Problem Avoidance for WebSphere Application Server for z/OS

Typical problem areas in WebSphere for z/OS

Potential pitfalls and useful hints and tips

Problem avoidance checklists

Rica Weller
Per Fremstad
Suresh Maddukuri
Kiet Nguyen
Rajesh Pericherla

ibm.com/redbooks
First Edition (March 2006)

This edition applies to Version 6, Release 0, Modification 1 of WebSphere Application Server for z/OS (program number 5655-N01).
Contents

Notices .................................................................v
Trademarks ............................................................vi

Preface .................................................................vii
The team that wrote this Redpaper .....................................viii
Become a published author ...............................................ix
Comments welcome ......................................................ix

Chapter 1. Phase 1: Installation, configuration, and migration ..........1
1.1 Preparing the Installation .........................................2
1.2 Installation and configuration .......................................3
1.3 Migration ..........................................................4
  1.3.1 Migrating from V5.x to V6.0.x ..................................4
  1.3.2 Migrating from V4.0.1 to V6.0.x ..............................5
  1.3.3 Migrating from V3.5 Standard Edition to Version 6.0.x ....5
  1.3.4 Checklist for migration .......................................5
1.4 Coexistence ......................................................6
1.5 Most common problems ............................................7
1.6 Related references ................................................9

Chapter 2. Phase 2: Application deployment ..........................11
2.1 Tools for the deployment phase ....................................12
  2.1.1 Install and deploy application files ..........................12
  2.1.2 Logging and tracing .........................................13
2.2 Problem avoidance checklist ......................................14
  2.2.1 Assembling an application ...................................14
  2.2.2 Deploying an application ....................................14
2.3 Most common problems ............................................15
2.4 Related references ................................................18

Chapter 3. Phase 3: Run applications .................................21
3.1 Request process overview .........................................22
3.2 Model-view-control model for problem determination ...........23
  3.2.1 Typical problems in the view tier ............................23
  3.2.2 Typical problems in the control tier ........................26
  3.2.3 Typical problems in the model tier ..........................29
3.3 Problem avoidance ...............................................30
  3.3.1 Designing, coding, and testing ...............................31
  3.3.2 Change control ..............................................31
3.4 Related references ................................................32

Chapter 4. Phase 4: System run time .................................33
4.1 The WebSphere for z/OS V6 runtime environment ...............34
4.2 Problem categories in runtime phase ................................36
4.3 Understanding your own runtime configuration ..................38
4.4 Troubleshooting tips for the runtime environment ..............38
4.5 Security issues and problems ....................................39
4.6 Problem avoidance checklist .....................................39
4.7 Typical problems ................................................42
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 U.S.A.

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.

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.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.
## Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

<table>
<thead>
<tr>
<th>@server®</th>
<th>DB2®</th>
<th>Rational®</th>
</tr>
</thead>
<tbody>
<tr>
<td>@server®</td>
<td>IBM®</td>
<td>Redbooks™</td>
</tr>
<tr>
<td>Redbooks (logo)™</td>
<td>IMS™</td>
<td>RACF®</td>
</tr>
<tr>
<td>z/OS®</td>
<td>MQSeries®</td>
<td>S/390®</td>
</tr>
<tr>
<td>zSeries®</td>
<td>MVS™</td>
<td>Tivoli®</td>
</tr>
<tr>
<td>CICS®</td>
<td>OS/390®</td>
<td>WebSphere®</td>
</tr>
</tbody>
</table>

The following terms are trademarks of other companies:

- EJB, Java, JavaServer, JavaServer Pages, JDBC, JDK, JMX, JSP, JVM, J2EE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.
Preface

This Redpaper is intended for system programmers and administrators working with WebSphere® Application Server for z/OS® V6. It is targeted towards those who need to identify problems, analyze them, and fix them efficiently so that they can deliver good support for the WebSphere environment. Another objective is to assist IBM® support and technical professionals as well as Java™ developers who work in this environment.

We identified the possible problem areas that can be encountered when using WebSphere for z/OS and arranged them into four specific problem phases that correspond to steps in the WebSphere for z/OS life cycle as shown in the following figure.

Problem categorization based on WebSphere life cycle

These phases are:

- Phase 1: Installation, configuration and migration
- Phase 2: Application deployment
- Phase 3: Testing and running applications
- Phase 4: System environment (production)

We give a general overview of the problem areas in the individual phases, explain how to analyze them, and provide valuable hints and tips about how to avoid problems.
The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

**Rica Weller** is a Project Manager at the International Technical Support Organization (ITSO), working in New Zealand and the U.S. She was a Systems Engineer for S/390® for two years and a Senior Consultant for IBM WebSphere Business Integration on z/OS in the Competence Center with IBM Germany for three years. She also taught classes, presented at several conferences, and coauthored several Redbooks™ about WebSphere for z/OS and textbooks about zSeries® basics. Rica holds a degree in Business Administration from TU Dresden, Germany, and a Master Degree in Business from Massey University, New Zealand.

**Per Fremstad** is an IBM certified IT specialist from IBM Systems and Technology Group, Norway. He has worked for IBM since 1982 and has extensive experience with zSeries and z/OS. His areas of expertise include the Internet, IBM WebSphere product family, and Web enabling applications on z/OS. He teaches frequently on WebSphere and Java topics and about zSeries and z/OS at several universities. He holds a Bachelor of Science degree from the University of Oslo, Norway.

**Kiet Nguyen** is an I/T Specialist with IBM Global Services/AMS CRM Siebel Development in North Carolina, USA. His over 20 years of experience ranges from MVS™ systems/application programming to building component software and end-to-end applications on distributed platforms for worldwide customers. He holds a degree in mathematics from Georgetown University in Washington, D.C. His areas of expertise include J2EE™ Development, WebSphere Application Server Administration, and, recently, writing about these subjects.

**Suresh Maddukuri** is an IT Specialist who assists customers in the United States. He worked as administrator for IBM WebSphere Application Server on z/OS and distributed platforms. His main responsibilities with customers are troubleshooting problems, performance monitoring, and application server tuning. His areas of expertise include IBM WebSphere MQ and WebSphere Business Integration Message Broker. He holds a post-graduate diploma in Computer Applications and a degree in Mechanical Engineering from Nagarjuna University, India.
Rajesh Pericherla is a system tester and lead strategist in the IBM WebSphere for z/OS SVT in WQCoC organization. He has been working with this group for the last eight years. He is responsible for planning the system tests for the latest WebSphere releases on all the supported platforms with the main focus on z/OS. He holds a Master of Science degree in Computer Engineering from Walden University.

Thanks to all the contributors of the previous Redbooks WebSphere for z/OS V5 Problem Determination, SG24-6880, and the support team of the IBM International Technical Support Organization, Poughkeepsie Center.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this Redpaper or other Redbooks in one of the following ways:

► Use the online Contact us review redbook form found at:

ibm.com/redbooks

► Send your comments in an email to:

redbook@us.ibm.com

► Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400
Phase 1: Installation, configuration, and migration

To prevent errors and successfully install WebSphere Application Server V6 for z/OS (WebSphere for z/OS), you must implement the necessary features, subsystems, and resources required for the runtime environment. Figure 1-1 shows common problem areas.

This chapter describes various methods for preventing common problems during the installation and migration processes of WebSphere for z/OS. We give hints and tips for the coexistence of WebSphere for z/OS V6 with previous releases and list common problems and their solutions. We also point to means and tools that help solving problems in this phase.
1.1 Preparing the Installation

This section is based on best practices and summarizes tasks that you might want to verify before installing your WebSphere for z/OS system.

Prepare all the necessary z/OS subsystems and complete the planning for customizing your system environment before attempting to install WebSphere for z/OS.

On the WebSphere for z/OS home page in the panel on the left, click library or go directly to:
http://www-306.ibm.com/software/webservers/appserv/was/library

Beneath the tab for V6 and WebSphere for z/OS, you can access WebSphere Application Server V6 for z/OS Program Directory, GI11-2825, or go to the WebSphere Application Server V6.0 Information Center Web site:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp

Search for Planning for installation in the Information Center to understand the various tasks for preparing the installation of WebSphere Application Server V6 for z/OS. Refer to “Steps for creating your implementation plan” from there and follow all the steps (unless it says optional) to ensure a smooth installation process.

To help prevent errors and successfully prepare the environment for WebSphere for z/OS, use the following checklist:

- Fill out the worksheet provided by the Information Center after you search for “Checklist: Preparing the base operating system.”
- Contact your security administrator to set up a RACF® user ID and authorize the ID so that it has read/write access to the WebSphere Application Server for z/OS files (BBO.* data sets and HFS files).
- Increase your paging by one 3390-3 volume if your storage is constrained, or two if your system does any paging of the WebSphere Application Server address spaces.
- Make sure that your address space is large enough. Some WebSphere Application Servers might require a 1 GB virtual region to run any workload.
- Set up the Workload Manager (WLM) dynamic application environment (DAE) so that WebSphere for z/OS can use the capacity automatically. Otherwise it will rely on the static WLM environment.

The dynamic WLM application environment is a function of WLM that became available when APAR OW54622 was made available for Workload Manager. It is incorporated into z/OS Version 1.5 and later. With the new WLM function, programs can dynamically create application environments on the fly. WebSphere for z/OS V6 is designed to make use of the dynamic WLM application environment if it sees the function available. Generally, this is a function of WebSphere that you cannot turn off. If the dynamic capability of WLM is not available, WebSphere relies on static WLM application environments. Static application environments that are till in existence when APAR OW54622 is applied simply stop being used in favor of dynamically created ones of the same name.

You also must collect and determine important information about your specific setup for components of WebSphere Application Server for z/OS. For planning related to naming conventions, TCP port allocation, shared Hierarchical File Structure (HFS) and clustering, refer to the white paper and spreadsheet available at:
http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1331

The following checklist can help you with your specific setup:
The recommended size allocations for the Hierarchical File System is 250 cylinders (3390) for primary and a secondary allocation of 100 cylinders (3390).

If possible, set up your HFS so that the root HFS is shared by all processors and so that the deployment manager configuration is in an HFS configuration on a system-generic mount point.

Understand the HFSs for the application servers, the nodes, the daemons, and the cells.

It might be necessary to resize your system dump data sets due to the size of WebSphere address spaces, and where possible, evaluate the use of dynamic dump data sets.

If you are running in a sysplex, set up your TCP/IP with Sysplex Distributor to make use of dynamic virtual IP addresses (DVIPAs).

If Automatic Restart Management (ARM) is enabled, you might want to disable ARM for the WebSphere Application Server address spaces during installation and customization to avoid unnecessary restarts of address spaces. After installation and customization are complete, you should consider re-enabling ARM.

Search for the “Ensuring problem avoidance” topic in the Information Center for additional information about USS/HFS configurations, SMP/E tasks, ISPF dialogs, TCP/IP configurations, and security information.

1.2 Installation and configuration

After the planning is completed and the z/OS base systems are ready for installation, go to the section “Installing your application serving environment” in the Information Center for more detailed information about installing WebSphere for z/OS.

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp

You can also refer to Installing your application serving environment, GA22-7957-03, from the library Web site at:

http://www-306.ibm.com/software/webservers/appserv/was/library

To help prevent errors and successfully install and configure WebSphere for z/OS, use the following checklist:

Fill in the Customization Dialog worksheets. These worksheets are provided for each task in the installation and configuration dialog boxes to help you determine what values you should enter in the Define Variables stage of customization. You can find the worksheets in the Information Center by searching for “Customization Dialog worksheet.” Consult your system programmers and administrators, and other WebSphere and z/OS experts to determine the values of all the variables.

When selecting server, cell, or node names, always avoid special non-alphanumeric characters because they are used as HFS directory names and are parsed in XML files that might have problems with special characters, such as blanks, slashes, dashes, tildes, question marks, or underscores.

Where possible, use the default names the first time that you install WebSphere Application Server to make the installation instructions easier to follow.

Initial customization of WebSphere Application Server V6.0.1 for z/OS requires that an installation be at a minimum Service Level of 6.0.1.2 (PTF UQ04304). Check for the latest maintenance requirements at the following Web site:

Check the product PSP bucket WASAS601 subset H28W601 to verify that all the suggested maintenance has been applied.

Make sure that the product code HFSs are mounted at the directories you have chosen in the planning session.

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

Navigating the configuration HFS with a UID of 0 can alter files or their ownership and permission attributes, thus making them inaccessible to the WebSphere for z/OS runtime servers and administrators. It is better to use the WebSphere for z/OS administrator user ID.

Always run the installation jobs from the same system on which WebSphere for z/OS is being installed. Use the JOBPARM card below the JOB card to avoid running the jobs from different system. Syntax for the JOBPARM is /*JOBPARM SYSAFF=SXX where SXX is the system name.

If you plan to use DB2®, search for “Plan to prepare your DB2” in the Information Center and follow the instructions, hints, and tips provided.

1.3 Migration

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

To better understand the migration process, go to the section called “Migrating, coexisting, and interoperating,” in the Information Center at http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp or see WebSphere Application Server V6 for z/OS, Migrating, coexisting, and interoperating, SA23-2207, which is available from the WebSphere for z/OS library Web site: http://www-306.ibm.com/software/webservers/appserv/was/library

1.3.1 Migrating from V5.x to V6.0.x

The major difference between V5.x and V6.0.x is that the Java Development Kit (JDK™) is embedded in V6.0.x, while it was external to V5.x. Users do not have to worry about the JDK change because the migration process takes care of it during the transformation process.

The migration utilities in WebSphere for z/OS 6.0.x support migration from V5.x. Search for the article “Migrating product configurations” in the Information Center and use it as a starting point for the planning information, customization dialogs, and V5.x to V6.0.x migration explanations for stand-alone application server nodes, deployment managers, and federated nodes.

Migration from V5.x to V6.0.1 is the same as that from V5.0 to V5.1 at the highest level. You copy the existing configuration HFS, transform it to V6.0.x, and write it to a new HFS.

Prior to migration, the old configuration is renamed so that if, for any reason, the migration fails, users should be able to go back to their previous configuration.
See Migrating from WebSphere for z/OS V5.x to V6 - An Example Migration, WP100559, which is available from:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100559

1.3.2 Migrating from V4.0.1 to V6.0.x

There is no migration path from V4.0.1 to V6.0.x. WebSphere for z/OS v6 must be installed from scratch. You must use the Rational® Application Development tool to repackage applications that were developed for V4.0.1 to make them compliant with the J2EE V1.4.2 specification and to be able to deploy them in WebSphere for z/OS V6.0.x.

1.3.3 Migrating from V3.5 Standard Edition to Version 6.0.x

WebSphere Application Server for z/OS V3.5 Standard Edition (SE) users that are upgrading to V6.0.x are moving to a platform that is fully compliant with J2EE specifications. J2EE technology clearly separates development and the creation of applications from application administration, deployment, and management. Migration from V3.5 SE involves changes in application structures, development, and deployment. Using the J2EE model, you can develop applications independently from their final deployment environment. This task separation simplifies the process of promoting an application from initial development through production, or moving an application from one server to another. The intent is to change only the application deployment parameters, and not the application code.

There is no migration path from Version 3.5 SE to Version 6.0.x. You must use the Rational Application Development tool to redevelop and package applications so that they can be deployed in WebSphere for z/OS V6.0.x.

1.3.4 Checklist for migration

To help prevent errors and successfully migrate WebSphere for z/OS, use the following checklist:

- Update the prerequisites to the levels required by Version 6.x. Prior levels of WebSphere for z/OS generally continue to run at the higher prerequisite levels.
- The authors highly recommend that you be at Service Level 6.0.2.2 (PTF UQ07342) before initiating the migration process.
- Be aware of other versions of WebSphere for z/OS that you have running on your system when going through the dialog, because the customization dialog does not detect them for you.
- Before you run any migration jobs, ensure that every application server node in the WebSphere for z/OS V6 system has a WebSphere administrator user ID and password in its soap.client.props file.
- If you are running other versions of WebSphere or z/OS, watch out for potential location service daemon port collisions or LPA issues that are caused by different versions in one system.
- Review the ports that have been defined to ensure that the WebSphere for z/OS V6 installation does not conflict with existing port definitions of previous WebSphere for z/OS releases. In particular, when you are installing V6 to coexist with V4.01 or V5.x, note that the default daemon port definition for all versions (V6, V5, and V4) is the same.
- The authors recommend setting up V6.x installation with the STEPLIB if V6.x must coexist with any prior releases. Because of naming conflicts, V5 and V6 product code cannot be in LPA at the same time. To support coexistence:
a. Place the V6 SBBOLPA data set in the STEPLIB of the V6 daemon. If a prior level of SBBLOAD is in LPA, add a V6 STEPLIB for SBBLOAD. BBORTSS5 needs to be in LPA to make CTRACE work. If V5.0x is already installed and running in the system, then BBORTSS5 must be in LPA already. To check whether BBORTSS5 is in LPA, use D PROG,LPA,MOD=BBORTSS5 from SDSF, syslog, or console. This does not cause any coexistence issues, because the DLL name is different from V4.0.1 and the module is the same for V5 and V6.

b. Only one set of PPT entries can be active at one time for a given program. WebSphere for z/OS V4.0.1 and V5 both use the BBOCTL program as a control region. If you are running V4.0.1 and V5 on the same system, their BBOCTL programs share the same PPT entry. Prior to V4.0.1, Service Level W401610, the PPT attribute SYST was required for the BBOCTL program; it is not required after that service level. Therefore, including the SYST keyword in the PPT entry for the V5 BBOCTL program causes an informational message (IEF188I PROBLEM PROGRAM ATTRIBUTES ASSIGNED) to appear when starting a V5 server (V5 does not require the SYST attribute). This message does not affect the functionality of the WebSphere address spaces. If you do not want this message to appear, and your V4.0.1 Service Level is at least W401610, you can delete SYST from the SCHEDxx member to stop the message from being generated. If you are not at this service level, you must leave SYST in the PPT for BBOCTL to start the V4.0.1 server. IEF188I is issued when V5 server is started as long as BBOCTL is defined as a system task.

c. The authors recommend using the WLM dynamic application environment (DAE) when configuring V6 so that a specific server name can be used by V6 and by a server on V5.

- A return code of zero means nothing in a migration job. Be sure to review the .err and .out logs carefully for diagnostic information.
- A migration cannot be restarted after the process is started. If something fails during the process, you must start from scratch.
- Always run the jobs from the same system where the node being migrated is located. Use the JOBPARM card below the JOB card to avoid running the jobs from different system. Syntax for the JOBPARM is /*JOBPARM SYSAFF=SXX where SXX is the system name.

### 1.4 Coexistence

WebSphere for z/OS V6 can coexist with any of the prior WebSphere releases on the same LPAR as two different releases.

V6.0.x can coexist with V5.0.x or higher in the same cell with a few known limitations, such as:

- The Deployment Manager must be at the highest release level in the cell that has mixed releases.
- If you have multiple V5.0.x nodes on the same LPAR, they all should be migrated simultaneously as Version 5.0.x nodes cannot coexist with Version 6.0.x nodes on the same LPAR. This restriction does not apply for V5.1.x and V6.0.x nodes.
- The V6.0.x Deployment Manager can manage Version 5.x nodes. In a coexistence situation, the V5.x configuration tree cannot be modified until Service Level W6012XX or higher of the V6.0x driver. At this level, the restriction is lifted but modification can be done to existing nodes and servers.

**Important:** There must only be one DAEMON per cell in an LPAR.
The following restrictions apply when multiple releases or versions are on the same LPAR:

- Cells cannot have same short names.
- Only one version of the code can exist in LPA/LNKLST on the same LPAR; the others must be included in the STEPLIB.

For successful coexistence, ensure that:

- The load modules are in LPA for one system, and the load modules are in STEPLIB for the other system
- The ports are unique between the two systems
- The daemon_group_name values are unique between the two systems. This is a known cause of the ABEND EC3 with reason code 02060018.

### 1.5 Most common problems

This section describes the most common problems that encountered by users during WebSphere for z/OS V6 installation, configuration, migration, and in a coexistence setup. The following list includes typical problems, solutions, and tools to use:

- **Problem:** GIM23901E messages during APPLY using CBPDO.
  - **Solution:** Apply the latest maintenance to solve the problem.
  - **Tools to use:** SYSLOG and job log from the SMP APPLY step.

- **Problem:** Error CEE3250C. The system or user abend SCC3 R=00020001 was issued from compile unit BBODBDLD at entry point BBODBDLD.
  - **Solution:** Place SBBOLPA in LPA or in STEPLIB when installing WebSphere Application Server for z/OS. The problem is a documentation error in the Customization Dialog. See APAR number PQ80728.
  - **Tools to use:** SYSLOG, job log, and editors.

- **Problem:** Message IGW01513T appears during SMP/E receive of WebSphere Application Server for z/OS and OS/390®. This message is produced by the utility IEBCOPY while copying files to TLIB (target library).
  - **Solution:** This error is the result of the output record format of the PDSE being forced to *Fixed Blocked* (FB) when it should be *Undefined*. If you have a DFSMS ACS routine that controls the allocation of the PDSE data sets and forces the PDSE record format to FB, the result is this problem. You can resolve it if you use a DFSMS ACS routine to ensure that PDSEs are created with RECFM=U, and then SMPE receive will complete successfully.
  - **Tools to use:** SYSLOG and job log from SMP receive.

- **Problem:** After building your WebSphere for z/OS V5 environment, you need to change the WebSphere configuration root directory.
  - **Solution:** It is part of the setup process to specify the WebSphere configuration root directory. You then run a series of configuration jobs to build the WebSphere for z/OS V5 environment. After building your V5 environment, there is no simple way to change this root directory. The only option has been to rebuild the V5 environment from scratch, specifying the new configuration root directory in the ISPF panels and rerunning all of the configuration batch jobs. If you have made significant changes to the V5 environment, such as defining several clusters, creating various definitions, installing a number of applications, you must redo all of this work, which can be a lengthy process. The value of the configuration root directory file is that it is stored in
several places. This value should be changed in several files and some definitions could be easily missed. A procedure that addresses this problem can be found on the WebSphere for z/OS support page. Enter **configuration root directory** in the search field.

- **Tools to use:** None.

  - **Problem:** Error message in console:
    
    SECJ4046E: Duplicate login configuration name system.wssecurity.IDAssertion. Will over write. Duplicate login configuration name system.wssecurity.Signature. Will over write.
    
    - **Solution:** Remove all duplicate entries in wsjaas.conf, and then restart WebSphere for z/OS. If the node is part of a WebSphere cell, you might need to remove duplicate entries in all four locations where wsjaas.conf is stored:
      - `install_root/bin/wsinstance/propdefaults`
      - `DeploymentManager/bin/wsinstance/propdefaults`
      - `install_root/properties`
      - `DeploymentManager/properties`

- **Tools to use:** None.

  - **Problem:** You do not know how to check the version and history information of your WebSphere Application Server for z/OS V5 environment.

    - **Solution:** Look in the SystemOut.log file of the specific installation for the base Instal_Root/logs/nodeagent/SystemOut.log and for the Deployment Manager Instal_Root/logs/dmgr/SystemOut.log. Alternatively, you can run the versionInfo command from the /bin directory of the specific installation, for example:

      `Install_Root/bin>versionInfo.sh`

- **Tools to use:** None.

  - **Problem:** When WebSphere for z/OS V5.0.2 is running, starting up a new V5.1 application server causes the control region to abend with SEC3 in module BBOSSACE.

    - **Solution:** Further review of the configuration setting in the job logs for both V5.0.2 and V5.1.0 shows that the names of the daemon group (cell name) are identical:

      daemon_group_name: PDCELL

      This is a coexistence issue. Rename daemon_group_name for V5.1.0 to a different name.

    - **Tools to use:** Job log.

  - **Problem:** When you are generating installation jobs, you receive the error message BBOMNINS stating that:

    BBOMNINS: BBOIPCSP does not exist

    Or:

    BBOSCHED does not exist.

    The instructions for the creation of a Deployment Manager node referred to two optional members, BBOSCHED and BBOIPCSCP. However, they were not being generated for use. Also, the START command for the node agent in the managed node generated instructions referenced the wrong node name.

    - **Solution:** The Customization Dialog was updated to generate the missing jobs, and the START command is corrected. APAR PK07293 is associated with Service Level (Fix Pack) 6.0.2.1 (Build Level cf10533.10) of WebSphere Application Server V6.0.1 for z/OS.
– **Tools used:** FTP tool

The release number no longer appears in the job output. It would be very useful to see both the release number and the build level information. The following message appears in the log:

BB0M0007I CURRENT CB SERVICE LEVEL IS build level cf10515.05

– **Solution:** The cf1 stands for Cumulative Fix 1. We should see a cf2 in the future when we roll a series of WebSphere for z/OS APARs into Cumulative Fix 2. *cf10515.05* stands for: Cumulative Fix1, and the service date is the fifteenth week of 2005 (0515), and the .05 is the day of the week that the PTF was cut. The .05 is for internal use and it could be any value from .01 to .05. So the next set of PTFs could be cf10521.xx because we expect a PTF to be available about once every six to eight weeks. As to whether the cf1 would change with each PTF release, the answer is no. Only a major series of PTFs (a level set across WebSphere family of products across platforms) would result in the cf1 changing to cf2. So you can anticipate cf10521.xx or cf10523.xx for the next series of fixes. It makes it harder, but you still need to keep track of WebSphere maintenance levels. See the APAR/PTF table for WebSphere Application Server V6.0.1 for z/OS at:


– **Tools used:** FTP tool

### 1.6 Related references

Learn about using IBM WebSphere Application Server with this reading list, compiled for customers, consultants, and other technical specialists, by IBM Software Services for WebSphere:


Consider using WebSphere Application Server for z/OS IBM services. For example, you could have “WebSphere Proof of Concept for z/OS.” IBM consultants design and implement a working solution to your business problem that is identified during the Architecture and Design Workshop by using preconfigured IBM hardware and software. This solution allows you to implement a WebSphere production environment without interfering with your day-to-day business functions. For more information, see:

http://www.ibm.com/software/webservers/appserv/zos_os390/services/
Phase 2: Application deployment

This chapter provides you with information about how to avoid the most common problems that occur while assembling and deploying an application. We mention some typical problems and their solutions, and list some helpful tools and commands for this phase.

**Note:** This chapter assumes that the WebSphere for z/OS run time has been installed correctly with the latest maintenance, and all the installation and configuration jobs have run successfully.
2.1 Tools for the deployment phase

Before installing an enterprise application or other installable module in an application server, you must develop and assemble the module and configure the target server. Before choosing a server as a target for the module, ensure that the node version for the server is compatible with your module.

Develop and assemble J2EE modules using one of the three tools supported by WebSphere Application Server for z/OS for assembly:

- Application Server Toolkit
- Rational Web Developer
- Rational Application Developer

For more information about these tools, search for “Assembling applications” in the Information Center at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp

To install and deploy the application files, you can use the wsadmin tool for production environments and unattended operations or command-line tools to start and stop application servers, check server status, add or remove nodes, and complete similar tasks. WebSphere for z/OS also supports a Java programming interface for developing administrative programs. All of the administrative tools that are supplied with the product are written according to the API, which is based on the industry standard Java Management Extensions (JMX™) specification.

2.1.1 Install and deploy application files

These are the tools supported by WebSphere Application Server V6 for z/OS for installation and deployment of application files:

- Administrative Console
  See the topics “Installing application files with the console” and “Starting and stopping applications” in the Information Center for information about using the Administrative Console to install and deploy applications. We focus on problems related to this tool.

- wsadmin scripts with startApplication
  The wsadmin tool only supports Jacl and Jython scripting languages. Jacl is the language specified by default. If you want to use the Jython scripting language, use the -lang option or specify it in the wsadmin.properties file. Start the wsadmin scripting client interactively, as an individual command, in a script, or in a profile. Then, refer to “Deploying applications using scripting” and “Starting applications with scripting” in the Information Center for more information.

  Because the wsadmin tool is mainly intended for production environments and unattended operations and because it is described in detail in the Information Center, we focus on problems that occur when the Administrative Console is used in the deployment process.

- Java administrative programs that use JMX APIs and ApplicationManager or AppManagement MBeans
  WebSphere for z/OS supports access to the administrative functions through a set of Java classes and methods. You can write a Java program that performs any of the administrative features of the WebSphere for z/OS administrative tools. You can also extend the basic WebSphere for z/OS administrative system to include your own managed resources.
Investigate these tools with the Java APIs to determine the best ways to administer WebSphere for z/OS and your applications. For information about the Java APIs, see “Java Management Extensions (JMX) API documentation” in the Information Center, which outlines the following procedure for taking advantage of these tools:

a. Create a custom Java administrative client program using the Java administrative APIs.
   This topic describes how to develop a Java program that uses the WebSphere Application Server administrative APIs to access the administrative system of WebSphere for z/OS.

b. Extend the WebSphere for z/OS administrative system with custom MBeans.
   This topic describes how to extend the WebSphere for z/OS administration system by supplying and registering new JMX MBeans in one of the application server processes. In this case, you can use the administrative classes and methods to add newly managed objects to the administrative system.

c.Deploy and manage a custom Java administrative client program for use with multiple J2EE application servers.
   This topic describes how to connect to a J2EE server and how to manage multiple vendor servers.

d. Manage applications through programming
   This topic describes how to use Java MBean programming to install, update, and delete a J2EE application in WebSphere for z/OS.

- Java programs that define a J2EE DeploymentManager object in accordance with J2EE Deployment API Specification (JSR-88)

JSR-88 defines a contract between a tool provider and a platform that allows tools from multiple vendors to configure, deploy, and manage applications on any J2EE product platform. The tool provider typically supplies software tools and an integrated development environment (IDE) for developing and assembling J2EE application modules. The J2EE platform provides application management functions that deploy, undeploy, start, stop, and otherwise manage J2EE applications.

WebSphere for z/OS is a J2EE 1.4 specification-compliant platform that implements the JSR-88 APIs. See the Information Center topic “Installing J2EE modules with JSR-88” for more information.

2.1.2 Logging and tracing

The tasks for developing, deploying, and maintaining applications are complex. When an application encounters an unexpected condition, it might not be able to complete a requested operation. You want the application to inform the administrator that the operation has failed and why it failed so that the administrator can take the proper corrective action.

Application developers need to gather detailed information that relates to the path of a running application to determine the root cause of a failure that is due to a code bug. The facilities that are used for these purposes are typically referred to as **logging and tracing** and you access them as follows:

1. In the Administrative Console, select **Troubleshooting → Logs and Traces → app_server → Change Log Detail Levels**.

2. Under the configuration tab, specify a log detail level for a predefined group of components.
2.2 Problem avoidance checklist

A smooth deployment process is your main goal. However, things do not always go as planned. The checklists in this section are based on best practices to minimize the number of problems during the deployment phase. Check them before deploying an application and increase your chances for successful deployment.

2.2.1 Assembling an application

When you assemble an application, use this checklist:
- Ensure that the application is J2EE 1.4 compatible.
- The bean names must be unique within a given JAR file.
- Generate and deploy code using the proper assembly tools. For detailed information about assembling applications, starting an assembly tool, and configuring an assembly tool, see the WebSphere for z/OS Information Center topic “Assembly tools” at: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp
- Verify your data sources and create them for application if you have not done it before.
- Configure your database, Enterprise Information System (EIS), and other components properly so that there is no subsystem problem.
- Disable automatic regeneration of the config files by updating the global.properties file with an editor and setting com.ibm.servlet.engine.disableAutoPluginCfg=true. The configuration information must be updated manually when servlets are added to a Web application so that the WebSphere for z/OS Web Server Plug-in can recognize the new servlets.
- Tune and set the JVM™ heap size using this path: Servers → Application Servers → server_name → Process Definition → Java Virtual Machine. The best way to determine the necessary size is to run a series of tests, each time increasing the initial size. When you are tuning a production system where the working set size of the Java application is not understood, a good starting value for the initial heap size is 25% of the maximum heap size. The JVM then tries to adapt the size of the heap to the working set size of the application.

2.2.2 Deploying an application

When you are deploying an application, use this checklist:
- Check to see if the correct version of the EAR file and the full path name of the EAR on the server or on the client machine have been chosen.
- On the first Preparing for application installation page, specify the context root if you are installing a stand-alone WAR file.
- On the first Preparing for application installation page, specify whether default bindings for any incomplete bindings you might have. should be generated.
- Specify whether to precompile JavaServer™ pages (JSP™) files as a part of the installation. The default is not to precompile JSP files. This option can only be installed in a V6.x deployment target; otherwise, the installation will be rejected.
- Specify the directory where the application EAR file is to be installed. In a network deployment configuration, by default the application is installed in the APP_INSTALL_ROOT/network_cell_name directory. In a base configuration, it is installed in the APP_INSTALL_ROOT/base_cell_name directory.
Choose **Deploy enterprise beans**, if:
- The EAR file was assembled with an assembly tool such as Rational Application Developer, and the EJBDeploy tool was not run during assembly,
- The EAR file was not assembled with an assembly tool,
- The EAR file was assembled using versions of the Application Assembly Tool (AAT) previous to Version 5.

This option allows the EJBDeploy tool to run during application installation and generates code that is required to run EJB™ files.

*Note:* Choosing this option might cause the installation program to run for several minutes.

Ensure that the application name is unique in a cell and does not contain characters that are not allowed in object names.

Select **Deploy WebServices** if the EAR file has modules that are using Web services and has not previously had the wsdeploy tool run on it. The wsdeploy tool then can run during installation of the application and can generate the code that is required to run applications that use Web services.

### 2.3 Most common problems

This section is a collection of common problems that occur during application installation. It is not meant to be a comprehensive list, but rather to solve some typical problems that you might encounter.

**Problem:** No response from the Administrative Console

- **Solution:** Use SDSF to find which server is utilizing more CPU. One factor for high CPU is enabling Performance Monitoring Infrastructure (PMI), Logs and Traces. Remove the unwanted selection and save the configuration. Then stop and start the application server. Restart the Administrative Console with the specific port number.

- **Tools to use:** Administrative Console message log, SDSF

**Problem:** Large applications to synchronize or memory is constrained

- **Solution:** Adjust the JVM options to limit memory usage and therefore reduce the possibility of receiving Out Of Memory errors. The instance synchronization JVM uses default settings, unless you change the JVM options. Set the JVM options using the `INSTANCE_SYNC_JVM_OPTIONS` property:

  \[
  \text{asadmin set domain.node-agent.node_agent_name.property.INSTANCE-SYNC-JVM-OPTIONS="JVM_options"}
  \]

  An example is:

  \[
  \text{asadmin set domain.node-agent.node0.property.INSTANCE-SYNC-JVM-OPTIONS="-Xmx32m -Xss2m"}
  \]

  The node agent is `node0` and the JVM options are `-Xmx32m -Xss2m`.

  For more information about JVM options see the Web site at: [http://java.sun.com/docs/hotspot/VMOptions.html](http://java.sun.com/docs/hotspot/VMOptions.html)

  **Important:** Restart the node agent after changing the `INSTANCE_SYNC_JVM_OPTIONS` property because the node agent is not automatically synchronized when a property is added or changed in its configuration.
Problem Avoidance for WebSphere Application Server for z/OS

Tools to use: Administrative console message log, SYSPRINT and job log of the server, and WebSphere for z/OS error log

Problem: The Plug-in of the HTTP Server is unable to recognize the availability of another server because one server is down.

Solution: To avoid this problem, tune the WebSphere HTTP Plug-in configuration parameters to fit your environment so that users can experience fewer delays and failover performance of the WebSphere environment improves.

Tools to use: HTTP Plug-in config files, error logs

Problem: When a large application starts, your application server hangs and then shuts down. You get an ABEND SEC3.

Solution: You are experiencing a timeout in the HTTP transport. The recommended way to solve this is to find the reason for the timeout by analyzing the SVC dump. See WebSphere for z/OS Problem Determination Means and Tools, REDP-6880. You might have to increase or even disable the deployment manager timeout variables to circumvent the problem for a while or to enable the SVC dump.

Tools to use: Administrative Console message log, system log, SYSPRINT and job log of the server, and WebSphere for z/OS error log.

Problem: Unable to install a large .ear file and the following message in the WebSphere for z/OS error log:

BB000271E HTTP REQUEST EXCEEDED 10485760 BYTE INPUT BUFFER

Solution: Increase the following variables as follows by selecting Environment → Manage WebSphere Variables:

- protocol_http_large_data_inbound_buffer = 20485700 (or some other large number)
- protocol_http_large_data_response_buffer = 20485700 (or some other large number)

After setting these variables, recycle the DMGR and install the EAR file. If you receive the same message with the larger byte input buffer referenced in the error message, increase the number in the variables again.

Tools to use: Administrative Console message log, SYSPRINT and job log of the server, and WebSphere for z/OS error log.

Problem: Session data integrity is lost when concurrent access to a session is made in different Web modules.

Solution: This problem can occur when two Web modules are installed in different servers. If Web modules are installed on different servers, the applications might share session attributes between Web modules using distributed sessions, but session data integrity is lost. Also, the use of some session management features such as TIME_BASED_WRITES is severely restricted. Install the Web modules in an enterprise application on one server to share session attributes to eliminate these problems as follows:

i. Start the assembly tool.

ii. In the assembly tool, right-click the application (EAR file) you want to share and select Open With → Deployment Descriptor Editor.

iii. In the application deployment descriptor editor of the assembly tool, select Shared session context under WebSphere Extensions. Make sure the class definition of attributes that are put in the session are available to all Web modules in the enterprise application. The shared session context does not fully meet the requirements of the specifications.
iv. Save the application (EAR) file. In the assembly tool, after you close the application deployment descriptor editor, confirm that you want to save the changes that you made to the application.

- **Tools to use:** WebSphere Administrative Console for z/OS, Assembly tool.

- **Problem:** When you deploy an application using wsadmin, the plugin-cfg.xml is not updated.

- **Solution:** You need to install the application with a target of -cluster. After installation and before the application is saved, use $AdminApp edit to add the additional mapping to the Web server. After the application is saved, plugin-cfg.xml is regenerated.

- **Tools to use:** wsadmin

- **Problem:** Documentation for Rollout Update for cluster deployment is not clear for use in scripting with wsadmin.sh

- **Solution:** The function for Rollout Update can be found under AdminTask for scripting with wsadmin.

Refer to “Commands for the AdminTask object” in the Information Center at:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp

The updateAppOnCluster command can be used to synchronize nodes and restart cluster members for an application update that is deployed to a cluster. After an application update, this command can be used to synchronize the nodes without stopping all the cluster members on all the nodes at one time. This command synchronizes one node at a time by stopping the cluster members to which the application is targeted, performing a node synchronization operation, and restarting the cluster members. This command might take more time than the default connector timeout period, depending on the number of nodes that the target cluster spans. Be sure to set proper timeout values in the soap.client.props file when a SOAP connector is used and in the sas.client.props file when a RMI connector is used. This command is not supported in local mode.

- **Tools to use:** wsadmin, Information Center

- **Problem:** The pre-compile JSP phase failed during deployment under WebSphere Application Server V6 for z/OS. Example 2-1 shows the trace output.

**Example 2-1 Trace output**

ADMA0021E: An error occurred in compiling JavaServer Pages (JSP) files - applic.war (rc=1)
at com.ibm.ws.management.application.task.CompileJspTask.compileWar(CompileJspTask.java:152)
at com.ibm.ws.management.application.task.CompileJspTask.performTask(CompileJspTask.java:86)
at com.ibm.ws.management.application.SchedulerImpl.run(SchedulerImpl.java:253)
at java.lang.Thread.run(Thread.java:568)

**Solution:** Generally, this error is seen when the JAR file that is being used by the compiler is not in a readable or complete state. It could be truncated or malformed in some other fashion.

Check the disk space. The JAR file might be placed in /tmp/app_1052f9db0a/ear. Ensure that there is enough free space for that directory and for the WebSphere for z/OS temp space that the application server is using to compile.

**Problem:** The following error occurs during synchronization when an EAR file is being deployed in WebSphere Application Server V6 for z/OS:

EDC5129I No such file or directory.
The configuration synchronization completed successfully but there is an error message in the trace as shown in Example 2-2.

**Example 2-2  EDC5129I error**

| Trace: | 2005/06/29 08:42:59.948 01 t=AC44F8 c=UNK key=P2 (13007002) |
| ThreadId: | 00000229 |
| FunctionName: | com.ibm.ws.management.repository.FileRepository |
| SourceId: | com.ibm.ws.management.repository.FileRepository |
| Category: | AUDIT |
| ExtendedMessage: | BB000222I: ADMRO0016I: User AHCPLEX/ASCR1T modified document cells/CELLDM1T/nodes/NODEB1T/servers/IMWEBPR6/plugin-BSYS-V61-cfg.xml. file:///Was601DB1T/V6R0/AppServer/profiles/default/config/cells/CELLDM1T/clusters/CLUSTER3/sib-engines.xml (EDC5129I No such file or directory. (errno2=0x05620062)) |

**Solution:** This message does not indicate any problem in the WebSphere environment. The sib-engines.xml file is missing in /cells/CELLDM1T/clusters/CLUSTERxxx after migration to V6.0.

See APAR PK07966 for more details:

http://www-1.ibm.com/support/docview.wss?rs=404&uid=swg1PK07966

After FixPack 6.0.2 is installed, the message will not appear.

**Tools used:** None.

### 2.4 Related references

For information about installing EAR, EJBs, WARs, resource adapter (connector or RAR), and application client modules, search the WebSphere for z/OS Information Center for “Ways to install applications or modules.”
For additional hints and tips about how to avoid problems in the deployment phase, search for “Troubleshooting deployment” in the WebSphere for z/OS Information Center.

The Web site for the WebSphere for z/OS information center is:
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp

See also Disabling the Deployment Manager HTTP Timeout, TD101703, available at:
http://www.ibm.com/support/techdocs

Refer to “Sample Scripts for WebSphere Application Server” at:
Chapter 3: Phase 3: Run applications

This chapter provides information about and guidelines for addressing application problems after an application has been successfully deployed to the hosting application server. When we say “deployed successfully,” we mean that the application runtime environment was correctly configured to map to all the components that the application requires to run. This chapter follows the flow of a request/response transaction. At each in the tier, we identify the problems that can occur and suggest actions that can take to fix them.

Consult this chapter to solve problems and to plan for and build better solutions. When you are designing the architecture, it is essential to know about the problems, the areas that they can be in, and their frequencies in your system. More time and resources can therefore be allocated to reinforce those areas that need them more during the design and test phases.

We summarize the chapter by revisiting some best practices, methods, and techniques that you can apply to build robust solutions and thus limit application problems.

We also provide sources for more in-depth information about addressing application problems after an application has been successfully deployed in 3.4, “Related references” on page 32.
3.1 Request process overview

When an application is perceived as not working properly, you have to do preliminary diagnostics to verify reported problems. The first step is to familiarize yourself with the flow of the request/response model as it flows through the layers of the J2EE topology (Figure 3-1). Knowing this helps you pinpoint which problem area (tier) a particular error falls into, and quickly discard any area that is not relevant.

A typical request flow, as shown in Figure 3-1, might be:

1. The browser or client issues a request for a resource from the J2EE application.
2. After the request is cleared and authenticated by the z/OS security component, it is routed to the Plug-in. The Plug-in forwards the request based on directives in plugin-cfg.xml file that are masked against resource requested, transport protocol (secured or non-secured), and destination Plug-in. This task is done by the Web server Plug-in.
3. The Controller validates the request for resource access and puts it in the WLM queue. If there is no preallocated servant task running to pick the request off the queue and process it, the controller starts one. This scenario can happen if the number of servant tasks has reached the default maximum.
4. The J2EE application server, also called the servant region, loads the components from the required WebSphere class libraries into the runtime environment and invokes the application.
5. The request is routed to a container and a servlet is loaded to service the request. Common events that take place during the servicing of a request are:
   - If the servlet is not already loaded, it is loaded.
   - If the servlet is packaged with “load on initialize,” it is loaded when the server is started.
   - If not, it is loaded when the first request hits.

6. Requests for data from the servlet are classified depending on the types and intended activities (read-only, update). Based on this classification, the appropriate EJBs are invoked. EJBs act as internal application data brokers and shield the applications from the mechanics of having to model and format data every time they require them.

7. Physical data is located on data servers. Database software models data to a predefined design and access patterns (hierarchical, relational, sequential). Organizations choose DB software that best fits their processing needs. J2EE connectors “connect” Java applications to data repositories with programming APIs. They do what EJBs do for internal applications that are accessing relational databases: they shield the clients from the mechanics of having to know the attributes of every piece of data needed. The two types of J2EE connectors are:
   a. JDBC™ for relational databases (inside a WebSphere environment)
   b. JCA (implemented as Resource Adapters) for EIS databases (outside a WebSphere environment)

For more information about J2EE connectors, visit:
http://java.sun.com/j2ee/connector/

8. Data is retrieved from the enterprise complex and returned to the EJB that is making the request. This data is processed (by the program code) and sent back to the requestor in HTML or XML format.

9. If the request has been serviced with no errors, it is posted back to the browser or client, and the HTTP return code is set to 200.

Knowing how far a request gets is critical in eliminating components that do not need to be addressed. To reinforce this concept, we take a request and superimpose it onto our J2EE framework. From there, we can easily identify the layer or tier where the problem area may lie.

3.2 Model-view-control model for problem determination

J2EE programming adheres to the model-view-control (MVC) standard. We apply this model to analyze areas with common (and potential for) problems. This enhances the understanding of the WebSphere for z/OS components together with thorough problem analysis and problem determination methodology. For the approach that we discuss in this section, keep Figure 3-1 on page 22 in mind.

3.2.1 Typical problems in the view tier

Based on the MVC model, anything that presents information to the user is located in the view tier, also called presentation tier or user interface layer. This tier is the starting point (where request is initiated) and endpoint (response) of a Web transaction. It is most often the only relevant contact (interface) that we have with the underlying system that services our business needs.

Components that belong to this tier are browsers, thick clients, Web servers, Web server plug-ins. See the components with the numbers 1, 2, 3, and 9 in Figure 3-1 on page 22.
The necessary logs for the IBM HTTP Server can be found in the directory 
<plugin_install_root>/logs/<web_server_name>/.

The files are:
- http_plugin.log  Confirms that the HTTP Server started and initialized
- error.log       Records errors within the server
- access.log      Records inbound and outbound requests

The Web server Plug-in software handles communication between a Web server and the 
application modules in the Web container. It acts as a somewhat intelligent router for HTTP 
requests based on directives from its configuration file.

We discuss typical problems in the order of a request/response flow from the diagram in 
Figure 3-1 on page 22.

**Problem: Cannot get to an application/application does not work**
Follow these steps to analyze and fix the problem:

1. Make sure that you don't have any typographical errors in the URI and that all HTTP 4xx 
codes are client error codes.

2. Search for the HTTP error code in the HTTP Server Information Center or consult the W3 
Consortium site at:
   http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

3. Check to see if the Web server has been started. Go to SDSF, select the SYSPRINT from 
   the controller job (installation dependent), procstep BBOCTL, and look for this message:
   PLGC0057I: Plug-in configuration service is started successfully.
   You can also access the top level of the URL from a Web browser. For example:
   http://<web_server_name>/

   Or, you can ping the server:
   ping <web_server_name>

4. If the Web server does not open a default welcome or error page, check in SDSF to see if 
   the Web server job is running. The name of the job is installation dependent, so verify it 
   with yours. If it is not running ensure that it is as follows:
   a. Start the Web server by issuing:
      ```
      START <web_server_procname>
      ```
   b. After the Web server is running, verify access to the application:
      ```
      http://<hostname>/snoop
      ```

5. If you still have problems, check to see if the application can be accessed directly with the 
   embedded HTTP server in the Web container:
   a. Go to Application servers → appserver link → HTTP transport. The Web container 
      default ports are listed on this screen, one for non-SSL and one for SSL.
   b. Invoke the application again using this URL format:
      ```
      http://<hostname>:port/snoop
      ```

6. If this did not work (you were not able to access the Web container), then the Web server 
   and its Plug-in have problems. Web servers and associated Plug-ins are simple and
solidly built components. Once they are running, they work. It is rare that bad code upgrade was released for the Plug-in.

Most problems in this area are related to incorrect changes that were made to the Plug-in file or files that were corrupted. With that in mind, incremental back-up of your Plug-in configuration is recommended.

In USS, use the -nostop and -nowait options because it is not necessary to stop the server to back up or restore configuration files:

```
backupConfig <backup_file> [options]
```

For example, you can issue:

```
backupConfig myFile.zip -nostop
```

The -nostop option does a backup in place; the server does not have to be stopped for backup to be performed.

```
restoreConfig <backup_file> [options]
```

For example, you can issue:

```
restoreConfig -nowait
```

The -nowait option does a restore in place; the server does not have to be stopped for restore to be performed.

7. To verify that the Plug-in is the problem, swap the configuration file you think is in error with a good backup copy. Use the Administrative Console to apply changes to the Plug-in file. Although it is possible to edit it manually, it is not recommended.

8. If this works, then you only have to analyze the two Plug-in files for differences to determine where the problem lies. When you know what causes the problem, you can fix it.

9. You can also regenerate the plugin-cfg.xml from the Administrative Console if you suspect that the copy on the local server is bad. Select Login → Servers → Web Servers and click Generate plug-in. If you are running in a network deployment environment, this action replaces the plugin-cfg.xml copy at the node server with the master copy stored at the deployment manager node.

```
Note: The Plug-in is an xml file. The directives that usually are changed when servers are remapped and reconfigured are: VirtualHostGroup, ServerCluster, VirtualHostGroup, and UriGroup.
```

10. After you get past the Web server and the application still does not respond to your requests, there are a few things that you can do to check on the server and its status.

Check for the servant job in SDSF (job name is installation dependent, look for Procstep=B805R). If it is not running, it does not show. In that case, you must start the application server as follows:

```
START <appserver_proc_name>,JOBNAME=<server_short_name>,
   ENV=<cell_short_name>.<node_short_name>.<server_short_name>
```

For example, we issued:

```
START WS6552C,JOBNAME=WS6552,ENV=CL6552.ND6552.WS6552
```

If you are running in a network deployment environment, you can start the application server from the Administrative Console. Select Login → Applications → Enterprise Applications → Select Application (checkbox) → Start.

11. If there is a servant job, select its SYSOUT (this log has the error trace turned on by default). Make sure that you do not see any exceptions logged. If the server is running but
the application is not responding, usually that means it has run into an Out Of Memory (OOM) problem. This can be caused by a limited heap size or other resource constraints. Check which processes use most of your memory and if this behavior is as expected or caused by configuration mistakes.

12. You can also use the Administrative Console to verify the status of the application server as shown in Figure 3-2. Go to Administrative Console and select **Expand Troubleshooting → Logs and Trace**.

The green arrow confirms that the application server is running. The startup of the application server automatically starts all the applications that it hosts.

---

**Problem: Browser rendering static content incorrectly**

A frequent browser problem that you can encounter is that the static content that is being served up is incorrectly rendered. This is because the source file is being transferred to z/OS in ASCII from workstations. HFS handles files in EBCDIC and any data at the presentation layer must be in ASCII, which might cause some confusion.

With your image, text, or source file available on your workstation, verify that the file has good contents there. Use a browser to try to open it. If the file content is good and its association is correct then it is viewable. If the file is not viewable, just retransfer the file in binary mode.

**Tip:** Issue the `bin` command to set the transfer mode to binary before transferring files between your workstation and HFS, and vice versa.

**Problem: Erratic browser response, inconsistent and quirky interface**

Sometimes, older (outdated) browsers also give erratic responses and render contents incorrectly.

To check on updates, fixes, and for a list of supported browsers see:


---

### 3.2.2 Typical problems in the control tier

After your request moves past the Web server, it arrives at the application server, which is in the control tier (see the components with the numbers 4, 5, and 6 in Figure 3-1 on page 22). This tier controls data that is entering or leaving the enterprise. The majority of application problems happen in this tier because it is where the WebSphere Application Server engine is located. It contains the two main containers for the Web and EJB components and these two containers make up the runtime environment for all applications deployed. They also act as the broker for all other required services.
Problem: Loss of session affinity
You can determine that you have session affinity problems when you must re-log on and enter passwords from page to page even though you are using the same function in the same record in the same application.

The Web container keeps track of sessions with cookies that carry session IDs that are passed back to the browser. Session data is maintained in the application server memory. The Plug-in configuration file is set up to enable session affinity by default and uses the Cloneld parameter for session IDs. The parameter can look like this:

```xml
<Server CloneID="80mn5ljkma" ...>
```

The Plug-in generation process adds the Clone ID parameter by default; this is how the Plug-in identifies each application server. When this ID is used, subsequent requests get routed back to the server that generated the session ID. This is the most efficient way of handling session affinity. Other ways to maintain session affinity throughout the servers in a cluster are:

- Persisting session information to a database
- Applying in-memory copying of objects between JVMs (domains replication)

The preferred and recommended way is to keep session objects on the server, the second option.

If you are experiencing a loss of session affinity, check to make sure that cookie writing is allowed in your browser and client firewall software. The browser software has different tabs or sub-menus for enabling cookie writing but, in general, it is in the privacy area.

Problem: Program loop
Program loops usually result in a request not responding, or a component hanging, usually followed by a timeout error. In some cases, things seem to work fine, but some tasks in the address space keep consuming system resources without producing a result. You can see this when CPU usage is high but nothing seems to justify this. A dump or trace might be necessary because the log information is not sufficient for determining the cause of the loop or the unusual high resource consumption.

Indications of a loop are:

- A repetitive message from a module waiting for work and nothing being done, such as
  
  ```
  ExtendedMessage: <component> waiting for next server work
  ```

- A repetitive message that a module is active and you are able to follow the executed address ranges, but the only thing changing is the time stamp.

- A repetitive message from a module processing work/requests but the thread ID stays the same. Notice the ThreadID and FunctionName in Example 3-1. They might stay the same, but the trace header line with the time stamp changes if a loop is occurring. There might be several other messages between the repetitions. The shorter the loop cycle, the more likely it is for you to recognize the loop.

**Example 3-1   Looping thread**

```plaintext
Trace: 2005/08/19 21:23:41.232 01 t=7D19C0 c=UNK key=P8 (13007002)
ThreadId: 0000006d

FunctionName: com.ibm.etools.validation.validationbuilder
SourceId: com.ibm.etools.validation.validationbuilder.UserStateRegistry
ExtendedMessage: closeUser - found UserPrefs: UserPreferences: nodeName:nd6552, serverName: ws6552, userId: waspd2, refreshRate
```
If you suspect an application loop or hang, you can use:

- IPCS to format a trace and analyze for recurring psw addresses (for more information, see *WebSphere for z/OS Problem Symptoms and Analysis*, REDP-6002 to be published in March 2006)

- The com.ibm.jvm.svc.dump.Dump utility to identify:
  - The thread under which a loop is occurring
  - The threads contending for resources or involved in a lockout
  - A thread waiting for some operation that is external to the server

For more information, see *WebSphere for z/OS Problem Determination Means and Tools*, REDP-6880.

Before you contact IBM for service, use these tools to identify the particular component or subcomponent that is responsible for the failure. In most cases, the components are the application program code rather than product code from IBM. Present the component name together with the class and method name from the trace to your application development team or IBM (in the case of IBM components). This enables them to fix the code quicker.

**Problem: JSP source code shown by the Web server**

The first indication that you have this problem is when you type in a URL (for example: `<hostname>/hitcount`), and there is no application page such as that shown in Figure 3-3.

![Figure 3-3 Hit Count application Web page](image)

Instead, you see a screen of source HTML in the browser window such as that shown Figure 3-4.

```html
<html>
<head><title>IBM WebSphere Hit Count Demonstration</title></head>
<script type="text/javascript">
function enableLookupButtons() {
 var myButtons = document.getElementsByTagName("lookup");
 for (i = 0; i < myButtons.length; i++){
 myButtons(i).disabled = false;
 }
}
</script>
</html>
```

![Figure 3-4 HTML source code instead of proper application](image)
Whenever the document root of the WebSphere Application Server is the same as the Web server document root, the Web server JSP source file is shown as plain text. Using the Plug-in directives, you can define to the Web server that a request is to be handled by the WebSphere Application Server. If an inbound request does not match any entry in these directives, the control returns to the Web server. In this case, the Web server searches the resource requested in the document root. Because the JSP file is stored in the document root, the Web server displays it as plain text.

To avoid the plain text display, move the application server JSP source file outside the Web server document root. When a request comes in with an unknown host header, the Plug-in returns control to the Web server and if the Web server cannot find this JSP source file in its root document, it returns an HTTP 404 error message instead.

### 3.2.3 Typical problems in the model tier

This is the last tier in the topology (see the components with the numbers 7 and 8 in Figure 3-1 on page 22). This (model) tier represents the data as the application would see it according to its access authorization (visibility and role). For example, a bank manager would see more details and have more options on a banking application account screen than a bank teller would.

If data is modeled properly, it has a built-in access control functionality (for example, a bank customer does not have the screen option, and therefore no access to it) and it maintains the state of the data (persistence), so that information such as account balances are accurate in real time.

**Problem: Application cannot access data**

This condition has more than one possible cause:

- The user of a request has insufficient authorization granted for a certain resource.

  If this is the case, you almost certainly find an authorization failure message such as that found in Example 3-2 in the job log. Request appropriate authorization from the security administrator so that the application can run with sufficient privileges.

**Example 3-2 Sample failure message**

ICH408I USER(WASPD2 ) GROUP(SYS1 ) NAME(USER1) /u/waspd2/.sh_history CL(FSOBJ )
FID(00000000000000004C03000000190000) INSUFFICIENT AUTHORITY TO OPEN
ACCESS INTENT(RW-) ACCESS ALLOWED(GROUP ---) EFFECTIVE UID(0000003174) EFFECTIVE
GID(0000000000)

- The connection to the back-end EIS or data sources has problems.

  There are two types of connectors to back-end data:
  - JDBC driver embedded with WebSphere if you are using DB2 and
  - EIS JCA connectors provided by IBM if using IMS™ or CICS®.

  Regardless of which connector you use, they all have traces and logs that can be collected to help diagnose errors.

  An incorrect configuration usually results in an exception logged in the SYSLOG when the resource is being accessed. A timeout exception occurs when the back-end system server (EIS or DB2) is unavailable.

  A JDBC trace is useful for diagnosing problems in the DB2 Structured Query Language for Java and Java Database Connector (SQLJ/JDBC). The output goes to an HFS file that is specified in the JDBC properties file. JDBC trace information shows Java methods,
database names, plan names, user names, or connection pools. Use the Administrative Console to check the configuration information and test the connection for DB2 JDBC:

a. Go to Login → Resources → JDBC Providers → and look for the link with the name of the database. The center pane displays the configured JDBC provider as seen in Figure 3-5.

b. Click the link and verify the name and properties of the database.

c. Test the connection to verify that the application can access the data source.

d. Ensure that you have the right data source name and properties configured in the Administrative Console.

e. Also check data source names in the resources.xml configuration file.

IMS and CICS (two EIS products) also produce their own traces and logs. These subsystems very likely run in their own LPARs. Contact the administrator to get the traces and logs required for further analysis or ask them for help.

If your application uses EIS resource adaptors:

a. Contact the EIS administrator to verify that the subsystem is up and available because there is no direct way to test the connection to an EIS resource from the WebSphere Administrative Console.

b. Go to the Administrative Console and select Login → Resources → Resource Adapters.

c. Drill down on the resource name link as you would with the JDBC providers. Verify the configuration properties (given by the administrator or developer of the application) such as spelling of resource names, class path information for libraries, and security information.

3.3 Problem avoidance

Having good standards in development and sound implementation methodology reduce the amount of time and resources being spent on problem resolution.

A good problem management system is indispensable because it can be used:

- To prevent problems from being introduced into steady state
- To feed solutions to problem found in steady state back into the repository of test cases.
It is outside the scope of this book to discuss design, build, and test philosophies in details. Instead, we briefly highlight best practices in the activities of building your system and controlling its deployment. Your problem domain is a function of your design. Poor design leads to poor usability. However, knowing the principles to pay attention to in the construction phase of a system can help prevent poor design.

3.3.1 Designing, coding, and testing

Designing, coding, and testing best practices that can help you avoid problems are:

- Understanding the requirements
- Spending time on design

**Understanding the requirements**

The better you understand the requirements of a solution, the better your design will be. That helps you trap and plan for conditions that are not considered normal operating procedures. By foreseeing these abnormal conditions, you can design better error and warning mechanisms that help speed the debugging process. The mechanisms can range from meaningful messaging and to even possibly and programmatically recovering from these conditions.

**Spending time on design**

You should spend time at the beginning of the design phase to come up with good design, build solid and comprehensive test cases, and follow them through. Proper testing is better than quantity testing, and, of course, proper design is always better than proper debugging.

3.3.2 Change control

Good change control management can help you avoid problems. You should have robust change control procedures and enforce them. A robust change control process can only be as good as its implementation. Do not take shortcuts when implementing changes no matter how small a change it is.

Always have a log of the changes that are going into the production systems. Hold regular audits of changes and verify all changes. If a change was implemented on a certain date, then follow up with its result. Did it or did it not meet the requirements?

Keeping access logs to systems also helps identify unauthorized accesses, security problems and so forth.

**Note:** Keeping change logs is a best practice, but do not use them as the sole source for identifying or considering where the root cause of a problem is. You must consider the problem from all aspects.

More information can be found at:

3.4 Related references

For more information about addressing the problems that might occur when you are running an application, see the following Redpieces:

- *Problem Determination Methodology for WebSphere on z/OS*, REDP-6001, at:
  

- *WebSphere for z/OS Problem Determination Means and Tools*, REDP-6880, at:
  

For additional learning and in-depth treatment of WebSphere components and related topics, consult the Information Center at:

Phase 4: System run time

In this chapter, we present an overview of the various system components and subsystems that make up a typical system run time for WebSphere for z/OS V6. WebSphere for z/OS V6 is fairly similar to WebSphere for z/OS V5 as far as the system run time goes. So, if you have a good understanding of the system run time for WebSphere for z/OS V5 you are well positioned for determining and solving problems in this phase.

We look at some key components and the underlying environment providing the structure for running applications. We then give a brief description of common processes along with some general sources of problems.

We also provide hints and tips that can help you avoid problems in checklist. In the last sections of this chapter, we provide a list of a few common problems that are related to the system run time and to the applications deployed to this runtime environment.

We assume that the user application has been thoroughly tested either in the WebSphere test environment of the IDE or in WebSphere V6 on the distributed platform. We also assume that the application has been deployed without problems to the WebSphere for z/OS V6 environment.

**Note:** We assume WebSphere for z/OS V6 has been installed and configured correctly and also that the latest maintenance has been applied. We focus on problems you might encounter in the WebSphere for z/OS V6 runtime servers when running your applications.
4.1 The WebSphere for z/OS V6 runtime environment

WebSphere for z/OS V6 is a runtime environment implementation for the J2EE 1.4 application programming model. It also supports the J2EE 1.3 and J2EE 1.2 application programming models.

z/OS (or z/OS.e) Version 1 Release 4 is the prerequisite minimum release level for running WebSphere for z/OS V6. WebSphere for z/OS V6 interacts with a number of components of the z/OS operating system in ways that are similar to those of the previous versions of WebSphere for z/OS. The following z/OS components are prerequisites for WebSphere for z/OS V6:

- z/OS Communication Server (TCP/IP)
- z/OS USS and HFS
- Security Server (like RACF or another SAF-compliant security product)
- System Logger
- WLM (in goal mode)
- Resource Recovery Services (RRS)
- System SSL required when using SSL

Additional optional components can be important parts of the complete runtime implementation necessary for supporting your application environment. Such optional components typically include:

- IBM HTTP Server for z/OS
- Lightweight Directory Access Protocol (LDAP)
- DB2 UDB for z/OS Version 7 or higher
- CICS TS V1.3 or higher
- WebSphere MQSeries® for z/OS V5.3.1 or higher
- WebSphere MQ Integrator Broker for z/OS V2.1 or higher

JDK 1.4 is now included with the WebSphere for z/OS V6 code base. This is different from previous versions of WebSphere for z/OS where you had to obtain and install the JDK separately. You no longer have to point to a separate copy and the JDK is now part of the WebSphere for z/OS maintenance stream.

A new messaging engine is included in the WebSphere for z/OS V6 runtime environment to provide a pure Java JMS provider. This function adds a new address space to the WebSphere for z/OS V6 runtime environment. This address space is called a control region adjunct (CRA) server. The CRA address space is started by WLM only when you are using messaging functionality. This new function helps avoid the possible installation of another product with its own prerequisites. The JMS messaging engine runs completely inside the JVM of the CRA.

Another new function in WebSphere for z/OS V6 is the option of administering your Web Server from the Administrative Console. From the Administrative Console, you can:

- View the Web server logs
- View and edit the Web server configuration file (httpd.conf)
- Start and stop the Web server
- Automatically build the plug-in configuration file (plugin-cfg.xml)

Figure 4-1 on page 35 shows the WebSphere for z/OS V6 runtime environment, including the underlying z/OS infrastructure and also some optional z/OS subsystems. Also refer to Chapter 1, “Phase 1: Installation, configuration, and migration” on page 1 for additional information about this structure.
Figure 4-1 shows the network deployment configuration, which was first introduced in WebSphere for z/OS V5. A network deployment configuration consists of

- A deployment manager for the cell
- A node agent for each node in the cell
- A number of servers in each node
- One daemon server per cell per z/OS image

The network deployment configuration is slightly more advanced in WebSphere for z/OS V6 because it might have additional address spaces. However, it is still very similar to the network deployment configuration in WebSphere for z/OS V5, especially if you are not using messaging.

“IBM HS” in Figure 4-1 stands for IBM HTTP Server for z/OS, which is an optional Web server if you are not using the HTTP Server embedded in the control region (which is indicated as CR in the figure). In Figure 4-2 on page 36, we also show the Web server for z/OS among the optional z/OS subsystems. This implementation assumes that we have configured and are using the IBM HTTP Server with the WebSphere Application Server Plug-in.
WebSphere for z/OS V6 also has a base application server configuration. It is called a stand-alone application server configuration because that is what it is. You can no longer add additional servers to the node like you did with WebSphere for z/OS V5. As shown in Figure 4-2, the stand-alone application server configuration is a node and a cell and it has a daemon server. The rule is one daemon per cell per z/OS image.

In WebSphere for z/OS V6, the stand-alone application server configuration supports multiple servant regions (SR in the illustration). This was not supported in WebSphere for z/OS V5. You could only have one servant region in the WebSphere for z/OS V5 base application server configuration.

4.2 Problem categories in runtime phase

Problems in this runtime environment can occur in many places. Problem determination is by no means a simple task in such an environment. We have categorized the problem areas as:

- z/OS infrastructure problems
  
The WebSphere for z/OS V6 runtime environment depends on the prerequisite z/OS components. Conscientious planning is necessary. Include experts from various areas like RACF (or another SAF-compliant security product), TCP/IP, and WLM, in your WebSphere for z/OS V6 administration team. A healthy z/OS operating system and subsystem infrastructure is necessary for WebSphere for z/OS V6 to run successfully. This helps avoid time consuming problem determination sessions later in your administration process.

- Problems related to optional subsystems
  
  One or more optional subsystems might be present in your total runtime environment. Most likely, some of your user applications that are deployed to WebSphere for z/OS V6...
are going to be accessing one or more of these subsystems. These subsystems can be in
the same LPAR as your WebSphere for z/OS V6 run time as indicated in Figure 4-1 on
page 35, but this is not required. They can even be on a separate server such as an LPAR
on a different zSeries platform. These subsystems (for example, CICS TS, DB2 UDB for
z/OS, or IMS) normally provide vital back-end functionality for your user applications.
When WebSphere MQ is involved, this subsystem might provide back-end as well as
front-end functionality to your applications.

Problems with connecting to the back-end subsystems usually create an application
timeout or some incorrect output from the application. Problems in the subsystems
themselves often impact the response times or the performance of your WebSphere for
z/OS V6 user application. In many of these cases, the WebSphere for z/OS V6 system
administrator works with his subsystem experts to resolve these types of problems.

WebSphere for z/OS environment problems

Having very briefly covered the z/OS basic infrastructure and the optional z/OS
subsystems, the WebSphere for z/OS environment basically is comprised of an
administrative part and the application server runtime part:

– The administrative part includes several tools like the Administrative Console, the
WebSphere administrative scripting program (wsadmin), administrative commands,
and possible administrative programs. Refer to the WebSphere for z/OS Information
Center for further information and description. In addition to the tools, the deployment
managers and the node agents are administrative components.

For the administrative part to work without problems, these tools must all be installed
correctly and the components must be configured correctly and up and running. For
example, in a network deployment configuration, the Administrative Console
application, the deployment manager, the node agent, and the configuration repository
(HFS or zFS) must all be present and working correctly for your administration to be
carried out. Any mismatch between the tools and the components can result in your
administration failing to work.

– The application server runtime part (either a stand-alone configuration or a network
deployment configuration) consists of, at minimum, a daemon, a control region, an
optional CRA, and one or more servant regions for each control region. The control
region, CRA, and the servant regions comprise a server. Inside each servant region is
a JVM where the J2EE applications are executed. Refer to Figure 4-1 on page 35 and
Figure 4-2 on page 36.

When analyzing problems in the WebSphere for z/OS V6 application server runtime
part, it is important to understand the functions of each component. It is also very
important to understand how the components interact with each other. Problems
arising in the runtime part should be approached in the standard way:

• What is the symptom?
• When does the problem occur?
• Has anything been changed in the runtime part recently?
• Has a new version of the application been deployed recently?
• Look at logs starting with the most likely component
• Look at the error log, look at SYSOUT, and look at SYSPRINT

4.3, “Understanding your own runtime configuration” on page 38 describes this
process in more detail.

For information about troubleshooting, search for “Troubleshooting and support” in the
WebSphere for z/OS Information Center at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp
4.3 Understanding your own runtime configuration

To be effective at problem determination, you need a good understanding of your current WebSphere for z/OS V6 runtime environment. Specifically, you need to understand:

- What system configuration files exist and what are their content?
- What logs and error logs are available?
- What variables and properties are available and what values can they take?
- What traces and trace options are available?
- What tools are appropriate to use for my current environment and symptoms?
- Which functions, administration, and problem determination tasks can I perform with the Administrative Console?

A good start is to take a close look at the job log for all your control and servant regions. Specifically, when these regions start, you can learn quite a lot about your configuration by examining the messages being issued.

Each WebSphere for z/OS V6 server has a configuration file called was.env. This file is a parameter file that contains important configuration settings. Each time you start your runtime environment servers, you must point to this file. The file allows the non-Java portion of the server to open and it is located far down the HFS directory tree.

If you are planning to enable global security in your runtime environment, it is necessary to have a good understanding of the implications. Refer to WebSphere Application Server for z/OS V5 and J2EE 1.3 Security Handbook, SG24-6086. This relatively large publication contains a lot of good information. With your security administrator, review the subjects that applies to your environment. Troubleshooting security issues is no different from troubleshooting other kind of problems, but sometimes it is harder to identify the cause of the problem.

4.4 Troubleshooting tips for the runtime environment

When you are troubleshooting problems in a runtime environment that previously ran without any error symptoms, it is important to use a structured approach.

The most common sources of problems in this environment include:

- Recent changes
  - Was any z/OS maintenance performed lately?
  - Was any maintenance to any of the z/OS subsystems performed?
  - Were any new versions of the production applications newly deployed?
  - Was any maintenance performed for WebSphere for z/OS V6 lately?
- Recent operational activities
  - Are you running any diagnostic tools or traces?
  - Have you activated any performance monitors?
  - Have you restarted any of the z/OS components or subsystems?
- Changes in the volume of traffic
  - Has number of WebSphere for z/OS V6 transactions increased?
  - Has the resource utilization increased?
  - Is there a lack of processor or memory capacity for the WebSphere for z/OS V6 LPAR?
Following such an approach leads you to the most likely cause of the problem in the shortest amount of time.

The WebSphere for z/OS V6 runtime environment differs from installation to installation. However, the initial troubleshooting process is almost identical and the authors recommend the following simple approach when you have identified a symptom:

1. Review the SYSLOG, the job log for the CR and the SR and the WebSphere for z/OS V6 error logs. Identify error messages.
2. If you can associate an error message directly with your initial symptom, take action as indicated by the error message.
3. If the error message cannot be directly associated with the symptom but requires you to obtain more diagnostic information, follow the indication in the error message.

If you still cannot identify the error messages that are associated with your symptom, the authors recommend that you use the flow charts for the symptoms that we have identified in the WebSphere for z/OS problem symptoms and analysis, REDP-6002, to find the root cause of the problem and solve it.

4.5 Security issues and problems

Security issues and problems are normally encountered and sorted out during the installation and customization phase, during the application deployment phase, or during the run application phase. However, problems can appear later during run time because of new functions that are being exercised or defects that were not discovered during the application testing period.

Troubleshooting security issues involves looking for symptoms as with any other troubleshooting activity. Look for symptoms in messages, in control region and servant region job logs and in the SYSLOG. If you cannot find any symptoms, turn on tracing and search the traces for symptoms. Additionally, you can use dumps. Security problems usually carry one or more of these symptoms:

- Security violation
- Access denied
- Permission bits problems
- Invalid user ID or password
- Not enough authority for the user ID

With your security administrator, you should review security logs and security traces. Sometimes a dump of the security system is also necessary. Refer to WebSphere Application Server for z/OS V5 and J2EE 1.3 Security Handbook, SG24-6086, for more information.

4.6 Problem avoidance checklist

The WebSphere for z/OS V6 runtime environment and configuration differs from installation to installation. It is therefore not possible to create a checklist that satisfies every configuration option. However, this check lists offers the following common and important problem sources:

- Software maintenance
  Establish a procedure for applying software maintenance to your system if you do not already have one. This procedure involves testing the maintenance that is being introduced on a test system image prior to putting it on the production system. The procedure must include the base z/OS system, its components, and all z/OS subsystems,
as well as the WebSphere for z/OS V6 product. Although many problems are caused by software components other than WebSphere for z/OS V6, many product defects are also found and maintenance for those defects is made available. As a standard procedure verify that your current runtime environment satisfies all the hardware and software prerequisites for the maintenance to be performed. Also verify that, when you perform maintenance for WebSphere for z/OS V6, you do not introduce any inconsistency between the z/OS load libraries and the HFS.

- **Change management**
  In the same context, establish a total change management procedure for your environment. This procedure should outline the proper processes for making any kind of changes to your environment. This prevents problems that occur as a result of one group or person in the organization introducing changes to the environment that others were not aware of. With a procedure to eliminate these occurrences, the chances of unexpected problems is reduced.

- **The test environment**
  Ideally, the test environment should be configured exactly like the production environment. However, this is sometimes not possible. When performing tests, the software maintenance level should be identical to your production environment. If the test environment and the production environment are kept in sync, you can easily do problem determination in the test environment if a problem occurs in the production environment. This is an advantage because it does not impact your production environment.

One of the reasons the authors recommend a testing environment with a configuration identical to the production environment is because it allows you to test performance without affecting your production environment.

- **Application testing**
  The best strategy for preventing problems from occurring in your WebSphere for z/OS V6 production runtime environment is through testing. A detailed testing strategy should be in place for all applications and it should be followed every time a new version of the application is being introduced into the production environment. This procedure must include both functional testing and performance testing. Prior to deploying an application into WebSphere for z/OS V6 runtime environment, the authors recommend that it be thoroughly tested in the WebSphere test environment of the IDE or in WebSphere V6 in a distributed environment.

- **Monitoring**
  A good monitoring strategy can help you identify problems before users experience them. WebSphere for z/OS V6 includes an enhanced Tivoli® Performance Viewer that is accessible from the Administrative Console. To use Tivoli Performance Viewer, you need to enable the Performance Monitoring Infrastructure, log on to the Administrative Console, select your application server, and select the PMI link. TPV can give the administrator clues about possible performance bottlenecks.

  It is also wise to monitor the WebSphere for z/OS V6 application server logs. The SystemOut and SystemErr logs for each application server should be monitored. Informational messages, warning messages, and error messages are in these logs.

- **Operational procedures**
  To avoid or reduce operational issues, make sure that you created and documented your own startup and shutdown procedure. They must clearly describe the sequence and commands for all components and subsystems. This can serve as a requirement specification for the automation of the operational process. Ideally, your operation should be fully automated to reduce human error.
Large HFS
Ensure that you allocate a large HFS for the WebSphere for z/OS V6 configuration. Leave a large spare space for your file system. This is always necessary to allow room for adding new applications, upgrading to new application servers, upgrading to a new WebSphere for z/OS V6 maintenance level with possibly a new version of the Administrative Console application, and so on.

Network deployment configuration
In your network deployment configuration, ensure that the deployment manager and the node agents are running prior to any application deployment or configuration changes. This helps ensure that the changes are effectively synchronized across all nodes.

Attributes and properties verification
For each server, using the Administrative Console or the job logs, verify the following attributes and properties:

- protocol_http_timeout_persistentSession
  Use the default values unless you know that your application requires different values.
- protocol_http_timeout_input
  Use the default values unless you know that your application requires different values.
- protocol_http_timeout_output
  Use the default values unless you know that your application requires different values. This timeout might also cause abend EC3-0413000n in the case of a long-running JSP compilation request.
- transaction_maximumTimeout
  The default is 300 seconds. Check the value and increase it.
- wlm_dynapplenv_single_server=1
  Use this value if you do not want to have multiple servant region instances per application server. Otherwise set it to 0.
- wlm_maximumSRCount=1
  If you need multiple servant region instances to be started, make sure that this is set to a desired value.
- wlm_minimumSRCount=1
  This determines how many servers to start as a minimum.
- ras_trace_defaultTracingLevel=1
  For new applications, the recommendation is 1 and:
  ras_trace_outputLocation=BUFFER SYS PRINT

Correct settings
For each server, use the Administrative Console or the servant job lob to verify that the following attributes and properties are set correctly:

- Verbose Garbage Collection (GC) is enabled.
  Verbose GC for the application server JVM is turned on. Turning on verbose GC will not be too costly and can help to monitor possible Java memory leaks.
- JVM maximum heap size is set.
  You can set this value to 256, -Xmx256m
4.7 Typical problems

In this section, we list a few common problems in the WebSphere for z/OS V6 runtime environment. These problems were discovered by WebSphere for z/OS V6 users and we document the solution for them:

- **Problem**: After applying maintenance to WebSphere for z/OS V6, runtime servers do not start or they abend while running applications.
  - **Solution**: Actually, the symptoms here can vary. We have seen this problem being caused by a mismatch in code levels between the z/OS load libraries and the HFS files. Typically, this happens if maintenance is applied to a service HFS and then later not copied to the production HFS.
    
    This problem can also occur if you do not remove old copies of WebSphere load modules from LPA or you do not refresh LLA.

- **Problem**: After performing maintenance to WebSphere for z/OS V6, applications do not respond as expected. Changes to the plugin-cfg.xml file are not being picked up.
  - **Solution**: This can be caused by using the wrong plugin-cfg.xml file. After maintenance, files might be overwritten or the pointers to these files might have changed. Make sure the correct `plugin-cfg.xml` file is located at the path specified in the `httpd.conf` file.

- **Problem**: Instructions for empty managed node are wrong.
  
  After an attempt to federate an empty managed node, it was discovered that the file ownerships were wrong; they are owned by UID=0. The federate completed rc=0 but there were RACF error messages where the node agent was unable to access parts of the HFS.
  
  It appears that the instructions that were generated for empty managed node creation and federation were wrong, because they indicated that a UID=0 user ID should be used, when in fact, it is likely that the admin user ID should be used, such as when a non-empty node is federated.
  
  - **Solution**: The soap.client.props file for the managed node was not being properly updated with the user-provided values from the Customization Dialog, specifically the section defining the SSL Configuration. The soap.client.props file can be fixed so that it is properly updated with the user-provided values if you apply latest maintenance. APAR PK04464 is associated with Service Level (Fix Pack) 6.0.2.1 (Build Level cf10533.10) of WebSphere Application Server V6.0.1 for z/OS.

- **Problem**: In the Administrative Console on troubleshooting, the users selected Configuration Problems. They found multiple entries that they believed were not caused by configuration work that they had performed and as a result, they thought there might be a problem with the product.
  
  The message was:

  `CHKW2062E: The executable target of a java process definition is absent. cells/CELLDM1T/nodes/NODEB2T/servers/nodeagent/server.xml`

  It occurred multiple times for server.xml documents.

  They also received:
CHKW3706E: Validation of cells/CELLDM1T/nodes/NODEB1T/serverindex.xml failed with exception java.lang.NullPointerException.
cells/CELLDM1T/nodes/NODEB1T/serverindex.xml

They only received it once.

Another message was:

CHKW2130W: The server Network Deployment Server has no configured transaction service. cells/CELLDM1T/nodes/NODEDM1T/servers/dmgr/server.xml

- Solution: All of these CHKW messages are being issued by the configuration validator. The CHKW2062E messages are usually the result of Java process definitions that have blank entries, specifically the executable arguments field in the process definition for the JVM of a server. These messages communicate nothing except that the field is blank. They are benign. There is a good argument for why these messages are being generated, however.

The next message is the CHKW2130W. This is a warning message (note the W suffix on the message code) that appears to check the value of the enable attribute for the TransactionService element in the server.xml. This warning is benign and can safely be ignored.

For the last error, it is difficult to determine what is occurring in that CHKW3706E message. However, this is simply an error in the validation of the WebSphere document repository, and does not necessarily represent any runtime error in WebSphere that would affect operation.

Problem: After they applied the cumulative fix cf20523.06 to their WebSphere for z/OS V6.0.1 system, the users noticed that an old problem regarding SMF, had resurfaced. Separator values were missing in the SMF 120 subtype 5 field, AMCName, that is used to parse the application, module, and class name.

For example, what used to appear as:

“ECperfEAR::OrdersJAR.jar::OrderLineEnt”

now appears as

“ECperfEAROrdersJAR.jarOrderLineEnt”

The APAR to fix the old issue was PQ85314.

- Solution: In WebSphere Application Server V4.0.1 for z/OS and OS/390, the Application-Module-Component (AMC) name was represented in SMF as the string “ApplicationName::ModuleName::ComponentName” where each sub-field was separated by “::”. However in version 5, the AMC name returned with SMF recording is “ApplicationNameModuleNameComponentName” where there is no separation of each sub-field.

The Application-Module-Component value that is returned by SMF should follow this format: “ApplicationName::ModuleName::ComponentName” and can now be tokened with “::” separating each sub-field. This value is now compatible with WebSphere Application Server V4.0.1 for z/OS and OS/390 SMF tooling. Apply the latest maintenance. APAR PK07137 is associated with Service Level (Fix Pack) 6.0.2.1 (Build Level cf10533.10) of WebSphere Application Server V6.0.1 for z/OS.

- Tools used: SMF
Index

A
ABEND EC3  7
ABEND SCC3  7
ABEND SEC3  8, 16
APAR  2, 7
application
  environment
  dynamic  2, 6
ASTK  15
Automatic Restart Management  3

B
buffer
  exceeded  16

C
checklist
  migration  5
coexistence  5, 8
problems  7
configuration
  change root directory  7
problems  3, 7
control region
  abend  8
migration  6
CTRACE
  setup  6

D
daemon
  group  8
  port collision  5
data set
  format  7
  permission  2
DB2
  hints  4
deployment
  application  5
  phase  14
deployment manager  19
  configuration  3
  version  8
DFSMS ACS  7
dump
  data set  3
  resize  3
duplicate login configuration name  8
DVIPA  3

E
EAR file  15
error
  log  6

H
HFS
  mount  4
  permission  2, 4
  plan  3
  shared  2
Hierarchical File Structure (HFS)  2

I
Installation
  plan  2
problems  3, 7
IP address
  dynamic virtual  3
ISPF
  reference  3

J
J2EE
  migration  5
Java
  heap  14
JCL  4

L
large EAR file  16
LPA  5, 7

M
migration  6
  checklist  5
problems  4, 7
Version 3.5 SE to 5.1  5
Version 4.0.1 to 5.1  5
Version 5.0 to 5.1  4

N
naming
  conventions  2

P
port
  conflict  5, 7
problem  33
production environment  9
**R**
- RACF 2
- Redbooks Web site
  - Contact us ix

**S**
- security
  - administrator 2
- SMP/E
  - problems 7
  - references 3
- STEPLIB 6–7
- Sysplex Distributor 3

**T**
- TCP/IP
  - DVIPA 3
  - references 3
  - timeout 16

**U**
- USS
  - references 3

**V**
- version 8

**W**
- WebSphere
  - Proof of Concept for z/OS 9
- WLM
  - dynamic 2
  - static 2
  - worksheet 3
  - wsdeploy tool 15
  - wsjaas.con 8
Problem Avoidance for WebSphere Application Server for z/OS

This Redpaper is intended for system programmers and administrators who need an efficient way to identify problems, analyze them, and fix them so that they can deliver good support for the WebSphere Application Server for z/OS environment. IBM support and technical professionals and Java developers who work in the WebSphere for z/OS environment are also part of this targeted audience.

We identified possible problem areas that can be encountered in WebSphere for z/OS and arranged them into four specific problem phases corresponding to steps in the WebSphere for z/OS life cycle:

Phase 1: Installation, configuration and migration
Phase 2: Application deployment
Phase 3: Testing and running applications
Phase 4: System environment (production)

We give a general overview of the problem areas in the individual phases, explain how to analyze them, and provide valuable hints and tips for avoiding them.