Each object method executed in the server has its own global transaction.

This policy ensures local-remote transparency; in other words, objects behave the same way regardless of where they are deployed. Using Hybrid Global provides performance improvements along with maximum flexibility for deploying server application objects.

Same-Server Hybrid Global

As with the Hybrid Global policy, the Application Server disregards the client transaction, if any, and allows RRS and other involved resource managers to manage the global transaction.

All methods executed in the same server instance, however, share the same global transaction.

All objects that the server application uses must be configured to run in the same server instance. Additionally, the server application:

- Assumes that it is running under a global transaction, and
- Does not attempt to manage its transactional context.

Supports Same-Server Hybrid Global

As with the policies other than *Required*, the Application Server allows RRS and other involved resource managers to manage the global transaction.

In this case, however, the server honors the client application's transactional context. If the client has a global transaction, all methods executed in the same server instance share that same global transaction.

All objects that the server application uses must be configured to run in the same server instance. Additionally, the server application:

- Assumes that it is running under a global transaction, and
- Does not attempt to manage its transactional context.

- For more details, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Assembling J2EE Applications*.

Guidelines for Container Properties

When defining container properties, you need to understand whether the objects that they affect are transient or persistent. The objects have been defined by the application developer. The following provides guidelines for defining container properties.

Transient objects

- **Activation isolation:** Container level
- **Passivation constraints:** Pinned
- **Managed object refresh policy:** Per transaction

Persistent objects

- **Activation isolation:** Transaction level
- **Passivation constraints:** Not pinned
- **Managed object refresh policy:** At activation

Actions:

The actions you can perform against a container are described below.

Add

- Creates a new container in the branch of the tree, and a label for required subordinate objects
- Available only against the label for Containers

Modify

Puts the container's properties form in edit mode, which allows you to make and save changes

Delete Removes the container from the tree

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the container belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Conversation

A conversation object lets you display and modify an Application Server configuration. You may have one or more conversations in your tree, representing the:

- Models that you create and modify. Through the commit and activate process you can make one of these models the active image.
- Image, that is, a model that has been committed and cannot be changed any more.
- Active image, that is, the actual configuration. This one cannot be modified or deleted.
- Previously active images that have been replaced by a new active image. These cannot be modified or re-activated, but they can be deleted.

If there are multiple administrators, each will see the conversations created by that administrator identity as well as the active conversation.

Location in the tree:

The conversation is the highest level object in the tree.

Properties:

The properties of a conversation are described below.

Conversation name

- The name of the conversation
- Up to 256 characters

Conversation description

Up to 4096 characters of descriptive text

Actions:

The actions that you can perform against a conversation are described below.

Add

- Creates a new conversation in the tree
- Available only against the label for conversations

Modify


Puts the conversation's properties form in edit mode, which allows you to make and save changes

Delete Marks the conversation and all of its subordinate objects as deleted

Validate

- Causes the Systems Management Server to check the integrity of the selected conversation and return errors if any
- Available only when the selected conversation has not yet been committed

Commit

- Verifies the selected conversation and returns errors if any, propagates the image of your conversation, and marks the conversation with  to indicate that you should complete the instructions
- Available only when the selected conversation has not yet been committed

Instructions

- Displays instructions in the right frame. Instructions describe manual steps you need to take to define Application Server applications to your system using the new model
- Available only when a committed conversation is selected

Complete..

- Displays the list of instructions perhaps with parts of it marked complete
- Allows to mark a task which has been completed manually

Activate

- Propagates the image contained in this conversation, making it the active image
- Available only when a committed conversation with completed instructions is selected

Prepare for cold start

This action is only available for the active image.

- Saves the configuration data of the active image and the data of the WebSphere for z/OS administrators on the host in the file `CBCONFIG/Sysplex-name/conversations/DdateTtime/configuration.xml` where
 - `CBCONFIG` is the path that was specified in the environment variable `CBCONFIG`. The default is `/WebSphere390/CB390`
 - `Sysplex-name` is the name of the sysplex
 - `date` is the current date
 - `time` is the current time.
- Allows to define a new IP address for the host.
- Carries over the environment file for each server instance.
- Stops all application servers on the sysplex.
- Requires a following cold start of WebSphere for z/OS.
- Disables any modification of a configuration until the Administration application is restarted.
- All configurations except for the active image will be lost.

DLL

A Dynamic Link Library (DLL) provides the implementation of one or more classes that are compiled and link-edited together in the DLL. A DLL can be loaded at run time. Packaged within an application, it is imported to WebSphere for z/OS by the Administration application.

Location in the tree:

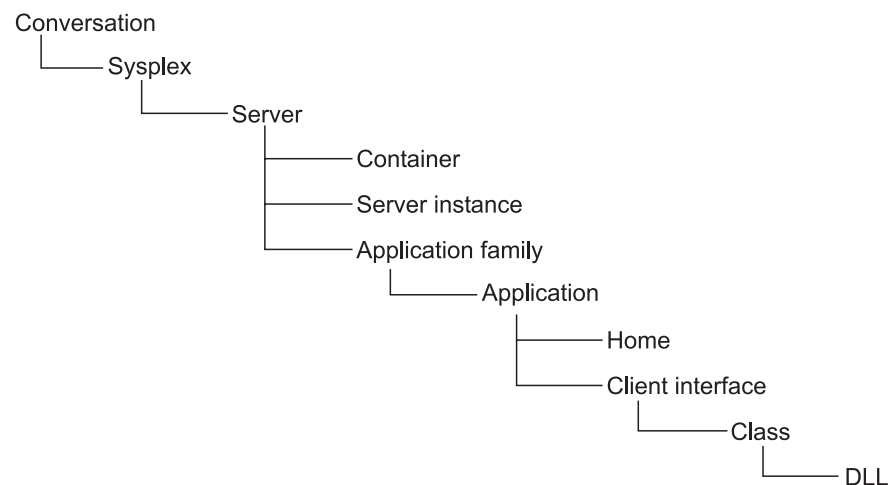


Figure 18. Location of a DLL in the Tree

Properties:

The properties of a DLL are described below. They cannot be modified through the Administration application.

DLL Name

- Used to reference the resources that are associated with the DLL
- Must be unique for the server

DLL Description

Descriptive text

Compiler Type

Compiler used to build the DLL

Containing Data Set

z/OS data set in which the DLL resides

Actions:

The actions that can be performed against a DLL are described below.

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the DLL belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Home

A home is the birthplace of managed objects. It is like a factory designed to manufacture only objects of a specific type. A home contains managed objects that are instances of the same class. In addition, a home serves as a collection for managed objects. A home provides object services for the instances in the home. Packaged within an application, it is imported to WebSphere for z/OS by the Administration application.

A home is associated with a container through Object Builder.

Location in the tree:

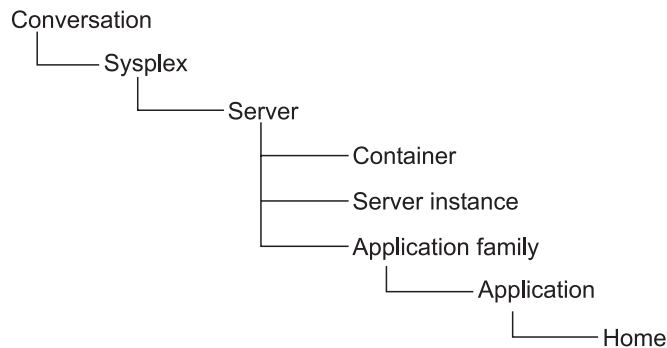


Figure 19. Location of a Home in the Tree

Properties:

The properties of a home are described below. They cannot be modified through the Administration application.

Home Name

- Used to reference the resources that are associated with the home collection in the Application Server Administration application
- Must be unique for the server

Home Description

Descriptive text

Managed Object Class Name

A class that contains the information on the Managed Object

Data Object Class Name

A class that contains the information on the Data Object. This class provides the data access function for the Managed Object

Mixin Class Name

A class that contains the information on the Mixin Object.

Primary Key Class Name

A class that contains the information on the Cursor Object. This class provides a class generic mechanism that can be used by the managed object framework to obtain and use the primary key information for the object

Cursor Class Name

A class that contains the information on the Cursor Object. This class provides a class generic mechanism that can be used to return the next instance in an iterable collection

Copy Helper Class Name

A class that contains the information on the Copy Helper Object. This class provides a class generic mechanism that can be used to pass all attribute data for a managed object to the home collection to allow it to be created

Visible In Workgroup

An indicator that the home should be registered in the workgroup portion of the name space when it is defined in the server

Visible In Cell

An indicator that the home should be registered in the Cell portion of the name space when it is defined in the server

Name As A Factory

The name that should be used to register the home as a factory in the name space

Name As A Home

The name that should be used to register the home in the name space

Business Object Class Override

Class name of the home object. For a specialized home, this is the subclass of the home interface that is used for the home.

Home Type Identifier

The type of object that is built by the home in its role as a factory. This piece of information is externalized to client ORBs that process the objects that are returned from the various create methods on the home interface.

Actions:

The actions that can be performed against a home are described below.

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the home belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

J2EE application

EJBs are composed into *J2EE applications*. J2EE applications organize EJBs (and WebComponents) into J2EE modules, and J2EE modules into J2EE components. A Web Component for our purposes is either a servlet or JSP.

A J2EE application consists of one or more J2EE modules and one J2EE application deployment descriptor.

For more information, refer to “Deploy J2EE applications” on page 32.

Location in the tree:

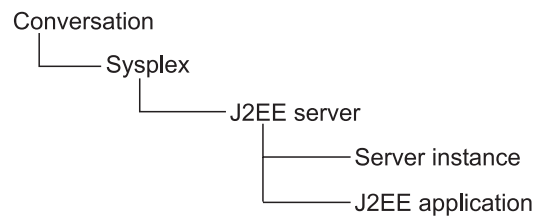


Figure 20. Location of a J2EE Application in the Tree

Properties:

With this release, no deployment information is displayed here. To see the information, you need to load the according EAR file into your application assembly tool.

The properties of a J2EE application are described below. They cannot be modified through the Administration application.

Application Name

Used to reference the resources that are associated with the J2EE application

Actions:

The actions that can be performed against an J2EE application are described below.

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the J2EE application belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Delete Deletes the J2EE application

J2EE component

In WebSphere for z/OS V4.0.1, *J2EE components* reflect Enterprise Java Beans, Servlets and JSPs. The components for Web modules (servlets, JSPs) are not displayed.

J2EE components are deployed as part of a J2EE application.

For more information, refer to “Deploy J2EE applications” on page 32.

Location in the tree:

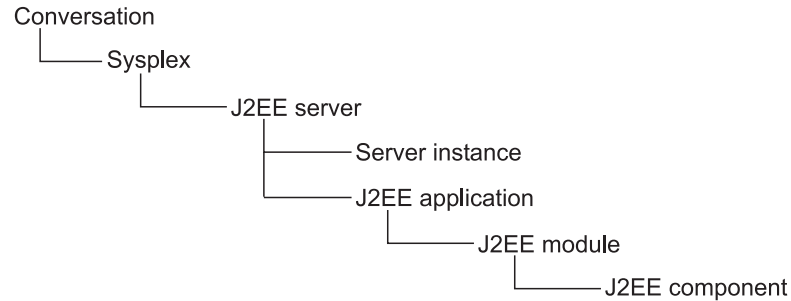


Figure 21. Location of a J2EE Component in the Tree

Properties:

With this release, no deployment information is displayed here. To see the information, you need to load the according EAR file into your application assembly tool.

The properties of a J2EE component are described below. They cannot be modified through the Administration application.

Component name

The logical name of the J2EE component

Reentrant

- If this box is checked, the entity bean which is held by this component is reentrant; that means the entity bean may be used concurrently.
- If this box is not checked, the bean which is held by this component may be used successively.

Home JNDI name

- The name and path under which the home for this component has been registered in JNDI
- This is the name that a client needs to use for a JNDI lookup when he wants to access this component

Actions:

The actions that can be performed against an J2EE module are described below.

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the J2EE component belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

J2EE module

A *J2EE module* is a collection of one or more J2EE components of the same container type with one component deployment descriptor of that type. This is either an EJB module (JAR) or a Web module (WAR).

For more information, refer to “Deploy J2EE applications” on page 32.

Location in the tree:

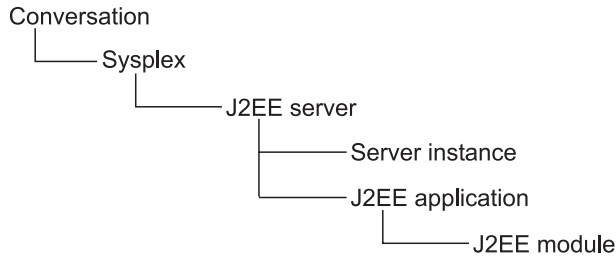


Figure 22. Location of a J2EE Module in the Tree

Properties:

With this release, no deployment information is displayed here. To see the information, you need to load the according EAR file into your application assembly tool.

The properties of a J2EE module are described below. They cannot be modified through the Administration application.

Module name

The logical name of the J2EE module

Actions:

The actions that can be performed against an J2EE module are described below.

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the J2EE module belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

J2EE resource

A *J2EE resource* is a logical grouping of J2EE resource instances. All J2EE resource instances are identical in structure: They inherit the J2EE resource's properties.

A J2EE resource enables a container which manages EJBs to connect to managed resources (see "Managed resources" on page 36), e.g. a JDBC API data source.

J2EE resources are connected to J2EE components with J2EE resource connections.

Note: A J2EE resource has to be defined before a J2EE application is installed that wants to use it.

Location in the tree:

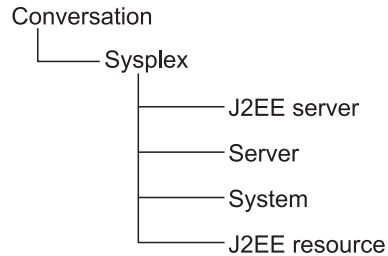


Figure 23. Location of a J2EE Resource in the Tree

Properties:

J2EE resource name

The name of the J2EE resource

J2EE resource description

Up to 4096 characters of descriptive text

Factory class name preview

- The factory class name depends on the J2EE resource type and is automatically filled in when the J2EE resource type is chosen. The factory supplies the J2EE resource class name to connect to managed resources
- Only indicated when a J2EE resource is added; this property will be inherited to the J2EE resource instances

J2EE resource class name preview

- The J2EE resource class name depends on the factory class name and is automatically filled in when the J2EE resource type is chosen. The J2EE resource class name enables the connection to managed resources
- Only indicated when a J2EE resource is added; this property will be inherited to the J2EE resource instances

J2EE resource type

- This drop-down list allows to choose different J2EE resource types; possible are predefined types (e.g. DB2Datasource, IMSDatasource, CICSDatasource) as well as self-defined or vendor-provided types. A J2EE resource type is defined as an XML file. This XML file also defines dedicated parameters that are needed when a J2EE resource instance is created.
- Can only be edited when the J2EE resource is added
- For more information about how to add a new J2EE resource type, refer to “Add a J2EE resource type” on page 45

Actions:

The actions that can be performed against a J2EE resource are described below.

Add

- Creates a new J2EE resource in the tree
- Available only against the label for J2EE resources

Modify

Puts the properties form in edit mode, which allows you to make and save changes

Delete Deletes the J2EE resource

J2EE resource connection

J2EE resource connections illustrate the external resource references of a J2EE component.

As these references are resolved during the installation of a J2EE application, the J2EE resource connections are then automatically created. The appropriate J2EE resource has to be created before the J2EE application is installed.

Location in the tree:

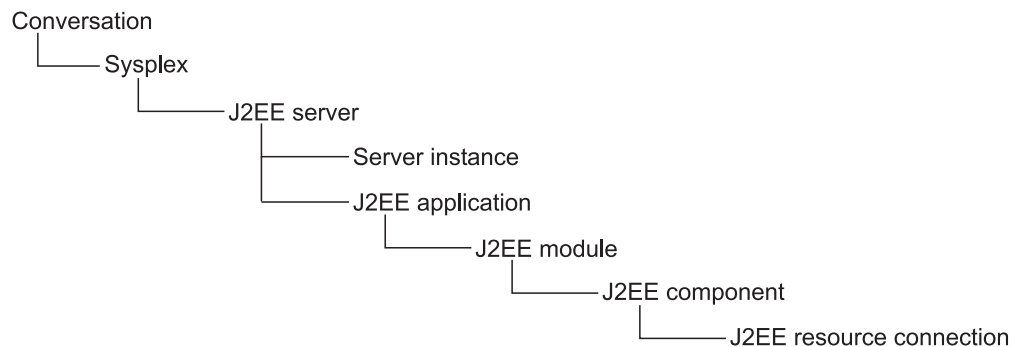


Figure 24. Location of a J2EE Resource Connection in the Tree

Properties:

The properties of a J2EE resource connection cannot be modified.

J2EE resource name

Name of the J2EE resource to which the J2EE server is connected.

J2EE resource instance

A J2EE resource instance is an instance of a J2EE resource that exists on a specific system. It provides information that allows a server instance which runs J2EE applications to connect to managed resources.

When a J2EE server has installed a J2EE application that references a J2EE resource, then each system with a server instance of this J2EE server must also have a J2EE resource instance of the referenced J2EE resource.

Each J2EE resource instance appears in two places in the tree, under the appropriate J2EE resource and under the appropriate system.

Location in the tree:

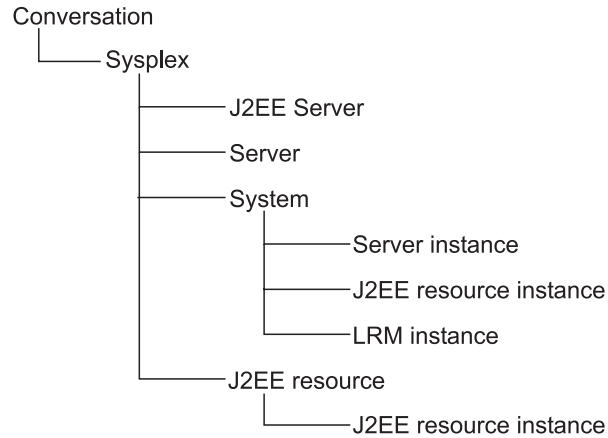


Figure 25. Location of a J2EE Resource Instance in the Tree

Properties:

The properties of a J2EE resource instance depend on its type. These types are added dynamically. They are not explained here.

There are two ways to obtain help:

- Either move the mouse on the entry field of the property and a help text will appear,
- or, if you are connected to the Web, you can obtain help from there: Click with the right mouse key on a property and choose **Web Help** in the context menu.

Actions:

The actions that can be performed against a J2EE resource instance are described below.

Add

- Creates a new J2EE resource instance in the tree
- Available only against the label for J2EE resource instances

Modify

Puts the properties form in edit mode, which allows you to make and save changes

Delete Deletes the J2EE resource instance

J2EE server

A WebSphere for z/OS server will host either J2EE applications or WebSphere for z/OS (MOFW) components; never both at once. A J2EE server is a server that hosts J2EE applications; in WebSphere for z/OS meaning it hosts EJBs, servlets, or JSPs. J2EE components' runtime execution is managed within a server by *containers*.

In WebSphere for z/OS V4.0.1, only a single container per server region will be supported. The server region initialization for a J2EE server will automatically initialize the EJB container.

Home registration will be done automatically when the server is initialized the first time (as part of the control region startup).

For more information, refer to “Deploy J2EE applications” on page 32.

Location in the tree:

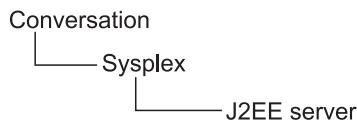


Figure 26. Location of a J2EE Server in the Tree

Properties:

The properties for a J2EE server are described below.

Server name

- The name of the server
- Must be unique within the sysplex
- 1 to 8 characters. The first character must be a letter; the others may be letters, numbers or #, \$ or @. Embedded blanks are not allowed

Server description

Up to 4096 characters of descriptive text

Control region identity

- The identity associated with the control region or system address space when the actual server image is generated
- Should be a trusted identity
- 1 to 8 characters. The first character must be a letter; the others may be a letter, number or #, \$ or @

Server region identity

- The identity associated with the server region address space when the actual server region image is created
- 1 to 8 characters. The first character must be a letter; the others may be a letter, number or #, \$ or @

Server region stack size

- The default size of the stack frame in the control region, in bytes
- A value of 0 causes the default size of 1,000,000 bytes to be used

Production J2EE Server

- Indicates that this is a production server. The Application Server run time limits certain real time debugging capabilities in production servers
- Default value is that this is a production server
- A production server supports more clients and uses more resources than a non-production server

Debugger allowed

- Specifies whether a debugger is allowed on the server
- Default value is that a debugger is not allowed


Object Level Trace hostname

- Object Level Trace (OLT) enables you to monitor the flow of a distributed application, and to seamlessly debug client and server code from a single workstation. OLT records method calls from the client application, or servlet, to distributed business objects, servlets, JSPs, or EJBs residing on WebSphere application servers
- Enter the fully-qualified name or TCP/IP address of the machine running your OLT server


Object Level Trace port

- Enter the port where the OLT server listens for connecting OLT clients
- The default port for the OLT server is 2102

Isolation policy

- Specifies how the server region should isolate user transactions from each other, that is, whether each transaction is assigned its own server region
- Click  to display a list of valid values
- Default value is Multiple transactions per server region
- Assigning a server region to each transaction provides greater security than allowing multiple transactions per server region

Replication policy

- Specifies how many server regions should be started
- Used when the server regions are replicated inside the server region
- Click  to display a list of valid values
- Default value is Replicate as needed
- One per server is recommended for transient objects

Local identity

- Identity assigned to a local non-authenticated client that connects to the server
- Required only if **allow non-authenticated clients** is selected
- 1 to 8 characters. The first character must be a letter; the others may be a letter, number or #, \$ or @

Remote identity

- Identity assigned to a remote non-authenticated client that connects to the server
- Required only if **allow non-authenticated clients** is selected
- 1 to 8 characters. The first character must be a letter; the others may be a letter, number or #, \$ or @

Register transaction factory

- Indicate that this server is a transaction factory, for use by a client that is starting a transaction
- Checked by default

Allow server region recycling

- Indicates that a server region is allowed to perform recycling
- For recycling the server region is stopped after completion or roll back of the last transaction indicated in the server recycling interval. The storage no longer in use is recovered. The server region is started again
- Checked by default

Server recycling interval

- Indicates the server recycling interval, that is the number of completed or rolled back transactions between recycling (garbage collections)
- The default value is 50000
- The valid range is from 1 to 2147483648(2^{31})

Logstream name

- The name of the server logstream. A valid logstream name consists of 1 or more qualifiers separated by periods, up to a maximum of 26 characters
- Each qualifier can contain up to 8 numbers, letters or #, \$ or @
- The first character of each qualifier must be a letter or #, \$ or @
- Each qualifier must be separated by periods, which must be counted as characters

Control region proc name

- The JCL PROC name is used to start control regions — which actually represent server instances in the Administration and Operations applications — for the corresponding server
- The default is the server name
- 1 to 8 characters. The first character must be a letter; the others may be letters, numbers or #, \$ or @. Embedded blanks are not allowed

Enable Setting OS Thread Identity to RunAs Identity

- Check this field to enable the J2EE server to access methods within J2EE applications under the authority of a specific role (requester rolename, server rolename or specified rolename)
- The rules for the methods and the rolenames which are allowed to access the methods are specified in the extended deployment descriptor of the J2EE application
- If checked, the Native OS Thread Security Identity can be modified to the user ID that corresponds to that method's RunAs identity while running an EJB method. This requires all system resources (files, databases, sockets) accessed during execution to be authorized for access by that identity
- Default is unchecked, that means, the security identity of the J2EE server is used, and it cannot be modified during the running of an application

Allow non-authenticated clients

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

If *non-authenticated clients* are allowed, clients that have not been authenticated may connect to this server.

- Default is to not allow non-authenticated clients to connect to this server
- If **allow non-authenticated clients** is selected without assigning a **local identity**, the default value is set to the value of environment variable `DEFAULT_UNAUTH_CLIENT_ID` on the sysplex object. If it is not set, the default value is `CBGUEST`. The message `BBON0560I` is displayed in the status bar.
- If **allow non-authenticated clients** is selected without assigning a **remote identity**, the default value is set to the value of environment

variable `DEFAULT_UNAUTH_CLIENT_ID` on the `sysplex` object. If it is not set, the default value is `CBGUEST`. The message `BBON0561I` is displayed in the status bar.

Refer to “Define the security class of a server” on page 27 for more information about security classes.

Userid password allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

If *userid password* is allowed, clients may connect to this server, with the MVS user ID and password being used for security.

- Provides the least security of the options **Userid password allowed**, **Userid passticket allowed**, **DCE allowed**, **SSL Type 1 allowed**, **SSL Client Certificates allowed**
- Check the box to allow this level of security to be used for this server

Refer to “Define the security class of a server” on page 27 for more information about security classes

Userid passticket allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

The *passticket* is a one-time-only, system-generated password.

- Check the box to allow this level of security to be used for this server

Refer to “Define the security class of a server” on page 27 for more information about passticket and security classes.

DCE allowed


Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

The client and server must contact the *DCE* security server, acting as a third party, before communications can occur.

- Check the box to allow this level of security to be used for this server and then enter a value for **DCE quality of protection** and **DCE keytab file**

Refer to “Define the security class of a server” on page 27 for more information about DCE and security classes.

DCE quality of protection

- Indicates the type of protection through DCE
- Click  to display a list of valid values, with increasing security:
 - No protection
 - Message integrity: Messages have been signed, which ensures that they have not been modified
 - Message confidentiality: Messages are encrypted
- Default is no protection

DCE keytab file

- File path in the HFS for the server’s DCE keytab file. The DCE keytab file contains the server’s DCE password
- Up to 1,029 characters

- Required if **DCE allowed** is checked

SSL Type 1 (Basic Authentication) allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

SSL Type 1 or *SSL Basic Authentication* is a security mechanism that authenticates the server using its digital certificate and encrypts messages flowing across the client/server connection. The server authentication entails ensuring that the server's certificate was granted by a certificate authority known to the client. The client's identity is established by user ID and password.

- Check the box to allow this level of security to be used for this server, and then fill out the values for the SSL related elements (**SSL RACF Keyring, SSL V2 timeout, SSL V3 timeout**)
- The server's certificate must be defined as the default certificate in the keyring specified in **SSL RACF Keyring**
- Default is not allowed

Refer to "Define the security class of a server" on page 27 for more information about SSL and security classes.

SSL Client Certificates allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

SSL Client Certificates ensure that the client authenticates the server and the server authenticates the client. Both client and server authentication mechanisms are done by SSL, each side presents a certificate. This aspect of authentication guarantees that servers can trust their clients.

- Check the box to allow this level of security to be used for this server, and then fill out the values for the SSL related elements (**SSL RACF Keyring, SSL V2 timeout, SSL V3 timeout**)
- The client certificate must be specified in the **SSL RACF keyring**
- All certificate authorities for servers you need to access, must have certificates defined to RACF and be connected to the client's keyring

Refer to "Define the security class of a server" on page 27 for more information about SSL and security classes.

Kerberos allowed

Indicates that this security class is desired for authenticating Application Server clients and servers.

SSL Kerberos is a security mechanism that allows a client to authenticate a server using the server's digital certificate. The client's identity is verified by the server using Kerberos authentication methods. Message protection, which may include data privacy and integrity, is supplied by the Secure Sockets Layer (SSL).

- The server's certificate must be defined as the default certificate in the keyring specified in **SSL RACF Keyring**
- Check the box to allow this level of security to be used for this server, and then fill out the values for the SSL related elements (**SSL RACF Keyring, SSL V2 timeout, SSL V3 timeout**)
- Default is not allowed

Refer to “Define the security class of a server” on page 27 for more information about Kerberos and security classes.

Send Asserted Identities allowed

Indicates the security class that is desired to prevent unauthorized client access to Application Server resources.

Outbound requests originating from this server can send RACF userids over an SSL connection to a remote Application Server without additional authentication information to impersonate an originating client.

- Check the box to allow this level of security to be sent by this server, and then fill out the values for the SSL related elements (**SSL RACF Keyring**, **SSL V2 timeout**, **SSL V3 timeout**)
- Available only if SSL is configured on this system
- Default is not allowed

Refer to “Define the security class of a server” on page 27 for more information about asserted identities and security classes.

Accept Asserted Identities allowed

Indicates the security class that is desired to prevent unauthorized client access to Application Server resources.

Accept Asserted Identities allowed enables a target server to accept SSL Asserted Identities.

- Check the box to allow this level of security to be used for this server, and then fill out the values for the SSL related elements (**SSL RACF Keyring**, **SSL V2 timeout**, **SSL V3 timeout**)
- Available only if **SSL Client Certificates** is allowed
- Default is not allowed

Refer to “Define the security class of a server” on page 27 for more information about asserted identities and security classes.

SSL Use Confidentiality Only

- If **SSL Use Confidentiality Only** is set, both encryption and authentication will be used by the peer systems
- If the peer system does not have these suites available, the SSL connection will fail
- Only available if one of the SSL options **SSL Basic Authentication allowed**, **SSL Client Certificates allowed**, **Kerberos allowed**, **Send Asserted Identities allowed**, or **Accept Asserted Identities allowed** is set

SSL RACF Keyring

- The name of the RACF keyring that contains the appropriate keys and certificates for SSL
- Up to 237 characters
- All characters supported by RACF for profiles are allowed
- Default is CBKeyring
- Required if **SSL Basic Authentication allowed**, **SSL Client Certificates allowed**, **Kerberos allowed**, **Send Asserted Identities allowed**, or **Accept Asserted Identities allowed** is checked

SSL V2 timeout

- Number of seconds for SSL Version 2 session data to time out

- A number from 1 to 100
- Default is 100
- Required if **SSL Basic Authentication allowed, SSL Client Certificates allowed, Kerberos allowed, Send Asserted Identities allowed, or Accept Asserted Identities allowed** is checked

SSL V3 timeout

- Number of seconds for SSL Version 3 session data to time out
- A number from 1 to 86400 (1 day)
- Default is 600
- Required if **SSL Basic Authentication allowed, SSL Client Certificates allowed, Kerberos allowed, Send Asserted Identities allowed, or Accept Asserted Identities allowed** is checked

Security preference list

- Defines which security types can be used when clients connect to this server, and in what order of preference. An order of preference can only be assigned to a security type if that type has been checked above
- In WebSphere for z/OS V4.0.1 the security order for a simple client for a remote call is
 1. SSL Client Certificates / SSL Asserted Identity
 2. Kerberos (over SSL)
 3. SSL Basic Authentication
 4. Passticket
 5. DCE
 6. Password
 regardless of what is specified.

Write Server Activity SMF Records

- If checked, enables SMF recording of the server activity
- For each activity that is run inside a server instance of this server, a single record is created
- Use these records to perform basic charge back accounting as well as profiling of own applications to determine in detail what is happening inside the transaction server
- **Note:** Make sure that, on z/OS, SMF record type 120 is specified to enable Application Server information recording

Write Container Activity SMF Records

- If checked, enables SMF recording of the container activity
- There is a single record for each container that is part of an activity
- Data that describe the actual business functions invoked within the server's containers are monitored
- **Note:** Make sure that, on z/OS, SMF record type 120 is specified to enable Application Server information recording

Write Server Interval SMF Records

- If checked, enables SMF recording of the server activity in intervals that you specify in **SMF Interval Length**
- There is a single record for each server instance that has interval recording active during the specified interval

- If the server has multiple server instances, then a record for each server instance is written
- **Note:** Make sure that, on z/OS, SMF record type 120 is specified to enable Application Server information recording

Write Container Interval SMF Records

- If checked, enables SMF recording of the container activity in intervals that you specify in **SMF Interval Length**
- There is a single record created for each active container located in the server within the interval being recorded. If there is more than one server instance associated with the server, there will be a record for the container from each server instance
- **Note:** Make sure that, on z/OS, SMF record type 120 is specified to enable Application Server information recording

SMF Interval Length

- Length of the recording intervals for SMF monitoring which repeat continuously
- Only enabled if **Write Server Interval SMF Records** or **Write Container Interval SMF Records** is specified
- A length from 15 seconds to 86400 seconds (24 hours)
- Specify the seconds of the interval length
- Default is 3600 seconds (1 hour)
- Specify a 0 to indicate the usage of the interval length from the SMF product settings

Environment variable list

- Contains the definitions of the environment variables that are common to the server.
- Each server instance of the server inherits these values.
- The Environment variable list contains three columns:
 - Type** The **Type** field tells you at which level this variable is defined: SY (Sysplex), SV (Server), or SI (Server instance).
 - Name** The name of the environment variable.
 - Value** The value of the environment variable as it is defined at the level which is specified by its **Type**.
- The current maximum length of environment variable values is currently limited to 4096 characters due to an LE restriction on z/OS.
- To edit the environment variable list, ensure you are in the **Modify** action. Then do one of the following:
 - Double click the left mouse button on any cell in a row. Refer to “Server instance run-time environment variables” on page 50 for a description on how to manage the Edit Variable dialog.
 - Click the left mouse button on any cell in a row to highlight the row, and then press Enter.
- To view the environment variable list when you are not in the **Modify** action, do one of the following:
 - Double click the left mouse button on any cell in a row.
 - Click the left mouse button on any cell in a row to highlight the row, and then press Enter.

- For a detailed description of environment variables, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*.

Actions:

Add

- Creates a new server in the branch of the tree, and labels for required subordinate objects
- Available only against the label for Servers
- Causes WebSphere for z/OS to create the associated Web container

Modify

Puts the server's properties form in edit mode, which allows you to make and save changes

Delete Marks the server and everything below it in the branch of the tree as deleted

Install J2EE application...

Deploys a J2EE application consisting of Enterprise JavaBeans which is archived in an Enterprise ARchive (EAR) file on your workstation to the J2EE server.

For more information, refer to "Deploy J2EE applications" on page 32.

Import server...

- Creates a new J2EE server with the server properties from the referenced J2EE server
- Available only against the label for J2EE Servers

See "Migrate a test server to a production system" on page 47 for more information.

Export server...

- Exports the J2EE server properties to the specified host path
- Only available for a J2EE server of an active image (refer to "States of a conversation" on page 20)

For more information see "Migrate a test server to a production system" on page 47.

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the server belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Logical resource mapping

A logical resource mapping (LRM) is used by containers that locate logical resources on the system. There may be one or more LRMs for each type of subsystem in the sysplex.

Information specific to the system on which a subsystem runs is found in the LRM instance.

LRMs are connected to containers with LRM connections. Each of the LRMs connected to a container must have an LRM instance for each system with a server instance on which the container runs.

Location in the tree:

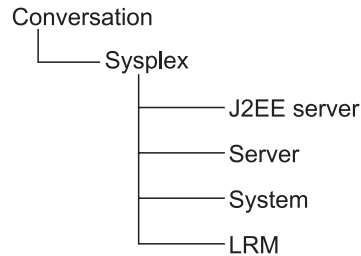


Figure 27. Location of a Logical Resource Mapping in the Tree

Properties:

The properties of an LRM are described below.

Logical Resource Mapping Name

- The name of the LRM
- Must be unique in the sysplex
- 1 to 64 characters. The characters may be letters, numbers or /, _ ,: , #, \$ or @

Logical Resource Mapping Description

Up to 4096 characters of descriptive text

Admin Object Class Name

- The name of the class for the admin object
- Modifiable only if you select an LRM subsystem type of GENERIC
- 1 to 242 characters. The characters may be letters, numbers or /, _ ,: , #, \$ or @

Admin Object DLL Name


- The name of the DLL for the admin object
- Modifiable only if you select an LRM subsystem type of GENERIC
- Up to 8 characters. The characters may be letters, numbers or /, _ ,: , #, \$ or @

Admin Object Class Create Function

- Function that, when called, causes the DLL to be loaded
- Modifiable only if you select an LRM subsystem type of GENERIC
- 1 to 256 characters. The characters may be letters, numbers or /, _ ,: , #, \$ or @

LRM Subsystem Type

- The type of subsystem with which this LRM is associated, for example, DB2
- Cannot be modified once the LRM has been saved

- When you are creating a new LRM, click  to display a list of valid values. If you select a type of `GENERIC`, you must also provide values for admin object class name, admin object DLL name, and admin object class create function
- The LRM subsystem types are:

Subsystem type	Description
DB2	Application adaptor that uses DB2
IMS_OTMA_PAA	Procedural application adaptor that uses IMS with the OTMA interface. This requires that Application Server and IMS be on the same system.
IMS_APPC_PAA	Procedural application adaptor that allows Application Server to communicate through MVS/APPc with IMS on a remote or local system.
CICS_EXCI_PAA	Procedural application adaptor that uses CICS with the EXCI interface
GENERIC	User defined LRM subsystem type

Actions:

The actions that can be performed against an LRM are described below.

Add

- Creates a new LRM in the branch of the tree, and a label for required subordinate objects
- Available only against the label for Logical Resource Mappings

Modify

Puts the LRM's properties form in edit mode, which allows you to make and save changes

Delete Marks the LRM and all of its subordinate objects as deleted. The associated LRM connections are removed from the tree

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the LRM belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Logical resource mapping connection

The Logical Resource Mapping (LRM) Connection provides a link between a container and one or more LRMs. The properties of an LRM connection are described below.

Location in the tree:

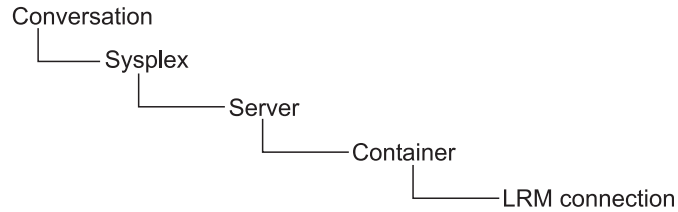



Figure 28. Location of a Logical Resource Mapping Connection in the Tree

Properties:

Logical Resource Mapping Name

- The name of the associated Logical Resource Mapping object
- When adding or modifying an LRM connection, click  to display a menu of LRM names and select one from the menu. The menu includes all LRMs defined in the sysplex and not connected to the container.

Actions:

The actions that can be performed against an LRM Connection are described below.

Add

- Creates a new instance of the LRM connection in the branch of the tree. An LRM must have already been created in the sysplex
- Available only against the label for LRM Connections
- Unavailable if the container is already connected to every LRM in the sysplex or if no LRMs have been defined in the sysplex

Delete

- Deletes the LRM connection and removes it from the tree
- Can be reversed, that is, an LRM connection that is deleted can be added back to the tree

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the LRM connection belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Logical resource mapping instance

An LRM instance is an instance of an LRM that exists on a specific system. It provides information that allows a server instance to connect to a subsystem on that system.

Each LRM instance appears in two places in the tree, under the appropriate LRM and under the appropriate system.

Each of the LRMs connected to a container must have an LRM instance for each system with a server instance on which the container runs.

Location in the tree:

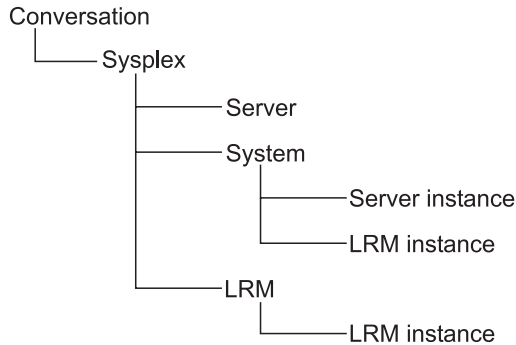


Figure 29. Location of a Logical Resource Mapping Instance in the Tree

Properties:

Logical Resource Mapping Instance Name

- The name of the LRM instance
- Must be unique in the LRM
- 1 to 32 characters. The characters may be letters, numbers or /, _ ,: , #, \$ or @

Logical Resource Mapping Instance Description

Up to 4096 characters of descriptive text

Logical Resource Mapping Name

- The name of the LRM of which this is an instance
- Cannot be changed once the LRM instance has been defined
- When you add an LRM instance under an LRM, this property cannot be modified. When you add an LRM instance under a system, click ▼ to display a list of LRMs. Selecting a different LRM from the list causes the connection data to be refreshed. This is because selecting a different LRM name may result in the LRM instance being connected to a different type of LRM, which may require different connection data to be specified

System Name

- The name of the system for which the LRM instance provides connection data
- Cannot be changed once the LRM instance has been defined
- When you add an LRM instance under a system, this property cannot be modified. When you add an LRM instance under an LRM, click ▼ to display a list of systems

Connection Data

- Information used by the server instance to connect to the appropriate subsystem
- To edit a cell in the connection data table, double click the cell. To get a cursor for this cell one more mouseclick is required. A triple click gets you directly in that editing mode.
- You can add values for names, or name and value pairs
- Names and values are each up to 256 characters

- Depending on the **LRM subsystem type**, which is a property of the logical resource mapping, subsystem type specific data have to be entered:

DB2:

Connection Data Table Entry	Description
DB2 Subsystem Name	DB2 subsystem name or group attachment name
CollectionID	ID which is used to reference DB2; the same ID is entered when DB2 is binded.

IMS_OTMA_PAA:

Connection Data Table Entry	Description
XCF group name	IMS-OTMA XCF group name. Equals the GRNAME entry in the DFSPBxxx proclib member used for initialization.
XCF partner name	Identifies the specific IMS with which the server communicates. The XCF partner name is the name specified by the OTMANM parameter in the IMS DFSPBxxx proclib member used for initialization. If no OTMANM parameter is defined, then the name specified by the APPLID1 parameter in the IMS DFSPBxxx member will be used as the default XCF partner name.
Number of sessions	Specify 1
TPIPE prefix	The transaction pipe name.

IMS_APPC_PAA:

Connection Data Table Entry	Description
Local LU name	Fill in the logical unit (LU) name associated with WebSphere for z/OS. This local LU name is defined in an LUADD statement in the APPCPMxx parmlib member for the system on which the Application Server runs. Look for the LUADD statement for the LU associated with WebSphere for z/OS (you might need the help of your system programmer to correctly identify the LU). Use the value specified on the ACBNAME parameter as the local LU name.
Partner LU name	Fill in the name of the LU with which the Application Server will initiate an APPC conversation. This partner LU is defined in an LUADD statement in the APPCPMxx parmlib member for the system on which IMS runs. The IMS subsystem may be, but does not have to be, on a system other than the one on which the Application Server runs. Look for the LUADD statement for the LU associated with IMS (an LU associated with IMS has the IMS subsystem name specified for the SCHED parameter on the LUADD statement). Use the value specified on the ACBNAME parameter as the partner LU name.
VTAM logmode name	Fill in the name of the VTAM logmode that designates the network properties to be associated with any APPC conversations between this local LU and its partner LU. Logmode names appear in the VTAM logon mode table, which exists in your installation's VTAMLIB data set.

Connection Data Table Entry	Description
APPC Conversation Timeout Value	Specify the length of time, in minutes, for the Application Server to wait for a response to the Allocate call and any subsequent calls the server issues during its conversation with IMS. Valid timeout values range from 0 through 1440, which is 24 hours. If you specify a value that is less than the value set for the OTS_DEFAULT_TIMEOUT environment variable, the APPC conversation timeout value will have no effect. Look for the OTS_DEFAULT_TIMEOUT environment variable setting for the application server's control and server regions.
APPC Sync Level (Syncpt, None, AutoTran)	<p>This value controls the type of APPC/MVS conversation the Application Server uses to communicate with IMS. Base your choice on the transaction policies you select for containers in this server configuration, and the characteristics of the applications to be deployed in this server. Use a sync level value that corresponds with the transactional context of the request that the server is currently processing.</p> <p>Syncpt With Syncpt, the server allocates a protected conversation, which preserves the global transactional context for the interaction between the server and the IMS subsystem, and allows the system to recover any resources if conversation errors or failures occur.</p> <p>Use Syncpt</p> <ul style="list-style-type: none"> - if this LRM is connected to one or more containers that all use the transaction policy Required, or - if you cannot guarantee that your server application will always run on the same z/OS system on which the IMS subsystem runs. <p>None With None, APPC/MVS, WebSphere for z/OS, and IMS do not coordinate any processing done on behalf of a distributed application; without the overhead of coordination, your application's performance improves.</p> <p>Use None judiciously. In this case, resources that the application uses might be in inconsistent states if conversation errors or failures occur.</p> <p>AutoTran</p> <p>The easiest way to match the sync level and context is to select AutoTran, so the system can determine which conversation type, Syncpt or None, is appropriate for the transactional context associated with the current thread of execution: If the current thread has a local transactional context, the server uses a sync level of None; for a global transactional context, the server uses Syncpt.</p> <p>Use AutoTran, if this LRM is connected to one or more containers that use a transaction policy other than Required.</p>

CICS_EXCI_PAA:

Connection Data Table Entry	Description
CICS applid	The CICS application ID.

GENERIC:

No predefined table entry

Actions:

The actions that can be performed against an LRM instance are described below.

Add

- Creates a new LRM instance in the branch of the tree for the LRM and the system
- Available only against the label for Logical Resource Mapping Instances
- At least one LRM and one system must have been defined for the sysplex

Modify

Puts the LRM instance's properties form in edit mode, which allows you to make and save changes

Delete Marks the LRM instance as deleted

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the LRM instance belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Server (Managed Object Framework)

A server is a logical grouping of server instances. All server instances within a server are identical in structure. Administration is usually done at the server level. You can manage a server through the Operations application.

Home registration will be done automatically when the server is initialized the first time (as part of the control region startup).

There are two different kinds of servers:

(MOFW) Server

A *server* or *MOFW server* manages Managed Object Framework (MOFW) applications.

J2EE Server

A *J2EE server* (see "J2EE server" on page 71) manages J2EE applications.

Location in the tree:

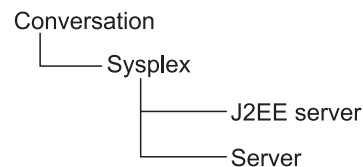


Figure 30. Location of a Server in the Tree

Properties:

The properties for a server are described below.

Server name

- The name of the server
- Must be unique within the sysplex
- 1 to 8 characters. The first character must be a letter; the others may be letters, numbers or #, \$ or @. Embedded blanks are not allowed

Server description

Up to 4096 characters of descriptive text

Control region identity

- The identity associated with the control region or system address space when the actual server image is generated
- Should be a trusted identity
- 1 to 8 characters. The first character must be a letter; the others may be a letter, number or #, \$ or @

Server region identity

- The identity associated with the server region address space when the actual server region image is created
- 1 to 8 characters. The first character must be a letter; the others may be a letter, number or #, \$ or @

Server region stack size

- The default size of the stack frame in the control region, in bytes
- A value of 0 causes the default size of 1,000,000 bytes to be used

Production server

- Indicates that this is a production server. The Application Server run time limits certain real time debugging capabilities in production servers
- Default value is that this is a production server
- A production server supports more clients and uses more resources than a non-production server

Debugger allowed

- Specifies whether a debugger is allowed on the server
- Default value is that a debugger is not allowed


Object Level Trace hostname

- Object Level Trace (OLT) enables you to monitor the flow of a distributed application, and to seamlessly debug client and server code from a single workstation. OLT records method calls from the client application, or servlet, to distributed business objects, servlets, JSPs, or EJBs residing on WebSphere Application servers
- Enter the fully-qualified name or TCP/IP address of the machine running your OLT server


Object Level Trace port

- Enter the port where the OLT server listens for connecting OLT clients
- The default port for the OLT server is 2102

Isolation policy

- Specifies how the server region should isolate user transactions from each other, that is, whether each transaction is assigned its own server region
- Click  to display a list of valid values
- Default value is Multiple transactions per server region
- Assigning a server region to each transaction provides greater security than allowing multiple transactions per server region

Replication policy

- Specifies how many server regions should be started
- Used when the server regions are replicated inside the server region
- Click  to display a list of valid values
- Default value is Replicate as needed
- One per server is recommended for transient objects

Local identity

- Identity assigned to a local non-authenticated client that connects to the server
- Required only if **allow non-authenticated clients** is selected
- 1 to 8 characters. The first character must be a letter; the others may be a letter, number or #, \$ or @

Remote identity

- Identity assigned to a remote non-authenticated client that connects to the server
- Required only if **allow non-authenticated clients** is selected
- 1 to 8 characters. The first character must be a letter; the others may be a letter, number or #, \$ or @

Register transaction factory

- Indicate that this server is a transaction factory, for use by a client that is starting a transaction
- Checked by default

Allow server region recycling

- Indicates that a server region is allowed to perform recycling
- For recycling the server region is stopped after completion or roll back of the last transaction indicated in the server recycling interval. The storage no longer in use is recovered. The server region is started again
- Checked by default

Server recycling interval

- Indicates the server recycling interval, that is the number of completed or rolled back transactions between recycling (garbage collections)
- The default value is 50000
- The valid range is from 1 to 2147483648(2³¹)

Logstream name

- The name of the server logstream. A valid logstream name consists of 1 or more qualifiers separated by periods, up to a maximum of 26 characters
- Each qualifier can contain up to 8 numbers, letters or #, \$ or @
- The first character of each qualifier must be a letter or #, \$ or @

- Each qualifier must be separated by periods, which must be counted as characters

Control region proc name

- The JCL PROC name is used to start control regions — which actually represent server instances in the Administration and Operations applications — for the corresponding server
- The default is the server name
- 1 to 8 characters. The first character must be a letter; the others may be letters, numbers or #, \$ or @. Embedded blanks are not allowed

Allow non-authenticated clients

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

If *non-authenticated clients* are allowed, clients that have not been authenticated may connect to this server.

- Default is to not allow non-authenticated clients to connect to this server
- If **allow non-authenticated clients** is selected without assigning a **local identity**, the default value is set to the value of environment variable DEFAULT_UNAUTH_CLIENT_ID on the sysplex object. If it is not set, the default value is CBGUEST. The message BBON0560I is displayed in the status bar.
- If **allow non-authenticated clients** is selected without assigning a **remote identity**, the default value is set to the value of environment variable DEFAULT_UNAUTH_CLIENT_ID on the sysplex object. If it is not set, the default value is CBGUEST. The message BBON0561I is displayed in the status bar.

Refer to “Define the security class of a server” on page 27 for more information about security classes.

Userid password allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

If *userid password* is allowed, clients may connect to this server, with the MVS user ID and password being used for security.

- Provides the least security of the options **Userid password allowed**, **Userid passticket allowed**, **DCE allowed**, **SSL Type 1 allowed**, **SSL Client Certificates allowed**
- Check the box to allow this level of security to be used for this server

Refer to “Define the security class of a server” on page 27 for more information about security classes

Userid passticket allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

The *passticket* is a one-time-only, system-generated password.

- Check the box to allow this level of security to be used for this server

Refer to “Define the security class of a server” on page 27 for more information about passticket and security classes.

DCE allowed


Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

The client and server must contact the *DCE* security server, acting as a third party, before communications can occur.

- Check the box to allow this level of security to be used for this server and then enter a value for **DCE quality of protection** and **DCE keytab file**

Refer to “Define the security class of a server” on page 27 for more information about DCE and security classes.

DCE quality of protection

- Indicates the type of protection through DCE
- Click  to display a list of valid values, with increasing security:
 - No protection
 - Message integrity: Messages have been signed, which ensures that they have not been modified
 - Message confidentiality: Messages are encrypted
- Default is no protection

DCE keytab file

- File path in the HFS for the server’s DCE keytab file. The DCE keytab file contains the server’s DCE password
- Up to 1,029 characters
- Required if **DCE allowed** is checked

SSL Type 1 (Basic Authentication) allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

SSL Type 1 or *SSL Basic Authentication* is a security mechanism that authenticates the server using its digital certificate and encrypts messages flowing across the client/server connection. The server authentication entails ensuring that the server’s certificate was granted by a certificate authority known to the client. The client’s identity is established by userid and password.

- Check the box to allow this level of security to be used for this server, and then fill out the values for the SSL related elements (**SSL RACF Keyring, SSL V2 timeout, SSL V3 timeout**)
- The server’s certificate must be defined as the default certificate in the keyring specified in **SSL RACF Keyring**
- Default is not allowed

Refer to “Define the security class of a server” on page 27 for more information about SSL and security classes.

SSL Client Certificates allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

SSL Client Certificates ensure that the client authenticates the server and the server authenticates the client. Both client and server authentication mechanisms are done by SSL, each side presents a certificate. This aspect of authentication guarantees that servers can trust their clients.

- Check the box to allow this level of security to be used for this server, and then fill out the values for the SSL related elements (**SSL RACF Keyring, SSL V2 timeout, SSL V3 timeout**)
- The client certificate must be specified in the **SSL RACF keyring**
- All certificate authorities for servers you need to access, must have certificates defined to RACF and be connected to the client's keyring

Refer to "Define the security class of a server" on page 27 for more information about SSL and security classes.

Kerberos allowed

Indicates that this security class is desired for authenticating Application Server clients and servers.

SSL Kerberos is a security mechanism that allows a client to authenticate a server using the server's digital certificate. The client's identity is verified by the server using Kerberos authentication methods. Message protection, which may include data privacy and integrity, is supplied by the Secure Sockets Layer (SSL).

- The server's certificate must be defined as the default certificate in the keyring specified in **SSL RACF Keyring**
- Check the box to allow this level of security to be used for this server, and then fill out the values for the SSL related elements (**SSL RACF Keyring, SSL V2 timeout, SSL V3 timeout**)
- Default is not allowed

Refer to "Define the security class of a server" on page 27 for more information about Kerberos and security classes.

Send Asserted Identities allowed

Indicates the security class that is desired to prevent unauthorized client access to Application Server resources.

Outbound requests originating from this server can send RACF userids over an SSL connection to a remote WebSphere for z/OS without additional authentication information to impersonate an originating client.

- Check the box to allow this level of security to be sent by this server, and then fill out the values for the SSL related elements (**SSL RACF Keyring, SSL V2 timeout, SSL V3 timeout**)
- Available only if SSL is configured on this system
- Default is not allowed

Refer to "Define the security class of a server" on page 27 for more information about asserted identities and security classes.

Accept Asserted Identities allowed

Indicates the security class that is desired to prevent unauthorized client access to Application Server resources.

Accept Asserted Identities allowed enables a target server to accept SSL Asserted Identities.

- Check the box to allow this level of security to be used for this server, and then fill out the values for the SSL related elements (**SSL RACF Keyring, SSL V2 timeout, SSL V3 timeout**)
- Available only if **SSL Client Certificates** is allowed
- Default is not allowed

Refer to “Define the security class of a server” on page 27 for more information about asserted identities and security classes.

SSL Use Confidentiality Only

- If **SSL Use Confidentiality Only** is set, both encryption and authentication will be used by the peer systems
- If the peer system does not have these suites available, the SSL connection will fail
- Only available if one of the SSL options **SSL Basic Authentication allowed**, **SSL Client Certificates allowed**, **Kerberos allowed**, **Send Asserted Identities allowed**, or **Accept Asserted Identities allowed** is set

SSL RACF Keyring

- The name of the RACF keyring that contains the appropriate keys and certificates for SSL
- Up to 237 characters
- All characters supported by RACF for profiles are allowed
- Default is CBKeyring
- Required if **SSL Basic Authentication allowed**, **SSL Client Certificates allowed**, **Kerberos allowed**, **Send Asserted Identities allowed**, or **Accept Asserted Identities allowed** is checked

SSL V2 timeout

- Number of seconds for SSL Version 2 session data to time out
- A number from 1 to 100
- Default is 100
- Required if **SSL Basic Authentication allowed**, **SSL Client Certificates allowed**, **Kerberos allowed**, **Send Asserted Identities allowed**, or **Accept Asserted Identities allowed** is checked

SSL V3 timeout

- Number of seconds for SSL Version 3 session data to time out
- A number from 1 to 86400 (1 day)
- Default is 600
- Required if **SSL Basic Authentication allowed**, **SSL Client Certificates allowed**, **Kerberos allowed**, **Send Asserted Identities allowed**, or **Accept Asserted Identities allowed** is checked

Security preference list

- Defines which security types can be used when clients connect to this server, and in what order of preference. An order of preference can only be assigned to a security type if that type has been checked above
- In WebSphere for z/OS V4.0.1 the security order for a simple client for a remote call is
 1. SSL Client Certificates / SSL Asserted Identity
 2. Kerberos (over SSL)
 3. SSL Basic Authentication
 4. Passticket
 5. DCE
 6. Passwordregardless of what is specified.

Write Server Activity SMF Records

- If checked, enables SMF recording of the server activity
- For each activity that is run inside a server instance of this server, a single record is created
- Use these records to perform basic charge back accounting as well as profiling of own applications to determine in detail what is happening inside the transaction server
- **Note:** Make sure that, on z/OS, SMF record type 120 is specified to enable Application Server information recording

Write Container Activity SMF Records

- If checked, enables SMF recording of the container activity
- There is a single record for each container that is part of an activity
- Data that describe the actual business functions invoked within the server's containers are monitored
- **Note:** Make sure that, on z/OS, SMF record type 120 is specified to enable Application Server information recording

Write Server Interval SMF Records

- If checked, enables SMF recording of the server activity in intervals that you specify in **SMF Interval Length**
- There is a single record for each server instance that has interval recording active during the specified interval
- If the server has multiple server instances, then a record for each server instance is written
- **Note:** Make sure that, on z/OS, SMF record type 120 is specified to enable Application Server information recording

Write Container Interval SMF Records

- If checked, enables SMF recording of the container activity in intervals that you specify in **SMF Interval Length**
- There is a single record created for each active container located in the server within the interval being recorded. If there is more than one server instance associated with the server, there will be a record for the container from each server instance
- **Note:** Make sure that, on z/OS, SMF record type 120 is specified to enable Application Server information recording

SMF Interval Length

- Length of the recording intervals for SMF monitoring which repeat continuously
- Only enabled if **Write Server Interval SMF Records** or **Write Container Interval SMF Records** is specified
- A length from 15 seconds to 86400 seconds (24 hours)
- Specify the seconds of the interval length
- Default is 3600 seconds (1 hour)
- Specify a 0 to indicate the usage of the interval length from the SMF product settings

Environment variable list

- Contains the definitions of the environment variables that are common to the server.
- Each server instance of the server inherits these values.

- The Environment variable list contains three columns:
 - Type** The **Type** field tells you at which level this variable is defined: SY (Sysplex), SV (Server), or SI (Server instance).
 - Name** The name of the environment variable.
 - Value** The value of the environment variable as it is defined at the level which is specified by its **Type**.
- The current maximum length of environment variable values is currently limited to 4096 characters due to an LE restriction on z/OS.
- To edit the environment variable list, ensure you are in the **Modify** action. Then do one of the following:
 - Double click the left mouse button on any cell in a row. Refer to “Server instance run-time environment variables” on page 50 for a description on how to manage the Edit Variable dialog.
 - Click the left mouse button on any cell in a row to highlight the row, and then press Enter.
- To view the environment variable list when you are not in the **Modify** action, do one of the following:
 - Double click the left mouse button on any cell in a row.
 - Click the left mouse button on any cell in a row to highlight the row, and then press Enter.
- For a detailed description of environment variables, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*.

Actions:

Add

- Creates a new server in the branch of the tree, and labels for required subordinate objects
- Available only against the label for Servers
- Causes WebSphere for z/OS to create associated containers and application families

Modify

Puts the server’s properties form in edit mode, which allows you to make and save changes

Delete Marks the server and everything below it in the branch of the tree as deleted

Import application...

- Imports an application. It displays a pop-up that lets you specify the location of the package files for the application that you want to import
- The server’s branch in the tree is collapsed
- See “Import an application” on page 31 for more information

Import server...

- Creates a new server with the server properties from the referenced server
- Available only against the label for Servers
- See “Migrate a test server to a production system” on page 47 for more information

Export server...

- Exports the server properties to the specified host path
- Only available a server of an active image (refer to “States of a conversation” on page 20). See “Migrate a test server to a production system” on page 47 for more information

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the server belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Server instance

In the Administration and Operations applications, a *server instance* consists of a *control region* and an arbitrary number of *server regions*.

It is a functional unit on which the WebSphere for z/OS applications run. All server instances within a server are identical in structure. Most of the properties are defined at the server level.

A particular server instance can exist on only a single server. Server instances appear under both servers and systems in the tree. At least one server and one system must be defined for the sysplex before a server instance can be added.

You can manage a server instance through the Operations application.

Location in the tree:

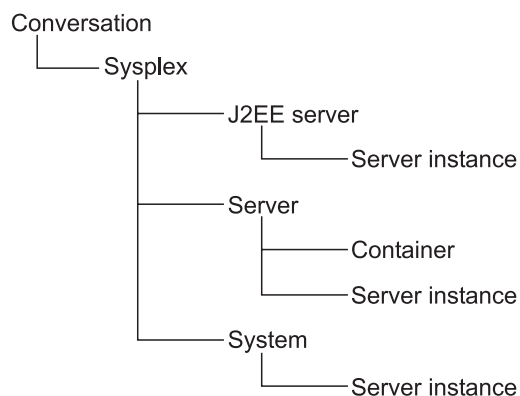


Figure 31. Location of a Server Instance in the Tree

Properties:

The properties of a server instance are described below.

Server instance name

- The name of the server instance
- Must be unique within the server
- 1 to 8 characters. The first character must be a letter. The others may be a letter, number or #, \$, @

Server instance description

- A description of the server instance
- Up to 4096 characters

System name

- The name of a system on which the server instance is running
- When you add the server instance to the tree under a System node, the system name is supplied and cannot be changed. When you add the server instance to the tree under a Server node, you specify a system by clicking ▼ to display a list of systems and then selecting a system from the list. Once the server instance is defined, the system name cannot be changed

Server name

- The name of the server to which this server instance belongs. The server is running within the sysplex
- When you add the server instance to the tree under a Server node, the server name is supplied and cannot be modified. When you add the server instance to the tree under a System node, you specify a server by clicking ▼ to display a list of servers and then selecting a server from the list. Once the server instance is defined, the server name cannot be changed

Logstream name

- The name of the server instance logstream. A valid logstream name consists of 1 or more qualifiers separated by periods, up to a maximum of 26 characters
- Each qualifier can contain up to 8 numbers, letters or #, \$ or @
- The first character of each qualifier must be a letter or #, \$ or @
- Each qualifier must be separated by periods, which must be counted as characters

IIOP Firewall port

If requests come through a firewall before coming to the server instance, and the requests are over Inter-ORB Protocol (IIOP), specify a unique fixed port on which the firewall listens for IIOP requests.

SSL Firewall port

If requests come through a firewall before coming to the server instance, and the requests are Secure Sockets Layer (SSL) requests over IIOP, specify a unique fixed SSL port on which the firewall listens for SSL requests over IIOP.

Environment variable list

- Contains the definitions of the environment variables that are unique for this server instance.
- This server instance inherits the values of the environment variables from its server and its sysplex.
- The Environment variable list contains three columns:

Type The **Type** field tells you at which level this variable is defined: SY (Sysplex), SV (Server), or SI (Server instance).

Name The name of the environment variable.

Value The value of the environment variable as it is defined at the level which is specified by its **Type**.

- When the conversation is activated or prepared for cold start, all values of the environment variables for the server instance are saved on the host in the file

*CBCONFIG/controlinfo/envfile/Sysplex_name/
ServerInstance_name/current.env*

where

- *CBCONFIG* is the path that you specified in the environment variable *CBCONFIG*. The default is */WebSphere390/CB390*.
- *Sysplex_name* is the name of the sysplex the server instance belongs to.
- *ServerInstance_name* is the name of the server instance.

Note that this file needs to be referenced by the job that starts this server instance in the *BBOENV DD* statement

- The current maximum length of environment variable values is currently limited to 4096 characters due to an LE restriction on z/OS.
- To edit the environment variable list, double click on an entry. Refer to “Server instance run-time environment variables” on page 50 for a description on how to manage the Edit Variable dialog.
- For a detailed description of environment variables, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*

Actions:

The actions that can be performed against a server instance are described below.

Add

- Creates a new server instance in the branches of the tree for the server and system
- Available only against the label for Server Instances
- At least one system and server must have been created

Modify

Puts the server instance’s properties form in edit mode, which allows you to make and save changes

Delete Marks the server instance as deleted

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the server instance belongs, and return errors if any
- Available only when the selected conversation has not yet been committed

Sysplex

A sysplex object represents a sysplex on which Application Server is installed.

Location in the tree:

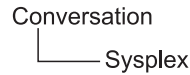


Figure 32. Location of a Sysplex in the Tree

Properties:

The properties of a sysplex are described below.

Sysplex name

- The name of the sysplex
- 1 to 8 characters. The first character must be a letter. The others may be a letter, number or #, \$, @
- Must be unique within the conversation

Sysplex description

Up to 4096 characters of descriptive text

Logstream name

- The name of the sysplex logstream. A valid logstream name consists of 1 or more qualifiers separated by periods, up to a maximum of 26 characters
- Each qualifier can contain up to 8 numbers, letters or #, \$ or @
- The first character of each qualifier must be a letter or #, \$ or @
- Each qualifier must be separated by periods, which must be counted as characters

Environment variable list

- Contains the definitions of the environment variables that are common to the sysplex.
- Each server instance of each server belonging to this sysplex inherits these values.
- The Environment variable list contains three columns:
 - Type** The **Type** field tells you at which level this variable is defined: SY (Sysplex), SV (Server), or SI (Server instance).
 - Name** The name of the environment variable.
 - Value** The value of the environment variable as it is defined at the level which is specified by its **Type**.
- The current maximum length of environment variable values is currently limited to 4096 characters due to an LE restriction on z/OS.
- To edit the environment variable list, ensure you are in the **Modify** action. Then do one of the following:
 - Double click the left mouse button on any cell in a row. Refer to “Server instance run-time environment variables” on page 50 for a description on how to manage the Edit Variable dialog.
 - Click the left mouse button on any cell in a row to highlight the row, and then press Enter.
- To view the environment variable list when you are not in the **Modify** action, do one of the following:
 - Double click the left mouse button on any cell in a row.

- Click the left mouse button on any cell in a row to highlight the row, and then press Enter.
- For a detailed description of environment variables, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*.

Configuration extensions

WebSphere for z/OS provides configuration extensions, such as connection management, in addition to the functional requirements for the current J2EE compliance level. Selected extensions apply for all J2EE servers defined to this sysplex. Use of these extensions does not cause any loss of function provided for J2EE compliance at the current level.

Note: WebSphere for z/OS is designed in compliance with the Sun Microsystems Java™ 2 Platform, Enterprise Edition (J2EE™) 1.2 specifications. This compliance has been verified by successful execution of Sun's J2EE 1.2 Compatibility Test Suite (CTS) against the V4.0.1 product.

The CTS verifies J2EE 1.2 compliance, and also checks that a product does not support application programming interfaces (APIs) that are outside the J2EE 1.2 specifications. To pass the CTS, a product must ship the J2EE 1.2 APIs— nothing more, nothing less. A consequence of this testing is that it also "flags" when a vendor ships J2EE capabilities that are beyond the J2EE 1.2 specifications.

One such capability is the ability to connect to existing IMS and CICS applications resident on the z/OS or S/390 platform. The J2EE Connection Architecture (JCA), which is part of the more advanced J2EE 1.3 specifications, provides this capability. WebSphere for z/OS does not currently support J2EE 1.3 or the JCA subset. However, WebSphere for z/OS V4.0.1 delivers an IBM implementation within its run-time, referred to as "connection management", that supports access to IMS and CICS connectors optimized for the z/OS and S/390 platform. Customers will need to activate this capability. Activation of this capability does not compromise the J2EE 1.2 function delivered by WebSphere for z/OS V4.0.1, nor does it compromise the portability of applications written to the J2EE 1.2 standard. However, because this capability provides function beyond the J2EE 1.2 specifications, if you elect to activate this capability, the presence of this run-time extension will be detected and flagged by J2EE 1.2 CTS. Customers who wish to run the CTS against WebSphere for z/OS V4.0.1 to verify IBM's compatibility, or who want to maintain a pure J2EE 1.2 environment, should not activate the run-time extension.

Actions:

The actions that can be performed against a sysplex are described below.

Modify

Puts the sysplex's properties form in edit mode, which allows you to make and save changes

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the sysplex belongs, and return errors if any

- Available only when the selected conversation has not yet been committed

System

A system is an z/OS system on which WebSphere for z/OS is running.

Location in the tree:

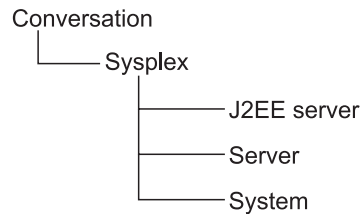


Figure 33. Location of a System in the Tree

Properties:

The properties of a system are described below.

System name

- The z/OS system name
- Must be consistent with the name specified in the IEASYSxx parmlib member of the system that WebSphere for z/OS will be running on
- Must be unique within the sysplex
- 1 to 8 characters consisting of letters, numbers and/or #, \$ or @

System description

Up to 4096 characters of descriptive text

Actions:

The actions that can be performed against a system are described below.

Add

- Creates a new system in the branch of the tree, and labels for required subordinate objects
- Available only against the label for Systems
- Causes WebSphere for z/OS to create associated server instances for WebSphere for z/OS servers
- The 'Servers' branch of the tree is collapsed

Modify

Puts the system's properties form in edit mode, which allows you to make and save changes

Delete Marks the system object and all of its subordinate objects as deleted

Validate

- Causes the Systems Management Server to check the integrity of the conversation to which the system belongs, and return errors if any
- Available only when the selected conversation has not yet been committed


Chapter 5. Instructions for z/OS tasks

This section describes the instructions that are provided as part of the WebSphere for z/OS configuration. It includes these topics:

- Instructions overview
- Instructions completion summary
- Instructions task detail
- Save the instructions

Instructions overview

Completing the Application Server configuration requires that some tasks outside of the scope of WebSphere for z/OS be performed. For example, security profiles must be defined through your security product. The Administration application displays instructions on how to complete these remaining tasks.

The instructions are provided after you have committed the model. When instructions are available, a  appears next to the name of the conversation in the tree.

You must complete the tasks described in the instructions and mark them complete before you can activate the image.



When working with the instructions, you can:

- Copy or move selected parts of the instructions to other locations, such as notes or files (Copy)
- Search the instructions for specific information (Find..)
- Save the instructions to a file (Save..)
- Refresh the instructions (Refresh)
- Switch back to displaying the properties for the selected object (Restore properties)

The actions to perform these tasks can be selected from the pop-up that is displayed when you press the right mouse button with the mouse pointer on the instructions.

Planning information related to the z/OS tasks can be found in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*.

Instructions completion summary


The instructions include a completion summary, which lists the types of tasks and indicates whether or not they were completed when you displayed the instructions. Completed tasks are marked with . Incomplete tasks are marked with .

The types of tasks are:

- Security tasks
- Workload management (WLM) tasks

- Automatic restart manager (ARM) tasks
- Automation tasks
- Resource management tasks
- Logstream tasks

The instructions always contain a section for each type of task, even if there are no tasks of that type required.

The completion summary is automatically refreshed to reflect tasks that have been completed while the instructions are displayed. Mark completed tasks in the pull-down menu of the **Complete** action of the **Build** menu bar choice. Completed tasks are marked with .

Planning information related to the z/OS tasks can be found in *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*.

Instructions task detail

The tasks detail contains a list of subtasks for each type of task. The sections group each set of tasks by type. The sections are:

- Security tasks
- Workload Manager (WLM) tasks
- Automatic Restart Manager (ARM) tasks
- Automation tasks
- Resource Management tasks
- Logstream tasks

Each section is always present, even if there are no tasks required for it.

The heading for a section ends with **(Completed)** if those tasks were complete when you displayed the instructions.

The task detail is automatically refreshed to reflect tasks that have been completed while the instructions are displayed. You can also see the status of instructions in the pull-down menu of the **Complete** action of the **Build** menu bar choice.

Completed tasks are marked with .

Save the instructions

1. Open the Instructions by selecting the **Instructions...** action of the **Build** menu bar choice from the main Administration window.
2. Click the right mouse button on the instructions to display the menu of actions.
3. Select the **Save...** action.
4. Complete the **Instructions File Path** dialog, either by typing the path and file to contain the instructions, or by clicking **Choose** and selecting the path. The file will contain HTML tags, so it should have an extension of htm or html.
5. Click **OK** to save the instructions to the file.

Part 3. Operations application

This part describes the Operations application for WebSphere for z/OS. It describes:

- The user interface of the Operations application,
- the operations tasks,
- the objects, their properties and actions that are available.

Chapter 6. Operation user interface

This chapter describes the graphical user interface and the objects that are displayed in the Operations application. These are:

- The main window with the operations objects and their properties
- The icons for operations
- The properties form in the right-hand frame of the operations window
- The work request list (not yet supported)

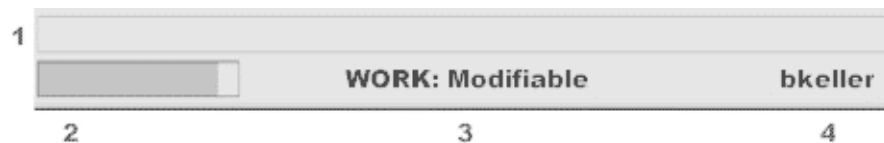
General user interface topics are described in Appendix A, “User Interface”, on page 139.

Main window

The main window for operations consists of two frames (see Figure 2 on page 3). The left frame shows WebSphere for z/OS objects as icons. The right frame shows details about a selected object using the properties form. It can also be used to show associated work requests, by selecting the appropriate tab at the bottom of the right frame.



At the bottom of the window is a variety of information, shown in the example below. A *message area* (1) is followed by an *informational line* that includes a *progress bar* (2), which indicates activity of a currently running process. The name and status of the selected conversation (3) is followed by the user ID (4) of the user.



Filter the operations window

Use the filter drop-downs to filter the window. There is one drop-down for servers and one for systems:




1. Click ▼ in the Server list to display a menu of Servers.
2. Select **All** to include all servers and server instances, or select a single server, to show only that server and its associated server instances.
3. Click ▼ in the System list to display a menu of Systems.
4. Select **All** to include servers and server instances running on all systems, or select a single system, to show only the servers and associated server instances running on that system.






When you select a single server and a single system, the filters work in combination. Only the server instances for that server running on that system are shown. The server itself is not shown.


Icons for operations

The Operations application shows an icon for each WebSphere for z/OS server and server instance in your configuration. The servers appear first, followed by the server instances. You can scroll the window up and down, if necessary, to see all the icons.

Each server is represented by the  icon. Each server instance is represented by an icon that shows the state of the server instance

 Active  In transition  Inactive  Unknown

A green dot in the upper right edge of a server or server instance indicates the state "ready for warm start".  for servers, and , , ,  for server instances in different states.

Note: The server instance icons do not automatically reflect a change in the status of the server instance. The status that is displayed is the status that was current when the window was accessed. To display the most current status, select the **Refresh** action of the **View** menu bar choice, or click  on the tool bar.

To display the status in text form, select a server instance. The name and status of the selected server instance are shown at the bottom of the window.

The following example shows a status of Active for server instance SI1.



Properties form

The properties form displays the values for the selected Application Server object. It appears in the right frame of the window.

Details for the fields on the property form are included with the help for each object or in Chapter 8, "Operation objects", on page 113 in this book.

You cannot edit the properties form with the Operations application, but with the Administration application.

To **display the properties form**,

1. Select the object by clicking it once with the left mouse button.
2. Click the Properties tab at the bottom of the right frame.
3. The Properties Form is displayed in the right frame, in browse mode.

Work request list (not yet supported)

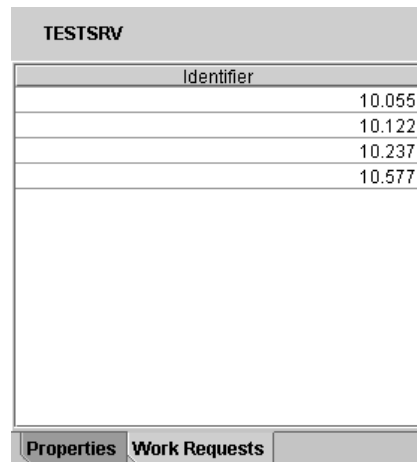
The work request list shows the work requests for the selected server or server instance. They are ordered by ID.

The work request list shows the name of the server or server instance that has been selected, followed by the list of work requests, in order by identifier.

To **display the work request list**,

1. Click the icon for the server or server instance.
2. Click the tab for Work Requests, at the bottom of the right frame, to display the list of work requests for that server or server instance.

The following example shows work requests for server TESTSRV.



TESTSRV	
Identifier	
	10.055
	10.122
	10.237
	10.577

Properties Work Requests

Chapter 7. Operation tasks


This topic describes how to use the Operations application to operate your WebSphere for z/OS.

It includes the tasks:


- Start a server or server instance
- Stop a server or server instance
- Perform a warm start for a server or server instance

You can also use commands to perform some operations tasks. Refer to Appendix B, “Commands for operations”, on page 155 for more information.

Start a server or server instance

1. Click the icon for the server or server instance.
2. Select the **Start** action of the **Selected** menu bar choice, or click  on the tool bar.

Stop a server or server instance

1. Click the icon for the server or server instance.
2. Select the **Stop** action of the **Selected** menu bar choice, or click  on the tool bar. You can also select the **Cancel with restart** or **Cancel** actions, to stop the server or server instance immediately.

Note: WebSphere for z/OS owned servers or server instances (their names have the prefix CB) cannot be stopped or cancelled.

Perform a warm start for a server or server instance

WebSphere for z/OS can be migrated from one release level to a new level without a disruption of service to the end users of applications that are deployed on the Application Server. This means that not all pieces of the WebSphere for z/OS network are changed at the same time - it can take weeks to get all systems changed. (Refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization* for more information.) Therefore, WebSphere for z/OS servers are capable of interoperating at different release levels within a sysplex at the same time. Associated to each server and server instance are two level attributes:

- The *capability level* is the release level the server instance is able to run at. It is hard coded in the WebSphere for z/OS code and set automatically during server instance start-up.
- The *function level* is the release level that the server instance is actually operating. This level is automatically calculated during server instance start-up and depends on the function level of the associated server.

During normal operation, all server instances of a server run the same function level, i.e. the function level of a server is equal to the function level of all of its

server instances. The capability level of a server is the minimum of the capability levels of its server instances. Each time a server instance starts up, this minimum is calculated again.

If the capability level of the server increases and is greater than its function level, then the server is "ready for warm start", i.e. its server instances could operate at the new capability level as function level. However, this upgrade is not done automatically, but requires an explicit operator action, the *warm start*. In the Operations application, warm start readiness is indicated by a green dot in the upper right edge of the servers and server instances.

For the server DAEMON (DMN),

a warm start can be issued only as console commands:

```
stop bbodmn  
start bbodmn,svname='DAEMON01',parms='-ORBCBI WARM'
```

The stop command stops all servers on the system, also the System Management Server. The start command starts the DAEMON and the WebSphere for z/OS servers, e.g. the System Management Server and the Naming Server. The application servers have to be started separately.

For all other servers,

warm start can be issued from the Operations application. WebSphere for z/OS stops each server instance of a server and restarts it with the increased function level. This process is repeated for each server instance, one at a time. To perform a warm start on a server:

1. Click the icon for the server
2. Select the **Warm start** action of the **Selected** menu bar choice
3. Confirm the **Confirm warm start** dialog box
4. WebSphere for z/OS stops and restarts all server instances of this server one after the other.

Because WebSphere for z/OS waits for running transactions to be completed, this action may take some time. (A timeout value is defined after which WebSphere for z/OS returns with an error message.)

5. The function level of the server and of each server instance of the server is increased

If an *error* occurs during the **Warm start** action, some server instances may be warm started and some not. After refreshing the Operations window, check the warm start readiness indicator for the server and server instances you warm started, and warm start the server instances that are still marked as "ready for warm start" separately.

Chapter 8. Operation objects

The objects that you control using the WebSphere for z/OS Operations application are

- servers and
- server instances.

These are the logical entities on which WebSphere for z/OS applications run. Most of the properties are defined at the server level. They include such things as whether it is a production server, and security properties that determine which clients can access the server.

J2EE server

Please note that the Operations application does not distinguish between J2EE servers and MOFW servers.

A J2EE server is a server that hosts J2EE applications. You can manage a J2EE server through the Operations application.

Properties:

The properties for a J2EE server are described below.

Server name

- The name of the server
- Is unique within the sysplex

Server description

Descriptive text

Control region identity

- The identity associated with the control region or system address space when the actual server image is generated
- Should be a trusted identity

Server region identity

- The identity associated with the server region address space when the actual server region image is created

Server region stack size

- The default size of the stack frame in the control region, in bytes

Production J2EE Server

- Indicates that this is a production server. The Application Server run time limits certain real time debugging capabilities in production servers
- A production server supports more clients and uses more resources than a non-production server

Debugger allowed

- Specifies whether a debugger is allowed on the server

Object Level Trace hostname

- Object Level Trace (OLT) enables you to monitor the flow of a distributed application, and to seamlessly debug client and server code

from a single workstation. OLT records method calls from the client application, or servlet, to distributed business objects, servlets, JSPs, or EJBs residing on WebSphere for z/OS application servers

- The fully-qualified name or TCP/IP address of the machine running your OLT server

Object Level Trace port

- The port where the OLT server listens for connecting OLT clients

Isolation policy

- Specifies how the server region should isolate user transactions from each other, that is, whether each transaction is assigned its own server region
- Assigning a server region to each transaction provides greater security than allowing multiple transactions per server region

Replication policy

- Specifies how many server regions should be started
- Used when the server regions are replicated inside the server region

Local identity

- Identity assigned to a local non-authenticated client that connects to the server

Remote identity

- Identity assigned to a remote non-authenticated client that connects to the server

Register transaction factory

- Indicate that this server is a transaction factory, for use by a client that is starting a transaction

Allow server region recycling

- Indicates that a server region is allowed to perform recycling
- For recycling the server region is stopped after completion or roll back of the last transaction indicated in the server recycling interval. The storage no longer in use is recovered. The server region is started again

Server recycling interval

- Indicates the server recycling interval, that is the number of completed or rolled back transactions between recycling (garbage collections)

Logstream name

The name of the server logstream

Control region proc name

The JCL PROC name is used to start control regions — which actually represent server instances in the Administration and Operations applications — for the corresponding server

Enable Setting OS Thread Identity to RunAs Identity

- If this field is checked the J2EE server is enabled to access methods within J2EE applications under the authority of a specific role (requester rolename, server rolename or specified rolename)
- The rules for the methods and the rolenames which are allowed to access the methods are specified in the extended deployment descriptor of the J2EE application

- If checked, the Native OS Thread Security Identity can be modified to the user ID that corresponds to that method's RunAs identity while running an EJB method. This requires all system resources (files, databases, sockets) accessed during execution to be authorized for access by that identity
- If unchecked, the security identity of the J2EE server is used, and it cannot be modified during the running of an application

Allow non-authenticated clients

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

If *non-authenticated clients* are allowed, clients that have not been authenticated may connect to this server.

- If **allow non-authenticated clients** is selected without assigning a **local identity**, the default value is set to the value of environment variable `DEFAULT_UNAUTH_CLIENT_ID` on the sysplex object. If it is not set, the default value is `CBGUEST`. The message `BBON0560I` is displayed in the status bar.
- If **allow non-authenticated clients** is selected without assigning a **remote identity**, the default value is set to the value of environment variable `DEFAULT_UNAUTH_CLIENT_ID` on the sysplex object. If it is not set, the default value is `CBGUEST`. The message `BBON0561I` is displayed in the status bar.

Userid password allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

If *userid password* is allowed, clients may connect to this server, with the MVS user ID and password being used for security.

- Provides the least security of the options **Userid password allowed**, **Userid passticket allowed**, **DCE allowed**, **SSL Type 1 allowed**, **SSL Client Certificates allowed**

Userid passticket allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

The *passticket* is a one-time-only, system-generated password.

DCE allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

The client and server must contact the *DCE* security server, acting as a third party, before communications can occur.

DCE quality of protection

- Indicates the type of protection through DCE

DCE keytab file

- File path in the HFS for the server's DCE keytab file. The DCE keytab file contains the server's DCE password

SSL Type 1 (Basic Authentication) allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

SSL Type 1 or *SSL Basic Authentication* is a security mechanism that authenticates the server using its digital certificate and encrypts messages flowing across the client/server connection. The server authentication entails ensuring that the server's certificate was granted by a certificate authority known to the client. The client's identity is established by userid and password.

- The server's certificate is defined as the default certificate in the keyring specified in **SSL RACF Keyring**

SSL Client Certificates allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

SSL Client Certificates ensure that the client authenticates the server and the server authenticates the client. Both client and server authentication mechanisms are done by SSL, each side presents a certificate. This aspect of authentication guarantees that servers can trust their clients.

- The client certificate is specified in the **SSL RACF keyring**
- All certificate authorities for servers you need to access have certificates defined to RACF and are connected to the client's keyring

Kerberos allowed

- Indicates that this security class is desired for authenticating Application Server clients and servers
- SSL Kerberos is a security mechanism that allows a client to authenticate a server using the server's digital certificate. The client's identity is verified by the server using Kerberos authentication methods. Message protection, which may include data privacy and integrity, is supplied by the Secure Sockets Layer (SSL)
- The server's certificate is defined as the default certificate in the keyring specified in **SSL RACF Keyring**

Send Asserted Identities allowed

- Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources
- Indicates that outbound requests originating from this server can send RACF userids over an SSL connection to a remote Application Server without additional authentication information to impersonate an originating client.

Accept Asserted Identities allowed

- Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources
- Enables a target server to accept SSL Asserted Identities

SSL Use Confidentiality Only

- If **SSL Use Confidentiality Only** is set, both encryption and authentication will be used by the peer systems
- If the peer system does not have these suites available, the SSL connection will fail

SSL RACF Keyring

- The name of the RACF keyring that contains the appropriate keys and certificates for SSL

SSL V2 timeout

- Number of seconds for SSL Version 2 session data to time out

SSL V3 timeout

- Number of seconds for SSL Version 3 session data to time out

Security preference list

- Defines which security types can be used when clients connect to this server, and in what order of preference. An order of preference can only be assigned to a security type if that type has been checked above
- In WebSphere for z/OS V4.0.1 the security order for a simple client for a remote call is
 1. SSL Client Certificates / SSL Asserted Identity
 2. Kerberos (over SSL)
 3. SSL Basic Authentication
 4. Passticket
 5. DCE
 6. Password
 regardless of what is specified.

Write Server Activity SMF Records

- If checked, SMF recording of the server activity is enabled
- For each activity that is run inside a server instance of this server, a single record is created

Write Container Activity SMF Records

- If checked, SMF recording of the container activity is enabled
- There is a single record for each container that is part of an activity
- Data that describe the actual business functions invoked within the server's containers are monitored

Write Server Interval SMF Records

- If checked, SMF recording of the server activity is enabled in intervals that are specified in **SMF Interval Length**
- There is a single record for each server instance that has interval recording active during the specified interval
- If the server has multiple server instances, then a record for each server instance is written

Write Container Interval SMF Records

- If checked, SMF recording of the container activity is enabled in intervals that are specified in **SMF Interval Length**
- There is a single record created for each active container located in the server within the interval being recorded. If there is more than one server instance associated with the server, there will be a record for the container from each server instance

SMF Interval Length

- Length of the recording intervals for SMF monitoring which repeat continuously

- A 0 indicates the usage of the interval length from the SMF product settings

Environment variable list

- Contains the definitions of the environment variables that are common to the server.
- Each server instance of the server inherits these values.
- The Environment variable list contains three columns:
 - Type** The **Type** field tells you at which level this variable is defined: SY (Sysplex), SV (Server), or SI (Server instance).
 - Name** The name of the environment variable.
 - Value** The value of the environment variable as it is defined at the level which is specified by its **Type**.
- The current maximum length of environment variable values is currently limited to 4096 characters due to an LE restriction on z/OS.
- To edit the environment variable list, double click on an entry. Refer to "Server instance run-time environment variables" on page 50 for a description on how to manage the Edit Variable dialog.
- For a detailed description of environment variables, refer to *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization*

Actions:

Start Starts the server instances associated with the server

Stop Stops the server instances associated with the server after currently running transactions complete

Cancel with restart

Stops the server instances associated with the server immediately. Currently running transactions do not complete. If the server instances are ARM-protected, the ARM component of z/OS will restart the server instances

Cancel

Stops the server instances associated with the server immediately. Currently running transactions do not complete. If the server instance are ARM-protected, the ARM component of z/OS will not restart the server instances

Warm start

Only available for servers in the state "ready for warm start".

- Increases the function level of a server
- Each server instance of this server is stopped and then restarted with the new function level

Server (MOFW)

A server is a logical grouping of server instances. All server instances within a server are identical in structure. You can manage a server through the Operations application.

Properties:

The properties for a server can only be changed by the administrations application (except for the function level). They are described below.

Server name

The name of the server

Server description

Descriptive text

Control region identity

The identity associated with the control region or system address space when the actual server image is generated

Server region identity

The identity associated with the server region address space when the actual server region image is created

Server region stack size

The default size of the stack frame in the control region, in bytes

Production server

- Indicates that this is a production server. The Application Server run time limits certain real time debugging capabilities in production servers
- A production server supports more clients and uses more resources than a non-production server

Debugger allowed

Specifies whether a debugger is allowed on the server

Object Level Trace hostname

- Object Level Trace (OLT) enables you to monitor the flow of a distributed application, and to seamlessly debug client and server code from a single workstation. OLT records method calls from the client application, or servlet, to distributed business objects, servlets, JSPs, or EJBs residing on WebSphere application servers
- The fully-qualified name or TCP/IP address of the machine running your OLT server

Object Level Trace port

- The port where the OLT server listens for connecting OLT clients

Isolation policy

Specifies how the server region should isolate user transactions from each other, that is, whether each transaction is assigned its own server region

Replication policy

- Specifies how many server regions should be started
- Used when the server regions are replicated inside the server region

Local identity

Identity assigned to a local non-authenticated client that connects to the server

Remote identity

Identity assigned to a remote non-authenticated client that connects to the server

Register transaction factory

Indicates that this server is a transaction factory, for use by a client that is starting a transaction

Allow server region recycling

- Indicates that a server region is allowed to perform recycling
- For recycling the server region is stopped after completion or roll back of the last transaction indicated in the server recycling interval. The storage no longer in use is recovered. The server region is started again

Server recycling interval

- Indicates the server recycling interval, that is the number of completed or rolled back transactions between recycling (garbage collections)

Logstream name

The name of the server logstream

Control region proc name

The JCL PROC name is used to start control regions for the corresponding server

Allow non-authenticated clients

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

If *non-authenticated clients* are allowed, clients that have not been authenticated may connect to this server.

- Default is to not allow non-authenticated clients to connect to this server
- If **allow non-authenticated clients** is selected without assigning a **local identity**, the default value is set to the value of environment variable `DEFAULT_UNAUTH_CLIENT_ID` on the sysplex object. If it is not set, the default value is `CBGUEST`. The message `BBON0560I` is displayed in the status bar.
- If **allow non-authenticated clients** is selected without assigning a **remote identity**, the default value is set to the value of environment variable `DEFAULT_UNAUTH_CLIENT_ID` on the sysplex object. If it is not set, the default value is `CBGUEST`. The message `BBON0561I` is displayed in the status bar.

Userid password allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

If *userid password* is allowed, clients may connect to this server, with the MVS user ID and password being used for security.

- Provides the least security of the options **Userid password allowed**, **Userid passticket allowed**, **DCE allowed**, **SSL Basic Authentication allowed**, **SSL Client Certificates allowed**

Userid passticket allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

The *passticket* is a one-time-only, system-generated password.

DCE allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

The client and server must contact the *DCE* security server, acting as a third party, before communications can occur.

DCE quality of protection

Indicates the type of protection through DCE

DCE keytab file

File path in the HFS for the server's DCE keytab file. The DCE keytab file contains the server's DCE password

SSL Basic Authentication (Type 1) allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

SSL Basic Authentication is a security mechanism that authenticates the server using its digital certificate and encrypts messages flowing across the client/server connection. The server authentication entails ensuring that the server's certificate was granted by a certificate authority known to the client. The client's identity is established by userid and password.

SSL Client Certificates allowed

Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources.

SSL Client Certificates ensure that the client authenticates the server and the server authenticates the client. Both client and server authentication mechanisms are done by SSL, each side presents a certificate. This aspect of authentication guarantees that servers can trust their clients.

Kerberos allowed

- Indicates that this security class is desired for authenticating Application Server clients and servers
- SSL Kerberos is a security mechanism that allows a client to authenticate a server using the server's digital certificate. The client's identity is verified by the server using Kerberos authentication methods. Message protection, which may include data privacy and integrity, is supplied by the Secure Sockets Layer (SSL)
- The server's certificate is defined as the default certificate in the keyring specified in **SSL RACF Keyring**

Send Asserted Identities allowed

- Indicates the security class that is desired to prevent unauthorized client access to WebSphere for z/OS resources
- Indicates that outbound requests originating from this server can send RACF userids over an SSL connection to a remote Application Server without additional authentication information to impersonate an originating client.

Accept Asserted Identities allowed

- Indicates the security class that is desired to prevent unauthorized client access to Application Server resources
- Enables a target server to accept SSL Asserted Identities

SSL Use Confidentiality Only

- If **SSL Use Confidentiality Only** is set, both encryption and authentication will be used by the peer systems
- If the peer system does not have these suites available, the SSL connection will fail

SSL RACF Keyring

The name of the RACF keyring that contains the appropriate keys and certificates for SSL

SSL V2 timeout

Number of seconds for SSL Version 2 session data to time out

SSL V3 timeout

Number of seconds for SSL Version 3 session data to time out

Capability level

Indicates the release level that WebSphere for z/OS is capable of running. WebSphere for z/OS is able to run different release levels below its capability level if necessary (each server instance of a server has to run the same release level). For more information, refer to “Perform a warm start for a server or server instance” on page 111.

Function level

Indicates the release level that WebSphere for z/OS currently runs on this server. For more information, refer to “Perform a warm start for a server or server instance” on page 111.

Security preference list

- Defines which security types can be used when clients connect to this server, and in what order of preference
- In WebSphere for z/OS V4.0.1 the security order for a simple client for a remote call is
 1. SSL Client Certificates / SSL Asserted Identity
 2. Kerberos (over SSL)
 3. SSL Basic Authentication
 4. Passticket
 5. DCE
 6. Passwordregardless of what is specified.

Write Server Activity SMF Records

- If checked, SMF recording of the server activity is enabled
- For each activity that is run inside a server instance of this server, a single record is created

Write Container Activity SMF Records

- If checked, SMF recording of the container activity is enabled
- There is a single record for each container that is part of an activity
- Data that describe the actual business functions invoked within the server’s containers are monitored

Write Server Interval SMF Records

- If checked, SMF recording of the server activity is enabled in intervals that are specified in **SMF Interval Length**
- There is a single record for each server instance that has interval recording active during the specified interval
- If the server has multiple server instances, then a record for each server instance is written

Write Container Interval SMF Records

- If checked, SMF recording of the container activity is enabled in intervals that are specified in **SMF Interval Length**
- There is a single record created for each active container located in the server within the interval being recorded. If there is more than one server instance associated with the server, there will be a record for the container from each server instance

SMF Interval Length

- Length of the recording intervals for SMF monitoring which repeat continuously
- A 0 indicates the usage of the interval length from the SMF product settings

Environment variable list

- Displays the environment variables that are defined for this server
- The Environment variable list contains three columns:

Type The **Type** field tells you at which level this variable is defined: SY (Sysplex), SV (Server), or SI (Server instance).

Name The name of the environment variable.

Value The value of the environment variable as it is defined at the level which is specified by its **Type**.

Actions:

Start Starts the server instances associated with the server

Stop Stops the server instances associated with the server after currently running transactions complete

Cancel with restart

Stops the server instances associated with the server immediately. Currently running transactions do not complete. If the server instances are ARM-protected, the ARM component of z/OS will restart the server instances

Cancel

Stops the server instances associated with the server immediately. Currently running transactions do not complete. If the server instance are ARM-protected, the ARM component of z/OS will not restart the server instances

Warm start

Only available for servers in the state "ready for warm start".

- Increases the function level of a server
- Each server instance of this server is stopped and then restarted with the new function level

Server instance

In the Administration and Operations applications, a *control region* of a server is called a *server instance*.

It is a functional unit on which the WebSphere for z/OS applications run. It is an instance of a replicated server defined as a server. All server instances within a server are identical in structure. Most of the properties are defined at the server level.

You can manage a server instance through the Operations application.

Properties:

The properties of a server instance can only be changed by the Administration application (except for the function level). They are described below.

Server instance name

- The name of the server instance
- Is unique within the server

Server instance description

A description of the server instance

System name

The name of a system on which the server instance is running

Server name

The name of the server to which this server instance belongs. The server is running within the sysplex.

Logstream name

The name of the server instance logstream.

IIOP Firewall port

If requests come through a firewall before coming to the server instance, and the requests are over Inter-ORB Protocol (IIOP), specify a unique fixed port on which the firewall listens for IIOP requests.

SSL Firewall port

If requests come through a firewall before coming to the server instance, and the requests are Secure Sockets Layer (SSL) requests over IIOP, specify a unique fixed SSL port on which the firewall listens for SSL requests over IIOP.

Capability level

Indicates the release level that this server instance is capable of running. WebSphere for z/OS is able to run different release levels below its capability level if necessary (each server instance of a server has to run the same release level).

For more information, refer to "Perform a warm start for a server or server instance" on page 111.

Function level

Indicates the release level that WebSphere for z/OS currently runs on this server instance

Environment variable list

- Displays the environment variables that are defined for this server instance
- The Environment variable list contains three columns:

Type The **Type** field tells you at which level this variable is defined: SY (Sysplex), SV (Server), or SI (Server instance).

Name The name of the environment variable.

Value The value of the environment variable as it is defined at the level which is specified by its **Type**.

Actions:

Start Starts the server instance

Stop Stops the server instance after currently running transactions complete

Cancel with restart

Stops the server instance immediately. Currently running transactions do not complete. If the server instance is ARM-protected, the ARM component of z/OS will restart the server instance

Cancel

Stops the server instance immediately. Currently running transactions do not complete. If the server instance is ARM-protected, the ARM component of z/OS will not restart the server instance.

Warm start

Only available for server instances in the state "ready for warm start".

- Stops and then restarts the server instance with an increased function level

Part 4. Messages and diagnosis

This section gives a summary of how to diagnose problems with the Administration and Operations applications. The following tools and helps are provided:

Message log

The message log provides a chronological list of messages issued by the Administration or Operations application. The log may provide additional information about a message that was displayed in a pop-up or the immediate message area. See Chapter 9, "Message log", on page 129 for more information.

Messages

Messages issued by the Administration or Operations dialog are displayed either in pop-ups or in the immediate message area at the bottom of the window. Help is available for every message. All messages are collected in the message log. See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Installation and Customization* for information about specific messages.

Trace and debug facilities

The Administration and Operations applications include trace and debug facilities that are intended to be used under the supervision of IBM support personnel. See Chapter 11, "Trace and debug facilities", on page 135 for more information.

Environment information

The **Environment** action of the **Help** menu bar choice displays a summary of the run-time and execution environment.

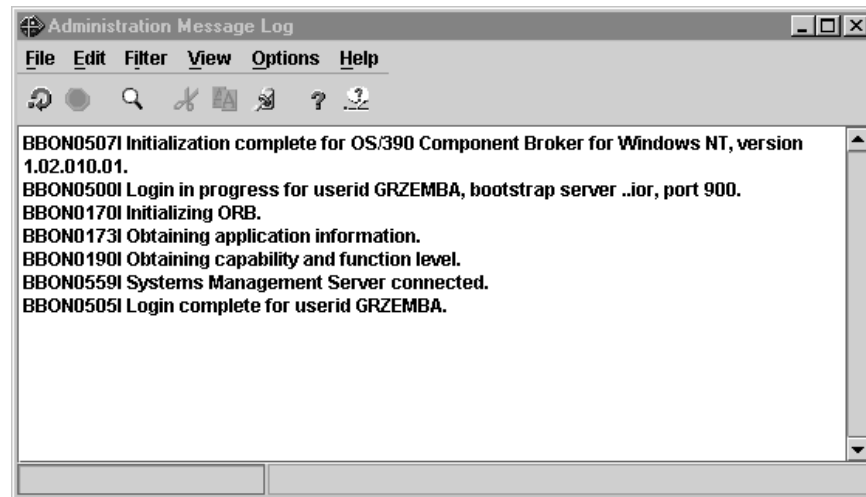
Release notes

For last-minute information, see the Release Notes (`Re1notes.htm`) shipped with the Administration and Operations applications.


Chapter 9. Message log

There is one message log for the Administration application and another message log for the Operations application.

This is an example of a message log window:



Display the message log

The message log is displayed by selecting the **Message Log...** action of the **File** menu bar choice, or clicking  on the tool bar.

Note: If you have an active screen reader, press **Alt+z** to have the status area messages read. Press **Alt+z** again to exit the message log status area.

Contents

The message log contains a list of messages issued by the application. Not every message issued by the Administration application appears in the message log. For example, messages issued in pop-ups that require a response from the user do not appear in the message log.

Order The messages in the message log are in chronological order. New messages are added to the bottom of the log. You can scroll the message log up and down to see older or newer messages.

Size The size of the message log is fixed. Once the maximum number of messages is reached, old messages are discarded as new messages are added. You control the size, and therefore the number of messages, with the **Log limit** action of the **Options** menu bar choice.

Format

The following example shows a message in the message log, with the optional (julian) date and time included:


```
1998.222 14:05:49 BBON0515I System SYS8 was added.
```

Date	Time	Message ID and text
------	------	---------------------

Persistency

When you close an application, the message log for that application is discarded. If you want to retain the message log, you can save it to a file or print it.

Refreshment

You can refresh the message log at any time by clicking  on the tool bar. New messages will be added to the bottom of the log. The **Dynamic mode** action of the **View** menu bar choice causes the message log to be updated when new messages are issued.

Options

The **Color** action of the **Options** menu bar choice of the main application window (not the message log window) lets you set the colors used for each message type.

Filtering and Searching

Actions available from the menu bar of the message log window let you filter and search the message log.

The following chapters describe how to

- filter the message log and
- print the message log.

For an overall description of the actions available for the message log, refer to “Menu bar actions for the message log” on page 145.

Filter the message log

Actions available from the Filter menu bar choice of the message log window let you filter the message log based on options you select.

1. Display the **Filter** actions from the menu bar.
2. Select the desired filter to put it into effect.
3. To combine filters, for example, to show only messages that are both Error and Server messages, repeat steps 1 and 2.

Filters are saved and will be in use the next time you access the message log.

Note that when you print the message log, filters are not applied. All messages are shown.

Turn off filtering with the **Reset** action of the **Filter** menu bar choice.

To see what filters are in effect, click **Filter** on the menu bar to display the filter actions. Filters that act as toggles (they are either on or off) have a next to them when they are in effect. Filters that use input from the user (for example, **String...**) are highlighted if they are in effect. To see the values for those filters, click on the associated action. For example, click on the **String...** action to display the text strings that are being used as filters.

Print the message log

To print the entire message log:

1. If it is not already open, open the message log by selecting the **Message log** action of the **File** menu bar choice.
2. Select the **Print** action of the **File** menu bar choice on the message log window.
3. Complete the resulting print dialog.

To print using the print function of another application:

1. If it is not already open, open the message log by selecting the **Message log** action of the **File** menu bar choice.
2. Copy or cut the portion of the message log that you want to print.
3. Paste the contents of the clipboard into a file that you can print.

To print a screen of the message log:

1. If it is not already open, open the message log by selecting the **Message log** action of the **File** menu bar choice.
2. Size the screen as desired.
3. Select the **Print screen** action of the **File** menu bar choice on the Message log window.
4. Complete the resulting print dialog.

Notes:

- When you print the message log, filters are not applied. All messages are shown.
- When printing a screen of the message log, be aware that the printer must have a page size adequate for the screen. For example, when using 8x10 inch paper you cannot print a width of greater than 8 inches.
- When you print the message log, the results have a slightly different format than what is displayed in the Message Log window. Each line begins with the Julian date and the time, even if date and time are not being displayed in the window. Following the time is a column reserved for information about the severity of the message (Warning, Error or Debug).

The following example shows the format:

```
1998.237 10:02:08 Error BB0N0402E Server SRVA#1 is already defined.
```

```
|           |           |           |
```

```
Date      Time    Severity  Message ID and text
```

In addition, a heading is added to each printed page indicating the date and time that the message log was printed.

Chapter 10. Messages


The messages for the Administration or Operations application may be displayed


- in the message area at the bottom of the window,
- in the message log,
- or in a pop-up window.

The message IDs are in the format *BBONNumberType*, where

- *BBON* is the prefix of the Administration and Operations applications in the WebSphere for z/OS product.
- *Number* is the message number
- *Type* is I for information or E for error.

To see a list of messages issued during the current session, display the message log.

To display a list of messages in the help window, if it is not already displayed, select the Messages tab, , above the left frame. Then, to get help on a message, select a message number from the list in the left frame. The message description is then displayed in the right frame of the window.

When a message is displayed in the immediate message area at the bottom of the application window, you can display help for that message by clicking  on the tool bar.

See *WebSphere Application Server V4.0.1 for z/OS and OS/390: Messages and Diagnosis* for messages and for error codes.

Chapter 11. Trace and debug facilities

The Administration and Operations applications include trace and debug facilities that are intended to be used under the supervision of IBM support personnel. These facilities can be triggered by login options and menu bar actions.

Traces

A *trace* contains information about the methods that are invoked within the Administration or Operations application to realize the graphical user interface.

Enable tracing

To enable tracing,

- either specify the **Trace writer** option in the login dialog to cause trace entries to be buffered. Refer to “Define workstation environment variables for login options” on page 6 and “Login options” on page 9 about information how to specify login options.
- or, if you prefer to view the trace on the standard output, start the Administration or Operations application from a DOS Command Prompt:
 1. Open a DOS Command Prompt window and switch to your installation directory, e.g. c:\Program Files\IBM\WebSphere for zOS
 2. Switch to the bin directory: cd bin
 3. Invoke the application using the command bbonrun
 4. Select the **Options...** choice on the login panel
 5. Ensure that the **Trace writer** option is selected
 6. Click the **Set** option
 7. Login normally

View traces

To view traces, specify the **Diagnose** action of the **Options** menu bar choice.

Trace file

The **Trace file** action copies the internal trace table and other diagnostic information to a pair of files that you specify in a dialog box.

Trace viewer

The **Trace viewer** action displays the trace table in a trace viewer window. The window provides actions to specify different trace levels in the **Filter** menu bar action.

Note: If you have an active screen reader, press **Alt+z** to have the status area messages read. Press **Alt+z** again to exit the status area.

Trace writer

The **Trace writer** action will impact performance. It controls the trace writer, which causes the trace entries that are being written to the internal trace table to also be written to stdout.

Remember that you have to start the Administration or Operations application from a DOS command prompt to view the standard output.

Communications trace

The *communications trace* contains the IIOP communication packages that are transferred between the application and the Systems Management Server on z/OS.

Enable communications tracing

To collect the communications trace data in two files, e.g. `trace.1st` and `trace2.1st`,

1. Open a DOS Command Prompt window and switch to your installation directory, e.g. `c:\Program Files\IBM\WebSphere for zOS`
2. Switch to the `bin` directory: `cd bin`
3. Invoke the application using the following command:
`bbonrun >trace.1st 2>trace2.1st`
4. Select the **Options...** choice on the login panel
5. Ensure that the **Communications trace** option is selected
6. Click the **Set** option
7. Login normally

When your session ends, the trace data will be in the `trace.1st` and `trace2.1st` files in the directory `c:\Program Files\IBM\WebSphere for zOS\bin`.

Debug facility

The *debug mode* enables various internal diagnostic facilities that may impact performance.

Enable debugging

Specify the **Debug mode** option in the login dialog or the **Diagnose -> Debug mode** action of the **Options** menu bar choice to activate the debug mode. Refer to "Define workstation environment variables for login options" on page 6 and "Login options" on page 9 about information how to specify login options.

View debug information

To view debugging information, open the *message log* (refer to Chapter 9, "Message log", on page 129). Initially, debug messages are depicted with green letters. The color may be changed using the **Color..** action of the **Options** menu bar choice of the *main* window.

The **Environment** action of the **Help** menu bar choice displays additional information about referenced class names and their service level.

Part 5. Appendixes

Appendix A. User Interface

The section describes the general user interface topics for the Administration and Operations applications:

- Menu Bar
- Tool Bar
- Pop-up Menus
- Customize the User Interface
- Get help
- Use the Mouse and Keyboard
- Select an Object or Area
- Display Actions
- Copy, Cut and Paste Information
- Search Instructions or the Message Log
- Refresh the window

The WebSphere for z/OS Administration and Operations applications use windows with elements common to many graphical user interfaces, such as a menu bar, a tool bar, and pop-up menus.

Menu bar

The menu bar, which is at the top of the main and message log windows, provides menus of all the actions you can perform using the application. As an example, the menu bar choices for the Operations application is shown below:

File Selected View Options Help

Click on a menu bar choice, such as **File**, to display the pull-down menu. The pull-down menus show the tool bar icon and accelerator keys associated with the action, if any.

Menu bar actions for the Administration application

This section describes the actions that are available from the menu bar of the Administration application. The actions are described in the order in which they appear on the menu bar.

File

The **File** choice of the menu bar contains the following actions:

- Message log...
- Print screen...
- Connect to server...
- Administrators
- Exit

Message log...: The **Message log** action displays the Message Log

Print screen...: The **Print screen** action prints an image of the current window. To print something that is larger than a single screen (for example, a tree), use **Print screen** multiple times. From the Message Log or Trace Viewer window, use the **Print** action.

Connect to server...: The **Connect to server** action attempts to reconnect the Administration application to the Systems Management Server on z/OS after that connection has been lost. It causes the Login dialog to be displayed.

Administrators...: The Administrator's action displays a dialog that lets you add, modify and delete administrators.

Exit: **Exit** closes and leaves the application.

The size and position of the message log and trace viewer windows are saved. However, the size and position of the Administration window is not saved automatically. To save them, use the **Save window** action of the Options menu bar choice.

Selected

The Selected choice of the menu bar contains the actions that can be taken against a selected object in the tree. An object must be selected for these actions to be available.


Modify: The **Modify** action changes the properties form to edit mode, which allows you to change the properties of objects in the model.

Add: The **Add** action creates a new object in the tree, under the selected label. In some cases, it also causes associated objects to be created by WebSphere for z/OS .

When you add a new object, a blank properties form is automatically displayed. You must fill in the properties form and save your changes.

Delete: The **Delete** action deletes the selected object from the model, and deletes any objects that are below it in the branch of the tree.

Delete cannot be undone.

The tree is refreshed after you confirm the delete. Deleted objects are marked with , with the exception of conversations and LRM connections, which are removed from the tree.

Save: **Save** causes changes made in a properties form to be retained.

Select the **Cancel** action, if you do not want to save your changes.

Cancel: The **Cancel** action discards any changes you may have made to the properties form and causes the properties form to be put into browse mode.

Import Application...: The **Import application...** action causes a DDL file for an application to be installed.

The import action may take some time to complete. A message is issued after the application is imported.

Install J2EE application...: The **Install J2EE application...** action deploys an EAR file (which represents a J2EE application) to a J2EE server.

Import Server...: The **Import server...** action creates a new server and imports the server properties from the specified host file.

Export Server...: The **Export server...** action exports the server properties to the specified host file.

Go to: The **Go to** action scrolls the window to the object that is selected. This is useful if you have been scrolling up or down from an object and want to quickly return to it.

Build

The **Build** choice of the menu bar contains the actions necessary to complete a build of a model. The actions are:

- Validate
- Commit
- Instructions
- Complete
- Activate

Validate: The **Validate** action causes the Systems Management Server to check the validity of the selected model.

You can validate a model at any time before you commit it. It is good practice to validate a model when you have made a significant change.

A message is issued after the model is validated.

Commit: The **Commit** action causes:

- The model to be validated
- Instructions for additional z/OS tasks to be made available
- The model to be locked so that it cannot be modified

Instructions...: The **Instructions** action displays instructions for completing additional z/OS tasks in the right frame of the Administration application window.

It is available only after the selected conversation has been committed.

Complete: The **Complete** action of the menu bar contains secondary actions that let you mark the instructions complete, in preparation for activating the image. You can mark a type of instruction complete or mark all instructions complete.

Activate: The **Activate** action moves a committed model, or image, to become the active image.

Prepare for cold start: The **Prepare for cold start** action is only available for the active image. This action saves the active image and the definitions of the WebSphere for z/OS administrators on the host. No further modifications will be allowed until a cold start of WebSphere for z/OS is accomplished and the Administration application is restarted. The active image then is the only available conversation.

View

The **View** choice of the menu bar contains actions that affect the appearance of the items displayed in the window.

Refresh: The **Refresh** action updates the window with current information from the Systems Management Server.

On the main window, **Refresh** refreshes and collapses the tree. Your place in the tree will not be preserved.

When selected from the pop-up for instructions, **Refresh** refreshes just the instructions. This will reflect the state of any tasks that you have marked completed.

Stop: The **Stop** action stops the expansion of the tree. You can also use the  icon on the tool bar.

Expansion of the tree is stopped at the next logical point. If a branch of the tree is partially expanded when you select **Stop**, the expansion of that branch completes. No more branches are expanded.

Expand: The **Expand** choice expands the branch of the tree below the selected object, to show lower level objects. Other branches of the tree are unaffected. You can stop the expansion at any time with the **Stop** action of the View menu bar choice.

Collapse: The **Collapse** action collapses the branch of the tree below the selected object, so that lower level objects are not shown. Other branches of the tree are unaffected.

Expand tree: The **Expand tree** action expands the entire tree, so that the entire structure is displayed. This action may take some time to complete. You can stop the expansion at any time with the **Stop** action of the View menu bar choice.

Collapse tree: The **Collapse tree** action collapses the entire tree.

Options

The Options choice of the menu bar contains actions that customize the user interface. The actions are:

- Confirm
- Alarm
- Color...
- Font...
- Show tool text
- Show toolbar
- Save window
- Diagnose
 - Debug mode
 - Trace file...
 - Trace viewer...
 - Trace writer

Confirm: The **Confirm** action turns confirmation on or off. When confirmation is on, a confirmation pop-up appears when you perform destructive actions, for example, when you delete an object.

Some confirmations remain in effect even when you turn confirmation off. These include confirmations for committing a model or activating an image.

Alarm: The **Alarm** action turns the alarm on or off. When it is turned on, the alarm sounds when an error message is issued.

Color...: The **Color** action displays a dialog to set colors for the user interface. You can choose foreground and background colors for such things as objects in the window and messages.

Font...: The **Font** action of the Options menu bar choice displays a dialog to set fonts used in the window and for printing.

You can set fonts for such things as objects in the window and printing of the message log.

Show Tool Text: The **Show tool text** action shows or hides labels on the tool bar. The labels identify the function of each icon.

Show Toolbar: The **Show toolbar** action shows or hides the tool bar.

Save window: The **Save window** action saves the size and position of the window, for the next time you open the window. (The size and position of the message log and trace viewer window are saved automatically.)

Diagnose: The **Diagnose** action is for support purposes only and is to be used under the direction of IBM support personnel. It includes actions to help diagnose problems.

Debug mode: The **Debug mode** action is for support purposes only and is to be used under the direction of IBM support personnel. It enables various internal diagnostic facilities that may impact performance.

Trace file: The **Trace file** action is for support purposes only and is to be used under the direction of IBM support personnel. It copies the internal trace table and other diagnostic information to a pair of files that you specify in a dialog box.

Trace viewer: The **Trace viewer** action is for support purposes only and is to be used under the direction of IBM support personnel. It displays the trace table using a trace viewer function.

Trace writer: The **Trace writer** action is for support purposes only. It will impact performance and is to be used under the direction of IBM support personnel. It controls the trace writer, which causes the trace entries that are being written to the internal trace table to also be written to stdout. You can start the trace writer, and specify a trace level that determines which trace entries are written to stdout.

Help

The Help choice of the menu bar contains actions that display information about the. The actions are:

- Contents
- Message
- Tutorial
- Environment...
- About

Contents: The **Contents** action displays general help. This includes a table of contents and a brief introduction to using help.

Message: The **Message** action displays help for messages. If a message is displayed in the immediate message area at the bottom of the window, the **Message** action displays help for that message. Otherwise, it displays general help for messages and lets you select a message from a list.

Tutorial: The **Tutorial** action displays a tutorial that introduces the Administration and Operations applications.

Environment...: The **Environment** action displays a summary of the run-time and execution environment. Information from this panel (such as the product number, release, and so on) will be required if there is a need to contact IBM for service.

About: The **About** action displays product and copyright information.

Menu bar actions for the Operations application

This section describes the actions that are available from the menu bar of the Operations application. The actions are described in the order in which they appear on the menu bar.

File

The **File** choice of the menu bar contains the following actions:

- Message log...
- Print screen...
- Connect to server...
- Exit

Message log...: The **Message log** action displays the Message Log

Print screen...: The **Print screen** action prints an image of the current window. From the Message Log or Trace Viewer window, use the **Print** action.

Connect to server...: The **Connect to server** action attempts to reconnect the Operations application to the Systems Management Server on z/OS after that connection has been lost. It causes the Login dialog to be displayed.

Exit: **Exit** closes and leaves the application.

The size and position of the message log and trace viewer windows are saved. However, the size and position of the Operations window are not saved automatically. To save them, use the **Save window** action of the Options menu bar choice.

Selected

The **Selected** choice of the menu bar contains the actions that can be taken against a selected object. An object must be selected for these actions to be available.

Start: The **Start** action starts a server or server instance.

Stop: The **Stop** action stops a server or server instance, after currently running processes have completed.

Cancel: The **Cancel** action stops a server or server instance immediately. Currently running processes do not complete. The server and server instance will not be restarted by ARM.

Cancel with restart: The **Cancel with restart** action stops a server or server instance immediately. Currently running processes do not complete. ARM-protected servers and server instances will be restarted.

Warm start: The **Warm start** action restarts all server instances of the server or the server instance and increases the function level of a server or server instance to its capability level.

Go to: The **Go to** action scrolls the window to the object that is selected. This is useful if you have been scrolling up or down from an object and want to quickly return to it.

View

The **View** choice of the menu bar contains actions that affect the appearance of the items displayed in the window.

Refresh: The **Refresh** action updates the window with current information from the Systems Management Server.

Options

For the **Options** choice of the menu bar have a look at the chapter Menu Bar Actions for Administration: “Options” on page 142

Help

For the **Help** choice of the menu bar have a look at the chapter Menu Bar Actions for Administration: “Help” on page 143.

Menu bar actions for the message log

The menu bar of the message log lets you perform actions and set options.

File

The **File** menu bar choice contains the following actions:

- Save
- Print
- Print Screen...
- Close

Save: The **Save** action lets you save the information to a file. It displays a dialog for specifying the name of the file.

When you save the message log, it includes the date and time for each message, even if you have used the actions of the **View** menu bar choice to turn off the display of that information.

Print: The **Print** action displays a dialog that lets you print data (either the message log or the trace entries).

Print screen...: The **Print screen** action prints an image of the current window. To print something that is larger than a single screen (for example, a tree) from the main administration window, use **Print screen** multiple times. From the Message Log or Trace Viewer window, use the **Print** action.

Close: **Close** exits the current window. The size and position of the window are saved.

If the message log is not closed when the application is closed, it will be automatically reopened the next time the application is started.

Edit

The Edit choice of the menu bar contains actions that let you manipulate the data being displayed (either the instructions, the message log, or the trace entries). The actions are:

- Find
- Cut
- Copy
- Paste
- Select all
- Deselect all

The Edit menu bar choice does not put the properties form in edit mode. For that, use the **Modify** action of the Selected menu bar choice.

Find: The **Find** action lets you search for a character string.

Cut: The **Cut** action cuts a selected portion of information from the window and puts it on the clipboard. You can then paste the contents of the clipboard to another location, for example, into a note.

Unlike the copy action, the **Cut** action removes the selected text from the window. However, the change affects only the current display of the window. The next time the information is displayed, it will be complete.

Copy: The **Copy** action copies a selected portion of information to the clipboard. The information in the window that you copy from is not changed. You can then paste the contents of the clipboard to another location, for example, into a note.

Paste: **Paste** puts the contents of the clipboard into the current window. Information is put into the clipboard by a cut or copy action from any application.

You can paste information into an Administration window, for example, into a field on a dialog box.

Select all: The **Select all** action selects all of the data being displayed for subsequent copying or cutting.

Deselect all: The **Deselect all** action results in nothing being selected.

Filter Choice

The Filter menu bar choice contains actions that let you filter the message log. The actions are:

- Error
- Server
- String
- Reset

The actions identify what criteria will be used to filter the log. For example, selecting Error means that a severity of error will be used as a filter. Only messages with a severity of Error will be shown.

A check next to Error or Server indicates that the filter is in use. The **String** action is highlighted when it is in use.

The filters can be used in combination. To pass filtering, a message must meet all of the filter criteria. For example, if you select Error and Server, a message must be both an error-type message and a message issued by the Systems Management Server on z/OS to be displayed.

Error: The **Error** action controls a filter that is based on the severity of the messages. When the filter is in effect, messages with a severity of error are displayed. Error messages typically indicate a problem that needs to be resolved. Informational messages, which typically provide information about the successful completion of an action, are not displayed when this filter is in effect.

Error messages can be identified by an E following the message number, as in the following example:

```
BBON0140E Unable to open file E:\TEMP\cb390msg.txt,  
reason: java.io.FileNotFoundException  
Exception: E:\TEMP\cb390msg.txt.
```

When the filter is in effect, the action is checked.

Server: The **Server** action controls a filter that is based on the source of the messages. When the filter is in effect, messages issued by the Systems Management Server on z/OS are displayed in the message log. Messages that are issued by the Administration or Operations application are not displayed.

When the filter is in effect, the action is checked.

String: The **String** action displays a dialog box that lets you set a filter based on one or more text strings. When the filter is in effect, records that contain those strings are displayed. Records that do not contain the strings are not displayed.


When the filter is in effect, the word **String** is highlighted.

Reset: The **Reset** action turns off all filtering. Filter actions in the pull-down are unselected. Any selected strings on the Filter String Selection dialog are discarded.

View Choice

The View menu bar choice contains actions that let you control the appearance of the data being displayed. The actions are:

- Refresh
- Stop
- Dynamic mode
- Show date
- Show time

Refresh: The **Refresh** action causes the data to be immediately refreshed with new entries. You can also click  on the tool bar.

The data is refreshed automatically whenever you change a filter or make a change that affects the view of the data (for example, by changing the setting for show date).

Stop: The **Stop** action of the View menu bar choice stops the loading of data. The data will be incomplete. A message in the window indicates how many entries are not displayed.

You can also click  on the tool bar.

Dynamic mode: The **Dynamic mode** action causes new messages to be added to the message log as they are issued.

Show date: The **Show date** action causes the date the message or trace entry was issued to be displayed. The date appears in Julian format, *yyyy.ddd*.

Show time: The **Show time** action causes the time the message or trace entry was issued to be displayed.

Options Choice

The Options menu bar choice contains actions that let you set the following options:

- Log limit
- Show tool text
- Show tool bar
- Save window

Log limit: The **log limit** action displays a dialog that lets you limit the size of the message log in terms of number of messages.

Show Tool Text: The **Show tool text** action shows or hides labels on the tool bar. The labels identify the function of each icon.


Show Toolbar: The **Show toolbar** action shows or hides the tool bar.

Save Window: The **Save window** action saves the size and position of the window, for the next time you open the window.

Tool bar

The tool bar, which is immediately beneath the menu bar, provides an easy way to perform common actions. The tool bar for the Operations application is shown below:



Each icon represents an action. To perform the action, click on the icon. For example, to display the Message Log, click the  icon.

When an action is not available, the icon is grayed out. An action might be unavailable because it is not supported for the selected object or is not valid in the current context. The selected icon is highlighted.

To display a short description of the icons on the tool bar, you can either:

- Position the mouse pointer over an icon for a few seconds to see a description for that icon.
- Select the **Show toolbar text** action of the Options menu bar choice.

You can control the display of the tool bar with the **Show toolbar** action of the Options menu bar choice.

Pop-up menus

Pop-up menus, displayed by pressing the right mouse button, provide an easy way to perform actions. They display the valid actions for a selected object, or for an area of the window, for example, the instructions. Actions that are valid for the type of object but unavailable because of the status of the object (for example, because the object is locked) are grayed out.

Most of the actions on the pop-up menu can also be displayed and selected from the menu bar. However, some actions, including some that you perform against instructions, are available only from the pop-up menu.

Customize the user interface

The actions in the Options menu bar choice let you set such things as whether the tool bar is displayed, which colors and fonts are used, and whether confirmation is required for destructive actions.

1. Select the Options choice on the menu bar.
2. Select the desired action.

These actions apply only to the application (Administration or Operations) that you are using. You set options for the Administration and Operations applications separately. The settings are saved and will be used the next time you open the Administration or Operations application.

A separator bar divides the administration and operations windows into two frames. You can change the size of the frames by sliding the bar.

1. Position the mouse pointer on the bar. The mouse pointer will change shape when it is in position over the bar.
2. Press and hold the left mouse button.
3. Drag the bar right or left.
4. Release the mouse button.

To save the size and position of the administration or operations window:

1. Select the Options choice on the menu bar.
2. Select the **Save window** action.

Get help

General help or help on any topic:

1. Press F1. The help window is displayed with the table of contents in the left frame.
2. Use the table of contents, index, search and other links to find the help you need.

Help for the properties form:

1. Click the right mouse button anywhere on the properties form.
2. Select **Help** from the pop-up. Help for the object and its properties is displayed.

For some J2EE resource instance properties, a help text from the web is available which is called **Web Help**. If you have a web browser and a web connection, this help text is displayed.

Help for an object in the tree:


1. Click the right mouse button on the object.
2. Select **Help** from the pop-up. Help for the object and its properties is displayed.

Help for a tool on the tool bar:


Position the cursor over the tool for a few seconds. A description is displayed in a window.

Help for a message:

From the message log:

- Select the message in the log by selecting any text in the message or positioning the cursor in the message.
- Click on the message help tool on the tool bar  or press F1.

From the main window:

For help on a message in the immediate message area of the Administration or Operations window click on the message help tool on the tool bar  or select the **Message** action of the Help menu bar choice. Help is displayed in a pop-up.


If no message is displayed,

1. Select the **Message** action from the Help menu bar choice.
2. Select the message by number from the menu.

Use the mouse and keyboard

You use the mouse and keyboard to work with the Administration and Operations applications.

Select objects and actions

- **Mouse:** Move the pointer to the thing you want to select and click the left mouse button. This applies to objects in the tree, menu bar choices, tools on the tool bar, window, and so on. A selected object in the tree is highlighted.
- **Keyboard:** Use the arrow  and Tab keys along with the Enter key and accelerator keys. **Select an object in the tree** by using the up and down arrow keys to move up and down in the tree. The selected object is highlighted. **Select a menu bar choice** by pressing the Alt key plus the letter that is underlined in the menu bar choice. For example, press Alt and f for the File choice. Select an action from the resulting menu by typing the underlined letter of the action. You can also press the up and down arrow keys until the action you want is highlighted, then press Enter to select the action. Some menu bar actions also have accelerator keys associated with them. You can press these keys at any time to perform the action, without first selecting the menu bar choice. For example, the F1 key performs the **Help** action; F11 performs the **Validate** action. The

accelerators for menu bar actions are displayed to the right of the action. **Select a tool bar choice** by pressing the Tab key until the tool you want is highlighted. Press Enter to select the tool.

Expand an object's branch in the tree

- **Mouse:** Click the left mouse button on the node to the left of the object. (The node changes to point down.) Click the left mouse button on the node again to collapse the branch.
- **Keyboard:** For a selected object, press Enter to expand or collapse the branch.

Clear an input field

- **Mouse:** While holding the left mouse button down, drag the mouse pointer across the text. Release the mouse button and then press the Delete key.
- **Keyboard:** Position the mouse pointer in the field and press Alt+Delete. Clear the field from the position of the cursor with Ctrl+Delete.

Use an active screen reader to have messages in the status bar read

- **Mouse:**N/A
- **Keyboard:** Press **Alt+z** to enter the status area and have messages read as they appear. To exit the status area, press **Alt+z** again.

Shortcut keys

Function	Keys
Add (creating a new object)	INSERT
Collapse	CTRL+L
Delete (the selected object)	DELETE
Expand	CTRL+E
Import application	CTRL+T
Install J2EE application	CTRL+I
Opening Message Log...	CTRL+M
Print Screen...	CTRL+P
Refresh	F5
Save a modified or new object	CTRL+S
Stop	PAUSE/BREAK

Select an object or area

To select an object or a frame of the window:

1. Place the mouse pointer on the object or frame of the window.
2. Press the left mouse button once. The selected object is highlighted.

To select a particular block of text:

1. Place the mouse pointer at the start of the text.
2. Press the right mouse button.

3. While holding the right mouse button down, drag the mouse pointer to the end of the text.
4. Release the mouse button.

Display actions

You can display the valid actions for an object or area of the window:

1. Select the object or area of the window.
2. Press the right mouse button. The actions are displayed in a pop-up menu. To perform an action, select it from the pop-up.

Select the **Help** action from the pop-up to get help on the object, including a description of the valid actions.

Actions that are valid for the type of object but unavailable because of the status of the object (for example, because the object is locked) are grayed out.

Copy, cut and paste information

To copy information, use **Copy**. To move or delete information, use **Cut**.

1. Select the information you want to work with by holding down the right mouse button while dragging the cursor over the text. When you release the mouse button, a pop-up menu of actions is displayed.
2. Copy or cut the information by selecting the appropriate action in the pop-up menu. The information is put onto the clipboard.
3. Paste the contents of the clipboard where you want it to go.

Hints and Tips:

You can also use these keys:

- Cut: Ctrl+X or Shift+Delete
- Copy: Ctrl+C or Ctrl+Insert
- Paste: Ctrl+V or Shift+Insert

The tool bar of the Message Log includes tools for the cut, copy and paste functions

 cut  copy  paste


Search instructions or the message log

You can use the Find function to search the Instructions or the Message Log for a particular string of characters.

1. Display the Instructions or the Message Log.
2. Display the Find dialog. For Instructions, place the mouse pointer on the Instructions and press the right mouse button. A pop-up menu of actions is displayed. Select **Find**. For the Message Log, select the **Find** action of the Edit menu bar choice.
3. Type the characters in the input field and select other search options.

Refresh the window

To refresh the entire window with the most current information from the Systems Management Server:

1. Click  on the tool bar or select the **Refresh** action of the View choice on the menu bar. In the Administration window, the tree is rebuilt and displayed with all branches collapsed and no conversation selected. In the Operations window, the icons for servers and server instances are displayed.

To refresh just the instructions in the right frame of the Administration application window:

1. Click the right mouse button on the instructions to display a pop-up menu of actions.
2. Select **Refresh**.

Appendix B. Commands for operations

This section describes MVS commands that you can use to perform some operations tasks. It explains how to use commands to:

- Control WebSphere for z/OS server instances
- Control application environments

Controlling server instances

You can use MVS commands to control WebSphere for z/OS server instances. MVS commands allow you to:

- Start a server instance
- Cancel a server instance, that is, end it immediately
- Stop a server instance, that is, end it after current work has finished
- Perform a warm start for a server instance

The WebSphere for z/OS Operations application that runs on a Windows/NT workstation is the primary means for interactive control of CB server and server instances. It is described in online help that accompanies the application.

Start a server instance

To start a server instance, use the Start command.

Syntax

```
► Start ControlRegionProcName ,
  S      [.—server-instance]
►-srvname='—server-instance'
  [,—parms='-ORBCBI [COLD]
  [WARM]']
```

ControlRegionProcName

is the JCL procedure name that is used to start the server

server—instance

is the name of the server instance to be started.

Note: If you start a server instance with *ControlRegionProcName.server-instance*, you also have to specify this name to cancel or to stop this server-instance

parms=' -ORBCBI COLD'
specifies a cold start

parms=' -ORBCBI WARM'
specifies a warm start

Notes

- The Operations application allows you to start servers as well as server instances.

Cancel a server instance

To cancel a server instance, use the f (modify) command.

Syntax

```
►► f—ControlRegionProcName [.—server-instance] ,—cancel [.—armrestart] ►►
```

ControlRegionProcName

is the JCL procedure name that is used to start the server

server— instance

is the name of the server instance to be started.

Note: If you have started a server instance with

ControlRegionProcName.server-instance, you also have to specify this name to cancel this server-instance

armrestart

specifies a cancel with restart. If this argument is omitted, the server/server instance is not restarted.

Stop a server

To stop a server, use the Stop command.

Syntax

```
►► Stop—ControlRegionProcName [.—server-instance] ►►  
p
```

ControlRegionProcName

is the JCL procedure name that is used to start the server

server— instance

is the name of the server instance to be stopped.

Note: If you have started a server instance with

ControlRegionProcName.server-instance, you also have to specify this name to stop this server-instance

Controlling application environments

Workload Manager restarts WebSphere for z/OS server regions if they fail or are canceled. However, there are circumstances under which Workload Manager will leave a server region terminated and will not send it any more work, for example, when:

- The server region has been canceled repeatedly in quick succession
- The application environment for the server region has been quiesced

You can use MVS commands to display the status of application environments and to restart an application environment. The commands are summarized below. For complete descriptions of the commands, see *OS/390 MVS System Commands*.

Display application environments

To display information about application environments, use the D WLM,APPLENV command.

This command has sysplex scope.

Syntax

►► `Display` WLM,APPLENV `application-environment-name`

where *application-environment-name* is the name of the application environment. The application environment name is the same as the server name. Use * to display all application environments.

Restart application environments

To restart application environments, use the V WLM,APPLENV command. After this command is processed, server address spaces are allowed to start, and work requests that are queued are eligible for selection.

This command affects all systems in the sysplex.

Syntax

►► `VARY` WLM,APPLENV `application-environment-name`, RESUME

where *application-environment-name* is the name of the application environment. The application environment name is the same as the server name.

Appendix C. DDL keyword naming conventions

The following table describes the naming conventions for DDL keywords, where:

- Letters =
{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z},
- Numbers = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9},
- Special characters = {/, :, @, _}, and
- National characters = {#, \$, @}.

Keyword/Name	Editing characteristics (Max Length)	Valid characters
Application Name	char (234)	The first character must be a letter; the others may be letters, numbers, special or national characters. Embedded blanks are allowed.
- <i>description</i>	char (4096)	any
- <i>requiredJavaVMName</i>	char (4096)	any
Application Family Name	char (234)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>description</i>	char (4096)	any
Container Name	char (234)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>description</i>	char (4096)	any
Database Alias Name	char (4096)	any
- <i>ContainerUsesDatabaseAlias</i>	char (4096)	any
- <i>databaseDriverName</i>	char (4096)	any
- <i>databaseName</i>	char (4096)	any
- <i>parmList</i>	char (4096)	any
DataObject Class Name	char (234)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>dataObjectCreateFunction Name</i>	char (256)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>description</i>	char (4096)	any
DLL Name	char (8)	The first character must be a letter; the others may be letters, numbers or national characters.
- <i>description</i>	char (4096)	any
Home Name	char (192)	The first character must be a letter; the others may be letters, numbers, special or national characters.

- <i>CollectsHome</i>	char (242)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>description</i>	char (4096)	any
- <i>nameAsFactory</i>	char (256)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>nameAsHome</i>	char (256)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>visibleInCellNameTree</i>	Y, N	
- <i>visibleInWorkGroupNameTree</i>	Y, N	
ManagedObject Class Name	char (234)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>copyHelperFunctionName</i>	char (256)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>description</i>	char (4096)	any
- <i>homeMOCreatFunctionName</i>	char (256)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>interfaceName</i>	char (230)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>keyCreateFunctionName</i>	char (256)	The first character must be a letter; the others may be letters, numbers, special or national characters.
- <i>primaryKeyClass</i>	char (234)	The first character must be a letter; the others may be letters, numbers, special or national characters.
Map Expression Name	char (4096)	any
- <i>expression</i>	char (4096)	any
- <i>UsesMapExpression</i>	char (4096)	any
Mapped Type Name	char (4096)	any
- <i>implementation</i>	char (4096)	any
- <i>mt</i>	char (4096)	any
- <i>parmList</i>	char (4096)	any
- <i>primaryKey</i>	char (4096)	any
- <i>signature</i>	char (4096)	any
Table Alias Name	char (4096)	any
- <i>databaseTableName</i>	char (4096)	any
- <i>TableAliasUsesDatabaseAlias</i>	char (4096)	any
- <i>UsesMappedType</i>	char (4096)	any

Appendix D. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

Examples in this book

The examples in this book are samples only, created by IBM Corporation. These examples are not part of any standard or IBM product and are provided to you solely for the purpose of assisting you in the development of your applications. The examples are provided "as is." IBM makes no warranties express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose, regarding the function or performance of these examples. IBM shall not be liable for any damages arising out of your use of the examples, even if they have been advised of the possibility of such damages.

These examples can be freely distributed, copied, altered, and incorporated into other software, provided that it bears the above disclaimer intact.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

- CICS
- DB2
- DB2 Universal Database
- IBM
- IMS
- MVS/ESA
- OS/390
- RACF
- S/390
- VTAM
- VisualAge
- WebSphere
- zSeries
- z/Architecture
- z/OS
- z/VM

Tivoli is a registered trademark of Tivoli Systems, Inc. in the United States, other countries, or both.

ActiveX, Microsoft, Visual Basic, Visual C++, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft, Inc. in the U.S. and other countries.

Some of this documentation is based on material from Object Management Group.

Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

WHILE THE INFORMATION IN THIS PUBLICATION IS BELIEVED TO BE ACCURATE, THE OBJECT MANAGEMENT GROUP, AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND WITH REGARDS TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. The Object Management Group and the companies listed above shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Other company, product, and service names may be trademarks or service marks of others.

Programming interface information

This publication documents information that is NOT intended to be used as Programming Interfaces of WebSphere for z/OS.

Glossary

For more information on terms used in this book, refer to one of the following sources:

- *WebSphere Application Server V4.0 for z/OS and OS/390 Glossary*, SC09-4450, located on the Internet at:
<http://www.ibm.com/software/webservers/appserv/>
- Sun Microsystems Glossary of Java Technology-Related Terms, located on the Internet at:
<http://java.sun.com/docs/glossary.html>

If you do not find the term you are looking for, refer to *IBM Glossary of Computing Terms*, located on the Internet at:

<http://www.ibm.com/ibm/terminology/>

or the Sun Web site, located on the Internet at:

<http://www.sun.com/>

Index

A

- actions, displaying 152
- active image 19
- activity record 45
- administration
 - objects 53
 - tasks 19
 - user interface 15
- Administration application 1, 11
 - access 10
 - overview 2
 - start 7
- administrator
 - define 10
- application 54
 - actions 55, 66, 67, 68
 - import 31
 - properties 55, 66, 67, 68
- application family 55
 - actions 56
 - properties 56
- archive files 36
- Asserted Identities 31
- Automatic restart manager (ARM)
 - tasks 103
- Automation tasks 103

B

- BBONPARM environment variable 6
- blue lock 20
- bootstrap server 5

C

- capability level 111
- CBCONFIG environment variable 25, 48, 63
- certificate authority 28, 76, 91, 116, 121
- certificates 28, 76, 91, 116, 121
- class 56
 - actions 57
 - properties 56
- client certificates 30
- client interface 57
 - actions 58
 - properties 57
- cold start 48, 63
- collapse the tree 17
- colors, setting 149
- commands for operation 155
- communication
 - between workstation and host 5
- communications trace 136
- component 33
- configuration 19
 - change 19
- confirmation, setting 149
- container 37, 58
 - actions 61

- container (*continued*)

- properties 58
- conversation 19, 61
 - actions 62
 - activate 25
 - add 22
 - add an object to 23
 - change 22
 - commit 24
 - delete an object from 23
 - icons for 20
 - modify an object from 24
 - properties 62
 - states of 20
 - validate 24
- copy data 152

D

- DCE 28, 75, 91, 115, 120
- DDL files 31
- DDL keyword naming conventions 157
- debug mode 136
- deinstallation 11
- deploy Enterprise JavaBeans 32
- deploying 37
- deployment descriptor 35
- diagnosis of problems 127
- Distributed Computing Environment 28
- DLL 63
 - actions 64
 - properties 63

E

- EAR file 36, 37
- EJB 33
 - EJB client view 33
 - EJB component 33
 - EJB container 33
 - EJB JAR file 36
 - EJB links 35
 - EJB module 36
 - EJB references 35
 - EJB server 33
 - EJB specification 33
- Enterprise archive file 37
- Enterprise Java Beans 33
- Enterprise JavaBeans, deployment 32
- entity beans 34
- environment file 25, 48
- environment variables 79, 94, 97, 99, 118
 - for default administrator 10
 - for login options 6
 - for the configuration data path on the System Management Server 48, 63
- EPAC 28
- expand the tree 17
- extended privilege attribute
 - certificate 28

F

- filter
 - operations window 107
- find data 152
- fonts, setting 149
- function level 111

G

- Generic Security Service Application Program Interface 30
- golden lock 20
- GSS_API 30

H

- help, getting 149
- home 37, 64
 - actions 65
 - properties 64
- Home interface 34
- host names 5
- HOSTS file 5

I

- icons
 - for administration 17, 20
 - for conversations 20
 - for operations 107
 - in tree 20
 - on tool bar 148
- image 19
 - activate 25
- informational line 15, 107
- installation 5
- instructions 103
 - complete 25
 - completion summary 103
 - display 24
 - overview 103
 - refreshing 153
 - task detail 104
- interval record 45

J

- J2EE 32
 - J2EE application 37, 66
 - J2EE component 36, 66
 - J2EE module 36, 67
 - J2EE resource 35
 - location in the tree 69
 - properties 69
 - J2EE resource connection 36
 - actions 69, 71
 - location in the tree 70
 - properties 70

- J2EE resource instance
 - location in the tree 70
 - properties 71
- J2EE resource reference 36
- J2EE server 37, 71
- JAR file 36
- Java 2 platform, Enterprise Edition 32
- Java Naming and Directory Interface 35
- JNDI 35
- JNDI names 35

K

- Kerberos 30, 76, 92, 116, 121
- keyboard, using 150
- keyring 28, 76, 91, 116, 121
- keys 151

L

- logical resource mapping 80
 - actions 82
 - properties 81
- logical resource mapping connection 82
 - actions 83
 - properties 83
- logical resource mapping instance 83
 - actions 87
 - properties 84
- login options 6
- Logstream tasks 103

M

- menu bar
 - about action 144
 - activate action 141
 - add action 140
 - administrators action 140
 - alarm action 143
 - cancel action (administration) 140
 - cancel action (operations) 145
 - cancel with restart action (operations) 145
 - close action 146
 - collapse action 142
 - collapse tree action 142
 - color action 143
 - commit action 141
 - complete action 141
 - confirm action 142
 - connect to server action 140, 144
 - contents action 144
 - copy action 146
 - cut action 146
 - debug mode action 143
 - delete action 140
 - deselect all action 146
 - diagnose action 143
 - dynamic mode action 148
 - environment action 144
 - error action 147
 - exit action 140, 144
 - expand action 142
 - expand tree action 142
 - export server action 141

- menu bar (*continued*)
 - find action 146
 - font action 143
 - goto action 141, 145
 - import application action 140
 - import server action 140
 - instructions action 141
 - log limit action 148
 - message action 144
 - message log action 139, 144
 - modify action 140
 - overview 139
 - paste action 146
 - prepare for cold start action 141
 - print action 145
 - print screen action 140, 144, 145
 - refresh action 142, 145, 147
 - reset action 147
 - save action 145
 - save action (administration) 140
 - save window action 143, 148
 - select all action 146
 - server action 147
 - show date action 148
 - show time action 148
 - show tool text action 143, 148
 - show toolbar action 143, 148
 - start action 144
 - stop action (administration) 142
 - stop action (message log) 148
 - stop action (operations) 144
 - string action 147
 - trace file action 143
 - trace viewer action 143
 - trace writer action 143
 - tutorial action 144
 - validate action 141
 - warm start action 145
- message area 15, 107
- message log 129
 - filter 130
 - format 129
 - print 130
 - printing 131
- messages 133
- model 19
 - add an object to 23
 - commit 24
 - create 22
 - delete an object from 23
 - modify an object from 24
 - validate 24
 - verify 24
- model object
 - properties of 108
- mouse, using 150
- move data 152

N

- naming server 5
- non-authenticated clients 27, 74, 90, 115, 120

O

- objects
 - WebSphere for z/OS owned 24
- operation
 - objects 113
 - tasks 109
 - user interface 107
- Operation application 105
- Operations application 1
 - overview 3
 - start 7
- Operations applications
 - access 10

P

- packaging 37
- passticket 28, 75, 90, 115, 120
- password 28, 75, 90, 115, 120
- performance recording 45
- persistent objects 61
- pop-up menus 149
- production system 47
- profile 10
- progress bar 15, 107
- properties form
 - for administration 18
 - for operation 108

R

- RACF Keyring 28, 76, 91, 116, 121
- release level 111
- Remote interface 34
- resource 35
- resource factory 35
- Resource management tasks 103
- resource manager connection factory 35
- resource manager connection factory
 - reference 36
- resource reference 36

S

- Secure Socket Layer 28, 76, 91, 116, 121
- security class 27, 75, 76, 90, 91, 115, 116, 120, 121
- Security Server 28, 76, 91, 116, 121
- Security tasks 103
- select an object 151
- server 87, 113, 118
 - actions 80, 95, 118, 123
 - export 47
 - import 47
 - migrate 47
 - properties 72, 88, 113, 118
 - start 111
 - starting 111
 - stop 111
 - stopping 111
 - work requests for 109
- server instance 96, 123
 - actions 98, 125
 - properties 96, 124
 - start 111

- server instance (*continued*)
 - stop 111
- session beans 34
- shortcut keys 151
- silver lock 20
- SM_DEFAULT_ADMIN environment variable 10
- SMF 45
- SSL 28, 76, 91, 116, 121
- SSL Asserted Identities 31
- SSL Basic Authentication 28, 76, 91, 116, 121
- SSL client certificates 30, 76, 91, 116, 121
- SSL Kerberos 30, 76, 92, 116, 121
- SSL Type 1 28, 76, 91, 116
- SSL Use Confidentiality Only 31
- stateful 34
- stateful session beans 34
- stateless 34
- stateless session beans 34
- sysplex 98
 - actions 100
 - properties 99
- system 101
 - actions 101
 - properties 101
- System Management EUI 1
- System Management Facilities (SMF) 45
- System Management User Interface 1

T

- TCP/IP 5
- test system 47
- tool bar 148
- trace 135
- trace facilities 133
- transaction
 - global 59
 - hybrid-global 59
- transient objects 61
- tree 15
 - collapse 16
 - expand 16
 - icons 17, 20
 - location of objects 53

U

- user ID 28
- user interface 139
 - customize 149

W

- WAR file 36
- warm start 111
- window
 - for administration 15
 - for administration, example 2
 - for operation 107
 - filter 107
 - for operation, example 3
 - refreshing 153
- work request list
 - fields on 108

- work request list (*continued*)
 - overview 108
- Workload management (WLM)
 - tasks 103

X

- X.509 certificates 28, 76, 91, 116, 121

Z

- z/OS tasks 103

Readers' Comments — We'd Like to Hear from You

WebSphere Application Server V4.0.1 for z/OS and OS/390:
System Management User Interface

Publication No. SA22-7838-06

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Deutschland Entwicklung GmbH
Department 3248
Schönaicher Strasse 220
D-71032 Böblingen
Federal Republic of Germany

Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5655-F31

SA22-7838-06

