

# WebSphere Application Server V6

## Problem Determination for Distributed Platforms

Learn how to approach the problem

Collect and analyze the data

Find a resolution



Carla Sadtler  
Simon Davitt  
Rana Katikitala  
Thu-Giang Pham  
Craig Scott  
David Titzler  
Hari Venkateshaiah  
Javier Voos





International Technical Support Organization

**WebSphere Application Server V6 Problem  
Determination for Distributed Platforms**

November 2005

Archived

**Note:** Before using this information and the product it supports, read the information in “Notices” on page ix.

### **First Edition (November 2005)**

This edition applies to WebSphere Application Server V6 on distributed platforms.

**© Copyright International Business Machines Corporation 2005. All rights reserved.**

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Notices</b> .....	ix
Trademarks .....	x
 <b>Preface</b> .....	xi
The team that wrote this redbook .....	xi
Become a published author .....	xiv
Comments welcome .....	xiv
 <b>Chapter 1. Approach to problem determination</b> .....	1
1.1 Introduction to problem determination .....	2
1.1.1 Causes of problems .....	3
1.1.2 Types of problem symptoms .....	3
1.2 Preparing for and preventing problems .....	4
1.2.1 Applying WebSphere maintenance .....	4
1.2.2 Checking the prerequisites .....	7
1.2.3 Testing the application .....	7
1.2.4 Setting up a test environment .....	10
1.2.5 Establishing safe operational procedures .....	12
1.2.6 High availability and failover .....	14
1.2.7 Monitoring .....	15
1.2.8 System documentation .....	19
1.2.9 Diagnostic data collection .....	21
1.3 What to do when a problem occurs .....	24
1.3.1 Revert to safe conditions .....	24
1.3.2 Identify problem symptoms .....	25
1.3.3 Investigate and research the problem .....	27
1.3.4 Problem determination strategies .....	29
1.4 Classify the problem and determine the root cause .....	31
1.4.1 Installation or migration .....	34
1.4.2 <b>Application packaging and deployment</b> .....	<b>35</b>
1.4.3 System management and configuration .....	37
1.4.4 Runtime .....	41
1.5 Contacting IBM for support .....	70
1.5.1 IBM support structure .....	70
1.5.2 Research the problem .....	72
1.5.3 Collect MustGather files .....	72
1.5.4 Determine the severity .....	73
1.5.5 Create a PMR .....	74

1.5.6 Send data to IBM	76
<b>Chapter 2. Diagnostic data</b>	77
2.1 JVM logs (SystemOut and SystemErr)	78
2.2 Tracing	82
2.3 Collector tool	87
2.4 First Failure Data Capture (FFDC)	87
2.5 Other logs	88
2.5.1 Process (native) logs	88
2.5.2 Service log (activity.log)	89
2.5.3 Installation logs	90
2.6 IBM HTTP Server and plug-in logs and traces	90
2.6.1 IBM HTTP Server logs	90
2.6.2 Web server plug-in logs	90
2.6.3 Web server plug-in trace	91
2.6.4 Network trace	91
2.7 System management logs	92
2.7.1 Output from wsadmin	92
2.7.2 Management scripts	92
2.7.3 Profile management logs	92
2.8 WebSphere Rapid Deployment logs	93
2.9 Summary of logs	93
<b>Chapter 3. Installation problem determination</b>	97
3.1 Introduction	98
3.2 Work the problem	100
3.2.1 Symptom: Launchpad or installation wizard will not start or fails	101
3.2.2 Symptom: Installation wizard hangs	103
3.2.3 Symptom: Profile creation failure	105
3.2.4 Symptom: IVT fails	108
3.3 Analyzing problem areas	110
3.3.1 Web browser requirements	110
3.3.2 Application server startup problems	111
3.3.3 Profile creation problems	112
3.4 The next step	113
<b>Chapter 4. System management problem determination</b>	115
4.1 Introduction	116
4.1.1 Collecting data	122
4.2 Work the problem	125
4.2.1 High-level symptom analysis	125
4.3 Analyzing problem areas	127
4.3.1 Problem: Unable to access the administrative console	127
4.3.2 Problem: wsadmin or management scripts can't access server	130

4.3.3 Problem: Unable to stop a server process .....	133
4.3.4 Problem: Unable to start a server process .....	135
4.3.5 Problem: Unable to access a node agent .....	137
4.3.6 Problem: Unable to manage a Web server .....	138
4.3.7 Problem: Unable to manage applications .....	141
4.3.8 Problem: Failure adding a node to a deployment manager .....	143
4.3.9 Problem: Repository synchronization .....	145
4.3.10 Problem: Save conflicts in the administrative console .....	150
4.3.11 Problem: enterprise applications missing .....	151
4.3.12 Problem: Invalid or expired certificates .....	153
4.3.13 Problem: WebSphere Rapid Deployment .....	158
4.4 The next step .....	162
<b>Chapter 5. Web container problem determination .....</b>	<b>165</b>
5.1 Introduction .....	166
5.1.1 Initial symptoms .....	168
5.2 Work the problem .....	169
5.2.1 Data to collect .....	169
5.2.2 High-level symptom analysis .....	170
5.2.3 Symptom: HTTP 404 error - The page cannot be displayed .....	172
5.2.4 Symptom: HTTP 404 error - Failed to find resource .....	175
5.2.5 Symptom: HTTP 404 error - WebGroup/virtual host not defined ..	177
5.2.6 Symptom: HTTP 500 error - JSP processing error .....	179
5.2.7 Symptom: HTTP 500 error - IllegalStateException .....	181
5.3 Analyzing problem areas .....	186
5.3.1 Application URL specification .....	186
5.3.2 Static resources not displayed .....	190
5.3.3 Web resources not reloading .....	192
5.3.4 Encoding and internationalization issues .....	195
5.3.5 HTTP session management .....	202
5.4 The next step .....	209
<b>Chapter 6. JCA connection problem determination .....</b>	<b>211</b>
6.1 Introduction .....	212
6.1.1 JCA technical overview .....	213
6.2 Work the problem .....	216
6.2.1 Symptom: A JDBC call returns incorrect data .....	217
6.2.2 Symptom: Failure to connect to a new data source .....	222
6.2.3 Symptom: Failure to connect to an existing data source .....	224
6.2.4 Symptom: Failure to access a resource through JDBC .....	225
6.2.5 Symptom: Failure to access a non-relational resource .....	226
6.3 Analyzing problem areas .....	227
6.3.1 Configuration problems .....	228

6.3.2	Naming problems	234
6.3.3	Authentication problems	235
6.3.4	Connection leak problems	236
6.3.5	Stale connection problems	243
6.3.6	SQLException and database errors	245
6.3.7	XAResource problems	246
6.3.8	Connection factory problems	247
6.4	The next step	249
<b>Chapter 7.</b>	<b>Web server plug-in problem determination</b>	<b>251</b>
7.1	Introduction	252
7.2	Work the problem	256
7.2.1	Collect the data	257
7.2.2	Analyze the high-level symptoms	259
7.3	Analyzing problem areas	261
7.3.1	Problem: Web server will not start	261
7.3.2	Problem: Failure between the Web server and plug-in	263
7.3.3	Problem: Sessions are being lost	272
7.3.4	Problem: The application works intermittently	277
7.3.5	Problem: Application load is not being evenly distributed	280
7.4	The next step	283
<b>Chapter 8.</b>	<b>Application server crash problem determination</b>	<b>285</b>
8.1	Introduction	286
8.2	Work the problem	286
8.2.1	High-level symptom analysis	287
8.2.2	Data to collect	288
8.2.3	Analyze the data	290
8.2.4	Analyze the javacore file	291
8.2.5	Finding a workaround for JIT problems	293
8.3	Analyzing problem areas	295
8.3.1	Stack overflow	295
8.3.2	Out of memory error	298
8.4	The next step	299
8.4.1	Sun Solaris	300
8.4.2	HP-UX	300
<b>Chapter 9.</b>	<b>Default messaging provider problem determination</b>	<b>301</b>
9.1	Introduction	302
9.1.1	Problem categories	305
9.2	Work the problem	305
9.2.1	Collect the data	306
9.2.2	Analyze the high-level symptoms	307
9.2.3	What to do if your symptom is not listed here	310



9.3 Analyzing problem areas . . . . .	310
9.3.1 Messaging engine startup problems . . . . .	310
9.3.2 Message flow problems . . . . .	324
9.3.3 Application configuration and resource problems . . . . .	344
9.3.4 Product errors . . . . .	349
9.4 The next step. . . . .	350
<b>Related publications</b> . . . . .	353
IBM Redbooks . . . . .	353
Online resources . . . . .	353
How to get IBM Redbooks . . . . .	356
Help from IBM . . . . .	356
<b>Index</b> . . . . .	357



# 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:

@server®

@server®

Redbooks (logo) ™

developerWorks®

z/OS®

AIX®

ClearCase®

Cloudscape™

CICS®

Domino®

DB2 Connect™

DB2®

Informix®

IBM®

MQSeries®

Netfinity Manager™

Netfinity®

OS/2®

OS/390®

OS/400®

Rational®

Redbooks™

SecureWay®

Tivoli®

WebSphere®

The following terms are trademarks of other companies:

iPlanet, Enterprise JavaBeans, EJB, Java, Java Naming and Directory Interface, JavaBeans, JavaScript, JavaServer, JavaServer Pages, JDBC, JDK, JMX, JSP, JVM, J2EE, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Win32, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

# Preface

This IBM® Redbook is intended to help customers through the process of identifying and resolving problems in WebSphere Application Server V6 on distributed platforms. Previously published as a series of IBM Redpapers, each chapter addresses a specific aspect of the problem determination process. Links to the original papers are provided throughout the book.

The first paper, *Approach to Problem Determination*, should be the first stop for IBM WebSphere® Application Server system managers. It provides information that is useful in preparing for and preventing problems. It then details what actions to take when a problem occurs and helps you define the problem. Once the primary symptoms are identified, it helps you narrow down the problem to terms that are similar to those used by the IBM support team, and leads you to more specific information that can help you pinpoint the problem.

The subsequent papers address selected topics that have been identified as some of the more common problem types that customers usually need to address.

## The team that wrote this redbook

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

**Carla Sadtler** is a certified IT Specialist at the ITSO, Raleigh Center. She writes extensively about the WebSphere and Patterns for e-business areas. Before joining the ITSO in 1985, Carla worked in the Raleigh branch office as a Program Support Representative. She holds a degree in mathematics from the University of North Carolina at Greensboro.

**Simon Davitt** works in the WebSphere MQ Java™ Messaging level 3 service. Previously he was an independent/free-lance programmer in the UK. He has over 20 years experience working in the IT field. Has been permanently employed by IBM since 2000 working in the WebSphere MQSeries® service area specializing in Java and JMS.

**Rana Katikitala** is an Advisory Software Specialist for Tivoli® Security in IBM Software Labs, India. He has over seven years of experience in IT industry in the areas of development, support and test of operating systems, systems management software and e-business solutions. He holds a masters degree in

Structural Engineering from Regional Engineering College (REC) Warangal, India. His areas of expertise include IBM OS/2®, Windows® 2K, Netfinity® Manager™, IBM Director, Healthcare domain solutions of HIPAA(Health Insurance Portability and Accountability Act) and HCN(Healthcare Collaborative Network).

**Thu-Giang Pham** Thu-Giang Pham is an Advisory Software Engineer working as a team leader for IBM WebSphere Application Server L2 support group in RTP, North Carolina.

Thu-Giang has more than seventeen years of experience in the IT industry where she spent the last thirteen years providing technical support and consulting in the fields of databases, development tools and application servers. Thu-Giang also had four years of experience as a software engineer involved in the development of an application code generator and various financial applications. Before joining IBM in RTP five years ago, Thu-Giang worked for DataDirect Technology (formerly known as MERANT, Intersolv), Oracle and Software AG of Canada. Thu-Giang holds a Bachelor of Mathematics degree, joint honors in the fields of Computer Science, Combinatorics and Optimization from the University of Waterloo in Waterloo, Ontario, Canada.

**Craig Scott** is a Software Support specialist in Australia. He has 16 years of experience in IT and over five years experience in WebSphere Application Server. He holds a degree in Computer Science from the University of Canberra. His areas of expertise includes WebSphere Application Server, WebSphere Edge Server, IBM DB2® and IBM Content Manager.

**David Titzler** is a Level 2 Support analyst for WebSphere Application Server based in Raleigh, North Carolina. He joined IBM in 2001 and has been in his current position for four years. In this position, he works directly with clients to resolve WebSphere Application Server problems. He specializes in problems relating to connection pooling, messaging, EJBs, transactions, Web services, and more. He is an IBM Certified Advanced System Administrator for WebSphere Application Server V5. He holds a degree in Computer Science from the Pennsylvania State University.

**Hari Venkateshaiah** is a Level 3 support for JIT and XE in India. He has four years of experience in the JVM™ and JIT areas. He holds a degree in Master of Computer Application from Bangalore University, India. His areas of expertise include Softswitch, VOIP, IPV6, JVM, Advanced Compiler. He has written extensively on application server crash problem determination.

**Javier Voos** is an IBM Certified Advanced System Administrator working for IBM Software Technical Support in Spanish South American region. He has twelve years of experience in the application development field. He has been working extensively with WebSphere Application Server and tools for the last six years,

helping customers to design, implement, and tune J2EE™ applications. For the past four years, he has been working as project leader for a J2EE development framework based on J2EE patterns in the Clinical Engineering R&D Center, Argentina. He holds a degree in Information Technology Engineering from the U.T.N. Córdoba University, Argentina. His areas of expertise include J2EE architecture design, WebSphere Application Server consulting services and legacy mainframe systems integration.

With special thanks to Ron Verbruggen for his guidance in designing and developing the content of this book.

Thanks to the following people for their contributions to this project:

Gustavo Bustos  
Grupo Leviminond

Thanks to the following IBMers:

Rama Boggarapu  
Rob Boretti  
Chris Clarke  
Matt Davis  
Sarah Eggleston  
James Fox  
Justin T Fries  
Jonathan Gibbs  
Jinwoo Hwang  
Daniel Julin  
Venkataraghavan Lakshminarayanachar  
Ricky Marley  
Rich Montjoy IV  
Federico Nachtigal  
Juan Pablo Napoli  
Rajeev Palanki  
Harry Rumley  
Gautam Singh

## 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](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

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

- Use the online **Contact us** review redbook form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- Send your comments in an email to:

[redbook@us.ibm.com](mailto:redbook@us.ibm.com)

- Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HZ8 Building 662  
P.O. Box 12195  
Research Triangle Park, NC 27709-2195



# Approach to problem determination

This paper introduces problem determination strategies for WebSphere Application Server V6. It discusses how to prevent problems, how to plan and prepare for problems that can occur, and what to do when a problem does occur so that it is resolved as quickly as possible. It then guides you to the more detailed information that can help you diagnose the cause of the type of problem that you are experiencing.

# 1.1 Introduction to problem determination

Keeping your enterprise applications highly available to your customers is crucial in today's on demand business environment. WebSphere Application Server V6 has many new features and tools that are designed to minimize problem occurrences. However, if a problem does occur that might negatively impact your business, you need to be able to respond quickly and effectively.

We have prepared a series of papers that provide guidelines for specific issues that you might experience with WebSphere Application Server V6. The goal of these papers is to help you form strategies to quickly identify the root cause of problems that do occur to minimize their business impact. You can find these papers, which may also be downloaded as a book, at:

<http://www.redbooks.ibm.com/abstracts/sg246798.html>

We show you the path to get from an initial problem symptom (for example, "Users cannot log onto our Web site!") to the root cause of the problem (for example, "Our newly installed application code caused a deadlock."). Although it is impossible to show a resolution for every conceivable problem that might occur, we outline a general problem determination process that you can use to discover the root cause of a problem.

**Note:** The methodologies discussed in this book will not necessarily provide a resolution to the problem. However, when you have determined the root cause, you will understand what needs to be changed or fixed to resolve the problem.

We assume that readers of this book already have a basic understanding of and experience with WebSphere Application Server V6. We focus specifically on problem determination. If you need more general information about using the product, consult the following material:

► IBM Training Courses

<http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=page&contentID=a0000048>

► IBM WebSphere Training and Certification

<http://www-306.ibm.com/software/inf01/websphere/index.jsp?tab=education/index>

► WebSphere Information Center

<http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp>

► IBM Education Assistant

<http://www-306.ibm.com/software/info/education/assistant>

- IBM Education Assistant: WebSphere Application Server problem determination

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27005460>

### 1.1.1 Causes of problems

When a problem occurs, your first inclination might be to call IBM Support so that they can provide a fix that resolves the problem. In some cases, contacting IBM is necessary. However, the experience of the WebSphere Application Server Support team has shown that a small percentage of client-reported problems are actually due to defects with WebSphere Application Server code. Most problems are caused by configuration issues, environment issues, application code defects, or a misunderstanding of WebSphere Application Server. Many of these problems can be resolved easily without calling IBM Support to open a problem management record (PMR). In addition, many issues can be resolved by following the problem determination procedures that are discussed in this book.

Each problem that you encounter has a different level of complexity and a different level of impact on your business. These factors determine how closely you follow the procedures in this book. For less complex problems, it might only be necessary to follow the most basic procedures. More complex problems can involve multiple components and maybe even multiple software products and systems. These problems require more time and effort with more thorough problem determination techniques. Obviously, the impact that the problem has on your business influences the urgency to resolve the problem and that also determines which problem determination techniques you follow.

### 1.1.2 Types of problem symptoms

When a user of an application that is running on WebSphere Application Server first notices a problem, a problem symptom is observed. Sometimes the problem symptom provides clues about the cause of the problem. Other times, a significant amount of problem determination is needed to determine the problem's root cause.

Here are the common types of symptoms that you might see. Almost every symptom falls into one of these categories:

- You cannot install or migrate WebSphere Application Server or install an application into WebSphere Application Server.
- You experience difficulties in WebSphere Application Server system management or configuration.
- An application or WebSphere Application Server process (for example, an application server, node agent, or deployment manager) is unable to start.

- ▶ An application does not respond to incoming requests.
- ▶ An application produces unexpected results (possibly errors or exceptions).
- ▶ An application cannot connect to an external system or resource.
- ▶ An application performs slowly or its performance degrades over time.

## 1.2 Preparing for and preventing problems

In any enterprise computing system, you can expect that some problems — large or small — will occur at times. In a best case scenario, the problems that you encounter will not be severe and will not result in critical business impact. However, it is a best practice to prepare and plan for the worst.

This section discusses how you can develop a detailed plan to prepare for problems. By developing such a plan, you can mitigate the effects of problems when they do occur, and you can even prevent problems from occurring in the first place.

### 1.2.1 Applying WebSphere maintenance

Although many problems are caused by factors other than WebSphere Application Server code defects, the WebSphere Application Server Support team does find product defects when working through PMRs with clients. When a defect is found, the support team opens an authorized program analysis report (APAR). Each APAR has a unique identifier, a string that contains two letters (either PQ or PK) and five numbers. You can search for a particular APAR or a problem symptom reported in an APAR on the WebSphere Application Server Support site (Figure 1-1 on page 5). The site is available at:

<http://www-306.ibm.com/software/webservers/appserv/was/support>



Figure 1-1 Searching for an APAR on the WebSphere Application Server Support site

Fixes for APARs are included in fix packs and refresh packs, which are regularly published to the support site. Fix packs for Version 6 are equivalent to cumulative fixes for Version 5 and 5.1. Refresh packs for Version 6 are equivalent to fix packs for Version 5 and 5.1. See Figure 1-2.

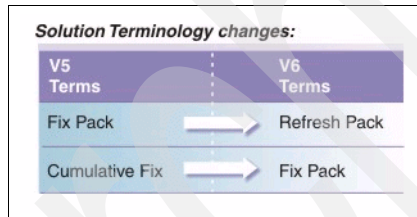


Figure 1-2 Changes in terminology for WebSphere maintenance between V5 and V6

Fix packs for Version 6 are delivered on a regular basis, about once every three months. They only contain fixes for APARs, and they do not introduce any new features or functionality to the product. You can think of a fix pack as preventive maintenance. Each fix pack includes a list of defects, which lists every APAR that is fixed in that fix pack. It contains all of the APAR fixes that were included in the previous fix pack as well as fixes for APARs that have been opened since the last fix pack. The fix packs add a fourth number to your WebSphere Application Server version. For example, if you were to install fix pack 2 for WebSphere Application Server 6.0.1, you would be upgrading to WebSphere Application Server 6.0.1.2.

Fix packs do not contain upgrades to the Java Software Development Kit (SDK). They are tested with the latest Java SDK service release, but the upgrades to the

Java SDK are delivered as separate fixes. You can also download the SDK fixes from the WebSphere Application Server Support site.

Because fix packs do not introduce new functionality to the product and because they have been fully regression tested, we recommend installing new fix packs as soon as they become available. However, we also recommend some level of testing with your applications.

Proactively installing fix packs as soon as they become available is an effective way to prevent problems from occurring. When you install a fix pack, you can be assured that you will not encounter any of the WebSphere Application Server code defects that are fixed in the fix pack. This saves the time and frustration of seeing one of these problems occur on your system.

Refresh packs for Version 6 are considerably larger upgrades than fix packs. In addition to containing fixes for APARs, all refresh packs contain some new features and functionality. They also upgrade the Java SDK. Refresh packs add a third number to your WebSphere Application Server version. For example, if you were to install refresh pack 1 for WebSphere Application Server 6.0, you would be upgrading to WebSphere Application Server 6.0.1.

We recommend that all clients install refresh packs when they become available. However, you should never install a refresh pack immediately on a production system. All applications should be fully regression tested with the new refresh pack. Upgrading to a new refresh pack level should be planned when there is sufficient time in the development cycle to test it.

For more information about the WebSphere Application Server V6 Update Strategy, you can review the Update Strategy document, which is available on the Support site:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21191989>

You can learn when fix packs and refresh packs are scheduled to be released by reviewing the *Recommended Updates for WebSphere Application Server Base and Network Deployment editions*:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27004980>

To determine which fix packs and refresh packs have been installed on your system, you can usually just check the version of the product. This is available on the Welcome page of the administrative console (Figure 1-3 on page 7) and at the top of all SystemOut.log files for each WebSphere Application Server process. You can also use the **historyInfo** and **versionInfo** commands in the `<WAS_install_root>/bin` directory.

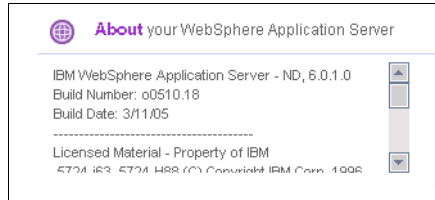


Figure 1-3 Checking the refresh pack and fix pack level from the administrative console

## 1.2.2 Checking the prerequisites

Another strategy for preventing problems is to ensure that all software and hardware in your environment meets the prerequisites of WebSphere Application Server V6. WebSphere Application Server has been tested with specific software and hardware configurations. It is known to work successfully in these configurations and to integrate well with the products with which it has been tested. You can find the software and hardware with which WebSphere Application Server has been tested and that it supports at:

<http://www-306.ibm.com/software/webservers/appserv/doc/latest/prereq.html>

If some of the software or hardware in your environment is not listed or if you are using older versions than the versions listed, you are at a greater risk for problems to occur. If you need to open a PMR, the support team will recommend moving to a supported configuration before proceeding with any other problem determination steps. Therefore, proactively ensuring that all of the software and hardware in your environment meets WebSphere Application Server's prerequisites is an important problem prevention and preparation technique.

## 1.2.3 Testing the application

The best strategy to prevent problems from occurring when running WebSphere Application Server in production is thorough testing. You should develop a detailed testing strategy for your application and make sure that the strategy is followed every time that you install a new version of the application or that you upgrade WebSphere Application Server.

There are countless methods of software testing that you can employ, and we could fill an entire book discussing every method. Instead, we discuss the basic types of tests that every client should perform. In addition, we provide links to other documentation that might be useful in developing your testing strategy. Testing methods include the following:

- ▶ Unit testing

Unit testing ensures that each method in each class of the application provides the expected output for all possible input. There are several unit testing frameworks that can be used to make unit testing easier. One popular framework is JUnit, which is open-source software.

- ▶ Functional testing

Functional testing ensures that the application performs as the user expects it to. It tests the entire application to make sure that every component works together correctly. Functional testing is sometimes called integration testing. There are several functional testing frameworks available. JFunc is an extension to JUnit for functional testing.

- ▶ Performance testing

Performance testing ensures that the application's performance is acceptable to users. WebSphere Application Server V6 includes an enhanced Tivoli Performance Viewer, which is accessible within the administrative console. In Version 5 and 5.1, the Tivoli Performance Viewer is run in a separate graphical user interface (GUI).

To use the Tivoli Performance Viewer, enable the Performance Monitoring Infrastructure (PMI) metrics that you want to view in the administrative console. Log onto the administrative console, select your application server, and then select **Performance Monitoring Infrastructure (PMI)**. PMI monitoring can be done in a production environment using the Basic set (default configuration) or the Extended set with minimal impact.

After selecting a statistic set, you can then use the Tivoli Performance Viewer to monitor the current performance of the system, log the performance data, or view data that had been previously logged. You can access the Tivoli Performance Viewer by expanding **Monitoring and Tuning** in the administrative console, then expanding **Performance Viewer**, and selecting to view the current activity or previously logged performance data.

The charts and graphics in the Tivoli Performance Viewer can give the administrator clues about where performance bottlenecks exist. They can then tune the appropriate WebSphere Application Server properties to relieve the problem.

Another feature included in WebSphere Application Server V6 is the Tivoli Performance Advisor. The Advisor analyzes the performance data from your system and provides suggestions on which WebSphere configuration



properties to change to improve the performance. To enable the Advisor, select your application server in the administrative console and then select **Runtime Performance Advisor Configuration**.

For more information about the Tivoli Performance Viewer and Advisor, review the following WebSphere Information Center sections:

- *Why use Tivoli Performance Viewer?*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cprf\\_tpv.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cprf_tpv.html)

- *Monitoring performance with Tivoli Performance Viewer (TPV)*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tprf\\_tpvmonitor.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tprf_tpvmonitor.html)

- *PMI data organization*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprf\\_datacounter6.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprf_datacounter6.html)

In addition to the Tivoli Performance Viewer and Advisor, there are several other tools, both open-source and proprietary, that are available to make performance testing easier. You can read an overview of the performance tools that are available in the technical article *Comment lines from Ruth Willenborg: Selecting WebSphere performance tools*, which is available at:

[http://www-128.ibm.com/developerworks/websphere/techjournal/0410\\_col\\_willenborg/0410\\_col\\_willenborg.html](http://www-128.ibm.com/developerworks/websphere/techjournal/0410_col_willenborg/0410_col_willenborg.html)

► **Load and scalability testing**

Load testing involves testing your application with a simulated workload that corresponds to the amount of load that you expect your application to be able to handle in production. Scalability testing involves testing your application with increasingly higher amounts of load to determine if the application is scalable for future growth.

There are several popular tools for simulating load during load and scalability testing. One such tool is Apache JMeter. You should coordinate your load and scalability testing with the tools that you use for performance testing so that you can tune WebSphere Application Server to provide better performance with higher levels of load.

We recommend using the WebSphere Application Server V6 Test Environment within Rational® Application Developer, Version 6 to develop and test all of your application code before installing it on your WebSphere Application Server environment.

As you design your tests, consider the following:

- ▶ Your test scenarios should focus on the most used code path, but you should comprehensively test all possible code paths.
- ▶ Testing should be done with multiple users (not just the same one, over and over).
- ▶ Tests should be done with multiple functions in parallel. In production, several functions are executed at the same time. Some problems might only occur when the functions are run in parallel. It is important to test your application's functions together and not in isolation.

Ensuring that your application has been rigorously tested before it is put into production greatly reduces the chance of application defects causing problems in your production environment. For more information about various software testing methods and advice about testing, review the article *Recommended reading list: Software testing* at:

[http://www-128.ibm.com/developerworks/websphere/library/techarticles/0402\\_polozoff/0402\\_polozoff.html](http://www-128.ibm.com/developerworks/websphere/library/techarticles/0402_polozoff/0402_polozoff.html)

## 1.2.4 Setting up a test environment

We strongly recommend that you maintain a test environment that is configured exactly the same as your production environment. The WebSphere Application Server maintenance level (including refresh packs and fix packs), the versions of your applications, and your configuration should all be the same on both systems. There are many benefits to doing this:

- ▶ When you successfully test an application in your test environment, it gives you an accurate reflection of how the application will perform in production.
- ▶ When you need to make a change to an application or apply maintenance to WebSphere Application Server, you can fully test these changes in the test environment to be sure that there are no problems before you make any changes to your production environment.
- ▶ When a problem does occur in production, you can reproduce the problem in the test environment and perform problem determination in the test environment. This ensures that your users will not experience down time with the production environment.
- ▶ You can collect any diagnostic data that is needed to determine the root cause of a problem in the test environment. Because some diagnostic methods can impact performance or necessitate restarting one or more WebSphere Application Server processes, it is usually beneficial to collect the data in your test environment.

The key to this strategy is ensuring that the test environment and production environment are configured *exactly* the same in every way. This includes:

- ▶ The hardware and network configuration.
- ▶ The operating system level and operating system patches.
- ▶ The other software that is used in conjunction with WebSphere Application Server. This might include Web servers, databases, and messaging systems.
- ▶ The WebSphere Application Server level, including refresh packs, fix packs, Java SDK fixes, and any individual APAR fixes that you might have downloaded or obtained from the WebSphere Application Server Support team.
- ▶ The versions of all applications that are installed.
- ▶ The WebSphere Application Server edition (Base, Network Deployment, or Express).
- ▶ The WebSphere Application Server configuration. WebSphere Application Server V6 includes functionality to create configuration archive files with a .car extension. You can export the configuration from one machine to a .car file and then import that .car file to another system. Any configuration information that is specific to one system (for example, the host name) is removed in the configuration archive. It is a good practice to use configuration archive files to replicate the same WebSphere Application Server configuration on your test environment and production environment.

Configuration archives are exported and imported with the wsadmin tool. To export the configuration of a WebSphere Application Server V6 profile or application server, use these wsadmin commands:

```
$AdminTask exportWasprofile {-archive c:\myDirectory\myCell.car}  
$AdminTask exportServer {-archive c:\myDirectory\myServer.car -nodeName  
node1 -serverName server1}
```

Use the target directory, node name, and server name that is appropriate for your system.

To import the configuration of a profile or application server into a WebSphere Application Server V6 environment, use these commands:

```
$AdminTask importWasprofile {-archive c:\myDirectory\myCell.car}  
$AdminTask importServer {-archive c:\myDirectory\myServer.car  
[-nodeInArchive node1] [-serverInArchive server1] [-nodeName  
node1] [-serverName server1]}
```

Again, use the target directory, node names, and server names that are appropriate for your system.

You can also automate changes to your test and production environments with scripts that run in wsadmin. Automation is advantageous if changes must be

done during non-peak hours (typically late nights or on weekends). The scripts can be scheduled to run at those times so that no one has to run the scripts manually during those hours. You might also want to run scripts to update the configuration of other software on your system at the same time. You can also use the scripts as part of your change log (see “Establishing safe operational procedures” on page 12).

Any time you make a change to either the test environment or the production environment, you should synchronize the two environments so that they remain identical.

It is a good idea to establish a baseline configuration, meaning a configuration that has been tested successfully and found to be stable. When you make any type of change, you can test it. If the change is successful and does not cause any problems, you can then add the change to the baseline. If the change does cause a new problem, you can revert to the safe baseline configuration.

If you set up a test environment following these guidelines, you can prevent many problems, because your testing is done in an environment that is identical to your production environment. In addition, you will be better prepared to follow the problem determination methodologies that are described in this book, because you have an ideal test environment ready for use.

### 1.2.5 Establishing safe operational procedures

Another important facet of problem prevention and preparation is establishing a set of safe operational procedures for your organization. These procedures should outline the proper processes for making any types of changes to your test and production environments. The WebSphere Application Server Support team has found that many problems occur as a result of configuration or code changes made by one person in an organization of which other people in the organization were not aware. A strategy to eliminate these occurrences greatly reduces the chance of unexpected problems.

It is important to define security roles for people in your organization and to ensure that only people who are authorized or part of the appropriate role can make changes to your configuration, upgrade your software, install applications, or do anything else that could potentially introduce problems. WebSphere Application Server V6 provides a comprehensive security infrastructure that you can use to define roles, authenticate, and authorize users. You should also take advantage of the security features of your operating system. This book does not discuss security. However, you can learn more about security in the security section of the WebSphere Information Center at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc\\_concepts\\_csec.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc_concepts_csec.html)

In addition to restricting the configuration of your environment, you should also implement a change control system for your application code. There are several change control software products available. However, we recommend Rational ClearCase®, which integrates seamlessly with Rational Application Developer. You can get more information about Rational ClearCase at its product Web site:

<http://www-306.ibm.com/software/awdtools/clearcase>

A key to any strategy for configuration or code changes is documentation. It is vitally important to document any and all changes that are made to your production and test environments. We call this documentation a *change log*. Many problems surface after an application has been running successfully in production for a long time. When you carefully document all changes that were made to your environment in one location, such as the change log, it is much easier to determine why a problem might have occurred at a certain time or date. Ensure that each person in your organization is aware of the process for updating the change log when any change is made to the environment and that they follow the process. They should include the precise time and date of any changes when they update the change log. Maintaining the change log and ensuring strict compliance with this process can save many hours of investigation and frustration when a problem occurs.

Another process that is important to establish is one for keeping the test and production environments synchronized at all times. If changes of any kind are rare, you might want to make the changes manually in both environments at the same time or make the change to production immediately after it has been tested in the test environment. On the other hand, if there are several changes during the course of a day, it might be easier to synchronize several changes at one time. As discussed in “Setting up a test environment” on page 10, you might want to create a configuration archive in one environment and then import it into the other environment.

Finally, you should select a procedure for scheduling changes to your production environment. You want to make sure that any changes to production do not cause your application to be unavailable to your users. You can determine the off-peak hours for your application. In many cases, the off-peak hours are late night and early morning hours or weekends. Schedule your production changes at a certain time, and make sure that everyone in your organization is aware of the schedule.

Establishing safe operational procedures is an essential part of the planning process. A good set of procedures that are agreed upon and acted upon by everyone in your organization can prevent problems from occurring and make you more prepared when problems do occur.

## 1.2.6 High availability and failover

As discussed earlier, even the most sophisticated problem prevention techniques cannot guarantee that problems will never occur in your WebSphere Application Server V6 environment. Given this, it is logical to develop a contingency plan for when a WebSphere Application Server process or service becomes unavailable. Fortunately, new features in the area of high availability and failover are included with Version 6.

Workload balancing enables your applications to serve more users and achieve better performance by distributing client requests to servlets and EJBs on multiple application servers in a cluster. Application servers that are clustered are called *cluster members*. Failover allows the work that is processed by one cluster member to failover to another cluster member if the first cluster member is down. Also, new requests are then routed to the other cluster members. These features make it easier for production environments to recover from failures.

New in WebSphere Application Server V6 is the high availability manager feature to provide failover for all applications and WebSphere Application Server services. It is configured automatically when you install WebSphere Application Server Network Deployment Version 6. The high availability manager runs critical WebSphere services (such as WLM and the transaction manager) on any available WebSphere Application Server processes.

In Version 6, WLM and other services can run on any deployment manager, node agent, application server, or cluster member process. The high availability manager tracks the status of each WebSphere process and the services that they run. When it detects a failure, it can start the service in another process in less than a second.

To make the WebSphere Application Server transaction manager highly available in Version 6, you must store the transaction logs (located by default in the `<WAS_install_root>/profiles/<profile>/tranlog` directory) on a network attached storage (NAS) system that is accessible to all of the WebSphere processes to which your transactions could failover (members of the core group). You also must select Enable high availability for persistent services for your cluster in the administrative console. When this is done, and a cluster member fails, its in-flight transactions are recovered on another cluster member.

A *core group* is a group of WebSphere Application Server processes (deployment managers, node agents, application servers, and cluster members) within a cell that can participate in high availability together. That is, a service running on one member of the core group can failover to another member of the core group. By default, all processes in a cell are part of one core group (called *DefaultCoreGroup*). This is also the recommended scenario for most production environments, although it is possible to have multiple core groups within one cell.

A process can only be a member of one core group. System services, such as the WLM service, PMI, and the high availability coordinator itself, can failover to any member of the core group. Other services, such as the transaction manager and messaging engine, must failover another cluster member in the same cluster in which they were running when they failed. This is because the transaction manager and messaging engine need the same applications to be installed on the process that they failover to in order to recover.

The WLM service in Version 6 provides tload balancing functionality that is comparable to Version 5 and 5.1. The addition of the high availability manager significantly enhances the failover capabilities in Version 6, making it no longer necessary to run third-party high availability tools with WebSphere Application Server.

Understanding and taking advantage of the high availability and failover features in WebSphere Application Server V6 is an essential part of problem planning strategies. In the event that a problem does occur, it is reassuring to know that your WebSphere Application Server environment is configured in such a way that the essential functions failover and users will not be impacted.

For more information about high availability in Version 6, review the following:

- ▶ *High availability manager*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun\\_ha\\_hamanager.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun_ha_hamanager.html)

- ▶ *Setting up a high availability environment*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun\\_ha\\_environment.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_ha_environment.html)

- ▶ *WebSphere Application Server V6 Network Deployment: High Availability Solutions, SG24-6688*

<http://www.redbooks.ibm.com/redpieces/abstracts/sg246688.html>

## 1.2.7 Monitoring

When you prepare for possible problems, you must consider how you will know when a problem has actually occurred. At first, this might seem trivial. However, a well-planned monitoring strategy can significantly increase the satisfaction of the application's users and customers.

For some enterprises, realization of a new problem occurs when a customer calls in or sends an e-mail explaining a problem that they have encountered with your production application. Usually, their explanation is simple, such as "I cannot log onto the Web site" or "the banking application does not do anything when I enter my account number and hit enter." It is normal for a customer to feel frustrated

and possibly dissatisfied with your company when these types of situations occur.

A good monitoring strategy can help you to identify problems *before* your customers experience them.

The most important part of your monitoring strategy is to monitor the WebSphere Application Server logs. You can find detailed information about managing the logs in *WebSphere Application Server V6: Diagnostic Data* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4085.pdf>

You should monitor the SystemOut and SystemErr logs for each WebSphere Application Server process in your environment. SystemOut and SystemErr logs are created for every WebSphere process (application server, cluster member, node agent, and deployment manager). These logs are also known as JVM logs. The System.Out and System.Err streams for each JVM are redirected to the SystemOut and SystemErr logs. You can find the logs in the `<WAS_install_root>/profiles/<profile>/logs/<process>` directory.

You notice several system messages in the SystemOut log. WebSphere Application Server V6 includes a wide variety of system messages that are designed to provide you with information, warnings, and notifications of errors. You also find many system messages in WebSphere Application Server traces. Each system message has a unique message identifier that is nine characters in length and is in the form CCCC1234X. The first four characters (CCCC) indicate the WebSphere Application Server component that issued the message. The next four characters (1234) indicate the specific message that is being issued by the component. The last character (X) indicates the severity of the message. Its value is either I (informational), W (warning), or E (error).

Here are examples of the different types of system messages:

```
SECJ0231I: The Security component's FFDC Diagnostic Module
com.ibm.ws.security.core.SecurityDM registered successfully: true.
ADMN0001W: The service is unable to parse the MBean descriptor file
com/ibm/ws/management/descriptor/xml/mbeans.xml.
SRVE0068E: Could not invoke the service() method on servlet
/com.ibm.ws.console.probdetermination/loggingSettingsGroups.jsp. Exception
thrown : java.lang.NullPointerException
```

These system messages are useful for monitoring purposes. Informational messages usually are not indicative of a problem. However, if an informational message is unexpected, it might alert you to an unusual occurrence that can result in a problem. Warnings and error messages are definitely signs that a problem has occurred.



The SystemErr log does not contain WebSphere system messages, but it does show exceptions that are thrown by WebSphere Application Server or by an application. It is a good idea to monitor the SystemErr log entries in addition to the system messages in the SystemOut log.

An administrator can monitor these logs and take appropriate action based on the output. It is also common for a production environment to be instrumented so that a major error or problem occurrence generates an e-mail message or a page to the required people, so that they are notified of the problem even if they are not actively monitoring the system.

It is also a good practice to add logging to your application. There are several APIs that you can use for logging. A popular API is the Java Logging API (the `java.util.logging` package), which is standardized in Java Specification Request (JSR) JSR-047. You can find detailed information about Java logging in *Logging and tracing with Java logging* at the WebSphere Information Center at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb\\_java\\_logging.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb_java_logging.html)

You can use this or another logging API to best suit your needs. You can design your application to output certain messages, errors, or exceptions when various problems occur. You can direct your application logging output to the WebSphere Application Server SystemOut and SystemErr logs or to separate logs that are used only by your application. You can log specific information that will make it easier for the system monitor to detect problems with the application.

By monitoring the SystemOut and SystemErr logs and adding your own application logging, you can discover proactively a majority of the problems that are covered in this book.

In addition to viewing the logs, you can monitor for configuration problems and runtime events within the administrative console. To do this:

1. Log onto the administrative console.
2. Expand **Troubleshooting**.
3. Expand **Configuration Problems or Runtime Messages**.

You can view errors, warnings, and informational messages directly in the administrative console (Figure 1-4).

Timestamp	Message Originator	Message
Oct 20, 2004 12:17:18 PM EDT	com.ibm.ejs.j2c.RALifeCycleManagerImpl	J2CA0052E: The lookup of the ActivationSpec with J
Oct 18, 2004 1:52:54 PM EDT	com.ibm.ws.odc.cell.TreeBuilder	ODCF0002E: Exception: NodeAgent.mbean not found: W
Oct 18, 2004 12:01:20 PM EDT	com.ibm.ws.webcontainer.webapp.WebApp	Error reported: java.lang.Exception: Mapping clash
Oct 18, 2004 12:01:20 PM EDT	com.ibm.ws.odc.cell.TreeBuilder	ODCF0002E: Exception:

Figure 1-4 Viewing Runtime Events in the administrative console

You can select each message that you see to get more details about the message, the source of the message, and the reason why it occurred (Figure 1-5 on page 18).

**Runtime Events > Message Details**

Runtime events propagating from the server

**General Properties**

**Message**  
handleServerChangeEvent(): PLGC0055E: Unable to get the list of Web servers defined on the node PlatoCell.ates/servertypes/APPLICATION\_SERVER. PLGC0026E: An exception occurred while reading the server index for the server ates/servertypes/APPLICATION\_SERVER.

**Message type**  
Error

**Explanation**  
No explanation found for ID=handleServerChangeEvent()

**User action**  
No user action found for ID=handleServerChangeEvent()

**Message Originator**  
com.ibm.websphere.pluginfg.initializers.ServerIndexChangePluginTask

Figure 1-5 Viewing message details in the administrative console

Another way to monitor your application is by using the Tivoli Performance Viewer, discussed in “Testing the application” on page 7. You can observe

performance metrics, such as average response time, number of requests, thread pool sizes, connection pool sizes, JVM memory, CPU, I/O, and system paging, to monitor the health of WebSphere Application Server and your applications. You can log the performance data from a time when everything is running normally and then compare that data to the current performance data to see if there are major discrepancies. Unexpected discrepancies are an indication of a problem. When you see signs that a problem might be occurring, you can perform the specific problem diagnostic steps that are described within this book.

Your monitoring needs might become more complex if your application runs in a multi-tiered environment involving many different software products. In this scenario, the IBM Tivoli Monitoring for Transaction Performance product might be useful to you. You can get more information about this at:

<http://www-306.ibm.com/software/tivoli/products/monitor-transaction>

A well-planned monitoring strategy, especially when combined with a good high availability and failover strategy, should enable you to diagnose and resolve problems that occur, possibly before your users and customers know about them.

## 1.2.8 System documentation

In the event that a problem occurs in your environment, it is possible that you will need to enlist the help of other people, either internal or external to your organization, to determine the root cause of the problem. When that happens, you will want everyone involved to understand thoroughly the details of the systems that are involved in your environment.

To this end, it is important to document the details of your configuration. In “Establishing safe operational procedures” on page 12, we discussed documenting all of the changes that have been made to your environment. In addition to that, you should maintain a high-level description of your basic topology. We call this *system documentation*. System document is useful in the following circumstances:

- ▶ A problem occurs and you need to get assistance from others who might not be as familiar with your application and topology as you are. The system documentation allows you to bring them up to speed as quickly as possible.
- ▶ A problem occurs and you want to identify from which parts of your environment you should collect diagnostic data or monitor. Your system documentation shows the software components that are involved and the flow of your application, that is how different software components are used when your application processes a request.

Your system documentation should consist of written documents and diagrams. Which information is included in the written documents and which is included in diagrams is a matter of preference. Overall, the information should be detailed and should show the specific versions and maintenance levels of the operating system and all software products involved, the hardware and network configuration, and specific host names and IP addresses of the systems that are involved.

A common and important component of system documentation is the topology diagram. It gives a quick overview of your system topology and application flow. Figure 1-6 on page 20 illustrates an example.

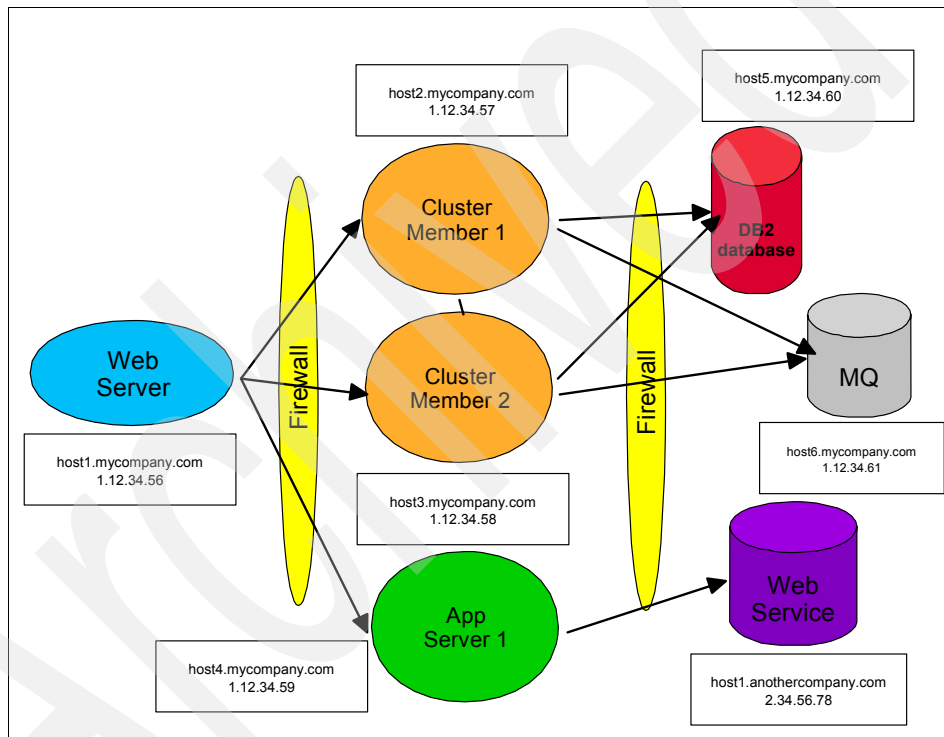


Figure 1-6 Example of a topology diagram

With a quick look at the diagram, you can see that the application is receiving HTTP requests from a Web server. Some requests are sent to a cluster, which in turn sends data to a DB2 database and an MQ messaging system. Other requests are sent to an application server which interacts with an external Web service.

In this scenario, the accompanying written documentation should include the specific software and hardware levels that are involved.

Detailed system documentation is an integral part of your problem planning strategy that should not be overlooked.

### 1.2.9 Diagnostic data collection

Finally, to prepare for a problem occurrence, you should plan what diagnostic data to collect for various problem scenarios. In “Types of problem symptoms” on page 3, we discussed several broad categories of problem symptoms. In “What to do when a problem occurs” on page 24 and other papers in this series, we elaborate on how to determine the cause of a problem for each of the problem symptoms. We also include information about what data to collect for different types of problems. It is a good idea to identify the most common problems that have occurred in the past in your environment and those that you believe might occur the most in the future. Then, you can form a diagnostic data collection plan so that you are prepared to collect the necessary data if a problem does occur.

There are some recommendations that apply to your diagnostic data collection plan regardless of the types of problems for which you are preparing. For example:

- ▶ Ensure that the clocks on all systems in your environment are synchronized. This helps in the analysis of diagnostic logs and traces. Often, a request is sent from one system to another, and it helps to match up the time stamps from both systems when analyzing the diagnostic data. If some systems are located in different time zones, make sure that this is documented in your system documentation (as discussed in “System documentation” on page 19).
- ▶ Configure WebSphere Application Server logging and tracing so that it captures a sufficient amount of data when a problem occurs. The WebSphere Application Server Support team has found that many times, the diagnostic data needed to determine the root cause of a problem has been overwritten before a client realizes that the problem has occurred. This type of situation can be prevented by increasing the amount of log and trace data that is saved before the files are rolled over. For both logging and tracing, you can configure file rotation properties for this purpose.

You can find the instructions for configuring logging and tracing, including the file rotation properties, in *WebSphere Application Server V6: Diagnostic Data* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4085.pdf>

- ▶ Plan to have extra disk space available on your system to store diagnostic data when problems occur. Many of the most severe problems, such as application server crashes, hangs, and out of memory conditions, require the largest amount of diagnostic data. If enough disk space is not available when a problem such as this occurs, you might need to reproduce the problem

several times to collect the necessary data. To avoid this situation, it is recommended that you have between 2 GB and 5 GB of extra disk space available on each system.

- ▶ After you resolve a problem, either delete or archive the diagnostic data that you collected for the problem. This will prevent the old diagnostic data from being confused with the new diagnostic data the next time that a problem occurs.
- ▶ Configure the thread monitor for hang detection. WebSphere Application Server V6 includes a thread monitor feature. This is also included in Version 5.1.1. The thread monitor is notified when the Web container, ORB, or asynchronous bean thread pools give work to a thread. By default, the thread monitor checks the status of all active threads every three minutes. If it finds a thread that has been active for more than ten minutes, it outputs a warning to the SystemOut log, similar to the following:

WSVR0605W: Thread threadname has been active for hangtime and may be hung. There are totalthreads threads in total in the server that may be hung.

The thread monitor makes it easier to determine that a problem has occurred. If you see a WSVR0605W warning in your SystemOut log, you now know that a thread has stopped responding. You can then perform further diagnostic steps to determine the cause of the hung thread. The thread monitor does not take any action to fix the problem beyond notifying you of the problem.

In preparing for diagnostic data collection, you might want to change the default thread monitor behavior. You can change the interval that the thread monitor checks the status of threads (this is three minutes by default) and the amount of time that a thread can be active before it is reported by the thread monitor (this is ten minutes by default). To alter these properties:

- a. Log onto the administrative console.
- b. Select your application server.
- c. Under Server Infrastructure, select **Administration**.
- d. Select **Custom Properties** and then click **New**.
- e. Create these properties and provide the desired values for them:
  - `com.ibm.websphere.threadmonitor.interval` (the interval that the thread monitor checks the status of threads)
  - `com.ibm.websphere.threadmonitor.threshold` (the amount of time that a thread can be active before it is reported by the thread monitor)

For more information about the thread monitor, see the following sections in the WebSphere Information Center:

- *Detecting hung threads in J2EE applications*

[http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb\\_hangdetection.html](http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_hangdetection.html)

- *Configuring the hang detection policy*

[http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb\\_confighangdet.html](http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb_confighangdet.html)

- ▶ Consider enabling verbose garbage collection on each application server. The performance impact of enabling verbose garbage collection is minimal, and the data is often useful when performance problems occur. To enable verbose garbage collection:
  - a. Log onto the administrative console.
  - b. Select your application server.
  - c. Under Server Infrastructure, expand Java and Process Management, and then select **Process Definition**.
  - d. On the resulting screen, select **Java Virtual Machine** under Additional Properties.
  - e. Select Verbose garbage collection.

When verbose garbage collection is enabled, the output appears in the `native_stderr.log` file for your application server.

- ▶ Become familiar with the WebSphere Application Server collector tool. The collector tool is run as an executable in the `<WAS_install_root>/bin` directory. It produces a Java archive (jar) file that contains all of the logs and XML configuration files from your system. The resulting jar file is very useful to the WebSphere Application Server Support team and any others who are involved in the problem determination process. It allows them to view quickly your WebSphere Application Server configuration and to see any errors or exceptions that have occurred.

You can find more information about the collector tool in *WebSphere Application Server V6: Diagnostic Data* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4085.pdf>

The WebSphere Application Server Support team has compiled a comprehensive list of MustGather documents for different types of WebSphere Application Server problems. For more information, see *IBM - MustGather: Read first for all WebSphere Application Server products* at:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21145599>

The files listed in the MustGather documents are useful in determining the cause of your problem. You can use these documents as part of your diagnostic data collection plan.

Consider all the types of problems that you might be likely encounter in your environment while reading this book. Based on the types of problems that have occurred in your environment in the past, you might be able to predict what types of problems can occur in the future. Take note of the diagnostic data that is needed to determine the root cause of each problem type. Make sure that this is documented so that everyone in your organization knows what data to collect. This will make the problem determination process simpler and less stressful.

## 1.3 What to do when a problem occurs

You have prepared, and you have planned. Your application is now running in your production environment, and it is being used successfully by your customers. Just as you are ready to celebrate, you get a call that a problem has occurred. What do you do now?

In this section, we start by discussing the first steps for any problem scenario. Then, we introduce some problem determination methodologies that you can use for any problem. This section leads to the following section, which helps you make that leap from symptom to problem description and points you to information that can help you determine the root cause of your specific problem.

### 1.3.1 Revert to safe conditions

When any problem occurs, your first action should be to consider the business impact of the problem. Depending on the business impact, it might be necessary to take steps to limit the business impact before beginning your problem determination efforts. If the problem occurred in your test environment, the business impact is relatively low when compared to the potential impact of a production outage. In that case, making an effort to work around the problem while looking for a permanent solution is probably not necessary. You can use the test environment to execute all of the necessary problem determination steps.

However, if the problem occurred in production, you will probably want to consider how to quickly alleviate the problem symptoms so that customers and users will experience the least possible negative effects. In this section, we refer to this process as *reverting to safe conditions*. You will want to do this in parallel with your problem determination efforts.

Techniques for reverting to safe conditions include:

- Making your test environment or another similar environment where the problem is not occurring your temporary production environment. Configure



your systems so that incoming requests are processed by an environment where the problem does not occur.

- ▶ Installing an older version of application code where the problem does not occur. If the problem started to occur after an application code change was introduced, it might be a good idea to go back to an earlier working version of the application.
- ▶ Changing any recent configuration changes back to your baseline configuration. As discussed in “Setting up a test environment” on page 10, the baseline configuration should be a configuration that is fully tested and is known to be stable.
- ▶ Removing any WebSphere Application Server maintenance that was recently installed before the problem occurred. This would temporarily resolve the problem if it is caused by a WebSphere Application Server code defect introduced by the latest maintenance package.
- ▶ Making the application function that produced the problem inaccessible to customers and users. You can post a notification on your Web site that the function is temporarily unavailable or under maintenance. You can provide an estimated time for when it will be available again.

Establishing safe conditions to which you can revert when a problem occurs and then reverting to those conditions when a production problem occurs is a good way to reduce the business impact of your problem. After you have evaluated whether reverting to safe conditions is necessary and taken the appropriate steps, you can begin the problem determination process.

### 1.3.2 Identify problem symptoms

When you are first notified about the problem, you might only receive a vague, non-specific set of problem symptoms. You might be told that users cannot access your application at all or possibly that a specific action taken by your users is resulting in an error message.

You should collect specific details about the problem symptom in order to form a detailed and thorough problem description. You can ask the following questions in order to get as specific a problem description as possible:

- ▶ What were the specific problem symptoms that were observed? Did an error occur? Did the application produce an unexpected result? Did the application fail to respond to an incoming request?
- ▶ What was the context under which the problem occurred? Did the user execute a specific function? Did the problem happen only after there was an unusually high workload on the system? Did it occur immediately after the application was restarted?

- ▶ How do you know when this particular problem occurs? Is there something specific to watch for in order to recognize if the problem occurs again?
- ▶ How would you know that the problem was resolved? Would it be that an error message no longer occurred? Would the application behave differently? What specifically would confirm a problem resolution?
- ▶ Where did the problem occur? Did the problem occur only in your test environment, only in your production environment, or on both? Did it only occur on one system in your environment? Did it occur on multiple systems? Did it occur on every cluster member or only one?
- ▶ When did the problem occur? What was the time stamp of the error or unexpected behavior? Did the problem occur only once or many times? How often did the problem occur? Did it occur at certain intervals, or did it seem to occur at random times? Was there some event that occurred that might have triggered the problem? For example, did the user attempt a specific application function?
- ▶ Why might the problem have occurred? You might not be able to answer this right away. Was this the first time something specific was tried? Was there a recent change to the application code or the configuration of your environment? Does it happen in all of your environments or only one? If it only happens in one environment, how is this environment different from the others?
- ▶ Has the diagnostic data that is identified in your diagnostic data collection plan been collected? Does the data provide any other details about the problem or offer immediate clues as to why the problem occurred?

Compile your answers in a document that gives as much specific detail about the problem occurrence as possible. We refer to this document as the *problem log*. Ensure that everyone who is involved in the problem determination process has access to the problem log and can update it with new information as it is uncovered. Also ensure that the problem log lists the location of the diagnostic data. It might be a good idea to coordinate the problem log with the change log as discussed in “Establishing safe operational procedures” on page 12.

It is possible that the answers themselves will reveal the cause of the problem. For example, the original problem symptom could have been that the user received an error message when accessing a specific servlet in the application. You might have found that a new version of this servlet was just installed in the environment where the problem occurs but was not installed in other environments that are still functioning as expected. This could lead you to determine that an application code change caused the problem.

On the other hand, the answer might require more investigation. If this is the case, you have developed a solid and complete problem description to use as the basis for more problem determination efforts.

### 1.3.3 Investigate and research the problem

While investigating the problem, you should form a list of all of the problem symptoms that occurred within your problem log. Sometimes, there is only one symptom. If this is the case, your job is easier. However, often several symptoms occur, and it can be difficult to determine which symptoms characterize the problem and which symptoms are simply the result of the problem. It is a good idea to organize the symptoms into a time line. You can also include details about what was happening in your environment in your time line. For example, your time line might look similar to the following:

```
10:00 - Peak workload reached
10:07:53 - ConnectionWaitTimeoutException in SystemOut.log for server1
10:08:24 - ConnectionWaitTimeoutException in SystemOut.log for server1
10:14:09 - ConnectionWaitTimeoutException in SystemOut.log for server2
10:20:46 - Users cannot log onto Web site, application is not responding
```

Because the `ConnectionWaitTimeoutException` was the first symptom that was observed, it is a good starting point for your investigation. However, it is possible that your investigation might reveal that a symptom is actually a different problem that is unrelated to the problem that you are investigating.

In this example, the problem that you want to resolve is that the application stops responding. It is possible that the `ConnectionWaitTimeoutException` is actually a second problem that happens to occur around the same time. On the other hand, it is also possible that multiple symptoms are all the result of the same underlying problem. During the course of your investigation, you will be able to identify which symptoms are related and which ones are not.

When forming your list of symptoms and your time line, you might notice several WebSphere system messages in your logs. These include informational messages, warnings, and errors. We discussed the details of WebSphere system messages in “Monitoring” on page 15.

WebSphere system messages are useful in your problem determination efforts. Informational messages and warnings provide context. They show you what was occurring immediately before or immediately after a problem. Error messages are significant problem symptoms. The message identifier for an error message is useful for entering into search engines.

When you choose a symptom to investigate, you can begin researching it. A good place to start is your organization’s internal documentation. It is possible

that someone else has already seen the same problem or a similar problem in your environment. If they have, it is likely that the root cause of the problem is the same. If this is the first time that a problem similar to this has been seen in your organization, the next place to research is online documentation that is available from IBM. We recommend using the following resources to research your symptoms:

- ▶ WebSphere Information Center

<http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp>

The WebSphere Information Center includes detailed information about the features and configuration of the product. It also includes descriptions of every WebSphere Application Server system message identifier, and it contains a troubleshooting guide that might aid your problem determination efforts:

[http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc\\_concepts\\_ctrb.html](http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc_concepts_ctrb.html)

- ▶ WebSphere Application Server Support site

<http://www-306.ibm.com/software/webservers/appserv/was/support>

You can search the support site for APARs and technotes that might be related to your problem symptom. As discussed in “Applying WebSphere maintenance” on page 4, APARs are reports of known WebSphere Application Server code defects. If an APAR description matches your problem symptom, you should install the WebSphere Application Server fix pack that contains the fix for the APAR. The APAR description explains which fix pack includes the fix. Technotes are documents that are created and maintained by the WebSphere Application Server Support team and knowledge engineering team. They document known problems and the solutions to those problems. Technotes are usually created for problems that are solved by a configuration change rather than a code change.

- ▶ *WebSphere developerWorks®*

<http://www-130.ibm.com/developerworks/websphere>

The developerWorks site contains many articles written by IBM developers and other technical staff members. Many articles discuss application programming best practices and provide tips for avoiding and resolving problems with WebSphere Application Server.

- ▶ *IBM Support Assistant*

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21192593>

The IBM Support Assistant is a downloadable tool that is used to simplify the problem determination process for many IBM software products. There are several product plug-ins for the Support Assistant, including one for WebSphere Application Server V6. It includes a federated search interface

that allows you to search for your problem symptoms at multiple IBM Web sites simultaneously. It also provides easy access to product education modules, including the IBM Education Assistant. If you need to open a PMR with IBM Support, the tool collects pertinent information that is needed by the support team. You can even open a PMR within the tool.

This book also provides valuable information as you research your problem.

### 1.3.4 Problem determination strategies

At this point, you have identified specific and detailed problem symptoms, reverted to safe conditions so that you have minimized the business impact of the problem, and begun your investigation of the problem. The next step is to start determining the root cause of the problem, which is the main focus of this book.

This section discusses general problem determination strategies that you can use for any type of problem. The following section discusses classifying the problem and then using specific problem determination methodologies for each problem classification.

For any problem that occurs, there are two major strategies that you can use to determine the root cause: the analysis strategy and the isolation strategy.

The *analysis strategy* involves analyzing the diagnostic data, possibly through several iterations, until the cause of the problem is found, as illustrated in Figure 1-7 on page 30. To be successful with this strategy, you must have a good understanding of the diagnostic data. This is the strategy used most often by the WebSphere Application Server Support team. There are many diagnostic tools that are available to help you. For example, the ThreadAnalyzer helps you to analyze Java thread dumps, and the HeapRoots tool is useful in analyzing out of memory issues. You can review and download the most current diagnostic tools at:

- ▶ *WebSphere Application Server Support site*  
<http://www-306.ibm.com/software/webservers/appserv/was/support>
- ▶ *WebSphere developerWorks*  
<http://www-130.ibm.com/developerworks/websphere>

Often, when analyzing one set of diagnostic data, you discover information that requires that you collect more diagnostic data for further analysis. You begin to follow an iterative process. After each iteration, you are closer to determining the root cause of the problem, and after enough iterations, you will find it.

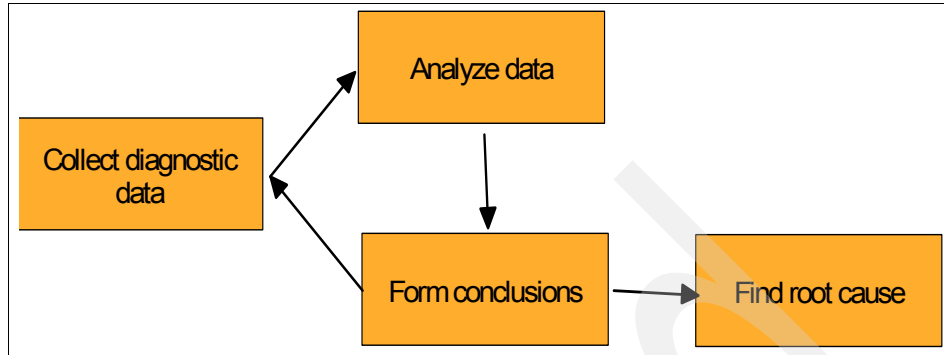


Figure 1-7 The analysis strategy

The *isolation strategy*, on the other hand, involves removing variables from the problem scenario until you have isolated the problem to the one variable that is causing the problem, as shown in Figure 1-8 on page 31. Variables can include application code, configuration, and software maintenance levels. You need to be able to reproduce the problem in order to use this strategy. Most often, you will reproduce the problem in your test environment. You need to have a thorough understanding of the topology of the environment. You can refer to the topology diagram as a reference.

The isolation strategy is often employed when there is an error or exception. In the SystemOut or SystemErr log for the application server, you see the Java stack trace of the exception. The stack trace is read from the bottom to the top. At the bottom, you see the thread that is allocated by the thread pool. Reading the stack trace upwards, you see every method that is called (including WebSphere Application Server code, application code, and possibly other third-party utilities) until the exception is thrown. You can use the stack trace to identify the different variables in the problem scenario and then remove them.

It might also help to insert print statements into the application to print debugging information as the application executes.

Similar to the analysis strategy, the isolation strategy is an iterative process. You keep removing variables until you have isolated the variable that is causing the problem.

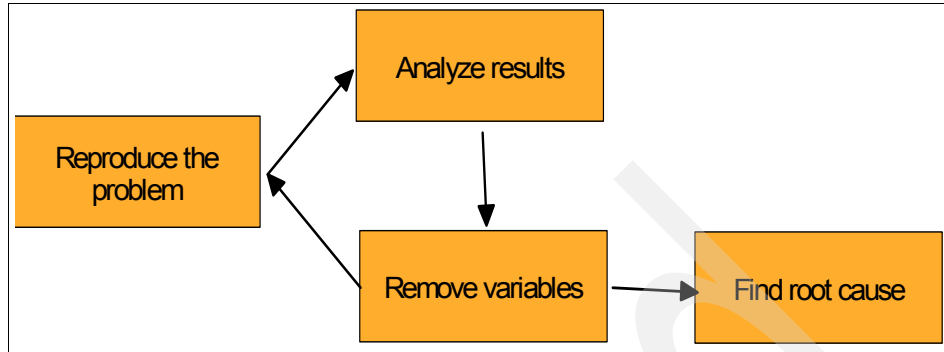


Figure 1-8 The isolation strategy

Now that we have discussed the general strategies for problem determination, we focus on the specific strategies for each problem classification.

## 1.4 Classify the problem and determine the root cause

The next step in the problem determination process, and a very important one, is classifying the problem. This book refers to many different classifications of problems. You need to identify which classification the problem fits into.

To get started, decide if your problem fits any of the following symptom categories:

- ▶ You cannot install or migrate WebSphere Application Server.  
This would occur during the installation or migration of the WebSphere Application Server product itself. See “Installation or migration” on page 34 for more information.
- ▶ You cannot install an application into WebSphere Application Server.  
This would occur when you attempt to install an application by using the administrative console or wsadmin interface. See “Application packaging and deployment” on page 35 for more information.
- ▶ You experience difficulties in WebSphere Application Server system management or configuration.

During normal administrative activities, an unexpected problem occurs. This might occur when configuring WebSphere Application Server through the administrative console or wsadmin, or when using any administrative scripts that are included with the product (for example, **addNode**, **collector**, **dumpNameSpace**, **ejbdeploy**, **keyman**, **Java2WSDL**, **wasprofile**, and so forth). This also includes problems with security, application assembly, and

performance monitoring. See “System management and configuration” on page 37 for more information.

- An application or WebSphere Application server process (for example, an application server, node agent, or deployment manager) is unable to start.

When running the **startServer**, **startNode**, or **startManager** commands to start a WebSphere Application Server process or when using the administrative console or wsadmin to start a process or an application, a problem occurs that causes the process or application to fail to start. See “Application or WebSphere process is unable to start” on page 42 for more information.

- An application does not respond to incoming requests.

This could occur due to a Web server, Edge component, or Web server plug-in problem, application server crash, hang, out of memory condition, or 100% CPU utilization condition. See “Application does not respond to incoming requests” on page 43 for more information about determining the root cause of the problem.

- An application produces unexpected results (possibly errors or exceptions).

An error is seen or an exception is thrown when certain application code is executed or when certain conditions (for example, heavy load) are met. The application could also behave differently than expected, but produce no error or exception. There are several components of WebSphere Application Server in which this type of problem could occur. See “Unexpected results from an application” on page 49 for more information.

- An application cannot connect to an external system or resource.

The application might need to access an external system or resource. This could be a database, a messaging system, an enterprise information system accessed through the Java Connector Architecture (JCA), an Enterprise JavaBeans™ (EJB™) running on a remote system, or a Web service. There might be a problem establishing a connection to the external system or resource, or an error might occur when the application interacts with it. See “Application cannot connect to an external system or resource” on page 63 for more information.

- An application performs slowly or its performance degrades over time.

Although performance problems fall outside the scope of this book, we do provide some external resources for performance problems. In general, performance problems can be corrected by tuning WebSphere Application Server, the other software products with which it interacts, and the operating system. Thorough performance testing and tuning should be completed before an application is put into production. See “Application is slow or its performance degrades over time” on page 69 for more information.



The WebSphere Application Server Support team has found that the majority of problems encountered by clients fit into one of these problem classifications. In this book, we provide comprehensive information about problem determination strategies for some of these common problem areas. For the problem classifications that are not covered in this book, we provide links to external resources that discuss problem determination for each problem type.

If you have already examined the WebSphere logs and have determined that a particular error message holds the key to the problem, use Table 1-1 as a quick reference to find information based on the message prefix.

*Table 1-1 Error message prefix to topic mapping*

Message prefixes	Topic
INST, ADMU (during profile creation) or WSVRT (during IVT)	"Installation" on page 34
MIGR	"Migration" on page 35
ADMA	"Application packaging and deployment" on page 35
SECJ, SECG, JSAS, JSSL, WSEC, or WSSK	"Security" on page 39
ADFS, ADMB, ADMC, ADMD, ADME, ADMF, ADMG, ADMK, ADML, ADMN, ADMR, ADMS, ADMU, BNDE, CHKC, CHKP, CHKS, CHKW, ECNS, ODCF, PROC, WACT, WSVM, WASX	"System management and configuration" on page 37
PLGN, PLGC, and PLPR	"Web server plug-in" on page 45
SRVE (Web container), JSPG (JSPs), or JSFG (JSFs)	"Web container" on page 50
CNTR, PMGR, and ACIN	"EJB container" on page 51
SESN	"Session management" on page 52
DYNA	"Dynamic cache" on page 54
WTRN, WLTC	"Transaction manager" on page 55
WWLM	"Workload management" on page 56
HMGR, CWRCB, CWWCW	"High availability manager" on page 57
CWWDR	"Data replication service" on page 58
WACS, APPR, ASYN, OBPL, SCHD, STUP, ACWA	"Program model extensions" on page 59

Message prefixes	Topic
I18N, LTXT	"Internationalization/Double byte character set" on page 61
CHFW, DCSV, HTPC, SSLC, TCPC, WSSC, and XMEM	"Transport channel service" on page 62
J2CA, WSCL (WebSphere client)	"JCA connection manager" on page 63
DSRA or CONM	"Database connections" on page 64
CWSIA, CWSIB, CWSIC, CWSID, CWSIE, CWSIF, CWSIH, CWSII, CWSIJ, CWSIK, CWSIL, CWSIM, CWSIN, CWSIO, CWSIP, CWSIQ, CWSIR, CWSIS, CWSIT, CWSIU, CWSIV, CWSIW, CWSIX, CWSIY, CWSIZ, CWSJA, CWSJB, CWSJC, CWSJD, CWSJO, CWSJQ, CWSJR, CWSJU, CWSJW, CWSWS, WMSG	"Messaging" on page 64
NMSV	"JNDI naming" on page 65
CORBA COMM_FAILURE ORBX	"ORB" on page 66
WSWS, SOAP, WSIF, CWWSG	"Web services" on page 67
PMON	"PMI and Tivoli Performance Viewer" on page 41

### 1.4.1 Installation or migration

This section addresses problems related to the installation of WebSphere Application Server or the migration from previous releases.

#### Installation

*Symptom:* You are having problems installing WebSphere Application Server, creating profiles, or running install verification tests (IVT). Symptoms include:

- ▶ The launchpad or installation wizard will not start or fails immediately.
- ▶ The installation wizard hangs.
- ▶ You have a problem creating a profile.
- ▶ The installation verification test (IVT) fails.
- ▶ You get error or warning messages during the installation process that begin with INST, ADMU (during profile creation), or WSVRT (during IVT).

**Note:** For problem determination strategies for installation problems, see *WebSphere Application Server V6: Installation Problem Determination* at:  
<http://www.redbooks.ibm.com/redpapers/pdfs/redp4068.pdf>

## Migration

*Symptom:* You are having problems migrating to WebSphere Application Server V6 from an earlier version of the product using the WebSphere Application Server migration tools. You might receive WebSphere system messages that begin with MIGR.

For problem determination strategies for migration problems, please review the following resources:

- ▶ WebSphere Information Center: *Migrating, coexisting, and interoperating*  
<http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6topmigrating.html>
- ▶ WebSphere Information Center: *Troubleshooting migration*  
[http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tmig\\_troubleshoot.html](http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tmig_troubleshoot.html)
- ▶ WebSphere Information Center: *Explanation of MIGR system messages*  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/MIGR.html>
- ▶ *MustGather: Migration problems*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21141284>
- ▶ *WebSphere Application Server V6 Migration Guide*, SG24-6369

## 1.4.2 Application packaging and deployment

*Symptom:* You are having problems deploying an application to WebSphere Application Server using the administrative console or wsadmin. You might receive WebSphere system messages that begin with ADMA.

For problem determination strategies for application installation and deployment problems, review the following resources:

- ▶ WebSphere Information Center: *Developing and deploying applications*  
<http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6topdeveloping.html>

- ▶ WebSphere Information Center: *Troubleshooting deployment*  
[http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb\\_deploy.html](http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb_deploy.html)
- ▶ WebSphere Information Center: Explanation of ADMA system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/ADMA.html>
- ▶ *MustGather: Problems during Deployment of EAR/JAR/WAR files*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21199344>
- ▶ *MustGather: Enhanced EAR file problems for V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21199181>

## WebSphere Rapid Deployment

*Symptom:* You are trying to use WebSphere Rapid Deployment to develop and are trying to test an application. You cannot connect to an application server or WebSphere Rapid Deployment does not create or update the applications.

**Note:** For issues with WebSphere Rapid Deployment, see *WebSphere Application Server V6: System Management Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4067.pdf>

## Application Server Toolkit

*Symptom:* You have a problem when using the Application Server Toolkit (AST) to assemble your applications. This encompasses all possible problems with the AST, including starting it, any problems that you experience with application assembly, and any errors that occur.

You can find problem determination strategies for AST in the following resources:

- ▶ WebSphere Information Center: *Overview of WebSphere Application Server Toolkit*  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.welcome.ast.doc/topics/astoverview.html>
- ▶ *MustGather: Application Server Toolkit V5 and V6 problems*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21194926>

### 1.4.3 System management and configuration

*Symptom:* You experience difficulties in WebSphere Application Server system management or configuration. This includes problems with the following system management tools or functions:

- ▶ Configuration and management using administration tools
- ▶ Security
- ▶ JMX clients
- ▶ PMI and Tivoli Performance Viewer

#### **Configuration and management using administration tools**

*Symptom:* You have a problem accessing or using the administrative console, wsadmin scripting tool, or command line scripts. This would include the following symptoms:

- ▶ You are not able to access the administrative console.
- ▶ You cannot access server processes using wsadmin or the management scripts such as stopServer.
- ▶ You are getting errors performing system management functions, for example managing application servers, node agents, Web servers, or applications.
- ▶ You cannot federate a node with a deployment manager.
- ▶ You are getting save conflict messages in the administrative console.
- ▶ Your enterprise applications no longer appear in the administrative console.

**Note:** For problem determination strategies for these system management problems, see *WebSphere Application Server V6: System Management Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4067.pdf>

#### **Resources for problems with scripting tools**

You can find problem determination strategies for issues with administrative scripting tools in the following resources:

- ▶ WebSphere Information Center: *Using Ant to automate tasks*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tovr\\_ant.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tovr_ant.html)
- ▶ WebSphere Information Center: *Using command line tools*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/txml\\_command.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/txml_command.html)

- ▶ WebSphere Information Center: *Troubleshooting administration*  
[http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb\\_admin.html](http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb_admin.html)
- ▶ WebSphere Information Center: Explanation of WASX system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/WASX.html>
- ▶ *MustGather: ws\_ant problems on V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196231>
- ▶ *MustGather: WebSphere configuration archive functionality*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21200348>

### ***Resources for other system management problems***

You can find problem determination strategies for other system management issues in the following resources:

- ▶ WebSphere Information Center: *Setting up the administrative architecture*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tagt\\_admin.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tagt_admin.html)
- ▶ WebSphere Information Center: *Administering application servers*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun\\_svr\\_conf.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_svr_conf.html)
- ▶ WebSphere Information Center: *Troubleshooting administration*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb\\_admin.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb_admin.html)
- ▶ WebSphere Information Center: Explanations of ADFS, ADMB, ADMC, ADMD, ADME, ADMF, ADMG, ADMK, ADML, ADMN, ADMR, ADMS, ADMU, BNDE, CHKC, CHKP, CHKS, CHKW, ECNS, ODCF, PROC, WACT, and WSVN system messages  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/welc\\_ref\\_trb\\_msg.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/welc_ref_trb_msg.html)
- ▶ *MustGather: System management functionality for V5.0, V5.1 and V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21199596>
- ▶ *MustGather: Synchronization problems in V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196219>
- ▶ *MustGather: Federation or Removal of a Node Issues for Version V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196227>
- ▶ *MustGather: Profile Creation/Removal Issues for V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196228>

- ▶ *MustGather: Usage and creation of templates fail on V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21195439>
- ▶ *MustGather: Node agent and Deployment Manager discovery problems for all releases and editions of V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196220>
- ▶ *MustGather: Port Management for V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196226>
- ▶ *WebSphere Application Server V6 System Management & Configuration Handbook, SG24-6451.*

## Security

*Symptom:* You have a problem configuring security in WebSphere Application Server, a problem that only occurs when security is enabled, or you have an error or exception related to security. You might receive WebSphere system messages that begin with SECJ, SECG, JSAS, JSSL, WSEC, or WSK.

**Note:** For information about problems with SSL configuration see *WebSphere Application Server V6: System Management Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4067.pdf>

You can find problem determination strategies for other security problems in the following resources:

- ▶ WebSphere Information Center: *Securing applications and their environments*  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6topsecuring.html>
- ▶ WebSphere Information Center: *Troubleshooting security configurations*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec\\_trouble.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec_trouble.html)
- ▶ WebSphere Information Center: *Troubleshooting authentication and authorization for Web services security based on Web Services for J2EE*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs\\_trbauth.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs_trbauth.html)
- ▶ WebSphere Information Center: *Explanation of SECJ system messages*  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/SECJ.html>

- ▶ WebSphere Information Center: Explanation of SECG system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/SECG.html>
- ▶ WebSphere Information Center: Explanation of JSAS system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/JSAS.html>
- ▶ WebSphere Information Center: Explanation of JSSL system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/JSSL.html>
- ▶ WebSphere Information Center: Explanation of WSEC system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/WSEC.html>
- ▶ WebSphere Information Center: Explanation of WSSK system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/WSSK.html>
- ▶ *MustGather: Problems using Global Security for all Releases of V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21199336>
- ▶ *MustGather: Problems using JAAS Web logins for all Releases of V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21199332>
- ▶ *MustGather: Problems using Web services Security for all Releases of V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21199335>
- ▶ *MustGather: Java Secure Socket Extension (JSSE), SSL or Java Cryptography Extensions (JCE) problems*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21162961>
- ▶ *WebSphere Application Server V6 Security Handbook, SG24-6316.*

## JMX clients

*Symptom:* Your organization has developed its own administrative client using the Java Management Extensions (JMX™) API. You have any kind of problem connecting to WebSphere Application Server or performing any JMX API operation.

You can find problem determination strategies for problems with JMX clients in the following resources:

- ▶ WebSphere Information Center: *Using administrative programs (JMX)*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tjmx\\_programming.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tjmx_programming.html)



- ▶ WebSphere Information Center: *Troubleshooting administration*  
[http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb\\_admin.html](http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb_admin.html)
- ▶ *MustGather: JMX API client for V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196218>

## PMI and Tivoli Performance Viewer

*Symptom:* You have a problem with the Performance Monitoring Infrastructure (PMI) or the Tivoli Performance Viewer. You might receive WebSphere system messages that begin with PMON. You might also have problems configuring PMI or performing any operation with the Tivoli Performance Viewer in the administrative console.

The PMI is instrumented by WebSphere Application Server V6 to monitor performance through several counters and statistics. You can view the performance data using the Tivoli Performance Viewer within the administrative console. For information about these features, see “Testing the application” on page 7.

You can find problem determination strategies for PMI and Tivoli Performance Viewer problems in the following resources:

- ▶ WebSphere Information Center: *Monitoring*  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6topmonitoring.html>
- ▶ WebSphere Information Center: *Tuning performance*  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6toptuning.html>
- ▶ WebSphere Information Center: Explanation of PMON system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/PMON.html>
- ▶ *MustGather: Performance monitoring infrastructure (PMI) and performance tool problems*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21141193>

## 1.4.4 Runtime

Problems that occur after installation of WebSphere Application Server or application deployment can include:

- ▶ Application or WebSphere process is unable to start
- ▶ Application does not respond to incoming requests

- ▶ Unexpected results from an application
- ▶ Application cannot connect to an external system or resource
- ▶ Application is slow or its performance degrades over time

## Application or WebSphere process is unable to start

*Symptom:* An application or WebSphere Application Server process (for example an application server, node agent, or deployment manager) is unable to start.

**Note:** For problems with the **startServer**, **stopServer**, **startNode**, **stopNode**, **startManager**, and **stopManager** commands, see *WebSphere Application Server V6: System Management Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4067.pdf>

You can find problem determination strategies for other startup issues in the following resources:

- ▶ WebSphere Information Center: *Starting servers*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/trun\\_svr\\_start.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/trun_svr_start.html)
- ▶ WebSphere Information Center: *Running application servers from a non-root user*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/trun\\_svr\\_nonroot.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/trun_svr_nonroot.html)
- ▶ WebSphere Information Center: *Running an Application Server and node agent from a non-root user*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun\\_nodeagent\\_nonroot.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_nodeagent_nonroot.html)
- ▶ WebSphere Information Center: *Running an application server from a non-root user and the node agent from root*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun\\_svr\\_msg\\_nonroot.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_svr_msg_nonroot.html)
- ▶ WebSphere Information Center: *The server process does not start or starts with errors*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rtrb\\_appsrvstart.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rtrb_appsrvstart.html)
- ▶ *MustGather: Application Server, dmgr and node agent Start and Stop Problems for all Releases of V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21204943>

## Application does not respond to incoming requests

*Symptom:* An application does not respond to incoming requests. The fault could lie with any of the following components or conditions:

- ▶ IBM HTTP Server
- ▶ Edge components
- ▶ Web server plug-in
- ▶ Application server crash
- ▶ Application server hang
- ▶ 100% CPU utilization
- ▶ Out of memory

### **IBM HTTP Server**

*Symptom:* An application server does not respond to incoming requests and static HTML pages are not being served by the IBM HTTP Server.

The IBM HTTP Server V6 is included with WebSphere Application Server V6. It can be administered within the administrative console. Other Web servers can be configured with WebSphere Application Server, and in general, the same problem scenarios occur with other Web servers.

When an application server does not respond to incoming requests, check to see if static HTML pages can be served by the HTTP Server. If not, the problem is probably caused by the HTTP Server. The HTTP Server process might have stopped unexpectedly (a crash), or it might be running but not responding to requests (a hang). You can check to see if the apache process is running on your operating system to determine if the process has crashed or if it is hanging. Other possible problems with HTTP Server include configuration issues, caching issues, authentication problems, and SSL problems. If the HTTP Server is able to serve static pages, the root cause of the problem is probably with the Web server plug-in or the application server itself.

**Note:** There is a possibility that the HTTP Server could fail to start due to a Web server plug-in problem. For more information about this, see *WebSphere Application Server V6: Web Server Plug-in Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4045.pdf>

You can find problem determination strategies for IBM HTTP Server issues in the following resources:

- ▶ IBM HTTP Server Support site

<http://www-306.ibm.com/software/webservers/httpservers/support>

- ▶ IBM HTTP Server, Version 6 Information Center  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/welcome\\_ihs.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/welcome_ihs.html)
- ▶ IBM HTTP Server, Version 6 Information Center, *Troubleshooting IBM HTTP Server*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/aes/ae/welc\\_troub.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/aes/ae/welc_troub.html)
- ▶ *MustGather: Read first for IBM HTTP Server*  
<http://www-1.ibm.com/support/docview.wss?rs=177&uid=swg21192683>

### **Edge components**

*Symptom:* An application does not respond to incoming requests. The application responds when accessed through the Web server but not when accessed through the Edge component.

The WebSphere Edge components (Load Balancer and Caching Proxy) are also included with WebSphere Application Server V6. The Edge components run on the *edge* of an enterprise system, that is the boundary between the external Internet and your Web server. The load balancer directs network traffic so that an equal workload arrives at each Web server in your environment. The caching proxy caches static and dynamic Web content from the Web server and WebSphere Application Server and serves the cached content in order to improve performance.

Potential problems from the Edge components include crashes, hangs, slow performance, unexpected patterns of workload distribution, and failover problems. To determine if your application is not responding due to an Edge problem, try issuing an HTTP request directly to your Web server (without going through either Edge component). If you get a response, the problem is likely with one of the Edge components. If not, then continue investigating the Web server, Web server plug-in, and application server.

You can find problem determination strategies for issues with Edge components in the following resources:

- ▶ WebSphere Edge Components Support site  
<http://www-306.ibm.com/software/webservers/edgeserver/support.html>
- ▶ IBM WebSphere Edge Components, Version 6 Information Center  
<http://www-306.ibm.com/software/webservers/appserv/doc/v60/ec/infocenter/index.html>

► Load Balancer, Version 6 *Troubleshooting*

<http://www-306.ibm.com/software/webervers/appserv/doc/v60/ec/infocenter/edge/LBguide.htm#HDRTRB>

### **Web server plug-in**

The Web server plug-in enables the Web server to send requests for dynamic content to Web applications (servlets and JSPs) that are installed in WebSphere Application Server.

*Symptom:* An application does not respond to incoming requests. The application responds when accessed directly through the application server but not when accessed through the Web server. High-level symptoms of a Web server plug-in problem include:

- Users cannot access an application through the Web server
- Load balancing and failover not working properly
- Session data is being lost
- Slow or intermittent application response
- The Web server will not start after plug-in installation or configuration

When your application fails to respond to requests, first check to see if your Web server is responding to requests for static HTML pages. If it is, you can narrow your focus to the plug-in and the application server.

To determine if it is a plug-in problem, try to access a servlet or JSP™ directly on the application server, bypassing the plug-in. You can do this by using the application server's HTTP transport port (port 9080 by default) in the URL to access the servlet or JSP. For example, you can try to access the snoop servlet, which is one of the WebSphere Application Server samples:

`http://localhost:9080/snoop`

If the servlet or JSP can be accessed through the HTTP transport but not through the Web server plug-in, you are likely experiencing a problem with the plug-in. On the other hand, if it cannot be accessed through the HTTP transport or the plug-in, it is probably an application server problem. Potential Web server plug-in problems include configuration issues (especially with the `plugin-cfg.xml` configuration file) and failures to send requests to an application server. WebSphere system messages for the plug-in begin with PLGN, PLGC, and PLPR.

**Note:** For problem determination strategies for the Web server plug-in, see *WebSphere Application Server V6: Web Server Plug-in Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4045.pdf>

## ***Application server crash***

*Symptom:* An application is not responding to incoming requests. The application server process is no longer running.

If an application is not responding to incoming requests and if you have ruled out the HTTP Server, Edge components, and Web server plug-in as the cause of the problem, you must be encountering an application server *crash* or *hang*. In both cases, the application server is unresponsive. The difference is that for a crash the application server process dies unexpectedly and for a hang the process is still running.

You should check to see if the application server process is running to determine if you are experiencing a crash or a hang. To do this, you need to know the process ID of the application server. You can find the process ID in the `<server>.pid` file in the `<WAS_install_root>/profiles/<profile>/logs/<server>` directory.

Open the `<server>.pid` file in a text editor. You see a four-digit number. This is the process ID. You can then use the appropriate operating system command to check to see if the process is running. If it is not running, the problem is a crash. If it is running, the problem is a hang.

**Note:** For problem determination strategies for application server crashes, see *WebSphere Application Server V6: Application Server Crash Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4059.pdf>

## ***Application server hang***

*Symptom:* An application is not responding to incoming requests but the application server process is still running.

If an application is not responding to incoming requests, you have ruled out the Web server, Edge components, and Web server plug-in as possible causes of the problem, and you have found that the application server process is still running, the classification of the problem is an application server hang.

You can find problem determination strategies for hang issues in the following resources:

- *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2 Diagnostics Guide*, SC34-6358

<http://www-106.ibm.com/developerworks/java/jdk/diagnosis>

► *Detecting hung threads in J2EE applications*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb\\_hangdetection.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_hangdetection.html)

► *Web module or application server dies or hangs*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_appdies.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_appdies.html)

► IBM Education Assistant: WebSphere Application Server problem determination

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27005460>

► You can find a list of MustGather documents for hangs with the following search argument

<http://www-1.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCMPB9&q=MustGatherDocument>

### **100% CPU utilization**

*Symptom:* You observe that a WebSphere Application Server process (an application server, cluster member, node agent, or deployment manager) reaches 100% or an unusually high percentage of CPU utilization.

You might notice this by using your operating system utilities to check the CPU use of each process on the system, or you might have an application server hang caused by the high CPU use.

You can find problem determination strategies for 100% CPU usage issues in the following resources:

► *MustGather: 100% CPU Usage on AIX® Platforms*

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21116458>

► *MustGather: 100% CPU Usage on HP-UX*

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21166017>

► *MustGather: 100% CPU usage on Linux®*

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21132699>

► *MustGather: 100% CPU Usage on Solaris™ platforms*

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21115625>

► *MustGather: 100% CPU Usage on Windows Platforms*

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21137447>

### **Out of memory**

*Symptom:* You observe that a WebSphere Application Server process (an application server, cluster member, node agent, or deployment manager)

consumes all of the available memory on your system, or you see a `java.lang.OutOfMemoryError` in the `SystemOut` or `SystemErr` log for the process.

You might notice this by checking the memory usage that is using your operating system utilities, by checking the logs, or by having an application server hang that is caused by an out of memory problem.

You can find problem determination strategies for out-of-memory problems in the following:

- ▶ For information about how IBM JVM components work and related diagnostic techniques see *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2 Diagnostics Guide*, SC34-6358  
<http://www-106.ibm.com/developerworks/java/jdk/diagnosis>
- ▶ For information about how the Virtual Memory Manager works on AIX, see *Performance Management Guide* article *Performance Overview of the Virtual Memory Manager (VMM)*  
[http://publib16.boulder.ibm.com/pseries/en\\_US/aixbman/prftungd/resmgmt2.htm](http://publib16.boulder.ibm.com/pseries/en_US/aixbman/prftungd/resmgmt2.htm)
- ▶ Articles on the IBM Research site about garbage collection  
<http://www.ibm.com/Search/?v=11&lang=en&cc=us&q=gc>
- ▶ *The Truth About Garbage Collection*  
[http://java.sun.com/docs/books/performance/1st\\_edition/html/JAppGC.fm.html](http://java.sun.com/docs/books/performance/1st_edition/html/JAppGC.fm.html)
- ▶ *Identifying memory leaks with the WebSphere Studio Profiler and the JDK™ Heapdump utility*  
[http://www-128.ibm.com/developerworks/websphere/library/techarticles/0411\\_persichetti/0411\\_persichetti.html](http://www-128.ibm.com/developerworks/websphere/library/techarticles/0411_persichetti/0411_persichetti.html)
- ▶ *How to read heapdump generated in .phd format*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21190476>
- ▶ *Memory Dump Diagnostic for Java technical preview*  
[http://www-128.ibm.com/developerworks/websphere/downloads/memory\\_dump.html](http://www-128.ibm.com/developerworks/websphere/downloads/memory_dump.html)
- ▶ MustGather documents  
<http://www-1.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCMPCY&q=MustGatherDocument>



## Unexpected results from an application

An application produces unexpected results (possibly errors or exceptions). This is a broad category that includes problems that can occur in any of the following runtime components or services:

- ▶ Just-In-Time (JIT) compiler
- ▶ Web container
- ▶ EJB container
- ▶ Classloader
- ▶ Session management
- ▶ Dynamic cache
- ▶ Transaction manager
- ▶ Workload management
- ▶ High availability manager
- ▶ Data replication service
- ▶ Program model extensions
- ▶ Internationalization/Double byte character set
- ▶ Transport channel service

### ***Just-In-Time (JIT) compiler***

*Symptom:* You receive intermittent unexpected results from an application. Disabling the JIT compiler resolves the problem.

Unexpected results or intermittent failures can indicate a problem with the JIT compiler. If possible, perform repeated runs through the application, both with a fresh restart and within an existing application process to determine if one of the following is true:

- ▶ Output is correct for initial few runs, and incorrect afterwards  
In this case, the code generated by the JIT compiler might be producing code that generates the incorrect results.
- ▶ Output is always incorrect  
This could be a JVM or JIT problem. A simple test is to narrow the problem to disable the JIT compiler and see if it goes away.

JIT uses the Mixed Mode Interpreter (MMI) execution mode to increase its performance. With MMI, methods are not compiled by the JIT until they have run the number of times specified by the `IBM_MIXED_MODE_THRESHOLD` setting. The default for this setting is 2000. That is why JIT problems might not show up immediately. If you are recreating a problem that you suspect might be a JIT-related problem, you can lower this threshold to speed up the time that the error occurs. Setting the threshold to 0 means that each method is compiled the first time it runs. Lowering the threshold has a performance impact, so remember to set it back when you are done.

For more information about mixed mode threshold, see:

<http://www.ibm.com/developerworks/java/jdk/diagnosis>

You can test to see if there is a problem with JIT by disabling JIT on the application server. It is important to note that disabling JIT causes performance degradation. Therefore, you should only disable JIT in your test environment. As soon as you test to see if the problem still occurs with JIT disabled, you should enable JIT again.

To disable JIT:

1. Log onto the administrative console.
2. Select **Servers** → **Application servers**.
3. Select your application server name.
4. On the resulting page, under Server Infrastructure, expand Java and Process Management, click **Process Definition**.
5. Select **Java Virtual Machine**.
6. Select **Disable JIT** and click **OK**.
7. Save your configuration and restart the application server.

If the problem does still occur when JIT is disabled, review the other problem classifications described here, determine into which classification the problem fits, and follow the problem determination steps for that classification.

### **Web container**

*Symptom:* You receive intermittent unexpected results from an application and the problem appears to originate with a Web resource (servlet, JSP). High-level symptoms include the following:

- ▶ Users cannot access a Web resource
- ▶ Unexpected behavior when running a Web resource
- ▶ Errors starting a Web module
- ▶ Problems with JSP compilation
- ▶ Errors or exceptions thrown in a Web module
- ▶ Error or warning messages that begin with SRVE (Web container), JSPG (JSPs), or JSFG (JSFs)

Problems areas that are related to the Web container also include session management problems and character encoding problems. For session management problem determination strategies, see the “Session management” on page 52. For character encoding problem determination strategies, see “Internationalization/Double byte character set” on page 61.

**Note:** For problem determination strategies for Web container problems, see *WebSphere Application Server V6: Web Container Problem Determination* at: <http://www.redbooks.ibm.com/redpapers/pdfs/redp4058.pdf>

### ***EJB container***

The EJB container is the runtime environment for EJBs in WebSphere Application Server and it handles all EJB requests from clients.

*Symptom:* You receive intermittent unexpected results from an application and the problem appears to originate with an EJB (entity beans, session beans, and message-driven beans). High-level symptoms can include any of the following:

- ▶ Unexpected behavior when an EJB runs
- ▶ Problems with the EJB life cycle and caching provided by the EJB container
- ▶ Any errors or exceptions that occur when running an EJB
- ▶ Error or warning messages with the following prefixes: CNTR, PMGR, and ACIN

Many errors might be related to data access and transactions.

You can find problem determination strategies for EJB container problems in the following resources:

- ▶ WebSphere Information Center: *EJB applications*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_ejb.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_ejb.html)
- ▶ WebSphere Information Center: *Enterprise bean and EJB container troubleshooting tips*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_ejbcntrcomp.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_ejbcntrcomp.html)
- ▶ WebSphere Information Center: *Cannot access an enterprise bean from a servlet, a JSP file, a stand-alone program, or another client*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_ejbaccessprobs.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_ejbaccessprobs.html)
- ▶ WebSphere Information Center: *Enterprise beans: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rejb\\_r4ln.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rejb_r4ln.html)
- ▶ WebSphere Information Center: Explanation of CNTR system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/CNTR.html>

- ▶ WebSphere Information Center: Explanation of PMGR system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/PMGR.html>
- ▶ WebSphere Information Center: Explanation of ACIN system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/ACIN.html>
- ▶ *MustGather: EJB container for releases of V4.0, V5.0, V5.1 and V6*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21153218>
- ▶ *MustGather: Persistence Manager*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21200344>

### ***Classloader***

*Symptom:* You experience a problem with the loading of classes and resources by WebSphere Application Server. Symptoms include:

- ▶ ClassCastException, ClassNotFoundException, NoClassDefFoundError, or UnsatisfiedLinkError in the SystemOut or SystemErr log. Your application might catch these exceptions and throw its own exceptions when they occur.
- ▶ Unexpected application behavior as a result of a class not being loaded or the wrong version of a class might be loaded if there are multiple classes with the same name on your system.

You can find problem determination strategies for classloader problems in the following resources:

- ▶ WebSphere Information Center: *Class loading*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/trun\\_classload.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/trun_classload.html)
- ▶ WebSphere Information Center: *Class loading: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rrun\\_classload\\_rlinks.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rrun_classload_rlinks.html)
- ▶ WebSphere Information Center: *Troubleshooting class loaders*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/ttrb\\_classload\\_viewer.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/ttrb_classload_viewer.html)
- ▶ *MustGather: Classloader Issues in V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196187>

### ***Session management***

A session (sometimes called an HTTP session) is a series of HTTP requests to a servlet from the same user using the same Web browser. WebSphere

Application Server provides session management functionality to keep track of each user and enable applications to provide personalized content.

*Symptom:* You experience a problem relating to the management of HTTP sessions by WebSphere Application Server. This can include any of the following:

- ▶ Unexpected session behavior
- ▶ Session time outs
- ▶ Problems with session storage (database persistence or memory-to-memory sessions)
- ▶ Session data being lost
- ▶ Errors or exceptions from the WebSphere Application Server session manager
- ▶ Problems with personalized Web content
- ▶ Error or warning messages with the prefix SESN

**Note:** For information about session management problems, see *WebSphere Application Server V6: Web Container Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4058.pdf>

**Note:** When there is a session management problem in a clustered environment, there might be an underlying Web server plug-in problem. For more information about this, see *WebSphere Application Server V6: Web Server Plug-in Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4045.pdf>

You can find other problem determination strategies for session management issues in the following resources:

- ▶ WebSphere Information Center: *Task Overview: Managing HTTP sessions*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tpsrs\\_sep1.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tpsrs_sep1.html)
- ▶ WebSphere Information Center: *Best practices for using HTTP Sessions*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprs\\_sess.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprs_sess.html)
- ▶ WebSphere Information Center: *HTTP session manager troubleshooting tips*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_httpsesncomp.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_httpsesncomp.html)

- ▶ WebSphere Information Center: *Problems creating or using HTTP sessions*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_httpsessprobs.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_httpsessprobs.html)
- ▶ WebSphere Information Center: *Managing HTTP sessions: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprs\\_r4ln.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprs_r4ln.html)
- ▶ WebSphere Information Center: Explanation of SESN system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/SESN.html>
- ▶ *MustGather: Sessions and session management problems in V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21192604>

Note that problems with the data replication service can cause session management problems (see “Data replication service” on page 58).

### ***Dynamic cache***

The dynamic cache feature caches the output of servlets, JSPs, and external Web services that are called by clients within WebSphere Application Server.

*Symptom:* You experience a problem with the dynamic cache feature of WebSphere Application Server. Potential problems can include any of the following:

- ▶ Unexpected or incorrect cache behavior
- ▶ Errors and exceptions from the dynamic cache component
- ▶ Dynamic cache configuration issues
- ▶ Slow performance
- ▶ Error or warning messages with the prefix DYNA

You can find problem determination strategies for dynamic cache issues in the following resources:

- ▶ WebSphere Information Center: *Dynamic caching*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_dyn.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_dyn.html)
- ▶ WebSphere Information Center: *Troubleshooting the dynamic cache service*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tdyn\\_probd.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tdyn_probd.html)

- ▶ WebSphere Information Center: *Troubleshooting tips for the dynamic cache service*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rdyn\\_trb.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rdyn_trb.html)
- ▶ WebSphere Information Center: Explanation of DYNA system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/DYNA.html>
- ▶ *MustGather: Dynamic cache problems in V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21193837>

Note that problems with the data replication service can cause dynamic cache problems (see “Data replication service” on page 58).

### ***Transaction manager***

A transaction is a unit of work that is done within your application where updates can be made to multiple resources so that the updates are either committed or rolled back together as one unit of work. The WebSphere Application Server service that coordinates transactions is called the transaction manager.

*Symptom:* You experience a problem relating to the management of transactions by WebSphere Application Server. This includes any of the following symptoms:

- ▶ Unexpected transaction behavior
- ▶ Transaction time outs
- ▶ Unexpected transaction rollbacks
- ▶ Problems with transaction recovery
- ▶ Any errors or exceptions from the transaction manager
- ▶ Error or warning messages that have the prefixes WTRN or WLTC

Transaction problems might also be related to database, messaging, or enterprise information system connection problems, which are described in “Application cannot connect to an external system or resource” on page 63.

You can find problem determination strategies for transaction manager problems in the following resources:

- ▶ WebSphere Information Center: *Transactions*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_jta.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_jta.html)
- ▶ WebSphere Information Center: *Troubleshooting transactions*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tjta\\_probd.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tjta_probd.html)

- ▶ WebSphere Information Center: *Tips for troubleshooting transactions*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rjta\\_prob0.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rjta_prob0.html)
- ▶ WebSphere Information Center: *Transaction service exceptions*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rjta\\_except.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rjta_except.html)
- ▶ WebSphere Information Center: Explanation of WTRN system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/WTRN.html>
- ▶ WebSphere Information Center: Explanation of WLTC system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/WLTC.html>
- ▶ *MustGather: Java Transaction Service (JTS) problems in V4.0, V5.0, V5.1 and V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21153216>

### **Workload management**

The workload management (WLM) functionality within WebSphere Application Server distributes client requests to the application servers that can process the requests most efficiently and also improves availability of applications by providing failover when application servers are not available. WLM is discussed more in “High availability and failover” on page 14.

*Symptom:* You experience a problem with the WLM features of WebSphere Application Server, including load balancing and failover of an EJB request or HTTP request. Potential problems include:

- ▶ Workload between cluster members not being balanced correctly
- ▶ A cluster member that is down does not failover
- ▶ Workload is not routed to a cluster member that is running
- ▶ An EJB client cannot reach any cluster members (a CORBA NO\_IMPLEMENT error is issued)
- ▶ Any errors or exceptions relating to WLM
- ▶ Error or warning messages with the prefix WWLM

**Note:** You can find problem determination strategies for issues with WLM of HTTP requests in *WebSphere Application Server V6: Web Server Plug-in Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4045.pdf>



You can find problem determination strategies for problems with WLM of EJB requests in the following resources:

- ▶ WebSphere Information Center: *Workload management (WLM) for distributed platforms*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun\\_wlm.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun_wlm.html)
- ▶ WebSphere Information Center: *Workload management component troubleshooting tips*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rtrb\\_wlmcomp.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rtrb_wlmcomp.html)
- ▶ WebSphere Information Center: *Workload is not getting distributed*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rtrb\\_wlmprobs.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rtrb_wlmprobs.html)
- ▶ WebSphere Information Center: *Workload management run-time exceptions*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rtrb\\_wlm\\_exceptions.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rtrb_wlm_exceptions.html)
- ▶ WebSphere Information Center: *Clustering and workload management: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rtrb\\_wlm\\_links.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rtrb_wlm_links.html)
- ▶ WebSphere Information Center: Explanation of WLM system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/WLM.html>
- ▶ *MustGather: Enterprise JavaBeans (EJB) workload management*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21052165>

### **High availability manager**

The high availability manager in WebSphere Application Server V6 eliminates single points of failure and provides failover for all applications and WebSphere Application Server services. It manages the availability of your applications and services. It is discussed in detail in “High availability and failover” on page 14.

*Symptom:* You experience a problem with the high availability manager in WebSphere Application Server, including

- ▶ One process in your core group fails and its services are not started on another process as expected.
- ▶ Errors or exceptions from the high availability manager.
- ▶ Error or warning messages that begin with HMGR, CWRCB, or CWWCW.

You can find problem determination strategies for high availability manager problems in the following resources:

- ▶ WebSphere Information Center: *Setting up a high availability environment*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun\\_ha\\_environment.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_ha_environment.html)
- ▶ WebSphere Information Center: *High availability manager*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun\\_ha\\_hamanager.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun_ha_hamanager.html)
- ▶ WebSphere Information Center: *Troubleshooting high availability environment problems*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rtrb\\_ha\\_env\\_trbl.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rtrb_ha_env_trbl.html)
- ▶ WebSphere Information Center: *Learning about high availability*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.pmc.nd.doc/tasks/tjt0014\\_.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.pmc.nd.doc/tasks/tjt0014_.html)
- ▶ WebSphere Information Center: Explanation of HMGR system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/HMGR.html>
- ▶ WebSphere Information Center: Explanation of CWRCB system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/CWRCB.html>
- ▶ WebSphere Information Center: Explanation of CWWCW system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/CWWCW.html>
- ▶ *MustGather: High Availability and the High Availability (HA) Manager*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21201016>

### ***Data replication service***

The data replication service transfers information between cluster members, enabling memory-to-memory session replication, dynamic cache, and stateful session bean failover.

*Symptom:* You experience a problem with the data replication service in WebSphere Application Server. Potential problems include:

- ▶ Unexpected data replication behavior (might appear as workload management, dynamic cache, or session management problems).
- ▶ Any errors or exceptions from the data replication service.
- ▶ Error or warning messages that begin with CWWDR.

You can find problem determination strategies for data replication service problems in the following resources:

- ▶ WebSphere Information Center: *Replicating data across application servers in a cluster*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun\\_drs\\_replication.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_drs_replication.html)
- ▶ WebSphere Information Center: *Replication*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun\\_drs\\_replication.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun_drs_replication.html)
- ▶ WebSphere Information Center: Explanation of CWWDR system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/CWWDR.html>

### ***Program model extensions***

Programming model extensions (PMEs) enhance application capability and performance, and make programming and deployment faster and more productive.

*Symptom:* You have any type of problem with the WebSphere Application Server extensions to the J2EE programming model. Potential problems include:

- ▶ Unexpected behavior, errors, or exceptions that occur when running an application that takes advantage of PMEs. These problems only occur with applications that use PMEs.
- ▶ Error or warning messages related to PMEs: (WebSphere system messages for each PME are in parentheses):
  - Activity session service (WACS)
  - Application profiling (APPR)
  - Asynchronous beans (ASYN)
  - Object pooling (OBPL)
  - Scheduler service (SCHD)
  - Startup beans (STUP)
  - Work area service (ACWA)
  - Internationalization service (I18N)

You can find problem determination strategies for PME problems in the following resources:

- ▶ WebSphere Information Center: *ActivitySessions*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_as.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_as.html)

- ▶ WebSphere Information Center: *Troubleshooting ActivitySessions*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ass/tasks/tas\\_probd.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ass/tasks/tas_probd.html)
- ▶ WebSphere Information Center: *ActivitySession service: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ass/ref/ras\\_rlinks.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ass/ref/ras_rlinks.html)
- ▶ WebSphere Information Center: Explanation of WACS system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/WACS.html>
- ▶ WebSphere Information Center: *Application profiling*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_appprof.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_appprof.html)
- ▶ WebSphere Information Center: Explanation of APPR system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/APPR.html>
- ▶ WebSphere Information Center: *Asynchronous beans*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_asb.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_asb.html)
- ▶ WebSphere Information Center: Explanation of ASYN system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/ASYN.html>
- ▶ WebSphere Information Center: *Object pools*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_objp.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_objp.html)
- ▶ WebSphere Information Center: *Object pools: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/asynbns/ref/rasb\\_objpoolrlinks.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/asynbns/ref/rasb_objpoolrlinks.html)
- ▶ WebSphere Information Center: Explanation of OBPL system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/OBPL.html>
- ▶ WebSphere Information Center: *Scheduler service*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_sch.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_sch.html)
- ▶ WebSphere Information Center: Explanation of SCHD system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/SCHD.html>

- ▶ WebSphere Information Center: *Startup beans*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_sub.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_sub.html)
- ▶ WebSphere Information Center: Explanation of STUP system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/STUP.html>
- ▶ WebSphere Information Center: *Work area*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_wa.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_wa.html)
- ▶ WebSphere Information Center: Explanation of ACWA system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/ACWA.html>

### ***Internationalization/Double byte character set***

*Symptom:* You have any type of problem with character encoding or the internationalization of your application (producing output in the language, time zone, currency, and cultural conventions for different regions or locales). This can include:

- ▶ Problems displaying double byte characters for certain languages.
- ▶ Problems that only occur with internationalized applications.
- ▶ Error or warning messages that begin with I18N and LTXT.

**Note:** Character encoding problems are discussed in *WebSphere Application Server V6: Web Container Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4058.pdf>

You can find problem determination strategies for other internationalization problems in the following resources:

- ▶ WebSphere Information Center: *Internationalization service*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_in.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_in.html)
- ▶ WebSphere Information Center: *Internationalization service errors*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/i18n/ref/rin\\_troubleshoot.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/i18n/ref/rin_troubleshoot.html)
- ▶ WebSphere Information Center: *Internationalization: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rin\\_resources.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rin_resources.html)

- ▶ WebSphere Information Center: Explanation of I18N messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/I18N.html>
- ▶ WebSphere Information Center: Explanation of LTXT system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/LTXT.html>
- ▶ *MustGather: i18n (Internationalization)/Double Byte Character Set (DBCS)*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21141732>

### ***Transport channel service***

The transport channel service is a new feature in WebSphere Application Server V6. It manages client connections and I/O processing for HTTP and JMS requests based on the new non-blocking I/O features in Java 1.4.

*Symptom:* Problems with the transport channel service would most likely manifest themselves as:

- ▶ Performance issues (where the transport channel service settings could be tuned)
- ▶ Unexpected results
- ▶ Error or warning messages that begin with CHFW, DCSV, HTPC, SSLC, TCPC, WSSC, and XMEM

You can find problem determination strategies for transport channel service problems in the following resources:

- ▶ WebSphere Information Center: *Configuring transport chains*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/trun\\_chain\\_transport.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/trun_chain_transport.html)
- ▶ WebSphere Information Center: *Transport chains*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/crun\\_chain\\_transport.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/crun_chain_transport.html)
- ▶ WebSphere Information Center: *Troubleshooting transport chain problems*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_chain\\_trbl.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_chain_trbl.html)
- ▶ WebSphere Information Center: Explanation of CHFW system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/CHFW.html>
- ▶ WebSphere Information Center: Explanation of DCSV system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/DCSV.html>

- ▶ WebSphere Information Center: Explanation of HTPC system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/HTPC.html>
- ▶ WebSphere Information Center: Explanation of SSLC system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/SSLC.html>
- ▶ WebSphere Information Center: Explanation of TCPC system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/TCPC.html>
- ▶ WebSphere Information Center: Explanation of WSSC system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/WSSC.html>
- ▶ WebSphere Information Center: Explanation of XMEM system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/XMEM.html>

## **Application cannot connect to an external system or resource**

These problems include:

- ▶ JCA connection manager
- ▶ Database connections
- ▶ Messaging
- ▶ JNDI naming
- ▶ ORB
- ▶ Web services

### ***JCA connection manager***

*Symptom:* You have any problem connecting to an external resource through the Java Connector Architecture (JCA) connection manager. This includes data sources (which are used to obtain JDBC™ database connections), JMS connection factories, and connections to enterprise information systems with an installed JCA resource adapter. The JCA connection manager pools and manages connections to these systems.

Potential problems include general connection pooling problems or configuration issues, problems establishing a connection, problems with activation specifications that get messages from your back-end systems. JCA connection symptoms are observed through WebSphere error messages with prefixes DSRA, WSCL, J2CA, WTRN, CONM, SQLException, or database error codes.

Problems caused by JCA components or JCA connection configuration errors can appear as one or more of the following initial symptoms:

- ▶ A connection hangs or incorrectly returns data to application
- ▶ An application connect to or access a database or EIS

**Note:** For problem determination strategies for JCA connection manager problems, see *WebSphere Application Server V6: JCA Connection Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4080.pdf>

### ***Database connections***

You have a problem that occurs when your application interacts with a database through JDBC. The connection to the database is obtained either through a data source (with the JCA connection manager) or a Version 4 data source (with the existing connection manager). Only J2EE 1.2 applications can use a Version 4 data source. These problems include JDBC Provider or data source configuration issues and any errors or WebSphere system messages that begin with DSRA or CONM.

**Note:** For problem determination strategies for database connection problems, see *WebSphere Application Server V6: JCA Connection Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4080.pdf>

### ***Messaging***

*Symptom:* You have a problem that occurs when your application interacts with the default messaging provider, a WebSphere MQ system, or a Generic JMS Provider through JMS. The connection to the messaging system is obtained through a JMS connection factory defined in the administrative console or wsadmin.

If you are using the Default messaging provider, problems might occur with the service integration bus. Symptoms of a problem in this area include :

- ▶ Problems sending and receiving messages
- ▶ Messaging engine startup problems
- ▶ Message data store connectivity or configuration issues
- ▶ Problems interacting with external messaging systems
- ▶ Configuration problems with a bus, messaging engine, destination, or mediation
- ▶ Error or warning messages generated by the service integration bus with the following prefixes:



CWSIA, CWSIB, CWSIC, CWSID, CWSIE, CWSIF, CWSIH, CWSII, CWSIJ, CWSIK, CWSIL, CWSIM, CWSIN, CWSIO, CWSIP, CWSIQ, CWSIR, CWSIS, CWSIT, CWSIU, CWSIV, CWSIW, CWSIX, CWSIY, CWSIZ, CWSJA, CWSJB, CWSJC, CWSJD, CWSJO, CWSJQ, CWSJR, CWSJU, CWSJW, and CWSWS. Note that these messages contain ten characters instead of nine.

- Problems with the WebSphere MQ JMS provider or a generic JMS provider might include JMS connection factory or destination configuration issues, message listener service and message-driven bean problems, and WebSphere system messages that begin with WMSG.

**Note:** For problem determination strategies for messaging problems, see *WebSphere Application Server V6: Default Messaging Provider Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4076.pdf>

### **JNDI naming**

The Java Naming and Directory Interface™ (JNDI) is used by applications to access EJBs and WebSphere resources such as data sources, JMS connection factories and destinations, and JCA connection factories.

*Symptom:* Your application experiences a problem when accessing any resource through a JNDI lookup. Indications of a JNDI problem include:

- JNDI naming error or exception in the SystemOut or SystemErr log.
- Problems using the dumpNameSpace tool.
- Error or warning messages that begin with NMSV.

You can find problem determination strategies for JNDI naming problems in the following resources:

- WebSphere Information Center: *Naming*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_nam.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_nam.html)
- WebSphere Information Center: *Troubleshooting name space problems*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tnam\\_troubleshoot\\_jndi.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tnam_troubleshoot_jndi.html)
- WebSphere Information Center: *dumpNameSpace tool*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rnam\\_dump\\_utility.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rnam_dump_utility.html)

- ▶ WebSphere Information Center: *Naming and directories: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.base.doc/info/aes/ae/rnam\\_r4ln.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.base.doc/info/aes/ae/rnam_r4ln.html)
- ▶ WebSphere Information Center: Explanation of NMSV system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/NMSV.html>
- ▶ *MustGather: Java Naming and Directory Interface (JNDI) and naming problems for all releases*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21143296>

## **ORB**

The ORB enables an EJB client to access an EJB running in WebSphere Application Server. An EJB client call to an EJB will go through the ORB unless the EJB client and EJB are running in the same WebSphere process.

*Symptom:* Your application experiences a problem when accessing an EJB through the Object Request Broker (ORB). Indications of an ORB problem include:

- ▶ ORB errors or exceptions in SystemOut or SystemErr log. Most likely, these would be CORBA error codes (such as a CORBA COMM\_FAILURE).
- ▶ Error or warning messages that begin with ORBX.

You can find problem determination strategies for ORB problems in the following resources:

- ▶ WebSphere Information Center: *Object Request Broker (ORB)*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_orb.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_orb.html)
- ▶ WebSphere Information Center: *Object Request Broker component troubleshooting tips*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_orbcomp2.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_orbcomp2.html)
- ▶ WebSphere Information Center: *Object Request Broker communications trace*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rorb\\_traceo.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rorb_traceo.html)
- ▶ WebSphere Information Center: *Object Request Brokers: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rorb\\_r4lno.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rorb_r4lno.html)

- ▶ WebSphere Information Center: Explanation of ORBX system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/ORBX.html>
- ▶ *MustGather: Object Request Broker (ORB) for all releases*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21157005>

### **Web services**

Web services are used to make an application or component available over the Internet, either within an organization or globally. WebSphere Application Server V6 supports the Web Services for J2EE standards.

*Symptom:* Your application experiences a problem when acting as a Web services client to a remote Web service or when acting as a Web service that is accessed by external clients. This would include:

- ▶ Web services errors and exceptions that appear in the SystemOut and SystemErr logs
- ▶ Unexpected Web services behavior
- ▶ Configuration issues
- ▶ Problems with Web services tooling, such as the Java2WSDL and WSDL2Java scripts
- ▶ Problems that occur when configuring Web services or Web services gateway instances with the service integration bus.
- ▶ Error or warning messages that begin with WSWS, SOAP, WSIF (for Web services), or CWSG (for Web services gateway)

You can find problem determination strategies for Web services problems in the following resources:

- ▶ WebSphere Information Center: *Web services*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech\\_wbs.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc6tech_wbs.html)
- ▶ WebSphere Information Center: *Troubleshooting Web services*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twbs\\_troubleshootwbs.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twbs_troubleshootwbs.html)
- ▶ WebSphere Information Center: *Troubleshooting Web services command-line tools*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs\\_trbcommand.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs_trbcommand.html)

- ▶ WebSphere Information Center: *Troubleshooting Web services compiled bindings*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs\\_trbjavacompiler.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs_trbjavacompiler.html)
- ▶ WebSphere Information Center: *Troubleshooting the runtime for a Web services client*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs\\_trbclientruntime.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs_trbclientruntime.html)
- ▶ WebSphere Information Center: *Troubleshooting serialization and deserialization in Web services*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs\\_trbserialize.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs_trbserialize.html)
- ▶ WebSphere Information Center: *Troubleshooting the Web Services Invocation Framework*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twsf\\_trouble.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twsf_trouble.html)
- ▶ WebSphere Information Center: *UDDI Registry troubleshooting*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twsu\\_probdet.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twsu_probdet.html)
- ▶ WebSphere Information Center: *Universal Discovery, Description, and Integration, Web Service, and SOAP component troubleshooting tips*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_svsccomp.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_svsccomp.html)
- ▶ WebSphere Information Center: *Errors returned to a client sending a SOAP request*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_soapprobs.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_soapprobs.html)
- ▶ WebSphere Information Center: *Tracing Web services*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twbs\\_tracewbscomp.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twbs_tracewbscomp.html)
- ▶ WebSphere Information Center: *Tracing SOAP messages with tcpmon*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twbs\\_tracewbs.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/twbs_tracewbs.html)
- ▶ WebSphere Information Center: *Frequently asked questions about Web services*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs\\_faq.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs_faq.html)

- ▶ WebSphere Information Center: *Web services: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs\\_resourceslearning2.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rwbs_resourceslearning2.html)
- ▶ WebSphere Information Center: Explanation of WWS system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/WWS.html>
- ▶ WebSphere Information Center: Explanation of SOAP system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/SOAP.html>
- ▶ WebSphere Information Center: Explanation of WSIF system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/WSIF.html>
- ▶ WebSphere Information Center: Explanation of CWSG system messages  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/CWSG.html>
- ▶ *MustGather: Web Services Engine problems for all releases and editions of WebSphere Application Server V5.0.2, V5.1 and V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21198363>
- ▶ *MustGather: Problems with the Web services gateway*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21159216>
- ▶ *MustGather: Problems with the Web Services Invocation Framework (WSIF)*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21159155>
- ▶ *MustGather: Problems with UDDI*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21164764>

## **Application is slow or its performance degrades over time**

This broad classification covers all problems with slow or degrading performance.

You can find problem determination strategies for performance problems in the following resources:

- ▶ WebSphere Information Center: *Tuning Performance*  
[http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc\\_concepts\\_cprf.html](http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/welc_concepts_cprf.html)
- ▶ WebSphere Information Center: *Troubleshooting performance*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tprf\\_troubleshoot.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tprf_troubleshoot.html)

- ▶ WebSphere Information Center: *Performance: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprf\\_resourceslearning.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprf_resourceslearning.html)
- ▶ *WebSphere Application Server Performance information*  
<http://www-306.ibm.com/software/webservers/appserv/was/performance.html>
- ▶ *WebSphere Application Server V6 Scalability and Performance Handbook*, SG24-6392.

Note that several runtime component failures can also cause overall system performance problems. These are discussed in:

- ▶ “Dynamic cache” on page 54
- ▶ “Workload management” on page 56
- ▶ “Transport channel service” on page 62
- ▶ “Web server plug-in” on page 45

## 1.5 Contacting IBM for support

As you might have realized by now, many potential problems with WebSphere Application Server V6 can be resolved without the help of the WebSphere Application Server Support team. By carefully preparing for and preventing problems and by following your problem determination strategy, you have the ability to resolve most problems on your own. However, there might be instances where invoking the Support team will be necessary, especially if a WebSphere Application Server code defect is the cause of a problem. This section explains the IBM support process and how you can work with the Support team most effectively.

In addition to the information that we provide here, there are two excellent resources on the Web about the IBM support process:

- ▶ *IBM Software Support Handbook*  
<http://techsupport.services.ibm.com/guides/handbook.html>
- ▶ *Steps to getting support for WebSphere Application Server*  
[http://www-106.ibm.com/developerworks/websphere/support/appserver\\_support.html](http://www-106.ibm.com/developerworks/websphere/support/appserver_support.html)

### 1.5.1 IBM support structure

In order to get the most out of the IBM support process, it is important to understand how the support organization is structured. There are several groups of support personnel with whom you might work for each problem:

► Front office (Problem entry) team

If you report a problem by phone, the first person you speak with is a member of the *front office* (also called *problem entry*) team. The people on this team work with you in your national language. They are usually located in the same region of the world from where you are calling.

The main goals of the front office team are to get a detailed problem description from you, confirm your contact information and availability, open a problem management record (PMR), and then route your PMR to the appropriate product support team.

► Front end (Level 1) team

The next step in the process is for the *front end* team (called the *Level 1* support team in the United States) to work on your PMR. If you open your PMR electronically, this would be the first team with whom you would work. This team also works with you in your national language, and they are usually located in the same region of the world as you.

The people on the front end team in many countries have broad skills in many IBM software products, and they work PMRs for several products. The Level 1 team in the United States only works WebSphere Application Server PMRs. They have broad skills in WebSphere Application Server, and they support the entire product. The support analysts obtain more details about the problem. They are likely to request files or logs that aid in determining the cause of the problem. They then conduct an investigation into the problem and provide possible solutions. If they are unable to provide a solution to the problem, they escalate the PMR to the next level of support.

► Back end (Level 2) team

Your PMR is escalated to the *back end* (Level 2) support team if it cannot be resolved by the front end or Level 1 team. The Level 2 team for WebSphere Application Server is located only in the United States, and they work only in English. The front end team asks you if you are able to work with the Level 2 team in English. If you would prefer to work in your own language, the front end team can translate the communication from the Level 2 team for you.

The Level 2 team is composed of a group of specialized teams. Each team works only on certain areas of the product, and they develop expertise in those areas. They conduct a more extensive analysis of the problem and the files that you have sent, and they obtain more information about the problem if it is required. The Level 2 team might also ask for more detailed diagnostic data, such as traces for a specific component, to help in their investigation. After analyzing all of the information and conducting research, the Level 2 team either provides a solution to the problem or works with the next level of support if they believe that the problem is caused by a WebSphere

Application Server code defect. In that case, an authorized program analysis report (APAR) is opened.

► Change team (Level 3)

If the Level 2 team's analysis shows that the problem is caused by a defect with WebSphere Application Server code, they send your PMR to the *change team*, also known as the Level 3 team. The change team is part of the WebSphere Application Server development team that is specifically focused on fixing defects. The members of the Level 3 team work only on one specific component of WebSphere Application Server. They have very deep knowledge of that particular component. The Level 3 team does not communicate directly with clients. Instead, they provide updates to the Level 2 team, and the Level 2 team communicates updates to the client.

## 1.5.2 Research the problem

Before you open a PMR, we recommend that you will follow all of the problem determination strategies discussed in this book. The most important thing to do is to research the problem. The WebSphere Application Server Support team has found that a large percentage of problems have already been reported by other clients. In these cases, you can save a great deal of time and effort by reviewing IBM documentation to make sure that the problem is not a known problem before you open a PMR. You can review the list of Web sites in "Investigate and research the problem" on page 27 to conduct your research.

## 1.5.3 Collect MustGather files

We first mentioned MustGather documents in "Diagnostic data collection" on page 21. These documents are useful guides when developing your diagnostic data collection plan, and they are also important when you open a PMR with the WebSphere Application Server Support team. The support team needs certain diagnostic data to investigate each type of WebSphere Application Server problem. In "Classify the problem and determine the root cause" on page 31, we listed each problem classification, and we included the links to the MustGather documents for each classification. You can review the specific document for the type of problem that you are experiencing, or you can start with the general MustGather document for all types of problems, *IBM - MustGather: Read first for all WebSphere Application Server products*, at:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21145599>

In either case, it is important that you provide the information and diagnostic data that is mentioned in the appropriate MustGather document when you open a PMR. You should expect that the front end or Level 1 support team will request it if they have not already received it.



## 1.5.4 Determine the severity

When you open a PMR, you provide a severity level for the PMR. The severity levels are in the range of 1 to 4, with 1 being the most severe and 4 being the least severe. The severity level reflects the business impact and, therefore, the urgency of the problem. Although it is tempting to open all PMRs at the Severity 1 level, it is very important to determine an accurate severity level for the situation. IBM support prioritizes PMRs according to the severity level. When you have a true Severity 1 issue, you do not want the support team to have to prioritize it at the same level as other issues that are not as critical. Table 1-2 provides the official descriptions of each severity level from the *IBM Software Support Handbook*.

**Note:** Although IBM support analysts might sometimes ask you if a severity level can be changed, it is ultimately your decision (or the decision of others in your company) as to what the severity level is.

Table 1-2 Guidelines of PMR severity levels

Severity level	Description
Severity 1	<b>Critical</b> business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
Severity 2	<b>Significant</b> business impact: The program is usable but is severely limited.
Severity 3	<b>Some</b> business impact: The program is usable with less significant features (not critical to operations) unavailable.
Severity 4	<b>Minimal</b> business impact: The problem causes little impact on operations, or a reasonable circumvention to the problem has been implemented.

To put these guidelines in WebSphere Application Server terms:

- **Severity 1** PMRs are typically situations where the availability of your production environment is affected.

In a case where your production system is down, the support team's first priority is to help you restore your production functionality, as discussed in "Revert to safe conditions" on page 24. It can be any situation where there is a high level of business impact. For example, it could be a problem that causes a loss of revenue or productivity for your organization. For Severity 1

PMRs, there is no work around or solution that is currently available. The support team works around the clock on your PMR, as long as they are able to contact you at any time of day, until the problem is resolved.

- **Severity 2** PMRs are the most common levels of PMRs. In this case, the problem has a significant impact, but the application is still running in your production environment. It can also include situations where your application is not yet in production.

These include situations where you have a deadline for resolving the problem, and situations where the problem delays when your application goes into production. Severity 2 PMRs are considered severe and are prioritized highly by the support team but the problems are not considered as critical as Severity 1 problems. The support team works with you during the normal business hours in your time zone.

- **Severity 3** PMRs indicate that a problem is not severely impacting your business.

Almost all Severity 3 problems occur in your test environment and do not affect production. If it does impact production, it has only a minor impact, and you might have a work around for the problem. Although you expect the problem to be resolved in a timely manner, you do not have a deadline for resolving the problem and the problem is not delaying the production date for your application. The support team works with you during the normal business hours in your time zone.

- **Severity 4** PMRs indicate that a problem has almost no business impact. This includes technical questions and requests to update documentation. It can also include problems that occur only in your test environment where a work around has already been provided.

The severity is sometimes lowered to Severity 4 when a solution is provided, but you need some time to test the solution before closing the PMR. It is expected that Severity 4 PMRs are not impacting your production environment. You do not have a deadline for resolving the problem and you are able to wait a longer period of time to receive a solution. The support team will work with you during the normal business hours in your time zone and updates by the support team might not be frequent.

### 1.5.5 Create a PMR

Now that you are familiar with the IBM support structure, you have researched the problem, collected the MustGather data, and determined the severity of the problem, it is time to open a PMR.

You can currently open a PMR in two ways, electronically or by phone. We have found that opening PMRs electronically is more convenient for most clients. We recommend that you use that method. However, phone support is still available.

### **Open a PMR electronically**

To open a PMR electronically on the Web, you can use the Electronic Service Request (ESR) problem submission tool at:

<http://www-306.ibm.com/software/support/probsub.html>

When you open a PMR with the ESR tool, the PMR is placed on the queue of the front end or Level 1 support team. A queue is a list of PMRs that need the attention of a particular support team, which are organized according to the PMR severity.

The ESR tool allows you to use your own words to describe the problem as thoroughly as you can. This ensures that the problem details are recorded accurately because you do not have to rely on someone else typing the description for you. You can also update the PMR whenever you want to, and you can log onto the ESR tool to view updates that have been made by the support team. You are notified by e-mail when the PMR has been updated. In addition, the ESR tool allows you to attach files to the PMR. These files are uploaded automatically to the FTP server that is used by the support team. The PMR is updated with the names and location of your files on the FTP server. This enables the support team to start reviewing your files immediately after you open the PMR.

### **Open a PMR by phone**

Your other option is to open a PMR by phone. The phone number that you call to open a PMR varies by country and changes from time to time. To find the phone number for your country, check this link from the *IBM Software Support Handbook*:

<http://techsupport.services.ibm.com/guides/contacts.html>

When you open a PMR by phone, you speak with a member of the front office or problem entry team. They open the PMR and provide you with the PMR number. They ask you for a problem description and then record that problem description in the PMR. They obtain your contact information, availability, and the severity level of the problem. Then, they forward the PMR to the front end or Level 1 team that will begin working with you on the problem.

If you do not use the ESR tool, you cannot view updates to the PMR. You can call the IBM support phone number for your country and ask them to update the PMR on your behalf when necessary. You also cannot attach files to the PMR. You need to either e-mail or FTP files to the support team.

## 1.5.6 Send data to IBM

In the course of working a PMR, it might be necessary to send many different types of data to the WebSphere Application Server Support team. As discussed in “Create a PMR” on page 74, using the ESR tool to attach files to your PMR makes this process easy. However, if you do not use the ESR tool, you can still send data through e-mail or FTP. As a general rule, you should use e-mail if the total size of the data is less than 10 MB, and you should use FTP if the data is more than 10 MB in size. Also, when you FTP data, your PMR is updated automatically so that the support team is aware that you sent the data and they can begin analyzing it right away. However, if you send an e-mail, be sure to either update your PMR electronically or call the IBM support phone number for your country and ask to have the PMR updated, so that the support team knows that you sent the data.

We recommend that you send all e-mail related to your PMR to `weblev2@us.ibm.com`. This is a shared e-mail address. The entire WebSphere Application Server Level 1 and Level 2 teams have access to this e-mail address. If you only send an e-mail to a specific support analyst, only that analyst has access to your e-mail. This can be disadvantageous, especially if the analyst is out of the office or if you need help outside of normal business hours.

For the complete and up-to-date instructions on sending e-mail or sending data via FTP to the WebSphere Application Server Support team, see *Submitting Diagnostic Information to IBM Technical Support for Problem Determination* at:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21153852>

## Diagnostic data

This paper contains information about the diagnostic data that is available in WebSphere Application Server V6. It contains information about the location of the data, how it is collected, and configuration options.

It includes information about the following:

- ▶ JVM logs (SystemOut and SystemErr)
- ▶ Tracing
- ▶ Collector tool
- ▶ First Failure Data Capture (FFDC)
- ▶ Process (native) logs
- ▶ Service log (activity.log)
- ▶ Installation logs
- ▶ IBM HTTP Server and plug-in logs and traces
- ▶ System management logs
- ▶ WebSphere Rapid Deployment logs

## 2.1 JVM logs (SystemOut and SystemErr)

SystemOut and SystemErr logs are created for every WebSphere Application Server process (application server, cluster member, node agent, and deployment manager). These logs are known as JVM logs. The System.Out and System.Err streams for each JVM are redirected to the SystemOut and SystemErr logs. WebSphere Application Server writes to these logs. Your applications can also write to them by using the `print()`, `println()`, and `printStackTrace()` methods.

You can find the logs in the following directory:

```
<WAS_install_root>/profiles/<profile>/logs/<process>
```

To configure the properties of these logs from the administrative console, select **Troubleshooting** → **Logs and Trace**. Select the process whose logs you want to configure, and then click **JVM Logs**. The General Properties window opens, as shown in Figure 2-1.

The screenshot shows the 'General Properties' window for JVM logs in the WebSphere Administrative Console. The window has two tabs: 'Configuration' and 'Runtime'. The 'Configuration' tab is selected. The window is titled 'General Properties' and has a sub-header 'System.out'. The 'File Name' field is set to 'R\_LOG\_ROOT/SystemOut.log'. The 'File Formatting' dropdown is set to 'Basic (Compatible)'. The 'Log File Rotation' section has two options: 'File Size' (checked) and 'Time' (unchecked). Under 'File Size', the 'Maximum Size' is set to '10 MB'. Under 'Time', the 'Start Time' is set to '24' and the 'Repeat Time' is set to '24 hours'. The 'Maximum Number of Historical Log Files' is set to '5'. The 'Installed Application Output' section has two options: 'Show application print statements' (checked) and 'Format print statements' (checked).

Figure 2-1 Changing the log file rotation properties

In the Configuration tab, you can edit the following:

- ▶ The file name (and the directory) of the SystemOut and SystemErr logs
- ▶ The file formatting

We recommend that you leave this at the default value of Basic to make the logs easier to read.

- ▶ Log file rotation

The SystemOut and SystemErr logs are self-managing. They write to the specified file until either the maximum file size or a certain time is reached. When that happens, the current log file is renamed as the current file name plus the current time stamp. Then a new SystemOut or SystemErr file is created for further logging. The older log files are called historical log files.

For example, after this occurs you might have the following SystemOut files in the `<WAS_install_root>/profiles/<profile>/logs/<process>` directory:

- SystemOut.log, the current log file
- SystemOut\_05.06.07\_10.28.48.log, the historical log file

Depending on your needs, you can choose to have the log files rotate (roll over) when they reach a specified size, a certain time interval, or both. If you choose a time, we recommend that you specify 24 hours as the Repeat Time. You can also set the Start Time to specify the time at which the logs rotate.

If you specify a file size, we recommend that you increase the Maximum Size above its default of 1 MB. You will want to coordinate the value of Maximum Size with the Maximum Number of Historical Log Files, based on the available disk space on your system. With either method, make sure that the amount of log data that is saved is enough so that the relevant log data is there when you identify that a problem has occurred.

- ▶ Maximum Number of Historical Log Files

The value that is entered here is the number of historical log files that are kept. If the value is reached and another historical log file needs to be created, the oldest one is removed from your system.

- ▶ Installed Application Output

These properties affect how `print` and `println` statements from your applications are output. There are two options:

- Show application print statements. This is enabled by default. If you deselect it, application `print` and `println` statements are not logged to the SystemOut and SystemErr log files.
- Format print statements. This is also enabled by default. You can deselect it if you do not want your application `print` and `println` statements to be

formatted similar to WebSphere Application Server messages in the log files.

All of these properties can be changed for both the SystemOut and the SystemErr logs. You can choose to use the same properties for both logs, which we recommend, or use different properties for them.

You can view the SystemOut and SystemErr files on the file system with a text editor or you can view them within the administrative console. It might be useful to view them in the administrative console if you need to view them from a remote system. To view them in the administrative console, select the Runtime tab and then click **View** (next to the SystemOut or SystemErr file name).

The entries in the output of the SystemOut.log are in the following format:

```
[7/12/05 14:46:00:264 EDT] 0000001a ApplicationMg A WSVR0221I: Application
started: adminconsole
```

Each entry can be deciphered as follows:

- Time stamp

In the example, the time stamp is [7/12/05 14:46:00:264 EDT].

The time stamp is formatted using the locale of the process where it is formatted. It includes a fully qualified date (for example MM/DD/YY), 24-hour time with millisecond precision, and a time zone.

- Thread ID

In the example, the thread ID is 0000001a.

The thread ID is an eight-character hexadecimal value that is generated from the hash code of the thread that issued the message.

- Short name

In the example, the short name is ApplicationMg.

The short name is the abbreviated name of the component that issued the message. This name is typically the class name of a WebSphere Application Server component and would be some other identifier for applications.



► Event type

In the example, the event type is A.

The event type is a one character field that indicates the type of the message. The possible values are:

- F - fatal message
- E - error message
- W - warning message
- A - audit message
- I - informational message
- C - configuration message
- D - detail message
- O - message that was written directly to System.out by an application or internal components
- R - message that was written directly to System.err by the user application or internal components
- Z - a placeholder to indicate that the type was not recognized

► Message Identifier

In the example, the message identifier is WSVR0221I.

The message identifier is a string that is nine characters in length and is in the form CCCC1234X. The first four characters (CCCC) indicate the WebSphere Application Server component that issued the message. The next four characters (1234) indicate the specific message that the component is issuing. The last character (X) indicates the severity of the message. Its value is either I (informational), W (warning), or E (error).

You can find descriptions of all WebSphere Application Server message identifiers in the WebSphere Information Center item *Troubleshooter reference: Messages* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/welc\\_ref\\_trb\\_msg.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/welc_ref_trb_msg.html)

► Message

In the example, the message is Application started: adminconsole.

The message is the data that is logged to the SystemOut.log by the component. It is meant to provide useful output for informational purposes, debugging, and troubleshooting.

## 2.2 Tracing

Trace logs can also be configured in a manner similar to the JVM logs. Traces must be explicitly enabled. They are disabled by default. Trace output gives very detailed information about the execution of WebSphere Application Server code. It provides time stamps, details about which WebSphere methods were called, and special diagnostic data that is included to make troubleshooting easier.

To configure the properties of the traces from the administrative console:

1. Select **Troubleshooting** → **Logs and Trace**.
2. Select the process whose trace logs you want to configure.
3. Click **Diagnostic Trace**. The General Properties window opens, as shown in Figure 2-2.

**General Properties**

☒ Enable Log

**Trace Output**

☐ Memory Buffer

\* Maximum Buffer Size  
8 thousand entries

☒ File

\* Maximum File Size  
20 MB

\* Maximum Number of Historical Files  
1

\* File Name  
\${SERVER\_LOG\_ROOT}/trace.log

Trace Output Format  
Basic (Compatible)

Apply OK Reset Cancel

Figure 2-2 Changing the trace properties

The configuration of the trace file properties is very similar to the configuration of the JVM logs. You want to ensure that Enable Log remains selected. For Trace Output, we recommend that you always select File instead of Memory Buffer so that the trace logs are easier to manage.

Trace files cannot be rolled over based on time. You must specify a Maximum File Size in conjunction with the Maximum Number of Historical Files. You should set these values appropriately, depending on how long it might take to reproduce the problem with trace enabled and how much disk space is available. With these properties set, the trace files roll over in the same manner as the JVM logs. You can also specify a File Name and specify a directory for the trace files.

As with the JVM logs, we strongly recommend selecting Basic (the default value) for the Trace Output Format. This makes the trace easier to read, and it is the preferred format of the WebSphere Application Server support team.

When you view the trace properties in the administrative console, you notice that there are two tabs, a Configuration tab and a Runtime tab. You can enable tracing on either tab. The difference is that when you use the Configuration tab, you must restart the WebSphere Application Server process before the tracing begins. When you use the Runtime tab, tracing begins as soon as you click **OK** or **Apply**. In many production environments, it is preferable to enable trace using the Runtime tab so that you do not have to restart the WebSphere process.

After the trace properties are configured, you must decide which WebSphere Application Server components to trace. To do this in the administrative console:

1. Select **Troubleshooting** → **Logs and Trace**.
2. Select the process whose trace logs you want to configure
3. Click **Change Log Level Details**.

As with the trace properties, the log level details can be set on the Configuration tab or the Runtime tab.

In the Change Log Detail Levels screen (Figure 2-3 on page 84), you can select which *components* and *groups* to trace.

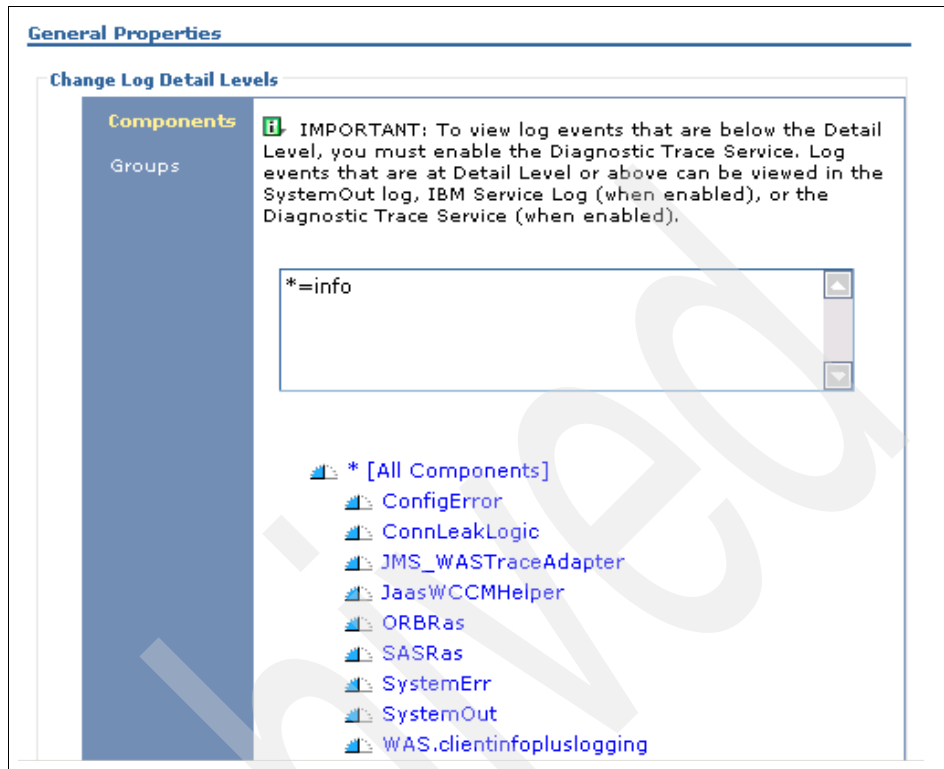


Figure 2-3 Changing the log detail level in the administrative console

Components can be any WebSphere Application Server packages or classes. Groups are predefined sets of packages and classes that are useful for troubleshooting a particular component.

The default log detail level is `*=info`. The log detail levels in WebSphere Application Server V6 are configured differently than the trace specifications in V5 and V5.1. If you use a V5 style trace specification as the log detail level in V6, it is mapped to the *most similar* V6 log detail level. However, to ensure that the correct tracing is enabled, we recommend only using the V6 log detail levels.

You can get a complete overview of the V6 log detail levels and how they relate to the V5 and V5.1 trace specifications in the *Log level settings* section of the WebSphere Information Center at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/utrb\\_loglevel.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/utrb_loglevel.html)

The MustGather documents for WebSphere Application Server components discuss which specific log detail level to set for different types of problems. It is a good idea to record the log detail level for different types of problems in your diagnostic data collection plan. When setting the log detail level, you should set the level to `all` in almost all cases. You should also include `*=info` in the beginning of the log detail level so that informational logging is enabled for components that are not being traced. For example, in order to enable a trace for the J2C connection manager component, you would set the log detail level to:

```
*=info:WAS.j2c=all
```

You can specify as many components and groups as you wish. You can view the components and groups that can be traced in the administrative console. However, the more components and groups you trace, the larger the trace output will be. It is a good idea to decide on a specific log detail level before enabling a trace.

After enabling the trace, clear out any old trace files from the `<WAS_install_root>/profiles/<profile>/logs/<server>` directory. Then, if you configured the trace properties and log detail level properties on the Configuration tab, restart the application server or process that you want to trace. If you configured the trace properties and log detail level on the Runtime tab, the tracing starts immediately without restarting the process. At this point, you can reproduce the problem and then disable tracing after you have reproduced it. This helps ensure that the trace file does not grow too large and makes it easier to find the time at which the problem occurs in the trace.

The trace output contains all of the messages that are also written to the `SystemOut.log` as well as the trace events. The trace events are in the following format:

```
[7/12/05 16:13:10:379 EDT] 00000032 DSConfigurati > getPooledConnection Entry
```

Each entry can be deciphered as follows:

- Time stamp

In the example, the time stamp is `[7/12/05 16:13:10:379 EDT]`.

The time stamp is formatted using the locale of the process where it is formatted. It includes a fully qualified date (for example `MM/DD/YY`), 24-hour time with millisecond precision, and a time zone.

- Thread ID

In the example, the thread ID is `00000032`.

The thread ID is an eight-character hexadecimal value generated from the hash code of the thread that issued the trace event.

► Short name

In the example, the short name is DSConfigurati.

The short name is the abbreviated name of the component that issued the trace event. This is typically the class name of a WebSphere Application Server component, and would be some other identifier for applications.

► Event type

In the example, the event type is a greater than symbol (>).

The event type is a one character field that indicates the type of the trace event. The possible values are:

- > - indicates the entry of the specified method name
- < - indicates the exit of the specified method name
- 1 - a trace entry of type fine or event
- 2 - a trace entry of type finer
- 3 - a trace entry of type finest, debug, or dump
- Z - a placeholder to indicate that the trace type was not recognized

The example indicates that the getPooledConnection method is entered.

► Class name

The class name is an optional part of the trace entry. It indicates the class that generated the trace event. In the example, the class name does not appear.

► Method name

The method name is another optional part of the trace entry. It indicates the method that generated the trace event. In the example, the method name is getPooledConnection.

► Text message

In the example, the message is Entry.

The text message is the data that is written to the trace output file. It is meant to provide advanced debugging and troubleshooting information.

► Parameters

Optionally, parameters can also be included in the trace entry. In the example, there are no parameters.

## 2.3 Collector tool

The WebSphere Application Server collector tool is a script that can be found in the `<WAS_install_root>/bin` directory (`collector.bat` or `collector.sh`). Running the script produces a Java archive (jar) file that contains all of the logs and XML configuration files from your WebSphere Application Server installation, as well as operating system information, Java information, and data on whether the software prerequisites were met and their levels.

You should always run the collector tool under the root or administrator user ID, and it must be run from a directory outside of the `<WAS_install_root>` directory. The resulting jar file is created in the current directory and the file name is `<host>-<cell>-<node>-<profile>-WASenv.jar`. For example:

```
C:\tmp\ServerHost1-Cell101-CellManager01-Dmgr01-WASenv.jar
```

The resulting jar file is very useful to the WebSphere Application Server support team and any others who are involved in the problem determination process. It allows them to view quickly your WebSphere Application Server configuration and see any errors or exceptions that have occurred.

You can find more information about the collector tool in the following WebSphere Information Center items:

- ▶ *Gathering information with the Collector tool*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb\\_ct.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_ct.html)
- ▶ *Running the collector tool*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb\\_runct.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb_runct.html)
- ▶ *Analyzing collector tool output*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb\\_readct.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_readct.html)

## 2.4 First Failure Data Capture (FFDC)

WebSphere Application Server V6 includes a feature called First Failure Data Capture (FFDC). The FFDC feature runs in the background and collects events and errors that occur during WebSphere Application Server runtime. The information that it collects are written to log files in the `<WAS_install_root>/profiles/<profile>/logs/ffdc` directory.

FFDC does not affect the performance of WebSphere Application Server and should not be disabled. The FFDC logs will not, most likely, be useful in your

problem determination efforts. However, they might be useful to the WebSphere Application Server support team if you open a PMR.

There are three FFDC configuration files in the `<WAS_install_root>/properties` directory. The only file that you should modify is the `ffdcRun.properties` file. You can add the `ExceptionFileMaximumAge` property to the file. This property specifies the number of days that an FFDC log remains in the `<WAS_install_root>/profiles/<profile>/logs/ffdc` directory before it is deleted. As part of your diagnostic data collection plan, you might want to modify the `ExceptionFileMaximumAge` property to ensure that the FFDC files remain on your system for a certain time period. You should not modify any other properties unless you are asked to do so by the WebSphere Application Server support team.

You can find more information about the FFDC feature in the WebSphere Information Center item *First failure data capture* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb\\_ffdc.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_ffdc.html)

## 2.5 Other logs

The following logs are not always useful for problem determination, but you might find that on occasion they will be required.

### 2.5.1 Process (native) logs

Native code running in a WebSphere Application Server process can write data to the process logs (also called native logs). Native code is non-Java code, typically found in files with `.dll`, `.exe`, and `.so` extensions. The process logs are named `native_stdout.log` and `native_stderr.log`. They are located in the `<WAS_install_root>/profiles/<profile>/logs/<server>` directory.

The only configuration that is possible for the process logs is changing the directory location or file names for the logs. You can do this in the administrative console:

1. Select **Troubleshooting** → **Logs and Trace**.
2. Select the WebSphere Application Server process.
3. Select **Process Logs**.



You can find more information about the process logs in the WebSphere Information Center item *Process logs* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb\\_stdlogs.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_stdlogs.html)

## 2.5.2 Service log (activity.log)

The service log is more commonly known as the activity.log and is found in the `<WAS_install_root>/profiles/<profile>/logs` directory. There is only one activity.log for each node. WebSphere Application Server runtime events are logged to the activity.log. It is written in binary format, so it cannot be viewed in a text editor. The main purpose of the activity.log is that it can be viewed with the Log Analyzer tool, is a graphical user interface that displays the events from the activity.log and uses a symptom database to analyze the events and diagnose problems.

You can find more information about the Log Analyzer in the WebSphere Information Center item *Log Analyzer* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb\\_jfla.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_jfla.html)

It is also possible to view the events in the activity.log outside of the Log Analyzer by using the showlog script in the `<WAS_install_root>/bin` directory.

You can find details about the usage of the **showlog** script in the WebSphere Information Center item *Showlog Script* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_showlog.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_showlog.html)

You can configure properties of the activity.log in the administrative console:

1. Select **Troubleshooting** → **Logs and Trace**.
2. Select the WebSphere Application Server process.
3. Select **IBM Service Logs**.

You can select whether to enable or disable the activity.log, choose the directory location and file name, set the maximum file size, and select what types of messages will be logged.

You can find more information about the service log in the WebSphere Information Center item *Viewing the service log* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_viewsvclog.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_viewsvclog.html)

### 2.5.3 Installation logs

When you have a problem installing WebSphere Application Server V6, you might need to view the following logs to determine the failure causes.

- ▶ `<WAS_install_root>/logs/log.txt`  
This log file records the installation status
- ▶ `<WAS_install_root>/profiles/<profile>/logs/pctlog.txt`  
This log file records the profile creation status
- ▶ `<WAS_install_root>/profiles/<profile>/logs/ivtClient.log`  
This log file records the events of install verification test

## 2.6 IBM HTTP Server and plug-in logs and traces

When you have a problem relating to the IBM HTTP Server or the Web server plug-in, you might need to view the logs or enable a trace. This section discusses the details about these logs and traces.

### 2.6.1 IBM HTTP Server logs

The IBM HTTP Server writes two log files: an access log that contains details of all accesses to the Web server and an error log that contains details of any errors. The default location of the logs is as follows:

- ▶ Windows
  - Access log: `<WAS_install_root>\logs\access.log`
  - Error log: `<WAS_install_root>\logs\error.log`
- ▶ UNIX®
  - Access log: `<WAS_install_root>/logs/access_log`
  - Error log: `<WAS_install_root>/logs/error_log`

### 2.6.2 Web server plug-in logs

The plug-in also writes its own log, which you can find in the Web server plug-in install directory path. The log file that you are looking for is under another directory structure, named for the logical Web server as defined in the WebSphere configuration.

You can find the location of the log file by first looking at the Web server configuration. This refers to the plug-in configuration file as shown in Example 2-1 on page 91. The plug-in configuration file then tells you where the

log file is as shown in Example 2-1. This example also shows you where you set the amount of detail that is logged.

*Example 2-1 Location of plug-in log file*

---

```
<Log LogLevel="Error"
Name="c:\ibm\was6\plugins\logs\webserver1\http_plugin.log" />
```

---

The default setting for LogLevel is Error, but you can set it to Trace to collect significantly more information. Should you need to raise this problem with IBM Support, they will request a plug-in trace.

### 2.6.3 Web server plug-in trace

To get an effective trace, you need to enable as much logging as possible on the Web server. For example, you can set the logging level to capture verbose output in the IBM HTTP Server by modifying the LogLevel directive in the configuration file as shown:

```
LogLevel debug
```

You need to restart the IBM HTTP Server for this to take effect.

Enable trace logging in the Web server plug-in by setting the LogLevel directive in the plugin-cfg.xml file as shown:

```
<Log LogLevel="Trace"
Name="c:\ibm\was6\plugins\logs\webserver1\http_plugin.log" />
```

You do not need to restart the IBM HTTP Server for this change to take effect.

**Tip:** The plug-in trace generates significant amounts of data. Make your test as specific as possible, and run it in isolation to reduce the number of lines generated.

### 2.6.4 Network trace

In rare cases, you might need to use a network protocol analyzer that allows you to capture an iptrace. This tool can help you to determine where the problem lies. WebSphere Application Server does not supply such a tool. However, there are third-party tools available (for example, Ethereal from <http://www.ethereal.com/>).

## 2.7 System management logs

When you have a system management problem, you might need to view certain logs or enable a trace. This section discusses the details about these logs and traces.

### 2.7.1 Output from wsadmin

Messages from wsadmin are written to the wsadmin.traceout log file:

```
<WAS_install_root>/profiles/<profile>/logs/wsadmin.traceout
```

You can also increase the amount of data that is logged to this file by tracing the wsadmin utility. To do so, update the following file:

```
<WAS_install_root>/properties/wsadmin.properties
```

Uncomment the following line:

```
com.ibm.ws.scripting.traceString=com.ibm.*=all=enabled
```

Note that the information that is logged is of limited use because wsadmin calls MBeans in the application server that is running the administrative console application. So, you usually need to trace that application server as well.

### 2.7.2 Management scripts

You can manage WebSphere Application Server services using the supplied management scripts. For example, each WebSphere Application Server installation has a script to start an application server, a script to stop an application server, and a script to show you the status of all application servers defined in a profile. Each of these scripts writes its own log file into the server's logs directory. For example, the stopServer script writes stopServer.log into the logs directory:

```
<WAS_install_root>/profiles/<profile>/logs/<server>/stopServer.log
```

### 2.7.3 Profile management logs

The profile creation and management tool wasprofile writes messages to the profile independent logs directory, that is:

```
<WAS_install_root>/logs/wasprofile/<profile>.log
```

This log file is in XML format.

The Java graphical interface that is used to create a profile simply calls the **wasprofile** command after collecting the information needed. By default, it does not write a log, but you can pass it a log parameter as shown:

```
pctWindows -is:log c:\temp\pct.log
```

## 2.8 WebSphere Rapid Deployment logs

The WebSphere Rapid Deployment tool works on a directory that you create and pass to WebSphere Rapid Deployment in the **WORKSPACE** environment variable. It logs Eclipse messages into two separate files within this directory:

- ▶ `<workspace>/.metadata/.log`
- ▶ `<workspace>/project/.metadata/.log`

In a manner similar to other WebSphere Application Server utilities, WebSphere Rapid Deployment calls MBeans on the application server. The application server logs can help you resolve a problem with WebSphere Rapid Deployment. There is no way to trace the WebSphere Rapid Deployment utility. However, you can trace the application server as described in “Tracing” on page 82.

## 2.9 Summary of logs

Table 2-1 on page 94 shows a summary of the WebSphere logs.

**Note:** In this table:

- ▶ `<WAS_install_root>` represents the installation root for WebSphere Application Server, for example:  
c:\WebSphere\Appserver
- ▶ `<profile_home>` represents the root directory for a specific WebSphere Application Server profile, for example:  
`<WAS_install_root>/profiles/<profile>`
- ▶ `<ihs_install>` represents the installation directory for the IBM HTTP Server, for example:  
c:\IBM HTTP Server

Table 2-1 Log file summary

Tasks	Logs	Format / tools
Installation tasks		
WebSphere Installation	<WAS_install_root>/logs/log.txt	text
IBM HTTP Server installation	<ihs_install>/ ▶ ihsv6_install.log ▶ gskitInstall.log	text
Sample and IVT application installation: ▶ DefaultApplication ▶ ivtApp ▶ query ▶ PlantsByWebSphere ▶ SamplesGallery	<profile_home>/logs/  ▶ defaultapp_config.log ▶ defaultapp_deploy.log ▶ ivt_config.log ▶ query_config.log ▶ samples_config.log ▶ samples_install.log	text
WebSphere system application installation: ▶ filetransfer ▶ filetransferSecured ▶ ManagementEJB ▶ SchedulerCalenders ▶ adminconsole	<profile_home>/logs/  ▶ filetransfer_config.log ▶ mejb.log ▶ scheduler.cal_config.log ▶ webui_config.log	text
Profile tasks		
Profile creation wizard	<profile_home>/logs/pctLog.txt	text
	<WAS_install_root>/logs/wasprofile/wasprofile_create_<profile>.log.	XML
Application and system tasks		
Application print() and println()	<profile_home> <sup>2</sup> /logs/<server>/: ▶ SystemOut.log <sup>1</sup> ▶ SystemErr.log <sup>1</sup>	Text.
JVM System.out and System.err streams		administrative console: <b>Troubleshooting → Logs and Trace → &lt;server&gt; →JVM Logs</b>
<sup>1</sup> Configurable <sup>2</sup> <profile_home> can represent the location of the profile for an application server, node agent, or deployment manager. If the profile is for a node agent, <server> is “nodeagent”. If the profile is for a deployment manager, <server> is “dmgr”		

Tasks	Logs	Format / tools
WebSphere processes	Process logs at <profile_home> <sup>2</sup> /logs/<server>/: ► native_stderr.log <sup>1</sup> ► native_stdout.log <sup>1</sup>	Text (the server must be stopped to view with a text editor)  administrative console: <b>Troubleshooting</b> → <b>Logs and Trace</b> → <server> → <b>Process Logs</b>
WebSphere System.out stream + messages that contain extended service information	IBM service log (aka activity log) at <profile_home>/logs/activity.log <sup>1</sup>	Binary format. View with Log Analyzer. Showlog tool can convert the contents to a text format.
<b>Operational tasks</b>		
Start / stop an application server.	<profile_home>/logs/<server>/: ► SystemOut.log <sup>1</sup> ► SystemErr.log <sup>1</sup>  If using <b>startServer &lt;server&gt; -trace</b> , see <profile_home>/logs/<server>/: ► startServer.log ► stopServer.log	text
Start / stop a deployment manager	<profile_home>/logs/dmgr/: ► SystemOut.log <sup>1</sup> ► SystemErr.log <sup>1</sup>  <profile_home>/logs/dmgr/: ► startServer.log ► stopServer.log	text
Start / stop a node agent	<profile_home>/logs/nodeagent/ : ► SystemOut.log <sup>1</sup> ► SystemErr.log <sup>1</sup>  If using <b>startNode -trace</b> , see <profile_home>/logs/nodeagent/ : ► startServer.log ► stopServer.log	text
<sup>1</sup> Configurable <sup>2</sup> <profile_home> can represent the location of the profile for an application server, node agent, or deployment manager. If the profile is for a node agent, <server> is “nodeagent”. If the profile is for a deployment manager, <server> is “dmgr”		

Tasks	Logs	Format / tools
Start / stop a cluster	When you start or stop a cluster, that action is taken on each server. The logging is the same as though you started or stopped each server.	
<b>Configuration tasks</b>		
Adding a node to a cell	<code>&lt;node_profile_home&gt;/logs/:</code> ► addNode.log ► runAddNode.log	text
<sup>1</sup> Configurable <sup>2</sup> <code>&lt;profile_home&gt;</code> can represent the location of the profile for an application server, node agent, or deployment manager. If the profile is for a node agent, <code>&lt;server&gt;</code> is “nodeagent”. If the profile is for a deployment manager, <code>&lt;server&gt;</code> is “dmgr”		



## Installation problem determination

This paper discusses how to diagnose issues associated with WebSphere Application Server V6 installation. It covers actions and resolutions related to the following installation issues:

- ▶ Launchpad or installation wizard will not start
- ▶ Installation failure or hang
- ▶ Profile creation failure
- ▶ Install Verification Test (IVT) failures

The installation routine for other products, such as IBM HTTP Server and Web Server plug-ins, is separate from the application server products. This paper only covers issues that are related to WebSphere Application Server V6 installation.

**Important:** We recommend that you start your problem determination process by reading *Approach to Problem Determination in WebSphere Application Server V6* at <http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>.

## 3.1 Introduction

The installation process for WebSphere Application Server V6 has changed from previous versions. The main change is that only one copy of the core product files is installed. Profiles are used to define multiple server runtime environments.

Before trying to determine the cause of an installation problem, make sure that you have read and met the hardware and software prerequisites. The latest information about these is available at:

<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>

The entry point into WebSphere Application Server installation is the launchpad (Figure 3-1). From here, you can choose to launch the installation wizard for WebSphere Application Server.

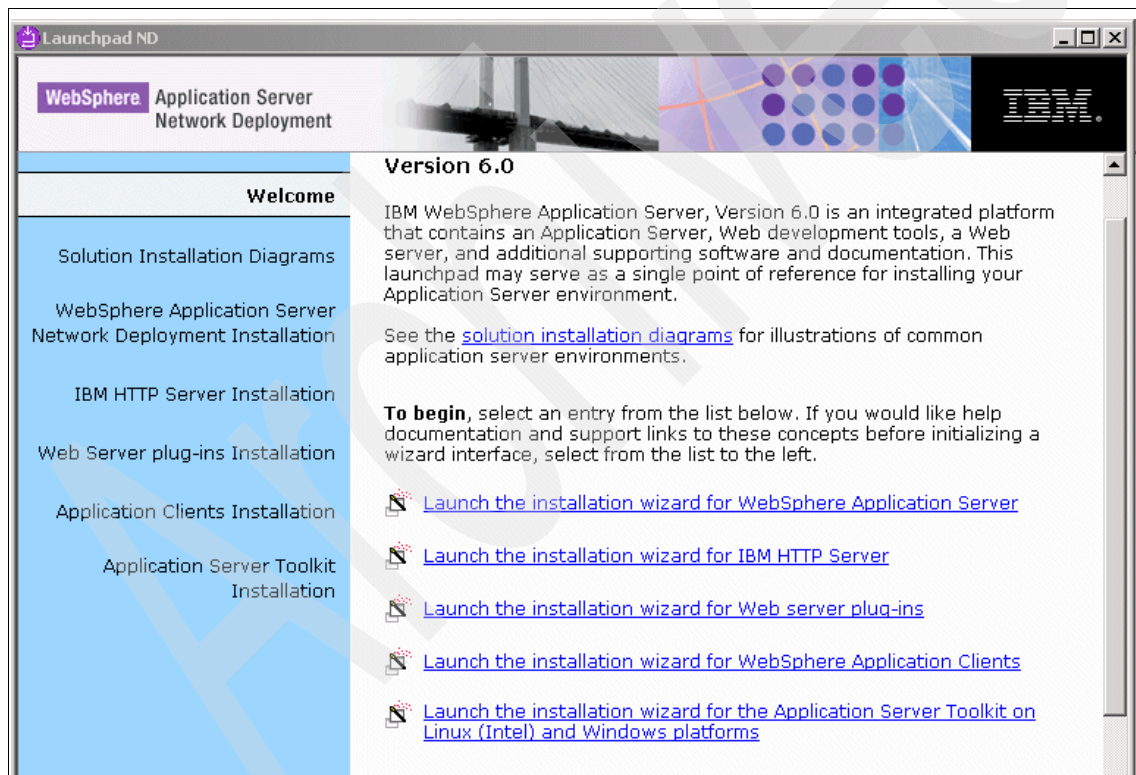


Figure 3-1 Launchpad for WebSphere Application Server installation

Figure 3-2 provides an overview of the installation events and common issues that are associated with installation.

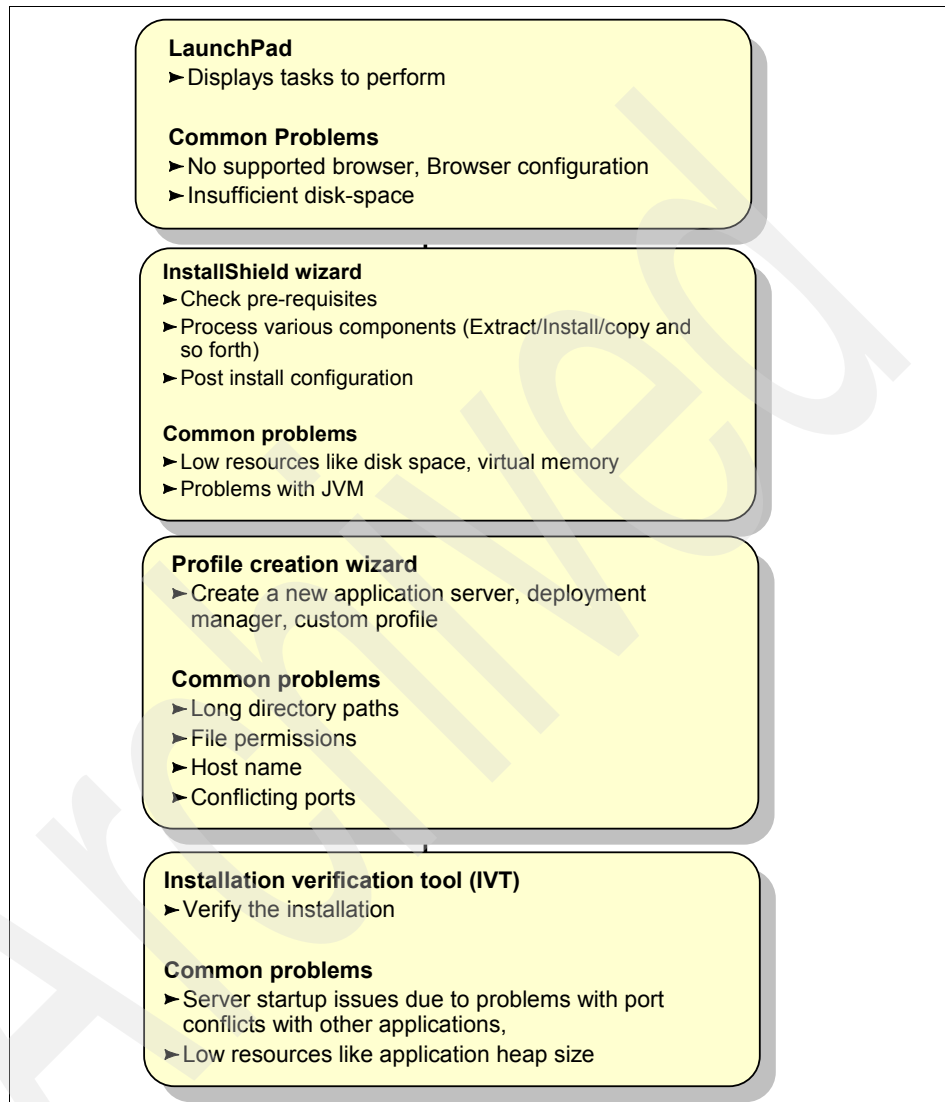


Figure 3-2 Overview of installation events, common issues, and logs

**Note:** In Figure 3-2 on page 99, the Installation wizard and Profile creation wizard are shown as different entities. This is what you will see if you are installing the Network Deployment version. In a WebSphere Application Server base or Express installation, profile creation for the default profile occurs in the Installation wizard itself.

## 3.2 Work the problem

You begin the problem determination process by evaluating the high-level symptoms to determine if one of these symptoms describes your problem. If it does not, you need to collect the appropriate data that is required to diagnose the problem.

Next, you go through the documentation to try to determine where the problem is. And lastly, we provide guidance on the next step to take for resolution, whether it be a support site, contacting IBM, information about configuration, or some other suggestion as to how to proceed.

Now, let's work the problem. You were led to this paper because you were experiencing one of the following impacts to an application or task:

- ▶ Symptom: Launchpad or installation wizard will not start or fails
- ▶ Symptom: Installation wizard hangs
- ▶ Symptom: Profile creation failure
- ▶ Symptom: IVT fails

If you do not find your symptom listed here, go to “The next step” on page 113.

### Recovering from a failed or hung installation:

Depending on the state that the system is in when an installation fails or hangs, you might need to uninstall WebSphere Application Server manually before you retry the process. The WebSphere Information Center has detailed, platform-specific instructions for uninstalling at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tins\\_uninstman.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tins_uninstman.html)

### 3.2.1 Symptom: Launchpad or installation wizard will not start or fails

This symptom covers the following situations:

- ▶ Executing the **launchpad.bat** command in windows or the **launchpad.sh** script in a UNIX environment does not start the launchpad.
- ▶ Selecting **Launch the installation wizard for WebSphere Application Server** from the launchpad fails to start the installation wizard or the wizard exits with or without an error message.

Possible causes for type of problem include Web browser requirements and disk space or permission requirements.

If the installation wizard starts and runs for a while but appears to hang, go to “Symptom: Installation wizard hangs” on page 103.

#### Data to collect

Installation events are logged in the following:

- ▶ The launchpad log

Events related to launchpad are logged in the file at the following location:

- UNIX

`/tmp/IBM_WebSphere_launchpad/IBM_WebSphere_LaunchPad_log.txt`

- Windows

`%TEMP%\IBM_WebSphere_launchpad\IBM_WebSphere_LaunchPad_log.txt`

If the launchpad fails to start or fails at an early stage, it is possible that no log files are generated. Take note of any error messages that are displayed before the wizard exits.

- ▶ The installation log

`<WAS_install_root>/logs/log.txt`

This log contains events that are related to the installation wizard and is created when the installation begins.

If the installer fails at a very early stage, then this log file might not be created or it might exist in the system temporary area, `%TEMP%\log.txt` in Windows or `/tmp/log.txt` in UNIX.

If this file does not exist, then run the installer from a command window using **-log** option to create a log of all events.

- UNIX

```
./install -log !logfile @ALL
```

- Windows

```
install -log !logfile @ALL
```

Where *logfile* is a fully qualified file name for writing the log events.

As you collect the log files, copy them to a place where you can view them. If you need to recreate the problem in order to collect this documentation, be sure to prepare (schedule off-shift hours, review prerequisites, and so forth).

## What to look for

Problems starting the launchpad or installation wizard can usually be traced back to missing prerequisite system or application levels. The first step in diagnosing this problem is to ensure that you have the proper prerequisites for installation. You can find this information at:

<http://www-306.ibm.com/software/webservers/appserv/doc/latest/prereq.html>

### ***Problems starting the launchpad or Installation wizard***

If the problem you are having is in starting the launchpad, look at the `IBM_WebSphere_LaunchPad_log.txt` log. This log normally contains minimal messages that indicate when the launchpad was run. In the event of an error during launch, it might also contain error messages. If the launchpad does not start at all, then no log file is created.

If you see any error messages or warnings in this file that are not self-explanatory or if the log does not exist, ensure that you have a supported Web browser and that it is configured correctly (see “Web browser requirements” on page 110).

### ***Problems in the installation wizard***

Installation events are installed in `log.txt`. If the installation wizard starts but fails during the installation process, look in `log.txt` for messages that contain `INSTCONFSUCCESS`, `INSTCONFPARTIALSUCCESS`, or `INSTCONFFAILED`, which might indicate the current status of the installation.

If you see the `INSTCONFPARTIALSUCCESS` or `INSTCONFFAILED` messages, then there should be error or warning messages preceding them (for example, messages that indicate problems with resources, such as not enough disk space, exceptions in the JVM, segmentation faults, and so forth).

An error that resembles the following indicates a problem with disk space:

A suitable JVM could not be found. Please run the program again using the option `-is:javahome < JAVA HOME DIR>` No space left on device

This error indicates there is not enough free space for the installer to run on. You can get this error even if enough space exists where you plan to install WebSphere Application Server (for example, drive D: or /usr) .

Verify that the location of C:\Temp or %TEMP% in Windows or /tmp directory in UNIX has enough free space for the installer to run. You need to check the install document to determine the exact amount of temporary disk space required, typically a minimum of 100 MB.

Another option is to use `-is:tempdir` with the installation wizard, where *tempdir* is the location of a temporary directory on a partition with enough free space.

If you meet the installation prerequisites and you still have not resolved your problem, go to “The next step” on page 113 for information about how to search for known problems with installation.

### 3.2.2 Symptom: Installation wizard hangs

During installation, a progress indicator is displayed to show you how far the install has progressed. If there is no change in the progress indicator for a very long time, the installation process could be hung.

Reasons that the install process might hang include the following:

- ▶ The system is very low on resources such as virtual memory or swap space.
- ▶ There might be heavy network traffic or network breakdown if installing from a remote location.
- ▶ A task or thread has gone into an infinite wait or loop.

In a WebSphere Application Server or WebSphere Application Server - Express installation, the installation wizard creates a default server profile. The installer first starts the file copy process and then the profile creation process. In WebSphere Application Server Network Deployment, this is a two-step process. The install completes, then you are asked if you want to create a profile.

## Data to collect

If the installation hangs, check the following:

- ▶ `<WAS_install_root>/logs/log.txt`

If the installer fails at a very early stage, then this log file might not be created or it might exist in the system temporary area, which is `%TEMP%\log.txt` in Windows or `/tmp/log.txt` in UNIX.

To ensure that you get a full listing of messages in this log, run the installer from a command window using the **-log** option to create a log of all events.

- UNIX

```
./install -log !logfile @ALL
```

- Windows

```
install -log !logfile @ALL
```

Where *logfile* is a fully qualified file name for writing the log events.

- ▶ `<WAS_install_root>/profiles/<profile>/logs/pctLog.txt`

This log file is created only when the profile creation wizard is executed. This log is not created when using the **wasprofile** command directly or during installation of the product.

## What to look for

If you think the install process is hung, check the log.txt file periodically to see if progress is being made. Messages such as `INSTCONFSUCCESS`, `INSTCONFPARTIALSUCCESS`, or `INSTCONFFAILED` in log.txt indicate the current status of the installation. If you see the `INSTCONFPARTIALSUCCESS` or `INSTCONFFAILED` messages, then there should be error or warning messages preceding them.

Also, check other system activities, such as CPU utilization, hard disk usage, or any network activity (if installing remotely), to make sure there are not external factors that are affecting the install.

If the installation does appear to be hung, look for the last recorded message in the log file. This message gives you an idea of what the installer was doing before it hung.



### ***Determine if errors occurred in the file copy process***

Look in log.txt for an entry such as the following:

```
(<Date and time stamp>), Install,  
com.ibm.ws.install.ni.ismp.actions.ISMPConfigManagerLaunchAction, msg1,  
INSTCONFSUCCESS: Post-installation configuration is successful
```

If you see this message, the file copy process has completed successfully. If not, inspect the messages in the log for an indication of the error.

### ***Determine if the errors occurred in the profile creation process***

If the file copy was completed successfully, any error messages after this indicate problems in profile creation or with other subsequent steps, including sample application and administrative console application deployment. Look for the following entry in the log (log.txt for WebSphere Application Server and WebSphere Application Server - Express, pctLog.txt for WebSphere Application Server Network Deployment):

```
(<Date and time stamp>), Install,  
com.ibm.ws.install.ni.ismp.actions.ISMPWSPprofileLaunchAction, msg1,  
INSTCONFSUCCESS: Post-installation configuration is successful
```

If you do not see this message, then there are problems with profile creation. To diagnose problems with profile creation, see “Symptom: Profile creation failure” on page 105.

On the other hand, if you see this message, the profile creation was successful, and it is time to re-evaluate your symptoms. Go to “The next step” on page 113.

## **3.2.3 Symptom: Profile creation failure**

Profiles allow you to define multiple runtime environments, each with its own administrative interface while sharing the same code base. Problems with profile creation might be due to long directory paths, file permissions, problems with the host name, and so forth.

A default server profile is created as part of the installation process in a WebSphere Application Server or WebSphere Application Server - Express installation. The Network Deployment installation wizard gives you the option of creating a profile. Profiles can also be created at any time after installation.

## Data to collect

If the profile creation fails, check the following:

- ▶ `<WAS_install_root>/logs/wasprofile/wasprofile_create_<profile>.log`

This log file is created when the installation phase has completed the file copy process and starts creating a default profile. This log file is also created whenever the profile creation wizard or **wasprofile** command is executed.

This log file traces all the events that occur during profile creation. It is an XML log file and is best viewed by a viewer that can format XML, for example a Web browser or WordPad in Windows. The entries in this log file consist of `<record>` entries. A sample log entry that indicates an error would look something similar to the following:

```
<record>
  <date>2005-07-19T14:18:53</date>
  <millis>1121762933500</millis>
  <sequence>2984</sequence>
  <logger>com.ibm.ws.install.configmanager.ConfigManager</logger>
  <level>WARNING</level>
  <class>com.ibm.ws.install.configmanager.ConfigManager</class>
  <method>executeAllActionsFound</method>
  <thread>11</thread>
  <message>Fatal configuration action failed:
    com.ibm.ws.install.configmanager.actionengine.ANTAction
    -C:\IBM\WAS\profileTemplates\managed\actions\
    executeManagedProfileSetup.ant</message>
</record>
```

- ▶ `<WAS_install_root>/logs/log.txt`

The following log files are created during profile creation. They are located in the `<WAS_install_root>/profiles/<profile>/logs` directory. Each log might or might not exist depending on the type of profile that is created.

- ▶ `pctLog.txt`

Created only when the profile creation wizard is executed. This log is not created when using the **wasprofile** command directly or during installation of the product.

- ▶ `amjrte_config.log`

Tivoli Access Manager configuration log for its Java Runtime Environment.

- ▶ `collect_metadata.log`

Collects metadata information about managed objects in the system to evaluate and prevent potential installation conflicts.

- ▶ createDefaultServer.log  
A log from wsadmin recording the creation of the server1 process in the default profile.
- ▶ createshortcutforprofile.log  
Windows tool log for creating menu entries and shortcuts.
- ▶ defaultapp\_config.log  
JACL script log from configuring default application resources.
- ▶ Service.log  
Start and stop events for server1.
- ▶ Application installation and configuration logs for system and sample applications:
  - filetransfer\_config.log for the file transfer application
  - ivt\_config.log for the ivtAPP application
  - mejb\_config.log for the ManagementEJB application
  - hamanager\_config.log for the high availability application
  - query\_config.log for the Query application
  - samples\_config.log for the PlantsByWebSphere sample application
  - samples\_install.log for the SamplesGallery and PlantsByWebSphere sample applications
  - scheduler.cal\_config.log for the SchedulerCalendars application
  - webui\_config.log for the administrative console application
  - defaultapp\_deploy.log for the DefaultApplication application
- ▶ SIBDefineChains.log  
Creation log for service integration bus endpoints, inbound channels and channel chains, outbound thread pool, and outbound channel and channel chains.
- ▶ SIBDeployRA.log  
Deployment log for the service integration bus function.
- ▶ winservice\_config.log  
Service log for the Windows service created for server1.
- ▶ addNode.log  
Log for federating a node to a cell (Custom profile).

## What to look for

Messages such as INSTCONFSUCCESS, INSTCONFPARTIALSUCCESS, or INSTCONFFAILED in log.txt or pctLog.txt indicate the current status of the

installation. If you see the INSTCONFPARTIALSUCCESS or INSTCONFFAILED messages, then there should be error or warning messages preceding them.

If you do not see an entry similar to the following, then there are problems with profile creation:

```
(<Date and time stamp>), Install,  
com.ibm.ws.install.ni.ismp.actions.ISMPWSProfileLaunchAction, msg1,  
INSTCONFSUCCESS: Post-installation configuration is successful
```

Look for an entry such as the following:

```
(<Date and time stamp>), Install,  
com.ibm.ws.install.ni.ismp.actions.ISMPWSProfileLaunchAction, err,  
INSTCONFFAILED: Cannot complete required configuration actions after  
installation. The configuration failed. The installation is not successful.  
Refer to \install_root\logs\wasprofile\wasprofile_create_profilename.log for  
more details
```

If you see this entry, look at the wasprofile\_create\_<profile>.log to try to determine what task was being performed when the profile creation failed. Most tasks, such as system or sample application installation, are logged to individual log files in the <WAS\_install\_root>/profiles/<profile>/logs directory. If you can determine which task the profile creation was doing, collect the file for that task. For example, if you had problems creating a custom node then look at the log file <WAS\_install\_root>/profiles/<profile>/logs/addNode.log for any errors.

For information about common problems, see “Profile creation problems” on page 112.

### 3.2.4 Symptom: IVT fails

The IVT looks at the profile configuration for the server and looks for a server running on the server port number. Note that if a server is up and running on that port, the IVT runs against that server, even if it is not the one you just installed. If no server is running on that port, the IVT attempts to start the server. When the server has started successfully, the IVT accesses the server and runs various tests, including servlet engine verification, JSP verification, EJB verification, and so forth.

The option to run the IVT is displayed in the First Steps console after the installation completes and after every profile creation, with the exception of a custom profile creation because no server is actually created.

You should run the IVT before making any configuration changes to WebSphere Application Server. This acts as a checkpoint to see if any problems exist as the result of the installation. If the IVT runs clean and problems show up later, they

are most likely due to configuration changes done after the installation. It is recommended that all instances of WebSphere Application Server be stopped before running the IVT.

A failure in the IVT is usually because the application server fails to start. Common reasons for this include port conflicts, not enough memory, and so forth.

### Data to collect

The logs most likely to be of interest are:

- ▶ `<WAS_install_root>/profiles/<profile>/logs/ivtClient.log`  
This log contains messages from the IVT execution.
- ▶ The following log files created in:  
`<WAS_install_root>/profiles/<profile>/logs/server1` directory  
They are created as a result of the IVT and start/stop of server.
  - `startServer.log` - Log of start server events
  - `SystemOut.log` - Log of all activity within the WAS environment
  - `SystemErr.log` - Record system errors

### What to look for

The first file to look at in case IVT has failed is `ivtClient.log`. Messages with IVTL and ADMU prefixes provide information about what the IVT application is doing and the status of each action.

Look for the following message in `ivtClient.log`:

ADMUXXXXX: Server *servername* open for e-business; process id is xxxx

If you find this message, then the server has started successfully, and the IVT failed while executing one of the tests. Examine the error messages in `ivtClient.log` to locate the failing process.

If another deployment manager is running with the same port, you see the following messages:

IVTL0110E: Log file error with  
C:\WebSphere\AppServer\profiles\Dmgr02\logs\dmgr\SystemOut.log,  
java.io.FileNotFoundException: C:\WebSphere\AppServer

If the server does not start, look at the `Server Port number is:` entry. This entry contains the server port number of the profile instance. Make sure that this port is not in use (see “Application server startup problems” on page 111).

If the port does not seem to be the problem, look at startServer.log, SystemErr.log, and SystemOut.log for information related to server startup, such as the following:

- ▶ Look for any error messages starting with WSVR (server runtime) or ADMU (management utility) in the startServer.log and SystemOut.log.
- ▶ Look for error messages in SystemErr.log. These messages will start with:  
[Date and time stamp] 0000000a SystemErr

If you cannot identify the problem, go to “The next step” on page 113.

## 3.3 Analyzing problem areas

Your analysis of the data that you gathered will most likely lead you to one of the following areas. If not, see “The next step” on page 113.

### 3.3.1 Web browser requirements

If you are having problems in starting the launchpad, it is possible that you do not have a supported Web browser installed or configured properly. If this is the case:

- ▶ Verify that the latest version of a supported Web browser is installed. Mozilla and Internet Explorer are supported for configuration and installation activities.
- ▶ On UNIX platforms, ensure that the location of the supported browser is exported. For example if the Mozilla executable is located in the /usr/bin directory, then export its location as follows:  
`export BROWSER=/usr/bin/mozilla`
- ▶ Ensure that JavaScript™ is enabled in the browser options or preferences. For example:
  - In Mozilla, select **Edit** → **Preferences** → **Advanced** → **Scripts & Plugins** - **Enable JavaScript for: Navigator, Allow Scripts to:** (select all boxes)
  - In Internet Explorer, select **Tools** → **Internet Options** → **Security** → **Custom Level for Internet Scripting** → **Active scripting** → **Enable**

The following link gives specific information about system readiness for installing WebSphere Application Server on RedHat Linux Enterprise Edition 4:

<http://www-1.ibm.com/support/docview.wss?uid=swg21201306>

### 3.3.2 Application server startup problems

During profile creation, it is important that the port numbers that are specified for each of the WebSphere Application Server services are unique. The Profile creation wizard, by default, tries to define port numbers that do not conflict with other profiles in the installation. However, it is also possible that there are other applications using the same port numbers (for example, older versions of WebSphere Application Server in a co-existence environment or the unit test environment of Rational Application Developer).

If an application server does not start during the IVT process, or later using the **startServer** command, look for the following message in `startServer.log`:

```
ADMYUXXXX: an instance of the server may already be running: server1
```

This message indicates that another instance of the server is already running or another process is using the same port number.

If you have a message such as the following, you most likely have a port configuration problem:

```
WSVRXXXX: Error occurred during startup com.ibm.ws.exception.RuntimeError:
com.ibm.ws.exception.RuntimeError: org.omg.CORBA.INTERNAL:
CREATE_LISTENER_FAILED_4
```

This message indicates that one of the port numbers specified in the following configuration file is in use and the listener thread cannot be created:

```
<WAS_install_root>/profiles/<profile>/config/cells/<cell>/nodes/<node>/serverindex.xml
```

To resolve the problem, do the following:

- ▶ Use the **serverStatus** command to determine if the server instance that you are trying to start is already started.
- ▶ Review the ports that are required by the application server process. You can see these in:  

```
<WAS_install_root>/profiles/<profile>/config/cells/<cell>/nodes/<node>/serverindex.xml
```
- ▶ List the ports that are currently being used by the system. One way of doing this is by issuing a **netstat -a** command in a command prompt window. If you have a port conflict, update the `serverindex.xml` file to use a non-conflicting port number.

### 3.3.3 Profile creation problems

The following are common problems with profile creation.

#### File path length errors

Windows 2000 has a length restriction of 258 characters for a command. A problem can occur that prevents the successful creation of a profile when a path is too long. The maximum length for the `<WAS_install_root>` directory is 60 characters. The maximum length for the profiles installation root directory is 80 characters.

If your log files have errors similar to `Input line is too long` then the length of the file path and node name in the command string has caused the entire command to exceed the operating system limit for command length.

This error can show up in a message box during the wizard or if using `wasprofile`, in `<WAS_install_root>/profiles/<profile>/logs/collect_metadata.log`.

Create the profile again using shorter directory paths and node names. If you are still in the installation process, consider re-installing using a shorter path for the installation root.

#### Host name error

If you see errors similar to `localhost is not a valid host name for remote access`, then you have entered `localhost` as the value in the host name field of the Profile creation wizard. Other machines in the network cannot reach your node using `localhost`, so you must provide a host name that can be resolved by other systems to your system's IP address.

#### Template path error

If in a UNIX environment you get an error similar to the following then verify the profile templates location and permissions are correct:

```
<message>Incoming command line is: { "-create" ,"-help" ,"-templatesPath"
,"/opt/WebSphere/AppServer/crso/profileTemplates/managed" }</message>
<message>Could not resolve templatePath from command line</message>
```

#### Custom profile error

If an error occurred creating a custom profile, look at the `addNode.log` file for any additional errors. Look for an error message similar to the following:

```
[Date and Time stamp] 0000000a AbstractNodeC E ADMU0006E: Exception creating
Deployment Manager connection:
com.ibm.websphere.management.exception.ConnectorException: ADMC0016E: The
system cannot create a SOAP connector to connect to host hostname at port 8879
```



If you see this message, it is likely that the deployment manager profile is not running. Ensure that the deployment manager profile is created and running on the specified port and host before creation of custom profile.

If neither of these messages appears and you do not see any other obvious problems, go to “The next step” on page 113.

### 3.4 The next step

The symptoms and problem areas included in this paper are some that you are more likely to experience. However, there are other things that can go wrong during installation.

If, after going through this process, you still have an undiagnosed problem, it is recommended that you go back to *Approach to Problem Determination in WebSphere Application Server V6* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

Review the problem classifications to see if there are any other components that might be causing the problem.

If you are sure the problem is in the installation process, there are things that you can do before contacting IBM support. First, you should review the documentation that you have gathered for errors that were not addressed in this paper and search support sites for information or fixes. Look for current information available from IBM support on known issues and resolutions on the following IBM support page:

<http://www-1.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCVS24>

Look also at the WebSphere Information Center documentation for additional resources for diagnosing and fixing issues:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb\\_allrfl.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_allrfl.html)

If these does not resolve your problem, then gather additional information as specified in the following MustGather document and raise a problem record with IBM:

<http://www-1.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCVS24&q=mustgather>



## System management problem determination

This paper discusses techniques for determining the cause of problems with WebSphere Application Server system management. The symptoms that this paper discusses are:

- ▶ You are not able to access the administrative console.
- ▶ You cannot access server processes using wsadmin or the management scripts such as stopServer.
- ▶ You are getting errors performing system management functions (for example, managing application servers, node agents, Web servers, or applications).
- ▶ You are not able to federate a node with a deployment manager.
- ▶ You are getting save conflict messages in the administrative console.
- ▶ Your enterprise applications no longer appear in the administrative console.
- ▶ You have problems with WebSphere Rapid Deployment.
- ▶ You are having trouble communicating using SSL and are getting messages that indicate your certificates have expired.

**Important:** We recommend that you start your problem determination process by reading *Approach to Problem Determination in WebSphere Application Server V6* at <http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>.

## 4.1 Introduction

In this paper, we look at resolving problems that occur during system management activities. WebSphere Application Server configuration is maintained in XML files. The XML files are maintained in a directory structure that reflects the topology of your WebSphere Application Server installation (Figure 4-1).

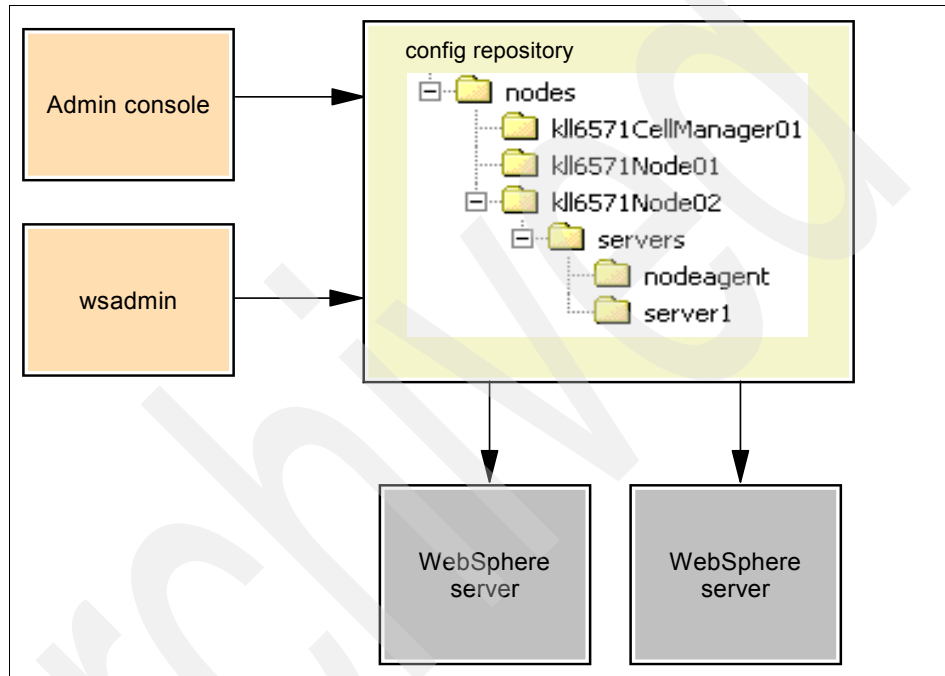


Figure 4-1 Overview of system administration

The WebSphere administrative console and the wsadmin command line administration tool read, maintain, and modify the contents of the XML files.

We strongly recommend that you back up the config repository on a regular basis. You can either back up the repository manually via a file system backup, or you can use the WebSphere Application Server supplied tool **backupConfig**. This script creates a zipped file that contains all the configuration files that you might need to restore the WebSphere Application Server configuration.

**Tip:** Editing the XML files in the repository directly is not supported and can lead to unexpected results.

At the most basic level, that is a stand-alone installation of a single WebSphere Application Server, the config directory is as shown in Figure 4-2. The XML files shown contain the data that gets formatted by the administrative console and displayed in the various boxes and text fields in the browser.

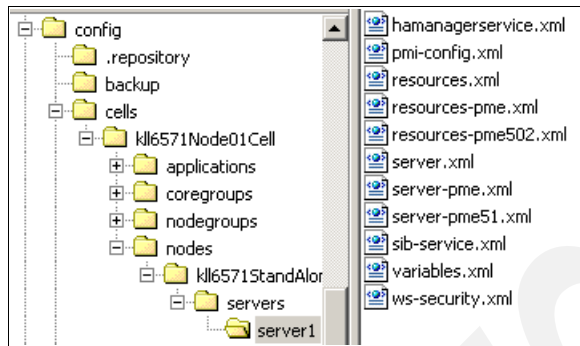


Figure 4-2 The config directories and files

The administrative console and other management tools call management beans (MBeans) in the WebSphere Application Server process. MBeans are used by WebSphere Application Server to perform system management functions.

In a Network Deployment environment, the deployment manager maintains the master repository for all of the WebSphere Application Server nodes and servers that it manages in the cell. Copies of the files that each node needs are replicated to that node by a process known as *synchronization*.

Figure 4-3 on page 118 shows a Network Deployment environment with two nodes. All of the configuration files relevant to both Node01 and Node02 are kept in the master repository along with the configuration files that are relevant to the deployment manager. Only those files that are relevant to Node01 are replicated to Node01, and only those files that are relevant to Node02 are replicated to Node02. Each node gets a copy of the `serverindex.xml` file for every other node because this contains connection information for the other nodes, that is host names and ports.

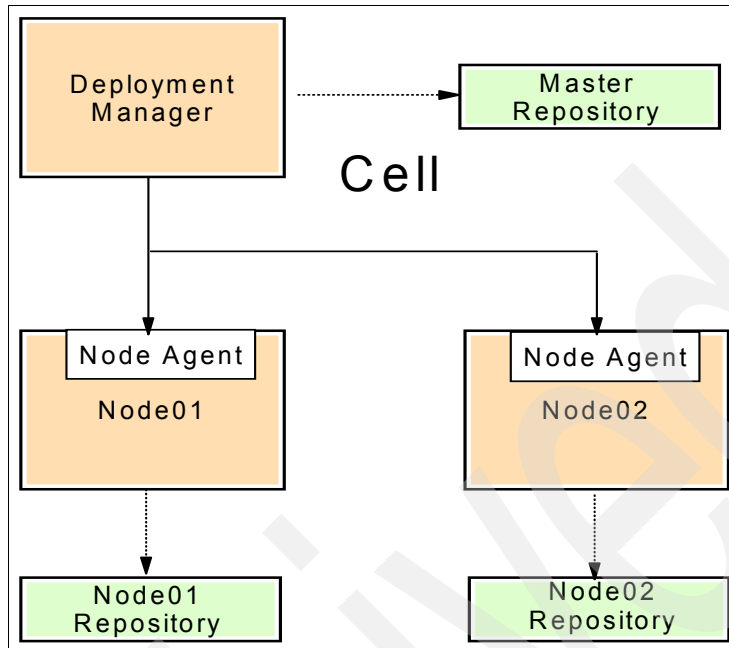


Figure 4-3 Replicated admin repository

You can set each node agent to perform automatic or manual synchronization and the interval at which each node agent will perform the synchronization. To set this in the administrative console, select **System administration** → **Node agents** → **nodeagent** → **File synchronization service**.

Only changed files are synchronized, so even if your system is performing automatic synchronization every minute (the default interval), this will have a minimal performance impact because the node agent will connect to the deployment manager and be told that nothing has changed.

You can manually initiate synchronization using the administrative console by selecting **System administration** → **Nodes**, putting a check by the node that you wish to synchronize, and then clicking either **Synchronize** or **Full Resynchronize**.

Choosing **Synchronize** only synchronizes changes that the administrative console is aware of using the normal synchronization mechanism. Choosing **Full Resynchronize** synchronizes all changes that have been made to the master repository by whatever means. For example, if you were to edit the master copy of the server.xml file with a text editor, clicking **Synchronize** does not replicate this file to the node. If you click **Full Resynchronize**, then your updates are replicated.

You can also perform synchronization from the node agent using the `syncNode.batlsh` script. You must stop the node agent to use this tool. You can pass parameters to the script to log messages to a specified file or to trace the synchronization.

**Note:** You can also perform synchronization using the `wsadmin` tool. See the WebSphere Information Center for further details.

**Tip:** Synchronization does not protect against losing your entire replicated repository. The node agent configuration files must exist for synchronization to work. It is not a replacement for a comprehensive backup policy.

## Profiles

A profile is a runtime environment for one or more WebSphere Application Server processes. The profile includes all of the files that the application server uses and can change.

Profiles save disk space and simplify product upgrades because they only use one set of product files for every profile that is created. Only those files that are needed for a profile are copied to the profile directory. The deployment manager is now a profile under the same installation directory as an application server.

Application developers can quickly create a new clean profile to test an application and then remove that profile when testing is complete.

WebSphere Application Server updates are not applied to the files in a profile. This means that any changes to config repository structure, XML file DTDs, or the sample applications will not be applied to existing profiles. Note that the scripts in a profile's `bin` directory typically call those in the common `bin` directory. So, changes to scripts generally are accessible to existing profiles.

Profiles are created either by the graphical profile creation wizard or the **`wasprofile`** command and are managed via the **`wasprofile`** command.

## Security

WebSphere Application Server can be secured with an LDAP user repository or the server operating system. You can customize WebSphere Application Server so that it can use any LDAP repository that you choose. Some common repositories that are used with WebSphere Application Server are the IBM SecureWay® Directory and Microsoft® Active Directory.

A user registry contains the names, groups memberships, roles, and passwords of users in an organization. At the most basic level, most operating systems

contain a user registry so that users can logon with a user name and password and have access to operating system resources. At the more complex level, a large distributed organization might maintain a complete list of users including their passwords, personal information, group memberships, and job roles in a distributed LDAP repository.

WebSphere Application Server can use the local operating system as a user registry only when the cell is configured on a single machine or when you use a Windows domain controller. Operating system user registries are not typically available across hosts, and so users defined on one server would not be available on other servers.

If you have multiple installations of WebSphere Application Server on multiple servers that you need to secure, then you need to use an LDAP registry so that the same registry is available to all servers.

When you enable global security in WebSphere Application Server, you also secure the administrative console and all administrative interfaces, including wsadmin and the various management scripts. For example, to stop a server when security is enabled, you must supply a user name and password to the script as shown:

```
stopServer.sh server1 -username user -password password
```

Alternatively, you can update two property files with the user name and password so that you do not have to use those parameters with the scripts, as follows:

```
<WAS_install_root>/profiles/<profile>/properties/sas.client.props  
<WAS_install_root>/profiles/<profile>/properties/soap.client.props
```

The first file is referenced by utilities that use RMI to communicate with the application server, and the second is used by utilities that communicate via SOAP. The wsadmin tool and the management scripts such as **stopServer** use SOAP by default.

You can encode the passwords in these files to prevent them from being read easily by using the password encoder utility:

```
<WAS_install_root>/bin/PropFilePasswordEncoder.bat|sh
```

When security is enabled, WebSphere Application Server uses SSL for communication from the browser to the administrative console and for other administrative communications. SSL communication relies on a key database file and password. WebSphere Application Server comes with a dummy key database file that is secured with the password WebAS. We highly recommend



that you create your own key database file for production servers. Refer to the WebSphere Information Center for details on customizing the key database:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec\\_preparetruststorefile.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec_preparetruststorefile.html)

After you replace the dummy key database files, you need to update the properties files with the new key database file name and password, or you will not be able to manage the servers with the scripts and wsadmin:

```
<WAS_install_root>/profiles/<profile>/properties/sas.client.props  
<WAS_install_root>/profiles/<profile>/properties/soap.client.props
```

Again, you can encode the passwords in these files.

## WebSphere Rapid Deployment

WebSphere Rapid Deployment is a new feature in WebSphere Application Server V6 that simplifies and speeds up the process of deploying enterprise applications or their components for development purposes. WebSphere Rapid Deployment is a cut-down version of the Eclipse IDE and runs in batch mode.

WebSphere Rapid Deployment is a foreground process that monitors a chosen directory. If the developer copies an enterprise application EAR file to that location, WebSphere Rapid Deployment detects the new EAR and automatically deploys it to the WebSphere Application Server. Figure 4-4 shows the WebSphere Rapid Deployment components.

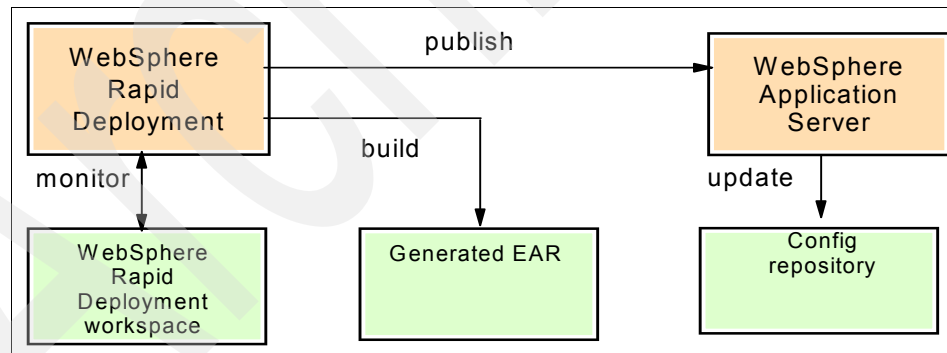


Figure 4-4 WebSphere Rapid Deployment components

WebSphere Rapid Deployment can also monitor the chosen directory for application components rather than complete EAR files and then create and update an application with the changes. For example, placing an updated Java program or JSP in the monitored directory causes WebSphere Rapid Deployment to create an application or to update an existing application with the compiled class or JSP.

WebSphere Rapid Deployment is only intended for development environments to simplify the process of testing updates to applications. It is not intended for production environments where you do not want production code updated easily. For that reason, the WebSphere Rapid Deployment process only runs in the foreground.

### 4.1.1 Collecting data

You need to collect the following log files and data for system management problem determination:

- ▶ WebSphere Application Server logs
- ▶ wsadmin output
- ▶ Management script logs
- ▶ Profile management logs
- ▶ WebSphere Rapid Development logs

If the problem is difficult to recreate or disruptive to business operations, see “The next step” on page 162 for a complete list of documentation to collect before continuing. In particular, you should review the MustGather documents for a complete list of documentation that is required by IBM support.

#### **WebSphere Application Server logs**

You need to look at the WebSphere Application Server log files when diagnosing system management problems.

##### ***Stand-alone server***

In a stand-alone server, the administrative console application and the administrative MBeans run in the server process. So, you need to look at the logs for that application server. The log files are:

```
<WAS_install_root>/profiles/<profile>/logs/<server>/SystemOut.log  
<WAS_install_root>/profiles/<profile>/logs/<server>/SystemErr.log
```

##### ***Deployment manager***

In a Network Deployment installation, system management involves more than one application server process. So, you need to look at the logs for each component. The deployment manager logs are:

```
<WAS_install_root>/profiles/<profile>/logs/dmgr/SystemOut.log  
<WAS_install_root>/profiles/<profile>/logs/dmgr/SystemErr.log
```

##### ***Node agent***

Node agent log files are:

```
<WAS_install_root>/profiles/<profile>/logs/nodeagent/SystemOut.log  
<WAS_install_root>/profiles/<profile>/logs/nodeagent/SystemErr.log
```

## **Application server**

With some problem types, you might also need to look at the logs for the application server that you are trying to manage. These log files are:

```
<WAS_install_root>/profiles/<profile>/logs/<server>/SystemOut.log  
<WAS_install_root>/profiles/<profile>/logs/<server>/SystemErr.log
```

## **System management trace**

Most system management problems can be traced by setting the following trace string:

```
Admin=all
```

If security is enabled, then add security related tracing directives to the trace string so that it looks similar to the following:

```
Admin=all:StartupService=all:SASRas=all:ORBRas=all:com.ibm.ws.security.*=all:SSL=all
```

For more detailed information about how to set up tracing, see the WebSphere Information Center section, *Configuring logging properties using the administrative console*, at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb\\_configjavalog.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb_configjavalog.html)

**Note:** When setting the trace string in administrative console, the console will stop responding while it populates the list of components. This action can take some time (around 30 seconds), during which the administrative console might appear to be hung. Be patient.

You can also set trace strings using wsadmin. For further details, see the WebSphere Information Center section, *Tracing operations with the wsadmin tool*, at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/txml\\_traceadmincontrol.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/txml_traceadmincontrol.html)

If you cannot connect to the WebSphere administrative console or wsadmin, you can set tracing by updating the server.xml file for the application server that you need to trace. You can find the server.xml file under the server's entry in the configuration file directory structure:

```
<WAS_install_root>/profiles/<profile>/config/cells/<cell>/nodes/<node>/servers/  
<server>/server.xml
```

Open this file in a text editor and update the services tag, changing the startupTraceSpecification to the trace string that you require, as shown in Example 4-1.

*Example 4-1 Setting the trace string in server.xml*

---

```
<services xmi:type="traceservice:TraceService" xmi:id="TraceService_1120228479019"
enable="true" startupTraceSpecification="Admin=all" traceOutputType="SPECIFIED_FILE"
traceFormat="BASIC" memoryBufferSize="8">
```

---

You need to restart the server for this to take effect.

### wsadmin output

Messages from wsadmin are written to the wsadmin.traceout log file:

```
<WAS_install_root>/profiles/<profile>/logs/wsadmin.traceout
```

You can also increase the amount of data that is logged to this file by tracing the wsadmin utility. Update the following file:

```
<WAS_install_root>/properties/wsadmin.properties
```

Uncomment the line:

```
com.ibm.ws.scripting.traceString=com.ibm.*=all=enabled
```

Note that the information that is logged is of limited use because wsadmin calls MBeans in the application server that is running the administrative console application. You usually need to trace the application server as well.

### Management script logs

WebSphere Application Server services can be managed using the supplied management scripts. For example, each WebSphere Application Server installation has a script to start an application server, a script to stop an application server, and a script to show you the status of all application servers that are defined in a profile. Each of these scripts writes its own log file into the server's logs directory. For example, the stopServer script writes stopServer.log into the logs directory:

```
<WAS_install_root>/profiles/<profile>/logs/<server>/stopServer.log
```

### Profile management logs

The profile creation and management tool **wasprofile** writes messages to the profile independent logs directory, that is:

```
<WAS_install_root>/logs/wasprofile/<profile>.log
```

This log file is in XML format.

The Java graphical interface that is used to create a profile simply calls the **wasprofile** command after collecting the information that is needed. By default, it does not write a log. However, you can pass it a log parameter as shown:

```
pctWindows -is:log c:\temp\pct.log
```

## WebSphere Rapid Development logs

The WebSphere Rapid Deployment tool works on a directory that you create and pass to WebSphere Rapid Deployment in the WORKSPACE environment variable. It logs Eclipse messages into two separate files within this directory:

- ▶ `<workspace>/metadata/.log`
- ▶ `<workspace>/project/.metadata/.log`

Similar to other WebSphere Application Server utilities, WebSphere Rapid Deployment calls MBeans on the application server. The application server logs can help you resolve a problem with WebSphere Rapid Deployment. There is no way to trace the WebSphere Rapid Deployment utility. However, you can trace the application server as described in “WebSphere Application Server logs” on page 122.

**Note:** In a UNIX environment, files and directories starting with a period (.) are hidden. You need to pass the **-a** parameter to the **ls** command to see them.

## 4.2 Work the problem

You begin the problem determination process by evaluating the high level symptoms to determine if one of these describes your problem. If it does not, you collect the appropriate data that is required to diagnose the problem.

Next, you go through the documentation to try to determine the location of the problem. And lastly, you receive guidance on the steps to take for resolution, whether it be a support site, contacting IBM, information about configuration, or some other suggestion as to how to proceed.

### 4.2.1 High-level symptom analysis

Let's start working the problem:

- ▶ If you are not able to access the WebSphere administrative console, go to “Problem: Unable to access the administrative console” on page 127.
- ▶ If you are using wsadmin or the management scripts such as **stopServer**, but these admin tools cannot access the server process, go to “Problem: wsadmin or management scripts can't access server” on page 130.

- ▶ If you are trying to perform administrative tasks against WebSphere Application Server processes (such as starting or stopping a server), consider the following situations:
  - If you are unable to stop an application server, go to “Problem: Unable to stop a server process” on page 133.
  - If you are unable to start an application server, go to “Problem: Unable to start a server process” on page 135.
  - If the deployment manager is not communicating with a node agent, go to “Problem: Unable to access a node agent” on page 137.
  - If you are unable to manage a Web server from the WebSphere administrative tools, go to “Problem: Unable to manage a Web server” on page 138.
  - If you are unable to manage (start, stop, install) an application, go to “Problem: Unable to manage applications” on page 141.
- ▶ If you are unable to federate a node with a deployment manager, go to “Problem: Failure adding a node to a deployment manager” on page 143.
- ▶ If you are making changes in the administrative console and getting messages warning you of a save conflict, go to “Problem: Save conflicts in the administrative console” on page 150.
- ▶ If enterprise applications no longer appear in the administrative console go to “Problem: enterprise applications missing” on page 151.
- ▶ If you are having a problem communicating with your system via SSL or are receiving messages that your certificates have expired, go to “Problem: Invalid or expired certificates” on page 153
- ▶ If you are having problems with WebSphere Rapid Deployment, go to “Problem: WebSphere Rapid Deployment” on page 158.
- ▶ If you cannot create, delete, or update WebSphere Application Server profiles, see *WebSphere Application Server V6: Installation Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4068.pdf>

**Where to go from here:** If, after finding your symptom and going to the appropriate section, or if none of these symptoms resemble your problem, go to “The next step” on page 162.

## 4.3 Analyzing problem areas

Your analysis of the data that you gathered will most likely lead you to one of the following problem areas. If not, see “The next step” on page 162.

### 4.3.1 Problem: Unable to access the administrative console

You are trying to access the WebSphere administrative console and the login page is not being displayed. Instead, you see a blank page or an HTTP 404 error, page not found error.

You experience this problem immediately after installing WebSphere Application Server, after enabling security, after adding a node to the deployment manager, or after restarting the WebSphere Application Server process.

#### **Data to collect**

The WebSphere Application Server logs for the process running the administrative console can be helpful in determining why you cannot access the administrative console.

In a Network Deployment environment, the process will run in the deployment manager. In a stand-alone server environment, the process will run in the server (server1).

You might also need to take an admin trace and a security trace, depending on the problem. See “Collecting data” on page 122 for details.

#### **What to look for**

This problem can occur because the server that hosts the administrative console is not running, or you are trying to connect to the wrong port number. It can also occur if the administrative console files have been accidentally deleted or security is incorrectly configured.

#### ***Verify the port number and URL***

By default, access to the administrative console is via the following URL:

`http://<host>:9060/ibm/console`

However, the port is configurable, and if multiple profiles have been installed on a machine, the port might be different from the default.

Review the WebSphere Application Server SystemOut.log to ensure that the administrative console has started. You will see messages similar to those in Example 4-2 on page 128, showing you that the administrative console has started and on which port the administrative console is listening.

#### Example 4-2 Administrative console startup messages

```
[6/22/05 9:58:31:342 EDT] 0000001b ApplicationMg A WSVR0221I: Application
started: adminconsole
[6/22/05 9:58:31:492 EDT] 0000000a TCPChannel A TCPC0001I: TCP Channel
TCP_1 is listening on host * (IPv4) port 9060.
```

You can also see the port number by looking at the port specified for WC\_adminhost in:

```
<WAS_install_root>/profiles/<profile>/config/cells/<cell>/nodes/<node>/serverin
dex.xml
```

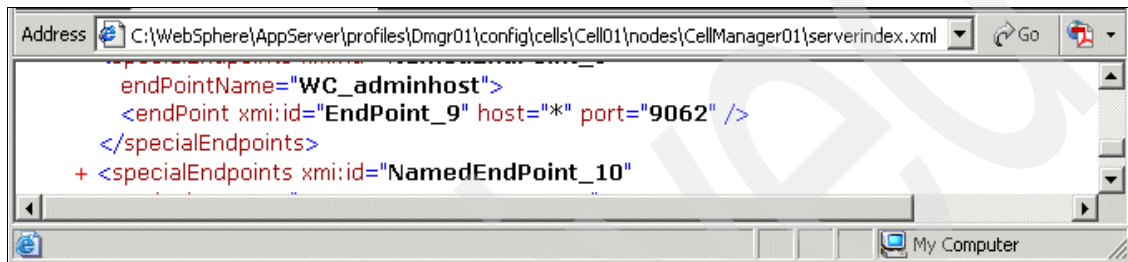


Figure 4-5 Administrative console port number

Ensure that you are connecting to the correct server on the correct port number, for example:

`http://localhost:9060/ibm/console`

#### **Verify that the adminconsole application has started**

If the administrative console fails to start, you will see errors similar to those shown in Example 4-3.

#### Example 4-3 Administrative console failure messages

```
[7/3/05 15:10:29:359 EDT] 0000000a DeployedAppli E WSVR0100W: An error occurred initializing,
adminconsole
java.io.FileNotFoundException: C:\IBM\WAS6\AppServer\systemApps\adminconsole.ear\deployment.xml
(The system cannot find the path specified)at java.io.FileInputStream.open(Native Method)
```

The error indicates the cause of the problem. In Example 4-3, a file is missing. Thus, the administrative console cannot start. The solution to the example problem is to either restore the missing files or to re-install WebSphere Application Server.



**Note:** In WebSphere Application Server V6, the adminconsole.ear and filetransfer.ear applications no longer appear as installed applications and cannot be managed in the same way as other applications. This is to prevent them from being uninstalled accidentally.

### ***Verify the security configuration***

You might not be able to access the administrative console application if security is enabled due to a problem with the digital certificates that are used to setup an SSL communication between your browser and WebSphere Application Server. If this is the case, you would be able to access other applications on the application server that did not use SSL without any problems. Refer to “Problem: Invalid or expired certificates” on page 153 for details.

If your problem is not resolved by fixing invalid certificates, you can confirm that it is related to security by turning security off and retesting.

**How to manually turn off security:** If you have a problem with security, you might not be able to access the administration tools. You can turn off security manually by editing the security.xml file. This allows you to access the administrative console so that you can fix the original problem.

You can find the security.xml file in the configuration directory structure under:

```
<WAS_install_root>/profiles/<profile>/config/cells/<cell>/security.xml
```

Edit the security tag to set enabled to false as shown in the following:

```
<security:Security xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:orb.securityprotocol="http://www.ibm.com/websphere/appserver/schemas/5.0/orb.securityprotocol.xmi"
xmlns:security="http://www.ibm.com/websphere/appserver/schemas/5.0/security.xmi" xmi:id="Security_1" useLocalSecurityServer="true"
useDomainQualifiedUserNames="false" enabled="false" cacheTimeout="600"
issuePermissionWarning="true" activeProtocol="BOTH"
enforceJava2Security="true" enforceFineGrainedJCASecurity="false"
activeAuthMechanism="SWAMAuthentication_1"
activeUserRegistry="LocalOSUserRegistry" defaultSSLSettings="SSLConfig_1">
```

You need to restart the server for this change to take effect.

### 4.3.2 Problem: wsadmin or management scripts can't access server

You are trying to use wsadmin to manage your system, but you are unable to connect to an application server process or are getting no response from wsadmin.

You have this problem when trying to use the supplied management scripts, such as **stopServer.bat** or **serverStatus.bat**.

You experience this problem immediately after installing WebSphere Application Server, after enabling security, after adding a node to the deployment manager, or after restarting the WebSphere Application Server server.

#### Data to collect

The following logs can be helpful in determining why you cannot administer WebSphere Application Server with wsadmin:

- ▶ Application server logs
- ▶ wsadmin trace log
- ▶ Management script log files if appropriate
- ▶ Trace from the management script if appropriate

If you are trying to manage a service using a management script, such as **stopServer.bat**, then you need the log file from that script (stopServer.log) and possibly a trace of that script.

#### What to look for

This problem can occur because the application server to which you are trying to connect is not running, or you are trying to connect to the wrong port number.

The wsadmin utility connects to an application server in a stand-alone environment or deployment manager in a distributed environment. The following error shows that no connection could be established:

```
WASX7023E: Error creating "SOAP" connection to host "hostname"
```

Make sure that the application server is running and that you are using the correct port number to connect to the application server or deployment manager.

#### Verify the port number

You can determine to which server wsadmin is trying to connect and on what port by looking at the wsadmin.properties file. This file is in the profile directory structure:

```
<WAS_install_root>/profiles/<profile>/properties/wsadmin.properties
```

Example 4-4 shows the host name and port number that wsadmin uses by default.

---

*Example 4-4 SOAP host and port for a stand-alone server*

---

```
com.ibm.ws.scripting.host=localhost
com.ibm.ws.scripting.port=8879
```

---

You can compare this to the entry in the SystemOut.log from the application server that tells you on which port the application server is actually listening for SOAP connections, as shown in Example 4-5.

---

*Example 4-5 SOAP connector port number from the SystemOut.log*

---

```
[7/8/05 14:19:55:445 EDT] 0000000a JMXSoapAdapte A   ADMC0013I: The SOAP
connector is available at port 8880
```

---

If the host and port are different, then your wsadmin.properties file is out of sync with the server. This can happen if you change the SOAP port after the profile has been created. When you create a profile, the SOAP port is hard coded into wsadmin.properties. If you later change the SOAP port, the administrative console updates the file that contains the port assignments for the server process (serverindex.xml) but does not update the wsadmin.properties file. You must do this manually.

Alternatively, you can pass the port number to wsadmin with the **-port** parameter:

```
wsadmin -host k1l6571 -port 8880
```

In a Network Deployment installation, wsadmin attempts to send requests to the deployment manager. Again, the wsadmin.properties file contains the host name and port number of the deployment manager process. Using the **-host** and **-port** parameters, you can use wsadmin to connect directly to a process other than the deployment manager. For example you could connect directly to a node agent or remote application server. Example 4-6 shows a connection to a node agent and then issuing a **stopServer** command.

---

*Example 4-6 Connecting directly to a node agent using wsadmin*

---

```
$ ./wsadmin.bat -host m23vnx60.itso.ral.ibm.com -port 8881
WASX7209I: Connected to process "nodeagent" on node m23vnx60Node01 using SOAP
connector; The type of process is: NodeAgent
WASX7029I: For help, enter: "$Help help"
wsadmin>$AdminControl stopServer nodeagent
WASX7337I: Invoked stop for server "nodeagent" Waiting for stop completion.
```

---

**Note:** In Example 4-6, you will not receive any further response because the server shuts down. Use **Ctrl+c** to exit the utility.

The WebSphere Information Center describes the full syntax of all wsadmin parameters in the section *Wsadmin tool* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rxml\\_commandline.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rxml_commandline.html)

### ***Verify that the application server is running***

You can use the **netstat** command to ensure that a process is listening on the port. Example 4-7 shows the output of the **netstat** command that has been filtered with the **grep** tool to only show lines containing the string 8880.

*Example 4-7 netstat output*

---

```
C:\IBM\WAS6\AppServer\profiles\StandAlone\bin>netstat -an|grep 8880
TCP    0.0.0.0:8880          0.0.0.0:0           LISTENING
TCP    9.42.171.145:8880    9.42.171.145:1114    CLOSE_WAIT
TCP    127.0.0.1:8880       127.0.0.1:1122       CLOSE_WAIT
TCP    127.0.0.1:8880       127.0.0.1:1161       CLOSE_WAIT
```

---

**Note:** The **netstat** command is available on both UNIX and Windows. The **grep** command is available on UNIX by default, and you can download free copies of **grep** for Windows.

You can also use the **telnet** command to try and get a connection to the host name and port as shown in Example 4-8.

*Example 4-8 Using telnet to test a port*

---

```
C:\IBM\WAS6\AppServer\profiles\StandAlone\bin>telnet localhost 8880
Connecting To localhost...Could not open a connection to host on port 8880 :
Connect failed
```

---

Note that even if you can connect, you will not be able to do anything because WebSphere Application Server does not respond to telnet commands. You can close your telnet session by pressing **Ctrl+c**.

If telnet is not able to connect, then the server is not started or not accepting any form of connection. Try restarting the application server. You probably need to kill the Java process because it will not accept the **stopServer** command.

### ***Verify the security configuration***

If security is enabled, then you need to pass a user name and password to wsadmin in order to connect. If you do not pass a user name or password, or get them wrong, you get an error connecting, as shown in Example 4-9

#### ***Example 4-9 Incorrect user name or password***

---

WASX7246E: Cannot establish "SOAP" connection to host "localhost" because of an authentication failure. Ensure that user and password are correct on the command line or in a properties file.

Exception message (if any): "ADMN0022E: Access is denied for the getProcessType operation on Server MBean because of insufficient or empty credentials."

---

You might not be able to connect to the application server or deployment manager if security is enabled, due to a problem with the digital certificates that are used to setup an SSL communication. See "Problem: Invalid or expired certificates" on page 153 for details.

If your problem is not resolved by fixing invalid certificates, you can confirm that it is related to security by turning security off and retesting. If you cannot access the administrative console, you can turn security off by editing the security.xml file as described in "Problem: Unable to access the administrative console" on page 127.

### **4.3.3 Problem: Unable to stop a server process**

You are trying to stop an application server through the administrative console but are getting failure messages.

#### **Data to collect**

The following logs and output can be helpful in determining why you cannot stop an application server:

- ▶ Deployment manager SystemOut.log, SystemErr.log
- ▶ Node agent SystemOut.log, SystemErr.log
- ▶ Application server SystemOut.log, SystemErr.log

If you are trying to manage a server using a management script, such as **stopServer.bat**, then you need the log file from that script (stopServer.log) and possibly a trace of that script.

## What to look for

Figure 4-6 shows the result that you would see in the administrative console if you tried to stop an application server that had hung. It takes some time for the stop request to time out.



Figure 4-6 Administrative console error stopping an application server

In this case, the SystemOut.log for the deployment manager will most likely not contain any relevant messages. However, in the SystemOut.log for the node agent, you might see messages similar to those that are shown in Example 4-10.

### Example 4-10 Node agent error stopping an application server

---

```
[7/11/05 13:51:54:207 EDT] 0000000d ThreadMonitor W   WSVR0605W: Thread
"SoapConnectorThreadPool : 6" (00000032) has been active for 607027 milliseconds and may be
hung. There is/are 1 thread(s) in total in the server that may be hung.
```

---

In this example, the problem is that the application server is hung and cannot respond to requests. You might also see a situation in which the node agent is hung, as shown in Example 4-11.

### Example 4-11 Hung node agent error messages

---

```
[7/11/05 14:19:00:918 EDT] 0000000f ThreadMonitor W   WSVR0605W: Thread "WebContainer : 1"
(00000038) has been active for 619290 milliseconds and may be hung.
There is/are 1 thread(s) in total in the server that may be hung.
```

---

In the case where the node agent is hung, the administrative console will eventually time out after at least 10 minutes. You will see a message in the deployment manager's SystemOut.log before the time out.

If an application server is hung, you can stop it by killing the Java process. You can get the process ID of the server from startServer.log in the server's logs directory. Make sure you scroll to the end of the file to get the process ID from the latest start. In a Windows environment, you can use the Task Manager to stop the task. In a UNIX environment, you can use the **kill** command to kill the Java process. After you have killed the hung Java process, you need to restart the server.

**Note:** In the Windows Task Manager, the process ID is not shown by default. However you can add the process ID (PID) using the **View** → **Select Columns** option.

Stopping a hung process is simply a work-around. The real problem is the hang.

#### 4.3.4 Problem: Unable to start a server process

If you are unable to start an application server, the problem is likely to be either a configuration or an application error. Both the administrative console and the **startServer** script output will return an error telling you where to look.

**Note:** The steps outlined here are also true for a deployment manager or node agent.

##### Data to collect

If you are trying to start the server through the administrative console, open a command window and try to start the server using the **startServer** command.

The following logs and output can be helpful in determining why you cannot start a server:

- ▶ Deployment manager SystemOut.log, SystemErr.log
- ▶ Node agent SystemOut.log, SystemErr.log
- ▶ Application server SystemOut.log, SystemErr.log
- ▶ startServer.log

If these logs do not make clear the cause of the problem, you might also need to trace the script.

##### What to look for

When a server fails to start, you need to review the output from the **startServer** command and the logs for that application server for messages that tell you at what point the server failed.

Example 4-12 shows the output from a **startServer** command that indicates why a server will not start.

*Example 4-12 Unable to start a server from the command line*

---

```
C:\IBM\WAS6\AppServer\profiles\StandAlone\bin>startserver server1
ADMU7701I: Because server1 is registered to run as a Windows Service, the
           request to start this server will be completed by starting the
           associated Windows Service.
ADMU0116I: Tool information is being logged in file

C:\IBM\WAS6\AppServer\profiles\StandAlone\logs\server1\startServer.log
ADMU0128I: Starting tool with the StandAlone profile
ADMU3100I: Reading configuration for server: server1
ADMU3028I: Conflict detected on port 8880. Likely causes: a) An instance of
```

```

        the server server1 is already running b) some other process is
        using port 8880
ADMU3027E: An instance of the server may already be running: server1
ADMU0111E: Program exiting with error:
        com.ibm.websphere.management.exception.AdminException: ADMU3027E: An
        instance of the server may already be running: server1
ADMU1211I: To obtain a full trace of the failure, use the -trace option.
ADMU0211I: Error details may be seen in the file:

C:\IBM\WAS6\AppServer/profiles/StandAlone\logs\server1\startServer.log

```

---

In Example 4-12, the output tells us that some other process is using one of the ports that this server needs in order to start. You can confirm that the port is in use with the **netstat** command.

Example 4-13 shows that some process is listening on port 8880. In UNIX, you can determine what process is listening on that port using the **lsof** command.

*Example 4-13 Output from the netstat command*

---

TCP	9.42.171.145: <b>8880</b>	9.42.171.145:2698	TIME_WAIT
TCP	9.42.171.145: <b>8880</b>	9.42.171.145:2739	TIME_WAIT

---

**Note:** The **netstat** command is available on both UNIX and Windows.

Example 4-14 shows the output from the **lsof** command. This output tells you that a Java process is listening on TCP port 8880. The process ID is 12968.

*Example 4-14 Using lsof to check for open ports*

---

```

[root@m23vnx60 root]# lsof -i|grep 8880
java      12968 wasadmin 135u  IPv4 655014      TCP *:8880 (LISTEN)

```

---

**Note:** There are Windows equivalents of the **lsof** command available for download from the Internet.

It is also possible that the server did actually start but that the administration utility lost contact with it and so reported that it could not start. This problem can occur in environments where there are network components between the application server and the command being issued, such as a firewall. This problem can also occur in a Windows environment because the **startServer** command calls the Windows services manager to start the service. These two components can also lose contact with each other as shown in Example 4-15 on page 137.



By checking the server logs, you can determine if the server has actually started.

*Example 4-15 Starting a server loses contact*

---

```
C:\IBM\WAS6\AppServer\profiles\StandAlone\bin>startserver server1
ADMU7701I: Because server1 is registered to run as a Windows Service, the
           request to start this server will be completed by starting the
           associated Windows Service.
ADMU0116I: Tool information is being logged in file

C:\IBM\WAS6\AppServer\profiles\StandAlone\logs\server1\startServer.log
ADMU0128I: Starting tool with the StandAlone profile
ADMU3100I: Reading configuration for server: server1
ADMU3200I: Server launched. Waiting for initialization status.
java.lang.reflect.InvocationTargetException
           at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
...
Caused by: com.ibm.websphere.management.exception.AdminException: Failed while
trying to start the Windows Service, retCode = -1
```

---

You might be able to resolve this by setting the SOAP timeout parameter in the following file:

```
<WAS_install_root>/profiles/<profile>/properties/soap.client.props
```

You can find information about this setting in the *Java Management Extensions connector properties* item in the WebSphere Information Center at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ragt\\_rconnector\\_customp.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ragt_rconnector_customp.html)

If the application server logs do not show the server trying to start, then it is possible that the deployment manager or node agent are not able to communicate with the application server. Reviewing the deployment manager or node agent logs should help you determine the cause of the failure.

### 4.3.5 Problem: Unable to access a node agent

You have recently restarted the deployment manager and discovered that one of your node agents does not appear to be available.

#### Data to collect

The following logs can be helpful in determining why your node agent is not available:

- ▶ Deployment manager logs
- ▶ Node agent logs

However, depending on the cause of the problem, you might or might not see any helpful messages in the logs.

### **What to look for**

Check the node agent to ensure that it is actually running by looking for its process and checking its SystemOut.log. You could also ensure that it is running by trying to connect to the node agent directly with wsadmin as shown in Example 4-6 on page 131.

If the node agent is running and the deployment manager is not reporting any errors but is still saying that the node agent is down, it is possible that a DNS error occurred as the deployment manager was starting.

By default, Java caches DNS entries for the life of a Java process. If a DNS error occurred as the deployment manager was starting, the deployment manager cannot resolve the name of the node agent. Even though, the DNS can be fixed and you might be able to resolve the node agent server name now from the command line, WebSphere Application Server has cached an incorrect or missing entry. This problem can be resolved by restarting the deployment manager.

## **4.3.6 Problem: Unable to manage a Web server**

WebSphere Application Server V6 has the ability to manage the IBM HTTP Server V6 from within the administrative console.

You are having problems starting, stopping, or getting the status of the Web server.

### **Data to collect**

The following logs can be helpful in determining why you cannot manage a Web server:

- ▶ Deployment manager logs
- ▶ Node agent logs
- ▶ Web server logs

### **What to look for**

You can start and stop any of the supported Web servers using the WebSphere administrative console. The Web server must be installed on a managed node. A WebSphere Application Server node agent executes the appropriate start or stop command.

When you are using IBM HTTP Server V6.0, you do not need to install the Web server on a managed node because administration can be performed through the IBM HTTP Administration Server. In addition to starting and stopping the IBM HTTP Server V6.0, you can also manage its configuration file and view the Web server logs.

### ***IBM HTTP Server on an unmanaged node***

If you cannot administer the IBM HTTP Server from the WebSphere administrative console, verify that the IBM HTTP Server administration server has been properly set up.

In order for the administrative console to access the IBM HTTP administration server, you must define a valid user ID and password to access the IBM HTTP Server administration server. The user ID and password are stored in the Web server's IBM HTTP Server administration server properties.

You can update your IBM HTTP Server administration server properties in the Web server definition through the Remote Web server management properties page of the administrative console. To set or change these properties, do the following:

1. Click **Servers** → **Web servers**.
2. Select the Web server.
3. Click **Remote Web server management** in the Additional Properties section.
4. Ensure that the user ID and password are correct by entering values that are defined to the IBM HTTP administration server.
5. Click **OK** and save the configuration.

#### **Setting the user ID and password in the IBM HTTP administration server:**

The IBM HTTP administration server is set, by default, to look at the following file to get the user ID and passwords to use for authentication:

```
<ihs_install>/conf/admin.passwd
```

To initialize this file with a user ID, use the **htpasswd** command. The following example initializes the file with the user ID **webadmin**:

```
C:\IBM HTTP Server\bin>htpasswd "C:\IBM HTTP Server\conf\admin.passwd"  
webadmin
```

```
Automatically using MD5 format.
```

```
New password: *****
```

```
Re-type new password: *****
```

```
Adding password for user webadmin
```

When you are managing an IBM HTTP Server using the WebSphere administrative console, you must ensure that the following conditions are met:

- ▶ Verify that the IBM HTTP Server administration server is running.
- ▶ Verify that the Web server host name and port that are defined in the WebSphere administrative console match the IBM HTTP Server administration host name and port.
- ▶ Verify that the firewall is not preventing you from accessing the IBM HTTP Server administration server from the WebSphere administrative console.
- ▶ If you are trying to administer the IBM HTTP Server over a secure SSL connection, verify that you have exported the IBM HTTP Server administration server key database personal certificate and imported it into the WebSphere key database as a signer certificate. The key database will be identified in the com.ibm.ssl.trustStore parameter in the sas.client.props file.
- ▶ Verify the IBM HTTP Server admin\_error.log file and the WebSphere Application Server logs (trace.log) do not contain any errors.

### ***IBM HTTP Server on a managed node***

When the IBM HTTP Server on a managed node is started using WebSphere administration tools. The node agent issues the same command that you would use from a command line, that is:

```
<Web_server_home>/bin/apachectl
```

Check to make sure the Web server can be started via other means, such as through the Windows control panel or from the command line. If it cannot be started, return to *Approach to Problem Determination in WebSphere Application Server V6* to see how to proceed with Web server problems. You can find this paper at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

The deployment manager does not log any messages when you try to start a Web server. The node agent logs a simple message to tell you that it is trying to start the Web server as shown in Example 4-16.

If the Web server fails to start, then you get a failure message in the administrative console and a message in the node agent log, as shown in Example 4-16.

#### ***Example 4-16 Node agent messages starting a Web server***

---

```
[7/12/05 14:56:15:658 EDT] 000000a6 AdminHelper    A    ADMN1001I: An attempt is made to launch
webserver1 on node m23vnx60Node01.
[7/12/05 14:56:15:661 EDT] 000000a6 NodeAgent      W    ADML0065W: A sync operation prior to
starting an application server failed.
```

You need to review the Web server logs to determine what went wrong.

Also, you can trace both the deployment manager and node agent process, as described in “Collecting data” on page 122, to gather further information about the source of a failure.

### 4.3.7 Problem: Unable to manage applications

You are trying to manage applications, for example starting, stopping, or installing an application, and are having problems.

You get these problems while installing an application, after installing an application, or after a server restart.

#### Data to collect

The following logs and output can be helpful in determining why you cannot manage WebSphere Application Server services:

- ▶ Deployment manager logs
- ▶ Node agent logs
- ▶ Application server logs

#### What to look for

When an error occurs when you are trying to manage an application, the following are things you can do:

- ▶ Verify that there are no resource or classpath errors
- ▶ Verify that the application was installed correctly
- ▶ Verify that the configuration repository is not corrupted

#### ***Verify that there are no resource or classpath errors***

The application server log files are the first place to look as any problems with the application itself, such as classpath issues or resource issues, are reported there. The error message from the administrative console tells you which node and server to check, as shown in Figure 4-7 on page 142. Should you find such an error, work with your application developers to resolve the application coding or resource problem.

#### ***Verify that the application was installed correctly***

If the application fails to start on a server after it has been installed, the installation might not have completed successfully. The installation might have

appeared to be successful, but when you try to start the application, it fails with a message in the administrative console as shown in Figure 4-7 on page 142.

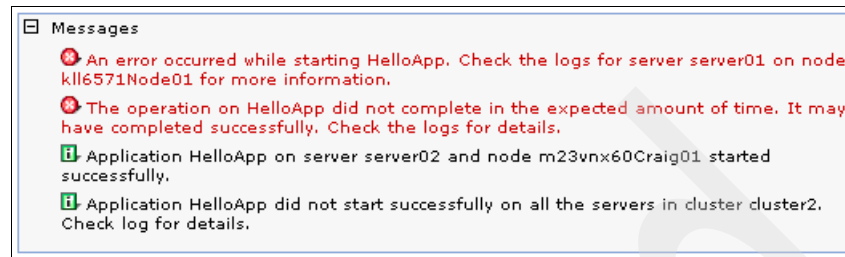


Figure 4-7 Message from administrative console when application fails to start

This message also appears in the deployment manager SystemOut.log, as shown in Example 4-17.

*Example 4-17 Message in deployment manager log when application fails to start*

---

```
[7/12/05 9:44:07:729 EDT] 000001ea MBeanHelper E Could not invoke an operation on object:
WebSphere:platform=dynamicproxy,cell=kll6571Cell01,version=6.0.1.2,name=ApplicationManager,mbeanIdentifier=ApplicationManager,type=ApplicationManager,node=kll6571Node01,process=server01
because of an mbean exception: com.ibm.ws.exception.ConfigurationWarning: Application HelloApp
not installed
```

---

The application server SystemOut.log shows a similar message, as shown in Example 4-18.

*Example 4-18 Message in application server log when application fails to start*

---

```
[7/12/05 9:44:07:379 EDT] 00000039 ApplicationMg W WSVR0215W: Starting application, HelloApp,
failed. The application is not installed.
```

---

In this example, the application was not installed on this node because automatic synchronization was disabled for the node. The administrator did not perform manual synchronization when saving the application installation changes to the master configuration repository. You can resolve this problem by synchronizing the node.

***Verify that the configuration repository is not corrupted***

Corruptions in the configuration repository can also cause enterprise applications to not start. For example, a node's serverindex.xml file contains a listing of all servers and ports in a cell with which the node might need to communicate. It also lists applications that are deployed on the servers in those nodes.

If this file becomes corrupt and the list of applications is lost, then the applications will not be started on the server in this node. When the server process starts, it thinks that the application is not installed and does not try and start it. If you try and start the application from the administrative console, you see the message that is shown in Figure 4-7 on page 142. The application server logs tell you that the application is not installed, as shown in Example 4-18 on page 142.

You can resolve this problem by restoring the `serverindex.xml` file from a backup or forcing the deployment manager to transfer the uncorrupted copy of the file from the master repository. You would do this by updating the master copy of the file but without changing it so that it appears to have been updated to the deployment manager.

For example:

1. Open the file in a text editor.
2. Add a single character anywhere in the file.
3. Delete the character you just added.
4. Save the file.

Then in the administrative console, navigate to **System Administration** → **Nodes**. Select the node and choose **Full Resynchronize**, as shown in Figure 4-8.

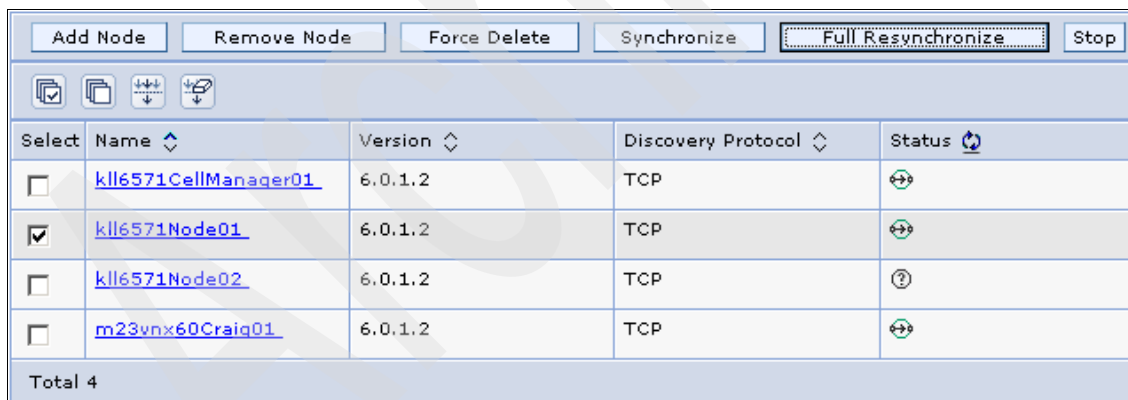


Figure 4-8 Full synchronization

### 4.3.8 Problem: Failure adding a node to a deployment manager

A WebSphere Application Server node is federated to a deployment manager using the `addNode` command. This can sometimes fail.

## Data to collect

The following logs can be helpful in determining why you cannot add a node to a deployment manager:

- ▶ Management script log file: addNode.log
- ▶ Deployment manager logs

The **addNode** command connects to the deployment manager and executes the MBeans that it needs to add the node to the cell. It also writes its own log file. You can increase the amount of data that is written by passing the **-trace** parameter. Trace output is written to the same log file:

```
addNode.sh DMhost port -trace
```

Depending on the cause of the failure, you might also need to look at the logs on the deployment manager.

## What to look for

Example 4-19 shows that the connection from the addNode script to the deployment manager timed out before the work completed.

### Example 4-19 Errors from the addNode script

---

```
ADMU0116I: Tool information is being logged in file
/opt/IBM/WAS6/AppServer/profiles/Craig02/logs/addNode.log
ADMU0128I: Starting tool with the Craig02 profile
ADMU0001I: Begin federation of node m23vnx60Node02 with Deployment Manager at
k116571:8879.
ADMU0009I: Successfully connected to Deployment Manager Server: k116571:8879
ADMU0507I: No servers found in configuration under:

/opt/IBM/WAS6/AppServer/profiles/Craig02/config/cells/m23vnx60Node01Cell/nodes/
m23vnx60Node02/servers
ADMU2010I: Stopping all server processes for node m23vnx60Node02
ADMU0024I: Deleting the old backup directory.
ADMU0015I: Backing up the original cell repository.
ADMU0012I: Creating Node Agent configuration for node: m23vnx60Node02
ADMU0014I: Adding node m23vnx60Node02 configuration to cell: k116571Cell01
ADMU0026I: An error occurred during federation; rolling back to original
configuration.
ADMU2018I: Node m23vnx60Node02 has been removed from the Deployment Manager
configuration.
ADMU0113E: Program exiting with error:
com.ibm.websphere.management.exception.ConnectorException:
ADMC0009E: The system failed to make the SOAP RPC call: invoke,
resulting from: [SOAPException: faultCode=SOAP-ENV:Client; msg=Read
timed out; targetException=java.net.SocketTimeoutException: Read
timed out]
ADMU1211I: To obtain a full trace of the failure, use the -trace option.
```



ADMU0211I: Error details may be seen in the file:  
/opt/IBM/WAS6/AppServer/profiles/Craig02/logs/addNode.log

---

You can get more detailed information from the addNode log as shown in Example 4-20 on page 145.

*Example 4-20 Errors from the addNode.log*

---

```
[7/12/05 15:20:02:155 EDT] 0000000a AdminTool      A   ADMU0113E: Program
exiting with error: com.ibm.websphere.management.exception.ConnectorException:
ADMC0009E: The system failed to make the SOAP RPC call: invoke
      at
com.ibm.ws.management.connector.soap.SOAPConnectorClient.invokeTemplate(SOAPCon
nectorClient.java:642)
...
resulting from: [SOAPException: faultCode=SOAP-ENV:Client; msg=Read timed out;
targetException=java.net.SocketTimeoutException: Read timed out]
      at org.apache.soap.transport.http.SOAPHTTPConnection.send(Unknown
Source)
```

---

In this instance, the problem was that the SOAP connection timed out before the node could be added. You would need to increase the SOAP time out parameter (com.ibm.SOAP.requestTimeout) in the following file and retry:

<WAS\_install\_root>/profiles/<profile>/properties/soap.client.props

Depending on where in the process the failure occurred, you might need to remove a partially federated node before trying again. To do this, you need the **-force** parameter:

removeNode.sh -force

### 4.3.9 Problem: Repository synchronization

You are making changes to your configuration in a Network Deployment environment and while you can save the changes, the synchronization request to a remote node is failing.

#### Data to collect

The following logs can be helpful in determining why you cannot synchronize the repository:

- ▶ Deployment manager logs
- ▶ Node agent logs

## What to look for

Review SystemOut.log from the deployment manager and from the node agent on the server that is not synchronizing. The administrative console message will tell you which node is failing to synchronize as shown in Figure 4-9.

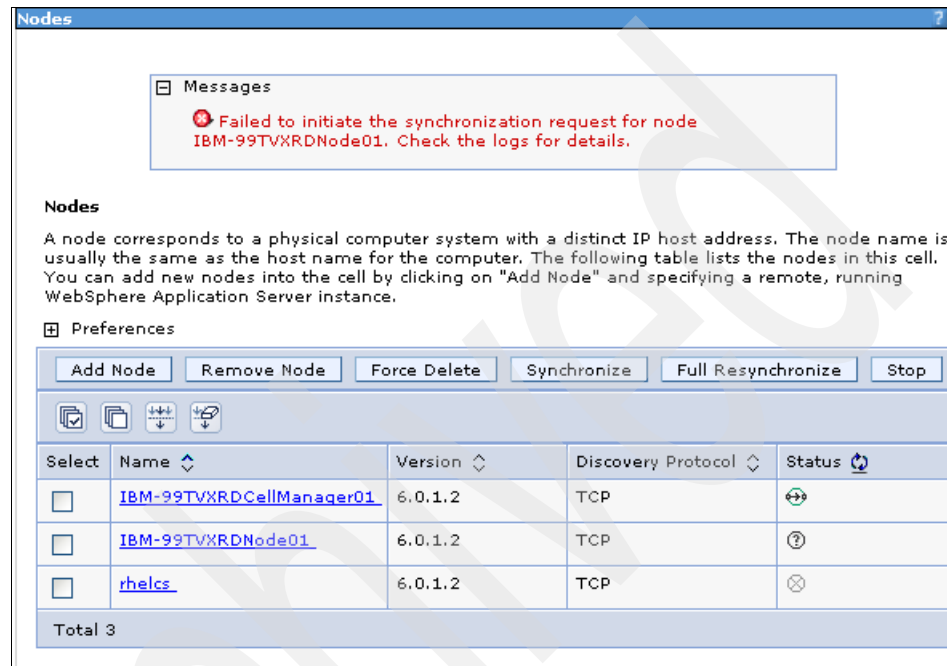


Figure 4-9 Output from a failed node synchronization

The message shown in Example 4-21 appears in the SystemOut.log of the deployment manager.

### Example 4-21 Unable to synchronize a node

```
[7/3/05 16:01:17:845 EDT] 00000026 CellSync      E    ADMS0104I: The system is unable to invoke
a synchronization request on node
WebSphere:platform=common,cell=IBM-99TVXRCell01,version=6.0.1.2,name=nodeSync,mbeanIdentifier=
nodeSync,type=NodeSync,node=IBM-99TVXRNode01,process=nodeagent.
javax.management.InstanceNotFoundException: MBeanServer cannot find MBean with ObjectName
WebSphere:platform=common,cell=IBM-99TVXRCell01,version=6.0.1.2,name=nodeSync,mbeanIdentifier=
nodeSync,type=NodeSync,node=IBM-99TVXRNode01,process=nodeagent
```

When synchronization works, you see the following message in the node agent SystemOut.log:

```
[7/3/05 16:05:03:096 EDT] 0000002c NodeSyncTask  A    ADMS0003I: The
configuration synchronization completed successfully.
```

The replicated copy of the file being modified might be corrupted or inaccessible on the node where the change is being made. For example, making a change to the Java heap size parameter updates the server.xml file in the repository. The change is made to the master copy of the file, and the administrative console returns a message that the change was saved and that the synchronization was successful, as shown in Figure 4-10.

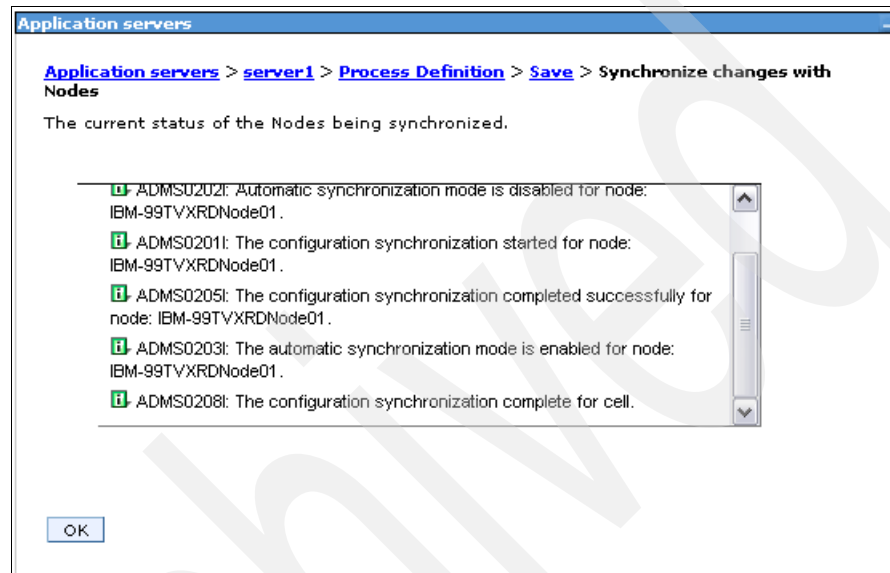


Figure 4-10 Output from a successful save and synchronization

The deployment manager's SystemOut.log also shows the synchronization as successful:

```
[7/3/05 16:11:13:343 EDT] 00000085 DeploymentMan A ADMS0208I: The
configuration synchronization complete for cell.
```

However, the node agent's SystemOut.log shows you that the node agent is not able to open the file due to permissions, as shown in Example 4-22.

*Example 4-22 Node agent fails to open server.xml*

```
[7/3/05 16:11:13:902 EDT] 00000040 FileDocument E ADMR0104E: The system is unable to read
document cells/IBM-99TVXRDCel101/nodes/IBM-99TVXRNode01/servers/server1/server.xml:
java.io.FileNotFoundException:
C:\IBM\WAS6\AppServer\profiles\AppSrv01\config\cells\IBM-99TVXRDCel101\nodes\IBM-99TVXRNode01\
servers\server1\server.xml (Access is denied)
[7/3/05 16:11:13:912 EDT] 00000040 NodeSyncTask A ADMS0016I: The configuration
synchronization failed.
```

In this case, the solution is to fix the file permissions and retry the synchronization.

In other cases, such as with a corrupt or empty file, the solution is to delete the file and allow automatic synchronization to replace the file with the good copy from the master repository.

If automatic synchronization is not enabled, you need to manually synchronize the node either from the administrative console or from the node itself.

Figure 4-11 shows the file synchronization settings for the node agent. In this instance, automatic synchronization is enabled.

**Node agents > nodeagent > File synchronization service**

The file synchronization service runs in the deployment manager and node agents, and ensures that configuration changes made to the cell repository are propagated to the appropriate node repositories. Specify the settings to configure the file synchronization service.

Configuration

**General Properties**

- ☒ Enable service at server startup
- \* Synchronization interval: 1
- ☒ Automatic synchronization
- ☐ Startup synchronization

**Exclusions**

Apply OK Reset Cancel

Figure 4-11 Setting up automatic or manual synchronization

The **Enable service at server startup** option means that when the node agent restarts, the file synchronization service is also started. If this is not enabled, the node agent simply starts the service when needed. You get to this page in the administrative console by selecting **System Administration** → **Node agents** → **nodeagent** → **File synchronization service**.

If you are unable to determine why synchronization is failing from the messages in the logs, try a command line synchronization initiated from the node that you cannot synchronize. You need to stop the node agent. From a command line, issue the **syncNode** command, as shown in Figure 4-23. You need to pass it the host name and SOAP port of the deployment manager. You can also specify the **-trace** option.

*Example 4-23 Running and tracing syncNode*

---

```
[wasadmin@m23vnx60 bin]$ ./syncNode.sh k116571 8879 -trace
ADMU0115I: Trace mode is on.
ADMU0116I: Tool information is being logged in file
/opt/IBM/WAS6/AppServer/profiles/Node01/logs/syncNode.log
ADMU0401I: Begin syncNode operation for node m23vnx60Node01 with Deployment
Manager k116571: 8879
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0111E: Program exiting with error:
com.ibm.websphere.management.exception.AdminException: ADMU0005E:
Error synchronizing repositories
ADMU0211I: Error details may be seen in the file:
/opt/IBM/WAS6/AppServer/profiles/Craig01/logs/syncNode.log
```

---

**Note:** You can stop a node agent without having to stop the application servers. The operational impact of stopping the node agent is that it will not monitor the application servers to make sure they are up.

Reviewing the log file shows the problem as shown in Example 4-24.

*Example 4-24 syncNode fails to open server.xml*

---

```
[7/11/05 12:14:00:448 EDT] 0000000c FileDocument E ADMR0105E: The system is unable to write
document cells/k116571Cell01/nodes/m23vnx60Node01/servers/server02/server.xml:
java.io.FileNotFoundException:
/opt/IBM/WAS6/AppServer/profiles/Node01/config/cells/k116571Cell01/nodes/m23vnx60Craig01/server
s/server02/server.xml (Permission denied)
```

---

In this example, it was not necessary to run the trace because the error message would have appeared anyway. A trace is useful for finding the problem when there are no obvious errors.

The message in Example 4-25 on page 150 often appears in the syncNode log and can usually be ignored. It happens because variables can be defined at several different scopes in the configuration. The synchronization code checks to see if the file exists at all levels in the scope and returns this message when one is not found. It is not an error.

```
java.io.FileNotFoundException:  
/opt/IBM/WAS6/AppServer/profiles/Node01/config/cells/k116571Cell01/nodes/m23vnx60Node01/servers  
/nodeagent/variables.xml (No such file or directory)
```

### 4.3.10 Problem: Save conflicts in the administrative console

You are seeing messages telling you that there is a save conflict in the administrative repository when you login to the administrative console.

#### Data to collect

The following logs and output can be helpful in determining why you are getting save conflicts:

- ▶ Deployment manager log (Network Deployment installation)
- ▶ Application server log (Stand-alone server installation)

#### What to look for

After you made changes to the WebSphere Application Server configuration in the administrative console and try and save those changes, you see a message telling you there is a save conflict as shown in Figure 4-12.



Figure 4-12 Save conflict

This error tells you that there is a conflict in updates to one or more files in the configuration repository. The save conflict is not reported in the SystemOut.log unless you choose to overwrite the conflicting change or have enabled tracing.

A save conflict can happen for a number of reasons:

- ▶ More than one administrator has made changes to the configuration at the same time.
- ▶ Changes have been made to the configuration repository files directly.
- ▶ You are deploying applications using WebSphere Rapid Deployment.

You can resolve this problem by either discarding your changes or by overwriting the changes that the other administrator might have made. If you discard your changes, you need to go back and start again. If you choose to overwrite the other changes, the other administrator is not notified and is not aware that their changes have been overwritten. However, the administrative console does log the overwrite in the SystemOut.log as shown in Example 4-26.

*Example 4-26 Overwriting configuration changes*

---

```
[7/11/05 11:20:03:348 EDT] 00000037 FileRepositor A   ADMR0016I: User WTRNTDM/config modified
document cells/k116571Node01Cell/nodes/k116571StandAlone/servers/server1/server.xml.
[7/11/05 11:20:10:569 EDT] 0000003f ServletWrapper A   SRVE0242I:
[/secure/tiles/synconflict.jsp]: Initialization successful.
[7/11/05 11:20:23:808 EDT] 00000038 FileRepositor W   ADMR0114W: The system is overwriting
document cells/k116571Node01Cell/nodes/k116571StandAlone/servers/server1/server.xml by request.
[7/11/05 11:20:23:848 EDT] 00000038 FileRepositor A   ADMR0016I: User WTRNTDM/wasadmin modified
document cells/k116571Node01Cell/nodes/k116571StandAlone/servers/server1/server.xml.
```

---

It is possible to have more than one administrator working on the repository at the same time and not have problems with save conflicts when those administrators are working with different files and objects in the repository. Unless you constantly check the SystemOut.log, you cannot be sure that the changes you are making are not being overwritten. Thus, we recommend that you only have one administrator making changes at a time.

**Note:** You can define roles in WebSphere, including a Monitor role, who can view the configuration and performance data but not make changes. Consider using the Monitor role to allow people to access the administrative console but avoid possible save conflicts.

### 4.3.11 Problem: enterprise applications missing

You have logged into the administrative console and are trying to manage enterprise applications but none are displayed. The applications might be working fine operationally, you just cannot manage them. This problem is most likely to happen after a server restart.

## Data to collect

The following logs can be helpful in determining why applications are missing:

- ▶ Deployment manager log
- ▶ Node agent log
- ▶ Application server log

## What to look for

From the administrative console, you should be able to see a list of installed applications under **Applications** → **Enterprise Applications**. If your applications are no longer listed here, start by checking the deployment manager log. This problem can be caused by a corruption to the configuration repository or problems with permissions, as shown in Example 4-27.

### *Example 4-27 Permissions problem in deployment manager log*

---

```
[7/13/05 14:09:48:993 EDT] 0000003a FileDocument E ADMR0104E: The system is unable to read
document cells/m23vnx60Cell01/applications/HelloApp.ear/deployments/IBMUTC/deployment.xml:
java.io.IOException: No such file or directory
```

---

Reviewing the node agent's log would show you a similar message, possibly referring to a different file, as shown in Example 4-28.

### *Example 4-28 Permissions problem in node agent log*

---

```
[7/13/05 14:12:03:979 EDT] 00000035 FileDocument E ADMR0109E: An error occurred restoring
document
cells/m23vnx60Cell01/applications/HelloApp.ear/deployments/HelloApp/META-INF/was.policy:
java.io.FileNotFoundException:
/opt/IBM/WAS6/AppServer/profiles/AppSrv01/config/cells/m23vnx60Cell01/applications/HelloApp.ear
/deployments/HelloApp/META-INF/was.policy (Permission denied)
```

---

The application server log also shows that the problem is caused by file permissions as shown in Example 4-29.

### *Example 4-29 Permissions problem in application server log*

---

```
[7/13/05 13:55:26:517 EDT] 0000001b FileDocument E ADMR0104E: The system is unable to read
document cells/m23vnx60Cell01/nodes/m23vnx60Node01/perftuners.xml: java.io.IOException:
Permission denied
```

---

In all three cases, the messages refer to the different files and different locations in the file system. Checking the file permissions for those files listed shows that the user who is running the processes does not have write access to those files.

In this example, the servers run on a UNIX platform and are configured to run as a non-root user under normal conditions. At some point, the servers were



restarted by the root user and the HelloApp application was deployed. Then the servers were restarted as the normal non-root user and this problem occurred. The non-root user was not able to write to the files that were created when the server was running as root. The problem is resolved by resetting the file permissions.

Incorrect file permissions on other files in the configuration repository and the wstemp directories can cause the administrative console to show incorrect results or error messages. Ensure your file permissions are consistent across the file system.

### 4.3.12 Problem: Invalid or expired certificates

If your problems seem to be specific to applications that require SSL access, it is a good probability that the problem is due to invalid or expired certificates. This can cause a variety of system management problems. Problems with expired certificates could occur at any time if you are not managing your certificates. You could also run into this problem if you have recently enabled security.

#### Data to collect

The following logs can be helpful in determining if you have problems with certificates:

- ▶ WebSphere Application Server logs for the process running the administrative console.
- ▶ Management script logs if appropriate.

You might also need to take an Admin trace and a security trace, depending on the problem. See “Collecting data” on page 122 for details.

#### What to look for

WebSphere Application Server writes a warning message (Example 4-30) into the SystemOut.log before a certificate is due to expire. This gives you the opportunity to renew the certificates before the actual expiration dates and before applications stop working. However, at present, WebSphere Application Server V6 does not notify you that a certificate has expired.

*Example 4-30 Certificate expiry warning messages*

---

```
7/6/05 11:37:15:860 EDT] 0000000a SASRas      W JSAS0456W: WARNING in sasOutboundSSLConfig:
The certificate with alias expires 07/07 from keyStore
C:\IBM\WAS6\AppServer/profiles/StandAlone/etc/ExpiresKeyFile.jks will be expired in 1 days.
```

---

Tracing the application server with the following tracestring shows you if the problem is with certificates, as shown in Example 4-31:

```
traceString=ORBRas=all
```

*Example 4-31 Trace messages showing an expired certificate*

---

```
[7/7/05 10:23:17:553 EDT] 00000020 ORBRas      1 com.ibm.rmi.transport.ListenerThread run:259
LT=3:P=897141:O=0:port=9402 The following exception was logged
javax.net.ssl.SSLException: No available certificate corresponds to the SSL cipher suites which
are enabled.
```

---

If you see messages that indicate a certificate has expired or has a problem, you should verify the certificate using the IBM Key Management tool (ikeyman).

In the event that you cannot access the administrative console, you can determine what certificates are being used by looking in the security.xml file. For a single application server, you can find the file at:

```
<WAS_install_root>/profiles/<profile>/config/cells/<cell>/security.xml
```

In a Network Deployment installation, you can find this file at:

```
<WAS_install_root>/profiles/<DMprofile>/config/cells/<cell>/security.xml
```

In security.xml, look for the `<setting>` XML tag as shown in Example 4-32.

*Example 4-32 Key database files defined in security.xml*

---

```
<setting xmi:id="SecureSocketLayer_1"
keyFileName="${USER_INSTALL_ROOT}/etc/DummyServerKeyFile.jks"
keyFilePassword="{xor}CDo9Hgw=" keyFileFormat="JKS"
trustFileName="${USER_INSTALL_ROOT}/etc/DummyServerTrustFile.jks"
trustFilePassword="{xor}CDo9Hgw=" trustFileFormat="JKS"
clientAuthentication="false" securityLevel="HIGH"
enableCryptoHardwareSupport="false">
```

---

This tag gives you the name of the key file databases that you need to check, for example DummyServerKeyFile.jks. Start the WebSphere Application Server key management utility and open the file. Refer to the following item in WebSphere Information Center for information about starting the key management utility (iKeyman):

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec\\_keytu.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec_keytu.html)

You will need to know the password on the key database file. If you have not replaced the supplied dummy key file databases, the default password is WebAS.

Figure 4-13 shows the DummyServerKeyFile key database open in the ikeyman utility. Click on the certificate name, websphere dummy server in this example, and then click **View/Edit**. If the certificate has expired, the utility displays a message box telling you this. Then, it displays the certificate details.

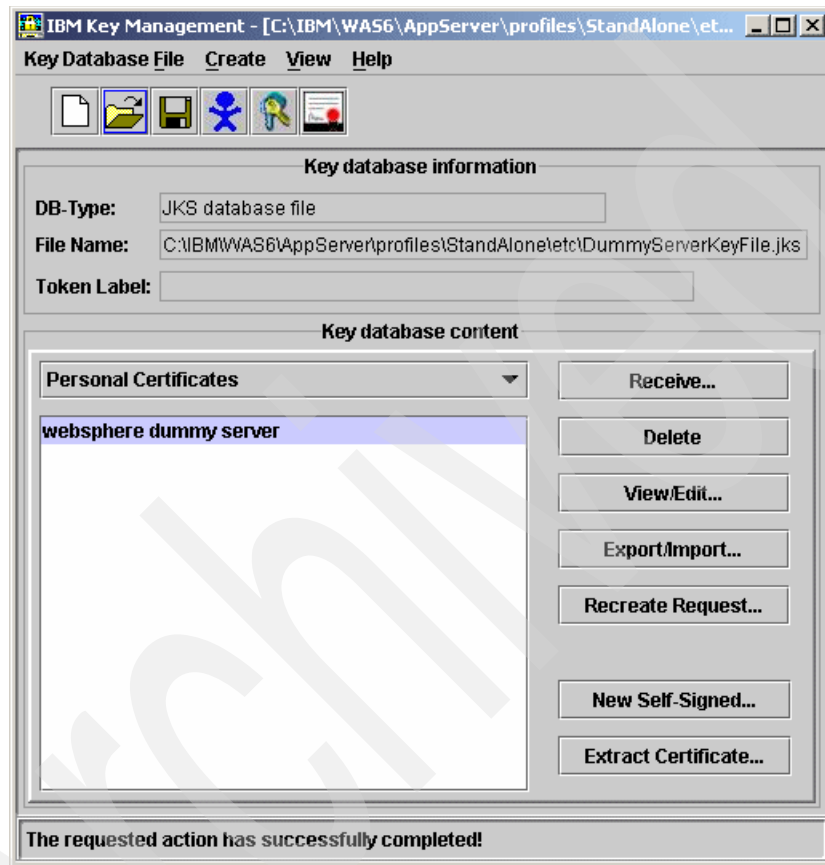


Figure 4-13 Open key database in the ikeyman utility

The Validity field also shows if the certificate has expired. In Figure 4-14 on page 156, the certificate is only valid up to July 7, 2005. Click **OK** to close the window.



Figure 4-14 Expired certificate details

### **Action: Request a new key from a certificate authority**

If the certificate has expired, you can resolve this problem by creating a new valid certificate in the existing key database file.

When a client program such as a browser is setting up an SSL connection with a server, the client can trust the contents of a certificate that is verified by a trusted third party. A certificate authority acts as a trusted third party. The key management utility allows you to request a certificate from a certificate authority. The certificate authority will charge you for this service.

### **Action: Create a self-signed certificate (not for production)**

Alternatively, you can create a self-signed certificate in the dummy key file database. This is a certificate that can be used to set up an SSL connection if the client chooses to accept the certificate even though it has not been verified by a certificate authority. In a production environment, you should not use self-signed certificates for securing SSL connections.

To create a self signed certificate, click **New Self Signed** as shown in Figure 4-13 on page 155.

**Create New Self-Signed Certificate**

Please provide the following:

Key Label	new websphere dummy server
Version	X509 V3 ▼
Key Size	1024 ▼
Common Name	hostname
Organization	IBM
Organization Unit (optional)	
Locality (optional)	
State/Province (optional)	
Zipcode (optional)	
Country or region	US ▼
Validity Period	365 Days

OK Reset Cancel

Figure 4-15 Creating a new self signed certificate

Enter the information for the self signed certificate. The maximum validity period you can set is 7300 days or approximately 20 years.

Alternatively, you could create a whole new set of key database files. WebSphere Application Server comes with a dummy key database file with the password WebAS. This password is the same for all organizations using WebSphere who have not replaced the dummy key database files and is a potential vulnerability in a production environment. For details on creating your own key database files, refer to the WebSphere Information Center section *Preparing truststore files* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec\\_preparetruststorefile.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec_preparetruststorefile.html)

After you have created your new certificates, restart WebSphere Application Server and retest.

**Where to go from here:** If you have exhausted all the possibilities in this section and are still having problems with SSL and certificates, refer to “The next step” on page 162.

### 4.3.13 Problem: WebSphere Rapid Deployment

You are trying to use WebSphere Rapid Deployment to develop and test an application, and you are not able to connect to an application server or WebSphere Rapid Deployment will not create or update the applications.

#### Data to collect

The following logs and output can be helpful in determining why you cannot manage WebSphere Application Server services:

- ▶ WebSphere Rapid Deployment console output
- ▶ WebSphere Rapid Deployment log file
- ▶ WebSphere Application Server logs

#### What to look for

If you have problems with WebSphere Rapid Deployment, you can take the following actions:

- ▶ Verify the WORKSPACE variable
- ▶ Verify the WebSphere Rapid Deployment configuration
- ▶ Verify the application is being built
- ▶ Verify the deployment manager and application server are running
- ▶ Look for Java coding errors

#### ***Verify the WORKSPACE variable***

You tell WebSphere Rapid Deployment what directory to monitor by setting the WORKSPACE environment variable. What directory you use is up to you. Ensure that you are putting the files that you want updated in the location that is specified by the WORKSPACE environment variable. For example:

```
WORKSPACE=C:\IBM\WRD
```

If you specify the directory incorrectly in the WORKSPACE variable, WebSphere Rapid Deployment starts but it does not build or deploy your applications because it is monitoring the wrong directory.

#### ***Verify the WebSphere Rapid Deployment configuration***

Ensure that you have run **wrld-config** to set up your environment for the WebSphere Rapid Deployment project that you are working on. You can check your project settings by looking in the project configuration file at:

```
<WRD_home>/<project>/wrldconfig.xml
```

Example 4-33 shows the contents of this file.

#### Example 4-33 WebSphere Rapid Deployment settings

---

```
<?xml version="1.0" encoding="UTF-8" ?>
<com.ibm.ws.rd.headlessmodel:WRDProjectConfiguration
xmlns:com.ibm.ws.rd.headlessmodel="http://com.ibm/ws/rd/headlessmodel.ecore"
projectName="Hello" styleID="freeform">
  <styleAttributes name="serverName" value="server1" />
  <styleAttributes name="serverJMXHost" value="m23vnx60" />
  <styleAttributes name="serverJMXPort" value="8880" />
  <styleAttributes name="earExportPath" value="c:\temp" />
</com.ibm.ws.rd.headlessmodel:WRDProjectConfiguration>
```

---

In a Network Deployment environment, the `serverName` should be the logical name of the application server that will run the application. The `serverJMXHost` and `serverJMXPort` should refer to the host name and SOAP port of the deployment manager.

The actual program that monitors the directory and does the deployment is `wrd`. Make sure that you have entered the `wrd` command with the `-monitor` parameter to start monitoring the directory and act on the application changes you introduce to it. For example:

```
wrd -monitor
```

#### **Verify the application is being built**

You can check the WebSphere Rapid Deployment log (Example 4-34) to ensure the environment has been initialized and that changes to the workspace are being recognized. The messages also shows if the changes are being handled properly.

#### Example 4-34 WebSphere Rapid Deployment log output showing successful config

---

```
!SESSION Jul 10, 2005 03:37:32.810
-----
eclipse.buildId=unknown
java.fullversion=J2RE 1.4.2 IBM build cxia32142sr1w-20041028 (JIT enabled:
jitc)
BootLoader constants: OS=unknown, ARCH=x86, WS=unknown, NL=en_US
Command-line arguments: -nouupdate -application
com.ibm.ws.rapiddeploy.core.WRDExec -config -project HelloApp -style FreeForm

!ENTRY org.eclipse.core.runtime Jul 10, 2005 03:37:32.816
!MESSAGE Product org.eclipse.platform.ide could not be found.

!ENTRY com.ibm.ws.rapiddeploy.core 0 0 Jul 10, 2005 03:40:49.16
!MESSAGE Creating a new project 'HelloApp'.

!ENTRY org.eclipse.core.resources 1 1 Jul 10, 2005 03:40:51.936
```

!MESSAGE Could not load library: libcore\_2\_1\_0b.so. This library provides platform-specific optimizations for certain file system operations. This library is not present on all platforms, so this may not be an error. The resources plug-in will safely fall back to using java.io.File functionality.

!ENTRY com.ibm.ws.rapiddeploy.core 1 1 Jul 10, 2005 03:40:51.994  
!MESSAGE Recording active directory ID to the workspace root.

!ENTRY com.ibm.ws.rapiddeploy.core 0 0 Jul 10, 2005 03:40:52.85  
!MESSAGE Configuring style '**WebSphere Free Form Project**' for project '**HelloApp**'.

!ENTRY com.ibm.ws.rapiddeploy.core 0 0 Jul 10, 2005 03:40:52.87  
!MESSAGE Recording the current style and project properties...

!ENTRY com.ibm.ws.rapiddeploy.core 0 0 Jul 10, 2005 03:40:52.782  
!MESSAGE Configuring build output location to : '/HelloApp/bin'

!ENTRY com.ibm.ws.rapiddeploy.core 0 0 Jul 10, 2005 03:40:52.861  
!MESSAGE Configuring source folder to '/HelloApp'

!ENTRY com.ibm.ws.rapiddeploy.core 0 0 Jul 10, 2005 03:40:52.863  
!MESSAGE Configuring source folder to '/HelloApp/gen/src'

!ENTRY com.ibm.ws.rapiddeploy.core 0 0 Jul 10, 2005 03:40:53.110  
!MESSAGE Configuring new build output location to '/HelloApp/bin'

---

### ***Verify the deployment manager and application server are running***

If WebSphere Application Server is not available when you start WebSphere Rapid Deployment, you see the following message in the WebSphere Rapid Deployment console:

ERROR! Failed to make connection to WebSphere Application Server

If the process that controls the deployment fails or is shutdown after you start WebSphere Rapid Deployment, the application will not be published. In a single server environment, the process that deploys the application is the application server. You see a message in the console as shown in Example 4-35 on page 160.

#### ***Example 4-35 WebSphere Rapid Deployment unable to publish application***

---

Error occurred during download. : publishrecord.remote  
[01:23:40 PM] Exporting Ear File.  
[01:23:41 PM] ERROR! [SOAPException: faultCode=SOAP-ENV:Client; msg=Error opening socket: java.net.ConnectException: Connection refused: connect;



```
targetException=java.lang.IllegalArgumentException: Error opening socket:  
java.net.ConnectException: Connection refused: connect]
```

---

In a Network Deployment environment, you publish the application to the deployment manager. However, the application is run on a separate application server. If the deployment manager is not running, you see the message that is shown in Example 4-35 in the console. If the target application server is not running, the application publishes but does not start. You see a message in the console as shown in Example 4-36.

*Example 4-36 WebSphere Rapid Deployment unable to start application*

---

```
[12:08:01 PM] ERROR! Application Failed to Start. HelloApp  
[12:08:01 PM] ERROR! MBeanServer cannot find MBean with ObjectName  
WebSphere:platform=dynamicproxy,cell=m23vnx60Cell01,version=6.0.1.2,name=Applic  
ationManager,mbeanIdentifier=ApplicationManager,type=ApplicationManager,node=m2  
3vnx60Node01,process=server1  
[12:08:01 PM] ERROR! Please see server logs for more details.
```

---

After you have restarted the process, you need to make a change to one of the application files so that WebSphere Rapid Deployment detects the update and publishes the application again.

**Note:** Using WebSphere Rapid Deployment causes save conflicts in the administrative console if you or another administrator are logged in. This is another reason why you should use WebSphere Rapid Deployment only for development.

### ***Look for Java coding errors***

If your application code has errors in it, WebSphere Rapid Deployment shows the coding error in the console, for example:

```
[01:28:05 PM] 'Syntax error, insert ";" to complete Statement' in resource  
'HelloServlet.java' on line number 37
```

Java coding errors are outside the scope of this book.

**Where to go from here:** If you have checked all of the problems described thus far and yet still cannot build and deploy an application with WebSphere Rapid Deployment, check the logs for the deployment manager if appropriate and the application server for any messages that are related to the application deployment.

If you have exhausted all the possibilities that are described in this section and are still not able to use WebSphere Rapid Deployment, refer to the next section.

## 4.4 The next step

The symptoms and problem areas included in this paper are some that you are more likely to experience. However, there are other things that can go wrong, or the cause of the problem might be related to something other than system management components.

If, after going through the process described here, you still have an undiagnosed problem, it is recommended that you go back to *Approach to Problem Determination in WebSphere Application Server V6* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

Review the problem classifications to see if there are any other components that might be causing the problem.

If you feel sure you have a system management related problem, there are things you can do before contacting IBM support. First, you should review the documentation you have gathered for errors that were not addressed in this paper and search support sites for information or fixes.

Next, you should collect all of the data that is outlined in the appropriate MustGather documents as follows and raise a problem record with IBM:

- ▶ *MustGather: System management functionality for V5.0, V5.1 and V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21199596>
- ▶ *MustGather: Application Server, dmgr and nodeagent Start and Stop Problems for all Releases of V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21204943>
- ▶ *MustGather: Synchronization problems in V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196219>

- ▶ *MustGather: Profile Creation/Removal Issues for V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196228>
- ▶ *MustGather: Federation or Removal of a Node Issues for Version V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196227>
- ▶ *MustGather: Port Management for V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196226>
- ▶ *MustGather: Node agent and Deployment Manager discovery problems for all releases and editions of V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21196220>
- ▶ *MustGather: Usage and creation of templates fail on V6.0*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21195439>



## Web container problem determination

If users receive unexpected results in a Web browser (such as errors or incorrect information), you might have a problem with the Web components in the application. The runtime environment for Web components is called the *Web container*. This paper discusses techniques for diagnosing problems that can occur in the Web container. Potential symptoms of a Web container problem can include:

- ▶ Users are not able to access a Web component
- ▶ Unexpected behavior when running a Web component
- ▶ Errors when starting a Web component
- ▶ Problems with JSP compilation
- ▶ Errors or exceptions thrown when running a Web component
- ▶ Messages starting with SRVE (Web container), JSPG (JSPs), or JSFG (JSF)

Problems areas that are related to the Web container also include session management problems and character encoding problems.

**Important:** We recommend that you start your problem determination process by reading *Approach to Problem Determination in WebSphere Application Server V6* at <http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>.

## 5.1 Introduction

The Web container is a runtime environment for Web applications. It processes servlets, JSP files, and other types of server-side includes. The Web container also provides session management, static content processing and an inbound transport chain for HTTP requests. Each application server runtime has one logical Web container, which can be modified but not created or removed.

Figure 5-1 illustrates the Web container and its place within the application server.

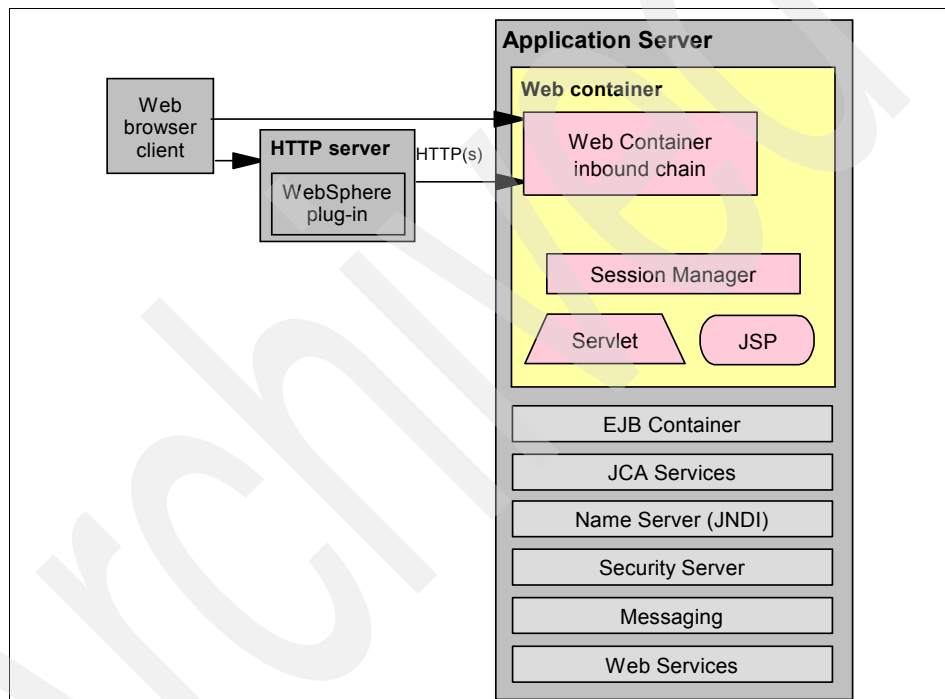


Figure 5-1 Web container overview

Each Web container provides the following:

- Web container transport chains

Requests are directed to the Web container using the Web container inbound transport chain. The chain consists of a TCP inbound channel that provides the connection to the network, an HTTP inbound channel that serves HTTP 1.0 and 1.1 requests, and a Web container channel over which requests for servlets and JSPs are sent to the Web container for processing.

- ▶ Servlet processing

When handling servlets, the Web container creates a request object and a response object and then invokes the servlet service method. The Web container invokes the servlet's destroy method when appropriate and unloads the servlet, after which the JVM performs garbage collection.

- ▶ HTML and other static content processing

Requests for HTML and other static content that are directed to the Web container are served by the Web container inbound chain. However, in most cases, using an external Web server and Web server plug-in as a front-end to a Web container is more appropriate for a production environment.

- ▶ Session management

Support is provided for the `javax.servlet.http.HttpSession` interface as described in the Servlet API specification.

## Web applications

Servlets and JavaServer™ Pages™ (JSP) files are referred to as Web components. Static content files (such as HTML pages, image files, and XML files) are bundled with Web components during application assembly to create a Web module. A Web module is the single deployable and usable unit of Web resources and has a specific structure or archived format known as a *Web archive* (WAR) file. A J2EE Web module corresponds to a Web application as defined in the Java Servlet specification.

The top-level directory of a Web module is the *document root* of the application. The document root is where JSP pages and static Web resources are stored. The document root contains a subdirectory named `/WEB-INF/`, which contains the Web application deployment descriptor (`web.xml`) and the server-side classes used by the module.

Figure 5-2 on page 168 illustrates the directory structure of a Web application.

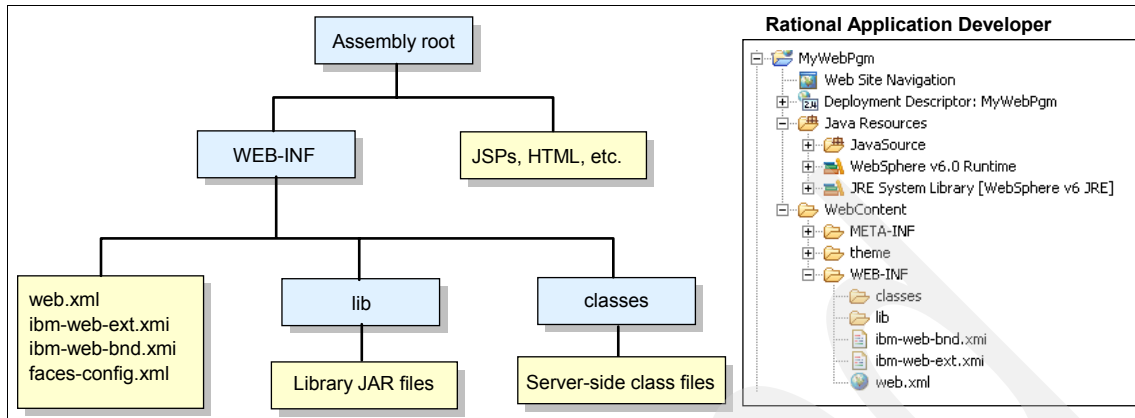


Figure 5-2 Web application components

### 5.1.1 Initial symptoms

A problem that occurs in the Web container usually causes unexpected results to be displayed in the Web browser. Three common indications are addressed here:

- ▶ HTTP 404 errors
- ▶ HTTP 500 errors
- ▶ Incorrect information

**Note:** If the application has an error page configured in the Web module to be delivered when HTTP errors occur, the user will see this page instead of an HTTP error page. In this case, the true error might not be apparent until the Web server log files are examined.

Figure 5-3 on page 169 shows a flow chart of the high-level symptoms and the potential problem areas that might apply to each.



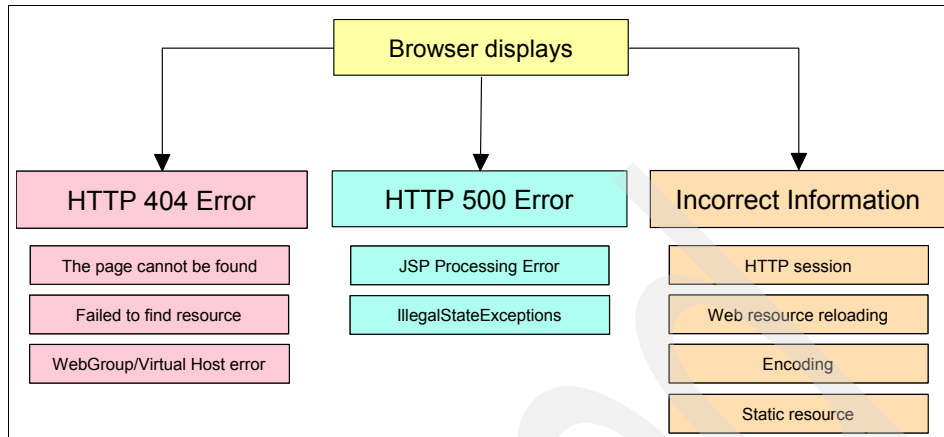


Figure 5-3 Initial symptoms for Web container problems

## 5.2 Work the problem

You begin the problem determination process by collecting the appropriate data that is required to diagnose the problem. We give you a list of all the documentation that might be required and how to collect it. If you have limited ability to recreate the problem, you might want to collect all the documentation at once, before starting the problem determination process.

Next, you go through a series of questions and actions that will help you define the high-level symptoms that you are experiencing. Each of these steps leads you to a more detailed procedure that takes you through the process of collecting and analyzing data to determine the most likely source of the problem.

And lastly, we provide guidance on the next step to take for resolution, whether it be a support site, contacting IBM, information about configuration, or some other suggestion as to how to proceed.

### 5.2.1 Data to collect

The following data will help you with the problem determination process in a Web container execution environment:

- ▶ WebSphere Application Server JVM logs: SystemOut and SystemErr files
- ▶ WebSphere Application Server process logs: native\_stderr.log and native\_stdout.log log files
- ▶ Web server log files

For information about collecting the JVM and Process logs, see *WebSphere Application Server V6: Diagnostic Data* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4085.pdf>

The Web server log files names and locations are specific to the product (IBM HTTP Server, Apache, SunOne, IIS, and Domino®) that is used for this function.

If the problem is difficult to recreate or disruptive to business operations, see “The next step” on page 209 for a complete list of documentation to collect before continuing. In particular, you should review the MustGather documents for a complete list of documentation that is required by IBM support.

## 5.2.2 High-level symptom analysis

Consider the following error types. If you do not find the description of your problem here, go to “The next step” on page 209.

### What to look for

Many indicators of a Web container problem first appear to a user as an unexpected Web browser page or page contents. The unexpected information can be an error page that results from an HTTP error or a page that has incorrect or incomplete information displayed.

When an HTTP error occurs, the Web browser displays the error page with the error code (in particular, HTTP error codes 404 and 500 are likely). If a custom error page has been configured for the Web module, you need check the Web server logs files to determine the specific HTTP error code.

### HTTP 404 errors

HTTP 404 errors can have different underlying causes. Table 5-1 shows the HTTP 404 errors that we address in this paper.

*Table 5-1 Where to go for information about HTTP 404 errors*

If the errors include	Then go to
Page can't be found or displayed	“Symptom: HTTP 404 error - The page cannot be displayed” on page 172
JSPG0036E: Failed to find resource message	“Symptom: HTTP 404 error - Failed to find resource” on page 175
WebGroup/virtual host not define	“Symptom: HTTP 404 error - WebGroup/virtual host not defined” on page 177

## ***HTTP 500 errors***

HTTP 500 errors can also have different underlying causes, which include the following:

- ▶ If the errors indicate JSP processor errors or incorrect syntax in JSPs, go to “Symptom: HTTP 500 error - JSP processing error” on page 179.
- ▶ If the symptoms include an HTTP 500 error page on the browser and in the SystemOut.log file you see an `IllegalStateException`, go to “Symptom: HTTP 500 error - `IllegalStateException`” on page 181.

## ***Incorrect information displayed on Web pages***

If the users are receiving the correct page on the Web browser but the page contains invalid information, check these symptoms:

- ▶ Users report that pages appear with missing elements.  
If a Web page appears but is missing static resources such as text, images, or file segments, go to “Static resources not displayed” on page 190.
- ▶ Users report seeing an old version of application pages.  
If a JSP file has been modified and deployed to the server but the changes do not appear to the browser interface, you might need to ensure that the application has been enabled for JSP reloading. Go to “Web resources not reloading” on page 192.
- ▶ Users report that pages contain invalid information such as multiple question marks (???) or garbage characters.  
It is possible that the application uses DBCS characters (Japanese, Chinese, Korean languages) or specific characters for other languages that are not included in the default character encoding.  
In these cases, a correct character encoding configuration is necessary to display and process this information without problems. For more details related to encoding configurations, go to “Encoding and internationalization issues” on page 195.
- ▶ Users report problems such as lost data during a session.  
Users repeatedly have to enter information that should be saved during the session, lose shopping cart information, or have other short-term data loss.  
This data loss might be caused by a problem with HTTP sessions. In a clustered environment, the problem could be with the Web server plug-in. However, there are other possible causes, such as configuration or application errors.  
To further analyze this problem, go to “HTTP session management” on page 202.

### 5.2.3 Symptom: HTTP 404 error - The page cannot be displayed

Figure 5-4 shows an overview of the steps that you can take to find the cause of an HTTP 404 error caused by a page cannot be displayed condition.

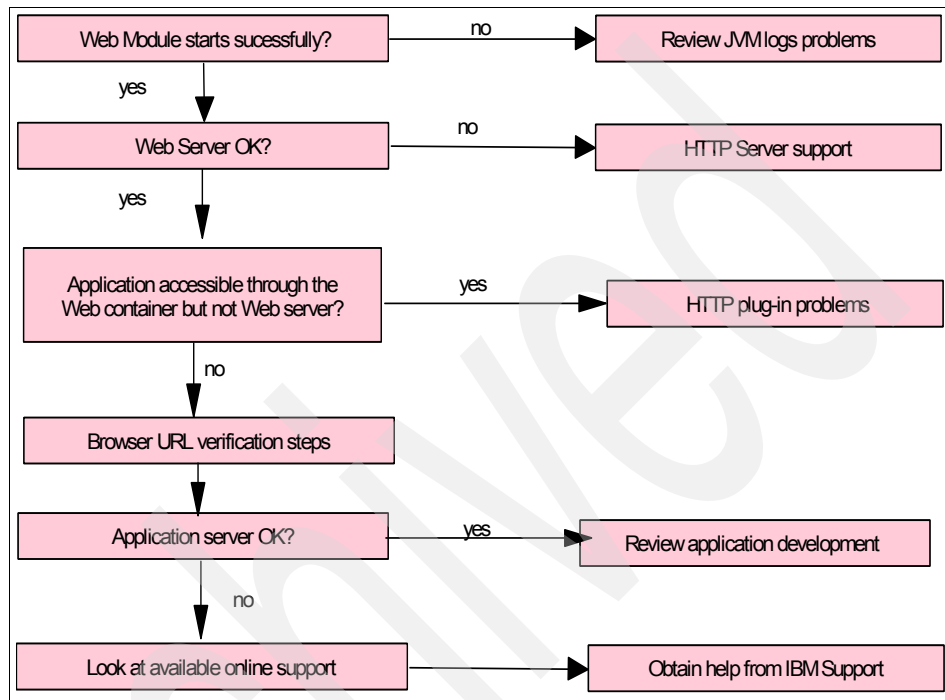


Figure 5-4 Page cannot be displayed verification steps

#### Verify that the Web module has started successfully

Check the JVM logs for the application server, and look for messages in the SystemOut.log file (Example 5-1) to verify that the Web container and Web module started successfully.

Example 5-1 Web Module start process results in SystemOut.log file

```
ApplicationMg A WSVR0200I: Starting application: [application_name]
WebContainer A SRVE0161I: IBM WebSphere Application Server - Web Container.
Copyright IBM Corp. 1998-2004
WebContainer A SRVE0162I: Servlet Specification Level: 2.4
WebContainer A SRVE0163I: Supported JSP Specification Level: 2.0
WebGroup A SRVE0169I: Loading Web Module: [web_module_name]
ApplicationMg A WSVR0221I: Application started: [application_name]
```

One possible reason that would keep the Web container from starting is a problem with the session manager. To determine if this is the case:

- ▶ Look for any errors or exceptions containing a package name of `com.ibm.ws.webcontainer.httpsession`. You will normally find these errors between the starting application message and the application started message.
- ▶ Look in the logs for session manager related messages. These messages will be in the format `SESNxxxxE` for errors and `SESNxxxxW` for warnings. You might also see transaction messages that indicate a problem with session data (see Example 5-2).

---

*Example 5-2 Session manager message error*

---

Error "SRVE0054E: An error occurred while loading session context and Web application."

---

You can look up the extended error definitions in the session manager message table:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/SESN.html>

If you find a session manager error but the explanation is not sufficient to solve the problem, go to "The next step" on page 209.

### **Verify that the Web server is working**

Verify that the Web server is healthy by accessing the URL `http://server_name` from a browser and seeing whether the Welcome page appears. This action indicates whether the Web server is up and running, regardless of the state of WebSphere Application Server.

If the Web server Welcome page does not appear (that is, if you get a browser message such as page cannot be displayed or something similar), the problem is most likely in the Web server. For information about how to approach Web server problems, see *Approach to Problem Determination in WebSphere Application Server V6* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

### **Verify that the Web server plug-in is working**

If you have recently updated or installed the application, ensure that the Web server plug-in was regenerated and propagated to the Web server. Also, ensure that the Web server is using the new plug-in configuration file.

If you have regenerated the plug-in and are sure it is in use but still have a problem, you can bypass the Web server and access the application directly

from the application server. This is not the recommended method of serving a production Web site. However, it provides a good diagnostic tool when it is not clear whether a problem resides in the Web server, WebSphere Application Server, or the Web server plug-in.

**Note:** To access the application directly through the Web container, do the following:

1. Find the port for the Web container:
  - a. In the WebSphere administrative console, select **Servers** → **Application Servers**.
  - b. Select the server name.
  - c. Under the Communications section, expand Ports.
  - d. Use the port number listed for WC\_defaulthost.
2. Use the port number to access the resource from a browser. For example, if the port is 9080, the URL is:

`http://hostname:9080/myAppContext/myJSP.jsp`

If you can access the application via the application server but not through the Web server, you are most likely experiencing a problem with the Web server plug-in. In this case, consult *WebSphere Application Server V6: Web Server Plug-in Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4045.pdf>

### Verify that the URL is correct

If you cannot access the page directly from the application server, verify that the URL being used to access the page is correct. For more information, see “Application URL specification” on page 186.

### Verify that the application server is working

If the URL appears to be correct but you cannot access the resource directly through the application server, verify the health of the hosting application server and Web module by doing the following:

1. View the hosting application server and Web module in the administrative console to verify that they are up and running. In a single server environment, you can check the application server process to see if it is running, or you can use the **serverStatus** command as follows:

```
c:\cd WebSphere\AppServer\profiles\AppSrv01\bin
serverStatus server1
```

2. Copy a simple HTML or JSP file to the Web module document root and try to access it. The Web module document root is located in:

```
<was_install_root>/profiles/<profile>/installedApps/<node>/  
<application>.ear\<web module>.war
```

If successful, the problem is with the resource. View the SystemOut.log file for the application server to discover why the resource cannot be found or served.

### **Look at available online support**

If none of these steps fixes your problem, check to see if the problem has been identified and documented by looking at the available online support (hints and tips, tech notes, and fixes) that are related to Web container problems:

<http://www.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCMPDF>

If the searches fail to find the problem, then go to “The next step” on page 209 for information about gathering the MustGather documentation for HTTP status code 404 “Not Found” problems.

## **5.2.4 Symptom: HTTP 404 error - Failed to find resource**

If the browser displays a JSPG0036E: Failed to find resource message, the JSP processor cannot find the specified JSP page in the Web module, as shown in Figure 5-5 on page 176.

## JSP Processing Error

**HTTP Error Code: 404**

**Error Message:**

JSPG0036E: Failed to find resource /Tests/index.jsp

**Root Cause:**

```
java.io.FileNotFoundException: JSPG0036E: Failed to find resource /Tests/index.jsp
    at com.ibm.ws.jsp.webcontainerext.JSPExtensionProcessor.findWrapper(JSPExtensionProcessor.java:246)
    at com.ibm.ws.jsp.webcontainerext.JSPExtensionProcessor.handleRequest(JSPExtensionProcessor.java:228)
    at com.ibm.ws.webcontainer.webapp.WebApp.handleRequest(WebApp.java:2841)
    at com.ibm.ws.webcontainer.webapp.WebGroup.handleRequest(WebGroup.java:220)
    at com.ibm.ws.webcontainer.VirtualHost.handleRequest(VirtualHost.java:204)
    at com.ibm.ws.webcontainer.WebContainer.handleRequest(WebContainer.java:1681)
    at com.ibm.ws.webcontainer.channel.WCChannelLink.ready(WCChannelLink.java:77)
    at com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.handleDiscrimination(HttpInboundLink.java:421)
    at com.ibm.ws.http.channel.inbound.impl.HttpInboundLink.handleNewInformation(HttpInboundLink.java:367)
    at com.ibm.ws.http.channel.inbound.impl.HttpICLReadCallback.complete(HttpICLReadCallback.java:94)
    at com.ibm.ws.tcp.channel.impl.WorkQueueManager.requestComplete(WorkQueueManager.java:548)
    at com.ibm.ws.tcp.channel.impl.WorkQueueManager.attemptIO(WorkQueueManager.java:601)
    at com.ibm.ws.tcp.channel.impl.WorkQueueManager.workerRun(WorkQueueManager.java:934)
    at com.ibm.ws.tcp.channel.impl.WorkQueueManager$Worker.run(WorkQueueManager.java:1021)
    at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:Compiled Code)
```

Figure 5-5 JSPG0036E: Failed to find resource

### Verify that the URL is correct

Follow the steps described in “Application URL specification” on page 186 to see if the URL being specified is the correct URL.

If the URL is incorrect and is being created as a link from another JSP file, servlet, or HTML file, try correcting it in the browser URL field and reloading the page. This ensures that you know the correct URL before updating the application to use it.

If the URL appears to be correct, it is possible that this page does not exist in the Web module deployment unit (WAR file). Verify that the requested page is located in the Web module directory structure:

```
<WAS_install_root>/profiles/<profile>/installedApps/<node>/
<application>.ear\<web module>.war
```

### Look at available online support

If these steps have failed to identify the problem, see “The next step” on page 209.



## 5.2.5 Symptom: HTTP 404 error - WebGroup/virtual host not defined

The error message SRVE0017W: WebGroup/Virtual Host has not been defined can appear in the SystemOut.log file for many reasons. Check the following to correct the problem.

### Verify that the Web module started

Verify that the Web module has started successfully (see “Verify that the Web module has started successfully” on page 172).

### Verify that the host alias is unique

A duplicate host alias that is defined in multiple virtual hosts can cause this type of problem. For example, in Example 5-3, the host alias test:80 is duplicated in both virtual hosts because a URI that contains test:80 would match aliases in both virtual hosts (\*:80 and test:80).

*Example 5-3 Virtual host definitions*

---

```
<VirtualHostGroup Name="default_host">
  <VirtualHost Name="*:80"/>
  <VirtualHost Name="*:9080"/>
</VirtualHostGroup>

<VirtualHostGroup Name="test_host">
  <VirtualHost Name="test:80"/>
  <VirtualHost Name="*:9081"/>
</VirtualHostGroup>
```

---

To resolve this problem:

1. In the WebSphere administrative console, click **Environment** → **Virtual Hosts**.
2. Select the target virtual host, then click **Host Aliases**, and select the \*:80 host alias.
3. Change the Host Name field to a specific alias instead of \* alias.
4. Restart the application server.

**Note:** The existence of localhost as an alias can cause duplicate entries.

If localhost is used as an alias entry, check the etc/hosts file to ensure that all host names (aliases) associated with the loop back address (127.0.0.1) are part of the same virtual host grouping (for example, default\_host).

## Verify that the URL is correct

See “Application URL specification” on page 186 to determine that the URL is correct.

**Note:** The virtual host matching that is performed by the application server is case sensitive. For example, if your virtual host alias is listed as myserver:80, a request to <http://MYSERVER:80/snoop> results in the following message in the browser:

SRVE0017W: A WebGroup/Virtual Host to handle MYSERVER:80 has not been defined.

## Favicon.ico reference

When accessing the WebSphere administrative console, you could see an error similar to the following:

Servlet Request Processor Exception: Virtual Host/WebGroup Not Found : The web group /favicon.ico has not been defined.

If you receive this error message, refer to Technote # 1193379 at the following Web address for the cause and solution details:

<http://www.ibm.com/support/docview.wss?uid=swg21193379>

## Look at available online support

If none of these steps fixes your problem, check to see if the problem has been identified and documented by looking at the available online support (hints and tips, technotes, and fixes) that are related to Web container problems:

<http://www.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCMPDF>

If your searches fail to find the problem, then go to “The next step” on page 209 for information about gathering the MustGather documentation for servlet engine and Web container problems.

## 5.2.6 Symptom: HTTP 500 error - JSP processing error

Figure 5-6 shows the diagnostic steps to take when the symptoms of the problem include an HTTP 500 - JSP processing error.

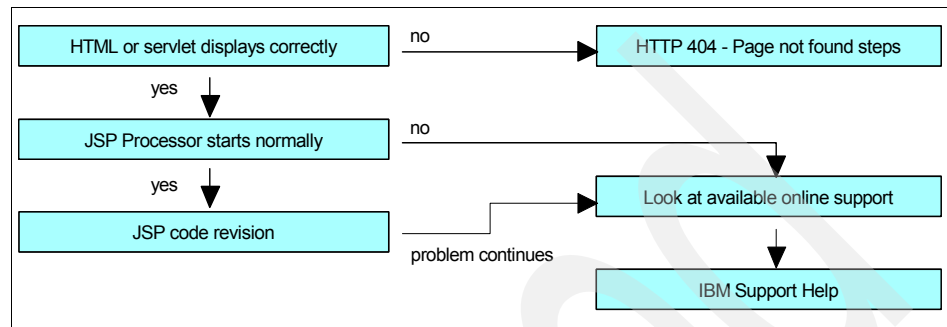


Figure 5-6 JSP processing error

### Determine if the failing file is the only failure

You should determine whether other resource types such as HTML files or servlets are being requested and displayed correctly. If the failure does not appear to be limited to one JSP file, go to “Symptom: HTTP 404 error - The page cannot be displayed” on page 172.

### Ensure that the JSP processor started normally

If other resources are being displayed correctly, determine whether the JSP processor has started normally. Browse the SystemOut.log file of the server that hosts the JSP files that you are trying to access. The messages shown in Example 5-4 indicate that the JSP processor has started normally.

#### Example 5-4 JSP processor messages in SystemOut.log file

```
WebContainer A SRVE0239I: Extension Factory [class  
com.ibm.ws.jsp.webcontainerext.JSPExtensionFactory] was registered  
successfully.  
WebContainer A SRVE0240I: Extension Factory [class  
com.ibm.ws.jsp.webcontainerext.JSPExtensionFactory] has been associated with  
patterns [*.jsp *.jspx *.jsw *.jsv ].
```

If the JSP processor fails to load, go to “The next step” on page 209 for information about gathering the MustGather documentation for JSP exceptions.

## Look for JSP code errors

If the JSP processor starts normally, the problem might be with the JSP file itself. The JSP might have invalid JSP syntax that cannot be processed by the JSP processor. To look for JSP code errors:

- ▶ Examine the SystemOut.log file of the target application server for invalid JSP directive syntax messages. Errors similar to that shown in Example 5-5 indicate this kind of problem.

### *Example 5-5 JSP directive syntax error message*

---

```
Message: /test.jsp(2,1)JSPG0076E: Missing required attribute page for jsp
element jsp:include
```

---

- ▶ Examine the SystemOut.log file for problems with invalid Java syntax. Errors that contain text similar to “failed to compile (shown in Example 5-6) indicate this kind of problem.

### *Example 5-6 Invalid Java syntax error message*

---

```
com.ibm.ws.jsp.JspCoreException: JSPG0049E: /test.jsp failed to compile :
JSPG0091E: An error occurred at line: 16 in the file: /test.jsp
JSPG0093E: Generated servlet error from file: /test.jsp
```

---

- ▶ Look up the extended error definitions for JSP messages at:  
<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/JSPG.html>
- ▶ Correct the error in the JSP file and retry the file.

## Look at available online support

If you have not identified the problem, check to see if the problem has been documented by looking at the available online support (hints and tips, technotes, and fixes) for JSP problems at:

<http://www-1.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCC2GL&rankprofile=8&dc=DB520+D800+D900+DA900+DA800&dtm>

If the searches fail to resolve your problem, go to “The next step” on page 209 for information about gathering the MustGather documentation for JSP exceptions.

## 5.2.7 Symptom: HTTP 500 error - IllegalStateException

If you have an illegal state exception error, the exception should give you an indication of the illegal state causing the problem. This section addresses the following:

- ▶ Invalid session object
- ▶ Response generation problems

### Invalid session object

The session manager component uses the `HttpSession` interface to create a session between an HTTP client and the server. When a new session object is created, a unique session ID is assigned to it. The session ID, which is then passed as part of every request, matches the user with the session object. Session tracking gives servlets the ability to maintain state and user information across multiple requests.

Sessions might get invalidated automatically due to a session timeout (see “Session timeout interval” on page 207) or can be ended explicitly by application code. The `HttpSession` interface provides the following method to terminate a session explicitly:

```
public void invalidate();
```

When a session terminates, the session object and the information that is stored in it are lost permanently. The session manager unbinds any objects bound to the session before it destroys the session.

Figure 5-7 illustrates the `HttpSession` interface life cycle.

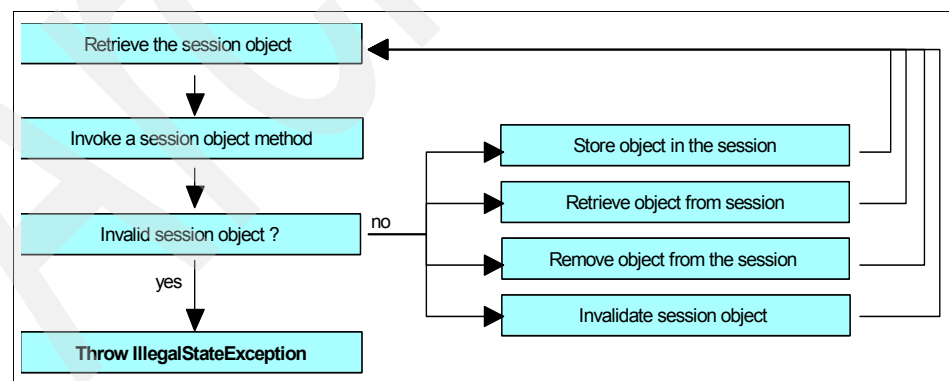


Figure 5-7 *HttpSession* interface life cycle

When the application tries to use an invalidated session object, the `IllegalStateException` occurs, as shown in Example 5-7.

*Example 5-7 `IllegalStateException` in invalid session object*

---

```
[7/7/05 16:41:30:627 ART] 00000028 ServletWrapper E SRVE0068E: Could not
invoke the service() method on servlet TestServlet. Exception thrown :
java.lang.IllegalStateException:
Session Object Internals:
id : pi55X7syi-ExTjyyhFn5Cu7
hashCode : 1449590567
create time : Thu Jul 07 16:24:54 ART 2005
last access : Thu Jul 07 16:41:30 ART 2005
max inactive interval : 1800
user name : anonymous
valid session : false
new session : true
overflowed : false
non-serializable app specific session data : {}
serializable app specific session data : {}
    at
com.ibm.ws.webcontainer.httpsession.SessionData.getValueGuts(SessionData.java(C
omplied Code))
    at
com.ibm.ws.webcontainer.httpsession.SessionData.getValue(SessionData.java(Inlin
ed Compiled Code))
    at
com.ibm.ws.webcontainer.httpsession.SessionData.getAttribute(SessionData.java(I
nlined Compiled Code))
    at
com.ibm.ws.webcontainer.httpsession.HttpSessionFacade.getAttribute(HttpSessionF
acade.java(Compiled Code))
```

---

Check the servlet or JSP code that threw the exception to make sure the session is not being invalidated too early. If that is not the case, check the session timeout interval to ensure it is not too short.

***Look at available online support***

If these steps do not identify the problem, the following resources can be of help:

- ▶ Review the Java Servlet Specification Version 2.4, Section SRV.15.1.7 related to the `HttpSession` interface and methods definitions, to obtain more details for `IllegalStateException` creation causes:

<http://jcp.org/aboutJava/communityprocess/final/jsr154/index.html>

- Look up the extended error definitions for session and user profiles messages:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.messages.doc/doc/SESN.html>

For current information that is available from IBM Support on known problems and their resolution related to session management, use the following URL:

<http://www.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCMPDS&rankprofile=8&dc=DB520+D800+D900+DA900+DA800&dtm>

If these actions do not identify the problem, go to “The next step” on page 209 for information about gathering the MustGather documentation for session and session management problems.

## Response generation problems

When an HTTP request from a client is delegated to a servlet, the `service()` method of the `HttpServlet` class is invoked. The `HttpServlet` class adds additional methods, such as `doGet()`, `doPost()`, `doPut()` and `doHead()`, for HTTP-based request processing. Figure 5-8 illustrates the life cycle of the `service()` method.

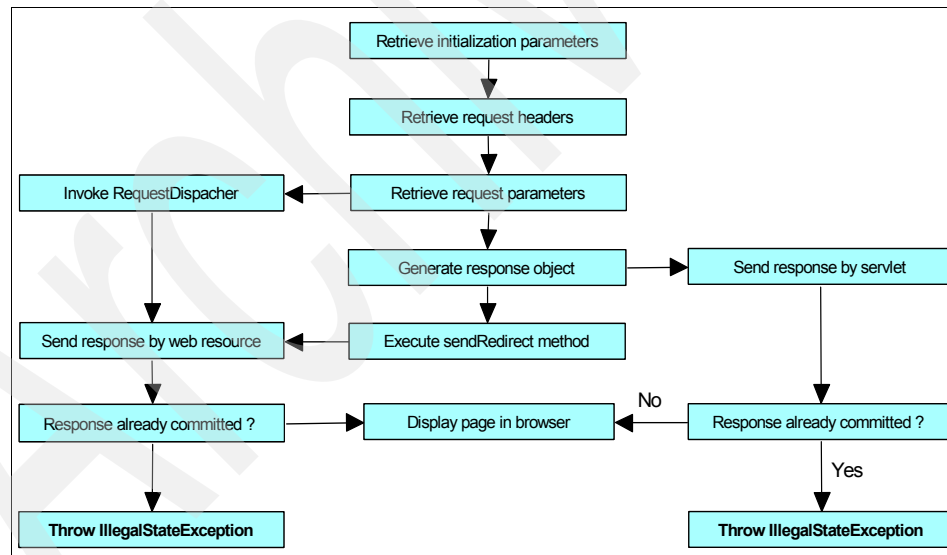


Figure 5-8 Servlet `service()` method life cycle

The `java.lang.IllegalStateException: Response already committed` exception is thrown by the `HttpServletResponse` interface during the response generation process (Example 5-8 on page 184). If the response has been committed, you cannot execute any method that is related to

HttpServletResponse object modification. For example, if you have written something in the response buffer, you cannot forward a page using the RequestDispatcher interface methods.

*Example 5-8 IllegalStateException in response generation*

---

```
[7/8/05 20:36:25:694 ART] 0000004f ServletWrapp E   SRVE0068E: Could not
invoke the service() method on servlet TestServlet. Exception thrown :
java.lang.IllegalStateException
    at
com.ibm.ws.webcontainer.webapp.WebAppDispatcherContext.sendRedirect(WebAppDispa
tcherContext.java:486)
    at
com.ibm.ws.webcontainer.srt.SRTServletResponse.sendRedirect(SRTServletResponse.
java:810)
    at web.TestServlet.doGet(TestServlet.java:56)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:743)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:856)

[7/8/05 20:36:25:764 ART] 0000004f LocalTranCoor E   WLTC0017E: Resources
rolled back due to setRollbackOnly() being called.
[7/8/05 20:36:25:774 ART] 0000004f WebApp      E   SRVE0026E: [Servlet
Error]-[TestServlet]: java.lang.IllegalStateException
    at
com.ibm.ws.webcontainer.webapp.WebAppDispatcherContext.sendRedirect(WebAppDispa
tcherContext.java:486)
    at
com.ibm.ws.webcontainer.srt.SRTServletResponse.sendRedirect(SRTServletResponse.
java:810)
    at web.TestServlet.doGet(TestServlet.java:56)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:743)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:856)

[7/8/05 20:36:25:784 ART] 0000004f SRTServletRes W   WARNING: Cannot set
status. Response already committed.
```

---

Other issues to look for in an application that can cause a `java.lang.IllegalStateException` are the following calls when the response has already been committed:

- ▶ Calling `setBufferSize()`
- ▶ Calling `ServletResponse.reset()` or `ServletResponse.resetBuffer()`
- ▶ Calling either `HttpServletResponse.sendError()` or `HttpServletResponse.sendRedirect()`.
- ▶ Calling `RequestDispatcher.forward()`, which includes performing a `jsp:forward`



**Note:** Remember that if you call `forward()` or `sendRedirect()` in your code, any lines of code following these will still execute.

Two common variants of this exception are:

- ▶ `java.lang.IllegalStateException: Header already sent`  
One or more headers have been committed to the client, so you cannot set that header again.
- ▶ `java.lang.IllegalStateException: Cannot forward as Output Stream or Writer has already been obtained`  
The calling servlet has called `response.getWriter()` or `response.getOutputStream()`. Because the response has been written, it is unsuitable for forwarding.

### ***Look at available online support***

If you still have not identified the cause of the problem, see the Java Servlet Specification Version 2.4 at the following URL to obtain more details for `IllegalStateException` generation causes in response generation process:

<http://jcp.org/aboutJava/communityprocess/final/jsr154/index.html>

Review the following sections:

- ▶ SRV.14.2.5 RequestDispatcher interface
- ▶ SRV.14.2.16 ServletRequest interface
- ▶ SRV.14.2.22 ServletResponse interface

For current information that is available from IBM Support on known issues and resolutions that related to session management, go to the following URL:

<http://www.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCMPDF&rankprofile=8&dc=DB520+D800+D900+DA900+DA800&dtm>

If these steps do not resolve your problem, go to “The next step” on page 209 for information about gathering the MustGather documentation for servlet engine and Web container problems.

# 5.3 Analyzing problem areas

Your analysis of the data you gathered will most likely lead you to one of the following areas. If not, please see “The next step” on page 209.

## 5.3.1 Application URL specification

In a new installation or new application, it is possible that the URL of the application is not being specified correctly. To determine the URL of the installed application, you need to use the administrative console to view the configuration of a number of items.

The format of the URL is as follows:

`http://<host>:<port>/<context_root>/<resource_name>`

### Finding <host>:<port>

When a Web module is installed, a virtual host is associated with it. To find the virtual host using the WebSphere administrative console:

- 1. Select **Applications** → **Enterprise Applications**.
- 2. Select the application name to open the details page.
- 3. In the Additional Properties section, click **Map virtual hosts for Web modules**.
- 4. Note the virtual host(s) that are used by the specified application Web module, as shown in Figure 5-9.

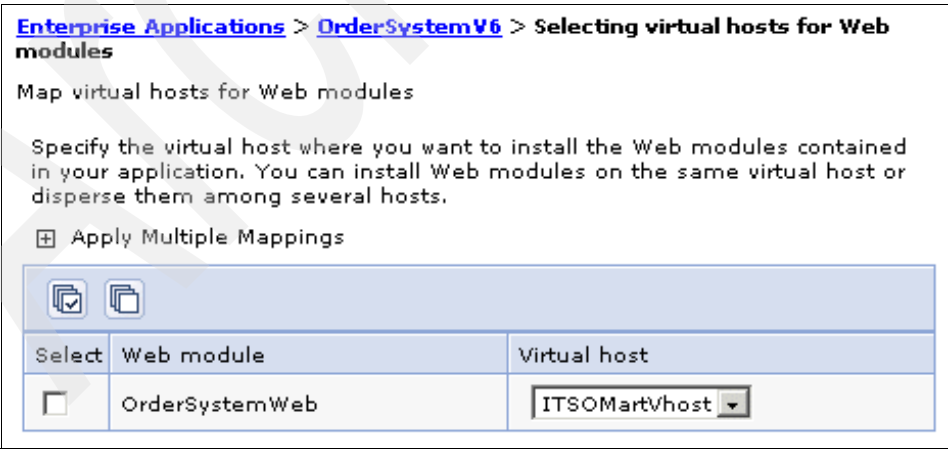


Figure 5-9 Listing the virtual host for a Web module

To find the host and port numbers that are valid for the virtual host:

1. Select **Environment** → **Virtual Hosts**.
2. Choose the virtual host, and then under Additional Properties, click **Host Aliases**.
3. The list contains the host name and port combinations that can be used to access this virtual host (Figure 5-10). The host column should contain values that are registered in a DNS server as a host name for the WebSphere server. An asterisk (\*) in the host column indicates that any name can be used. In this case, use the server host name.

If port 80 is listed, usually the request is being forwarded from a Web server. (The user specified the URL of the Web server, which is normally 80.)

The WC\_defaulthost (for example, 9080) and WC\_defaulthost\_secure (for example, 9443) ports for the application server should also be listed. You can see the corresponding ports by looking at the list of ports in the Communications section of the application server.

Use either the host/port combination that accesses the virtual host through the Web server or the host/port combination for using WC\_\* ports to access the application directly through the application server.

**Virtual Hosts** > **ITSOMartVhost** > **Host Aliases**

A list of one or more DNS aliases by which the virtual host is known.

**Preferences**

☐ ☐ ☐ ☐

Select	Host Name	Port
<input type="checkbox"/>	<a href="http://www.itsomart.com">www.itsomart.com</a>	80
<input type="checkbox"/>	<a href="http://www.itsomart.com">www.itsomart.com</a>	9080
<input type="checkbox"/>	<a href="http://www.itsomart.com">www.itsomart.com</a>	9443

Total 3

Figure 5-10 Finding the alias and ports for the virtual host

If you need SSL, you can check the Web container transport chains for the corresponding ports to see if SSL has been enabled by doing the following:

- a. Select **Servers** → **Application servers**.
- b. Select the application server name.
- c. Under Container Settings, open the Web Container Settings list.
- d. Click **Web container transport chains** and check the status and SSL enablement for the port that is specified in the virtual host alias.

### Finding `<context_root>`

The context root is the Web application root and is used to identify the application in the URI. To find the context root:

1. In the WebSphere administrative console, click **Applications** → **Enterprise Applications**.
2. Select the target application, then under Additional Properties, click **View Deployment Descriptor**.
3. Review the `<context-root>` tag for the specified Web module in the related deployment descriptor tag (Figure 5-11).

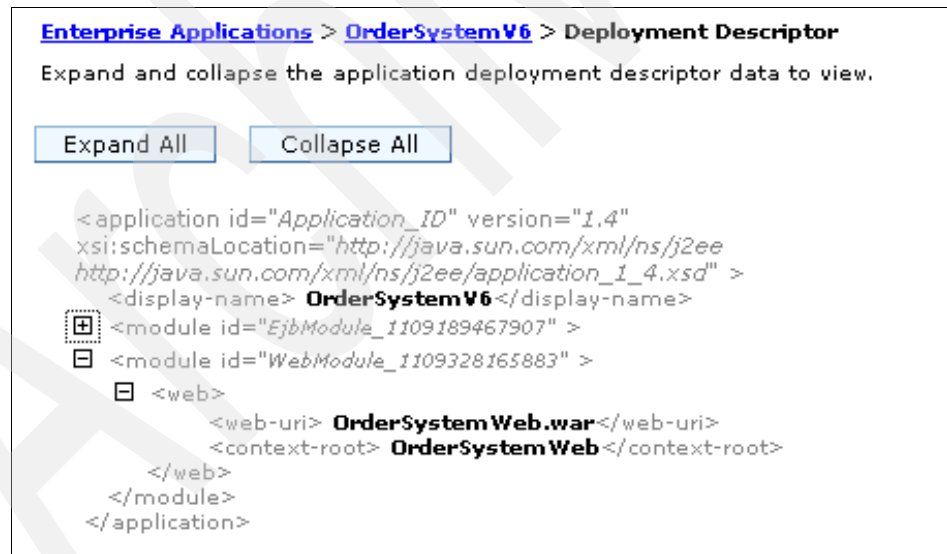


Figure 5-11 Finding the context root

**Important:** Remember that the Web container context root value “/” is used by the sample application named DefaultApplication. For this reason, if you need use this value and this sample application is installed, first uninstall the sample application, install your application, and regenerate the Web server plug-in.

### Finding `<resource_name>`

To find the initial file for the application (for example, index.html), you can look at the welcome file list in the Web deployment descriptor. If you are looking for a particular servlet, the servlet mappings in the Web deployment descriptor tell you the URL pattern to use for each servlet. To find the initial file:

1. In the administrative console, click **Applications** → **Enterprise Applications**.
2. Select the target application, and then under Related Items, click **Web modules**.
3. Choose the specified Web module, and then under Additional Properties, click **View Deployment Descriptor**.
4. Examine the `<servlet-mapping id>` and `<welcome-file-list>` deployment descriptor tags to determine the URL pattern and file name configured respectively for this purpose (Example 5-9).

*Example 5-9 Servlets and welcome files in Web deployment descriptor*

```
<servlet-mapping id="ServletMapping_1" >
  <servlet-name>Main Servlet</servlet-name>
  <url-pattern>*.kht</url-pattern>
</servlet-mapping>
<servlet-mapping id="ServletMapping_2" >
  <servlet-name>Action Servlet</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
<servlet-mapping id="ServletMapping_3" >
  <servlet-name>Process Servlet</servlet-name>
  <url-pattern>/process</url-pattern>
</servlet-mapping>
<welcome-file-list id="WelcomeFileList_1" >
  <welcome-file> index.html</welcome-file>
  <welcome-file> index.htm</welcome-file>
</welcome-file-list>
```

5. If the Web module has security configured, also you need check the `<security-constraint>` and `<security-role>` deployment descriptor tags for the role that is needed for access to the selected Web resource (Example 5-10 on page 190).

```
<security-constraint id="SecurityConstraint_1" >
  <web-resource-collection id="WebResourceCollection_1" >
    <web-resource-name>Main Servlet</web-resource-name>
    <description>Access for Main Servlet.</description>
    <url-pattern>*.kht</url-pattern>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
  </web-resource-collection>
  <auth-constraint id="AuthConstraint_1" >
    <description>Main Servlet Security</description>
    <role-name>All Role</role-name>
  </auth-constraint>
  <user-data-constraint id="UserDataConstraint_1" >
    <transport-guarantee>NONE</transport-guarantee>
  </user-data-constraint>
</security-constraint>
<security-role id="SecurityRole_1" >
  <description>All Authenticated Users Role.</description>
  <role-name>All Role</role-name>
</security-role>
```

---

### 5.3.2 Static resources not displayed

If the browser displays text output that is related to a JSP or servlet Web page but images or HTML files are not displayed, you could have a problem in the Web module packaging or in how the files are referenced within the application. Another possibility is that the file serving enablement feature needs to be turned on for your application.

#### Verify the static resource file locations

The first step is to verify that your files are in the right place and that the document root directory of the Web application module follows the J2EE standard, that is that the document root is in the `<web_module>.war` directory of the deployed application ear file. Typically this directory is in this location:

```
<WAS_install_root>/profiles/<profile>/installedApps/<node>/<application>.ear/
<web_module>.war/
```

If the image files are in a subdirectory of the document root, verify that the reference to the image reflects that, as shown in Example 5-11.

#### Example 5-11 Image reference in HTML tags

---

File Location: `<web_module_name>.war>/images/test.gif`  
Correct HTML tag: `<img SRC="images/test.gif">`  
Incorrect HTML tag: `<img SRC="test.gif">`

---

**Note:** Do not place files to be served to the client in the WEB-INF directory.

## Check the file serving enablement feature

File serving allows a Web application to serve static file types (HTML, images, and style sheets) using the enable file servlet, also known as the file serving servlet or file serving enabler. This servlet serves up any resource file that is packaged in the WAR file, and the file serving enabled attribute is set to **true** by default. For more information about this feature, see:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cweb\\_flserv.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cweb_flserv.html)

This behavior is implemented by setting the `fileServingEnabled` property to `true` when configuring the Web module. If it is set to `false`, the Web server plug-in does not send requests for static content to the application server but leaves it up to the Web server to serve them. This is sometimes done for performance reasons to ensure that the content is served by the Web server. Serving the page from the Web server provides a shorter path to the page and usually provides more customization options than the file servlet can offer.

For more information, see:

- *Customizing SimpleFileServlet* to disable file serving at:

<http://www.ibm.com/support/docview.wss?uid=swg21116838>

- Supported assembly tools in WebSphere Application Server V6:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/catk\\_assemblytools.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/catk_assemblytools.html)

The `fileServingEnabled` property is located in the `ibm-web-ext.xml` configuration file at:

```
<WAS_install_root>/profiles/<profile>/installedApps/<node>/<application>.ear\  
<web_module>.war/WEB-INF/ibm-web-ext.xml
```

To update the property using the Application Server Toolkit or a Rational Developer tool:

1. Go to the J2EE Hierarchy view (Project Explorer in Rational), and select the target Web application module.
2. Double-click the Web deployment descriptor (`web.xml`), and select the Extensions tab to see the IBM Web module extensions (Figure 5-12 on page 192).
3. Under the General section, select or deselect **File serving enabled** to enable or disable the static file serving.

4. Save the Web deployment descriptor file.
5. Redeploy the Web module.
6. Regenerate the Web server plug-in, and propagate it to the Web server.
7. Stop and restart the Web server or allow enough time for the new plug-in configuration to be reloaded.
8. Retry the Web request.

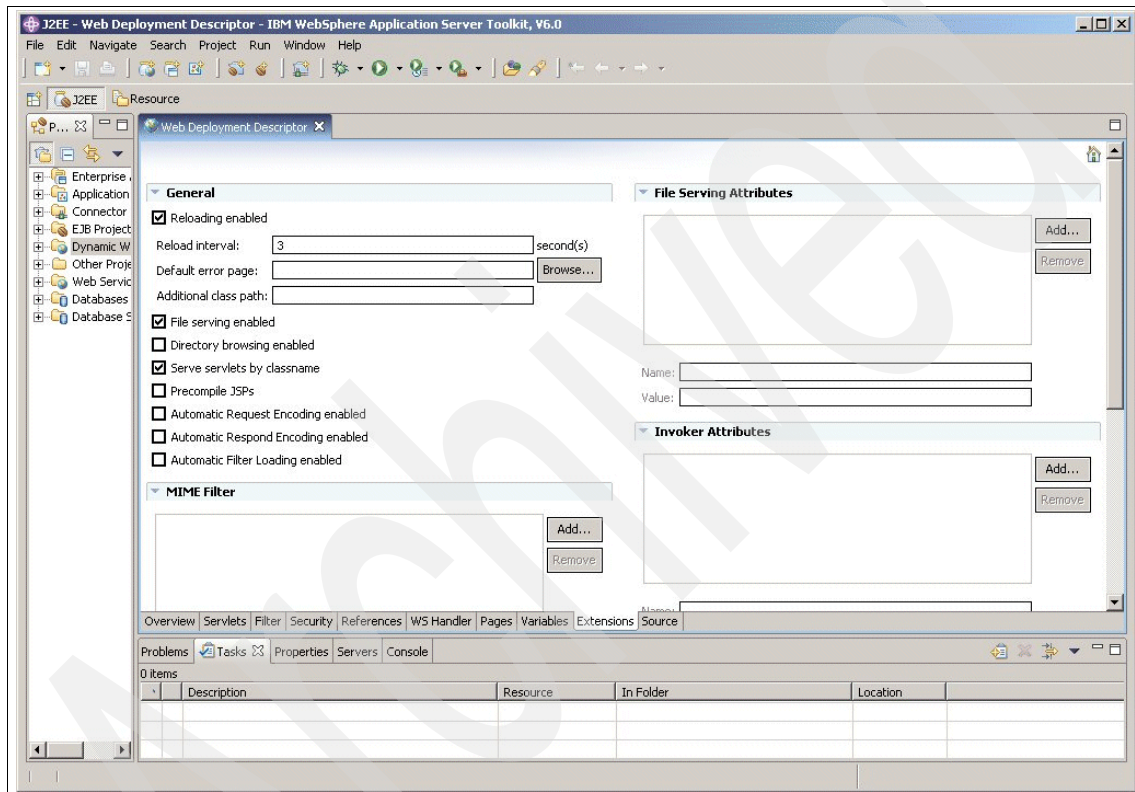


Figure 5-12 IBM Web module extensions

### 5.3.3 Web resources not reloading

If, after modifying and saving a servlet or JSP file, the change does not show up in the browser, you need to check the reload settings in the Web module configuration and the JSP runtime reload settings.



## Web module reloading

For Web resources such as servlets and JSPs, the Web container reloads a Web module only when the IBM extension reloadingEnabled in the ibm-web-ext.xml file is set to true. You can set reloadingEnabled to true when editing the Web module's extended deployment descriptors in an assembly tool.

To review these settings with the Application Server Toolkit or Rational Application Developer:

1. Go to the J2EE Hierarchy view (Project Explorer for Rational) and select the target Web application module.
2. Double-click the Web deployment descriptor (web.xml), and select the Extensions tab to see the IBM Web module extensions (Figure 5-12 on page 192).
3. In the General section, check the Reloading Enabled flag and the Reload Interval value.
4. Select **Reloading Enabled**, or if it is already selected, then set the **Reload Interval** lower.
5. Save the Web deployment descriptor file.
6. Redeploy the Web module.
7. Regenerate the Web server plug-in and propagate it to the Web server.
8. Stop and restart the Web server or wait for the new plug-in file to take effect.
9. Retry the Web request.

The entries in ibm-web-ext.xml look similar to that shown in Example 5-12.

*Example 5-12 Web module reloading settings in ibm-web-ext.xml*

---

```
<webappext:WebAppExtension xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:webappext="webappext.xml" xmi:id="WebAppExtension_1109270589179"
reloadInterval="3" reloadingEnabled="true" fileServingEnabled="true">
```

---

## Configuring JSP runtime reloading

You have the ability to modify the JSP processor behavior for different JSP stages (such as development, testing, or production environments). This done by configuring specific attributes in the IBM Web module extensions that affect the JSP runtime reload behavior.

JSP files can be translated and compiled at runtime when the JSP file or its dependencies are modified. This is known as JSP reloading. JSP reloading is enabled through the reloadEnabled JSP engine parameter in the ibm-web-ext.xml configuration file.

To review these settings with the Application Server Toolkit or Rational Application Developer:

1. Go to the J2EE Hierarchy view (Project Explorer for Rational), and select the target Web application module.
2. Double-click in Web deployment descriptor (web.xml), and select the Extensions tab.
3. Under the JSP Attributes section (Figure 5-13), click **Add**.
4. Enter `reloadEnabled` in the Name field, `true` in the Value field, and click **Finish**.
5. Save the Web deployment descriptor file.
6. Redeploy the Web module.
7. Regenerate the Web server plug-in and verify it is working (see “Verify that the Web server plug-in is working” on page 173).
8. Stop and restart the Web server.
9. Retry the Web request.

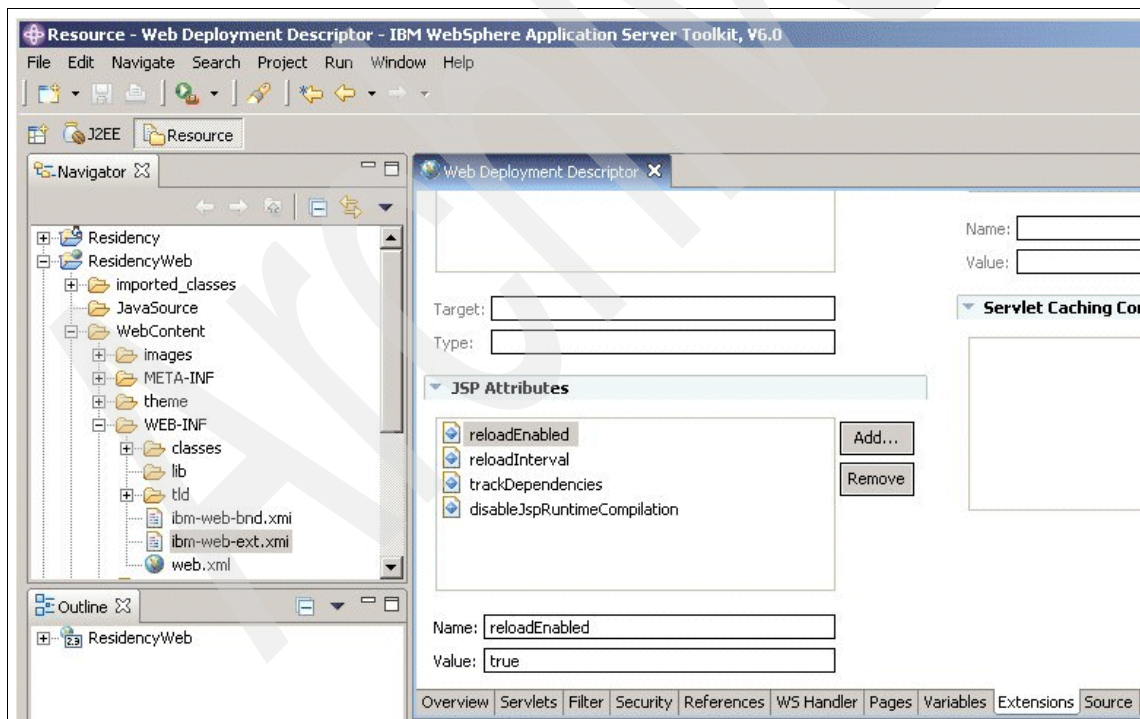


Figure 5-13 JSP attributes in IBM Web module extensions

The JSP engine settings are stored in `ibm-web-ext.xmi`, as shown in Example 5-13.

*Example 5-13 JSP engine settings in `ibm-web-ext.xmi`*

---

```
<jspAttributes xmi:id="JSPAttribute_1" name="reloadEnabled" value="true"/>
<jspAttributes xmi:id="JSPAttribute_9" name="reloadInterval" value="3"/>
```

---

For more available JSP attributes and details that are related to reload processing sequence, see the following article in the WebSphere Information Center:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rweb\\_jspreloading.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rweb_jspreloading.html)

### 5.3.4 Encoding and internationalization issues

Encoding and internationalization problems usually surface as garbage characters in Web pages (Figure 5-14 on page 196). Another common symptom is that input from the user is interpreted incorrectly.

Double Byte Character Set (DBCS) character processing is a common issue for encoding problems. DBCS is used for languages such as Chinese, Korean, and Japanese, where a single byte is not sufficient to represent all characters in the alphabet. Proper coding and configuration usually resolves these problems



## ***Request encoding***

Request encoding is the character encoding in which parameters in an incoming request are interpreted. Currently, many browsers do not send a request encoding qualifier with the content-type HTTP header. In such cases, a Web container uses the default encoding: ISO-8859-1 to parse request data. If the client has not set character encoding and the request data is encoded with a different encoding from the default, the data will not be interpreted correctly.

To correct this situation, you can use the `setCharacterEncoding(String enc)` method to override the character encoding supplied by the container, as shown in Example 5-14.

### ***Example 5-14 setCharacterEncoding() method implementation***

---

```
public class TestServlet extends HttpServlet {

    public void doPost(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException {

        req.setCharacterEncoding("UTF-8");
        String name = req.getParameter("name");
        resp.setContentType("text/html; charset=UTF-8");
        PrintWriter out = resp.getWriter();
        out.println("<html><body><h1>");
        out.println("Your name is "+name);
        out.println("</h1></body></html>");

    }
}
```

---

You must call the method before parsing any request parameters or reading any input from the request. Calling the method or tag after data has been read will not affect the encoding.

## ***Page encoding***

For JSP pages, page encoding is the character encoding in which the file is encoded. For JSP pages in standard syntax, the page encoding is determined from the following sources:

- ▶ The `pageEncoding` attribute of the page directive of the page
- ▶ The `charset` value of the `contentType` attribute of the page directive

If none of these is provided, ISO-8859-1 is used as the default page encoding.

#### Example 5-15 Page directive implementation

---

```
<HTML>
<HEAD>
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="ISO-8859-15" %>

<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
<TITLE>Receiving parameters</TITLE>
</HEAD>
<BODY>
<H1>
Your data is <%= request.getParameter("name") %>
</H1>
</BODY>
</HTML>
```

---

The `pageEncoding` and `contentType` attributes determine the page character encoding of only the file that physically contains the page directive.

#### **Response encoding**

Response encoding is character encoding of the textual response that is generated from a Web component. A Web container sets an initial response encoding for a JSP page from the following sources:

- ▶ The `charset` value of the `contentType` attribute of the page directive
- ▶ The encoding specified by the `pageEncoding` attribute of the page directive

If none of these is provided, ISO-8859-1 is used as the default response encoding.

The `setContentType()` method in a servlet can be called to change the character encoding. Calls made after the `getWriter()` method has been called or after the response is committed have no effect on the character encoding.

#### **MVC/Struts character encoding settings**

When using a Model-View-Controller architecture, including the use of Struts, consider the following for encoding.

### ***Request character encoding***

In a Struts environment, call the `setCharacterEncoding()` method in the `ActionForm`, as shown in Example 5-16.

*Example 5-16 Request encoding in Struts implementation*

---

```
.
public class PostMessageForm extends ActionForm {
    public void reset(ActionMapping mapping, HttpServletRequest request) {
        try {
            request.setCharacterEncoding("UTF-8");
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
    }
}
.
```

---

### ***Response character encoding***

Specify the response encoding using the `contentType` attribute with a `charset` value in the JSP page directive.

### **WebSphere Application Server configuration**

WebSphere determines the character encoding used for request/response by parsing the client input values in `getParameter()` and writing the output per the value in the `accept-language` header of the incoming request.

The language value and corresponding character encoding names are associated in the `encoding.properties` file, which is located in the `<WAS_install_root>/properties` directory (Example 5-17).

*Example 5-17 encoding.properties file*

---

```
.
en=ISO-8859-1
fr=ISO-8859-1
de=ISO-8859-1
es=ISO-8859-1
cs=ISO-8859-2
hr=ISO-8859-2
th=windows-874
vi=windows-1258
ja=Shift_JIS
ko=EUC-KR
zh=GB2312
zh_TW=Big5
hy=UTF-8
.
```

---

There might be cases where you want to override the definition in `encoding.properties`. For example, when you want to use UTF-8 for the entire application server, use the JVM command line argument `client.encoding.override` for the selected application server.

Specify `-Dclient.encoding.override=UTF-8` for the generic JVM arguments field in the Java Virtual Machine section.

For more information, see:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/trun\\_svr\\_utf.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/trun_svr_utf.html)

### ***autoRequestEncoding and autoResponseEncoding***

Starting with WebSphere Application Server Version 5, the Web container no longer sets request and response encodings and response content types automatically. The default value for both extensions is false, then the request and response character encoding is set to the Servlet 2.4 Specification default, which is ISO-8859-1.

Use an assembly tool (Figure 5-12 on page 192) to change the default values for the `autoRequestEncoding` and `autoResponseEncoding` extensions.

Review the `autoRequestEncoding` and `autoResponseEncoding` encoding examples for a description of Web container behavior when these values are set to true:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rweb\\_autoreq.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rweb_autoreq.html)



## Summary

Figure 5-15 shows a summary of how the encoding values are determined.

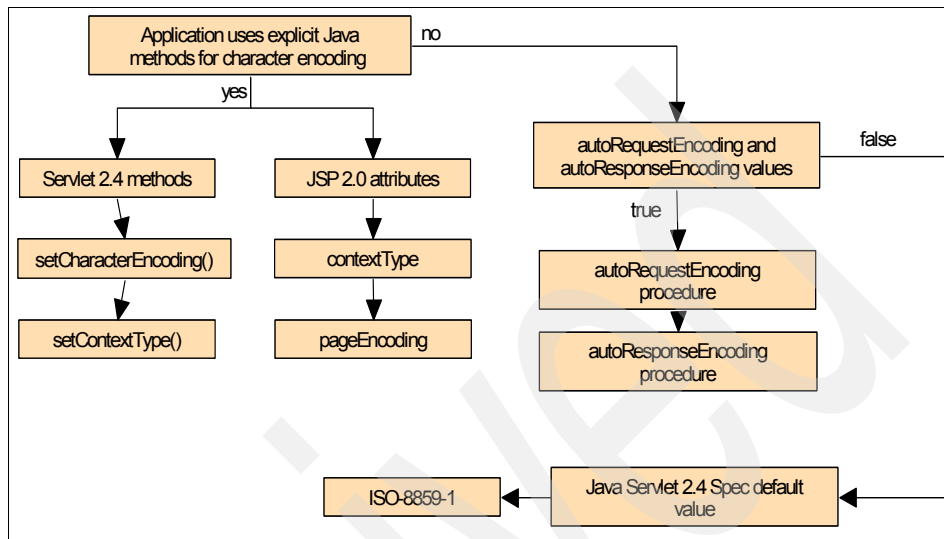


Figure 5-15 Encoding determination process

## Look at available online support

If none of these steps fixes your problem, the following resources might be helpful:

- ▶ *Globalize your On Demand Business* for tips on character encoding:  
<http://www-306.ibm.com/software/globalization/j2ee/encoding.jsp>
- ▶ The Java Servlet Specification Version 2.4 for details about the character encoding available methods:  
<http://jcp.org/aboutJava/communityprocess/final/jsr154/index.html>
- ▶ The JavaServer Pages Specification Version 2.0, section JSP.4 that are related to internationalization issues:  
<http://jcp.org/aboutJava/communityprocess/final/jsr152/index.html>
- ▶ *Internationalization: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rin\\_resources.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rin_resources.html)

For current information available from IBM Support on known issues and resolutions that are related to encoding, see:

<http://www.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCVRZX&rankprofile=8&dc=DB520+D800+D900+DA900+DA800&dtm>

If these steps do not resolve your problem, go to “The next step” on page 209 for information about gathering the MustGather documentation for internationalization problems.

### 5.3.5 HTTP session management

Figure 5-16 outlines the problem determination steps for HTTP session management problems.

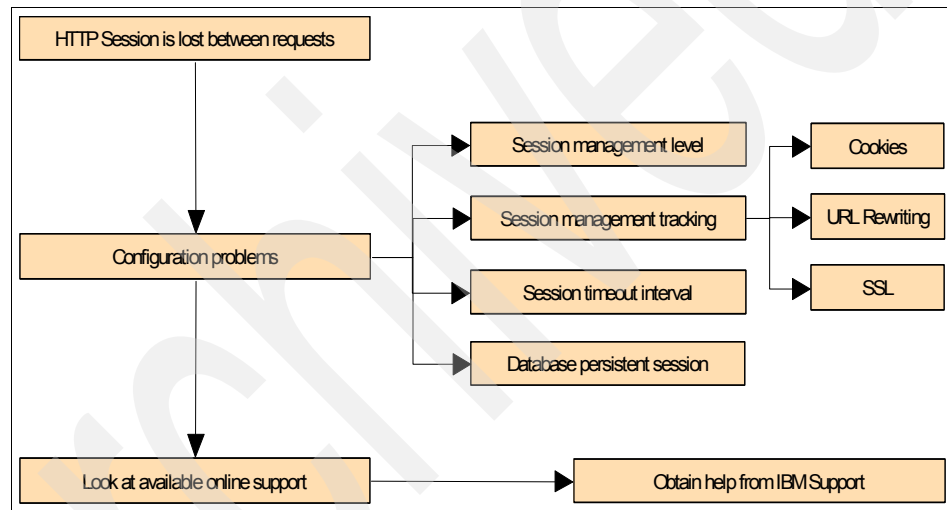


Figure 5-16 HTTP Session problems steps

When session data is lost between requests, the session manager configuration settings are the most likely suspect.

### IBMTrackerDebug:

A special servlet, `com.ibm.ws.webcontainer.httpsession.IBMTrackerDebug`, can be invoked to display the current configuration and statistics related to session tracking. This servlet has all the counters that are in performance monitor tool and has some additional counters.

It can be invoked from any Web module that is enabled to serve servlets by class name using the following URL:

<http://localhost:9080/servlet/com.ibm.ws.webcontainer.httpsession.IBMTrackerDebug>

Note that a module that can be viewed via the `serve servlets by classname` feature can be seen by anyone that can view the application. You might want to map a specific, secured URL to the servlet instead and disable `serve servlets by classname`.

Session management settings can be set at the following levels:

- ▶ Web container
- ▶ Enterprise application
- ▶ Web module

When you configure session management at the Web container level, all applications and the respective Web modules in the Web container normally inherit that configuration, setting up a basic default configuration for the applications and Web modules below it. However, you can set up different configurations individually for specific applications and Web modules that vary from the Web container default. These different configurations override the default for these applications and Web modules only.

**Note:** When you overwrite the default session management settings on the application level, all the Web modules below that application inherit this new setting, unless they too are set to overwrite these settings.

To begin session management configuration, you first need to get to the settings for the selected level.

1. Web container level:
  - a. In the WebSphere administrative console, click **Servers** → **Application servers**.
  - b. Select the application server name.

- c. Under Container Settings, expand the Web Container Settings list, and click **Session management**.
2. Enterprise application level:
  - a. In the WebSphere administrative console, click **Applications** → **Enterprise Applications**.
  - b. Click the application name, and under Additional Properties, click **Session management**.
3. Web module level:
  - a. In the WebSphere administrative console, click **Applications** → **Enterprise Applications**.
  - b. Select the target application, and then under Related Items, click **Web modules**.
  - c. Choose the specified Web module, and then under Additional Properties, click **Session management**.

**Note:** If you are working on the Web module or application level and want these settings to override the inherited Session Management settings, under General Properties, select **Override session management**.

WebSphere Application Server offers the following methods of session manager tracking:

- ▶ Session management tracking with cookies
- ▶ Session management tracking with URL rewriting
- ▶ Session management tracking with SSL

If you are having session management problems, there are specific things to check for each type of session manager tracking. In addition, the following configuration items are common to all three tracking types:

- ▶ Session timeout interval
- ▶ Database persistent session

### **Session management tracking with cookies**

The session manager uses cookies to store the session ID on the client between requests. To check the session management cookie settings, go to the appropriate session management configuration level and ensure that the **Enable cookies** option is set.

To view or change the cookie settings, click the **Enable cookies** link.

To ensure that cookies are flowing between the application server and the browser:

- ▶ Make sure cookies are enabled on the browser you are testing from or from which your users are accessing the application.
- ▶ Check the cookie flow between the browser and server. On the browser, enable cookie prompt. Enter the URL of the servlet and make sure the cookie is being prompted.
- ▶ If you use a Netscape browser, check the following URL for information related to cookie domains that are not recognized with Netscape:

<http://www.ibm.com/support/docview.wss?uid=swg21163880>

Check the cookie domain definition (see Example 5-18).

*Example 5-18 Cookie domain definition*

---

Cookie domain: “.myCom.com”

Resources should be accessed using that domain name:

<http://www.myCom.com/myapp/servlet/sessionServlet>

---

- ▶ Check the cookie path that is specified in the session manager settings. Check whether the problem URL is hierarchically below the cookie path that is specified. If it is not, correct the cookie path.
- ▶ If the cookie maximum age property is set, ensure that the client (browser) machine's date and time is the same as the server's, including the time zone. If the client and the server time difference is over the maximum age, every access would be a new session because the cookie will expire after the access.
- ▶ If you have multiple Web modules within an enterprise application that track sessions:
  - If you want to have different session settings among Web modules in an enterprise application, ensure that each Web module specifies a different cookie name or path.
  - If Web modules within an enterprise application use a common cookie name and path, ensure that the HTTP session settings, such as cookie maximum age, are the same for all Web modules. Otherwise, cookie behavior is unpredictable and depends upon which application creates the session. Note that this does not affect session data, which is maintained separately by the Web module.
- ▶ Configure Internet browsers to present a warning before accepting cookies:

<http://www.ibm.com/support/docview.wss?uid=swg21157880>

## Session management tracking with URL rewriting

Using URL rewriting for session management means that the session management facility uses rewritten URLs to carry the session IDs. If URL rewriting is enabled, the session management facility recognizes session IDs that arrive in the URL if the `encodeURL` method is called in the servlet.

Enabling protocol switch rewriting specifies that the session ID is added to a URL when the URL requires a switch from HTTP to HTTPS or from HTTPS to HTTP. If rewriting is enabled, the session ID is required to go between HTTP and HTTPS.

You can check if URL rewriting is enabled by going to the appropriate session management configuration level and verifying that the `Enable URL rewriting` option is selected. You can also see if `Enable protocol switch rewriting` option is selected.

If you are using URL rewriting:

- ▶ Ensure that there are no static HTML pages on the application's navigation path.
- ▶ Review that the servlet and JSP files are implementing URL rewriting correctly. For details and an example see:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprs\\_sesd.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprs_sesd.html)

- ▶ If the application uses Struts, check this URL to understand the URL rewrite behavior in WebSphere Application Server V6:

<http://www.ibm.com/support/docview.wss?uid=swg21205259>

## Session management tracking with SSL

No special programming is required to track sessions with Secure Sockets Layer (SSL) information. You can check to see if SSL session management is enabled by going to the appropriate session management configuration level and verifying that the `Enable SSL ID tracking` option is selected.

- ▶ The SSL session ID is negotiated between the Web browser and Web server. This ID cannot survive a Web server failure. However, the failure of an application server does not affect the SSL session ID if an external Web server is present between WebSphere Application Server and the browser.
- ▶ SSL tracking is supported for the IBM HTTP Server and iPlanet™ Web servers only.
- ▶ You can control the lifetime of an SSL session ID by configuring options in the Web server. For example, in the IBM HTTP Server, set the configuration variable `SSLV3TIMEOUT` to provide an adequate lifetime for the SSL session

ID. An interval that is too short can cause a premature termination of a session.

## Session timeout interval

The session manager invalidation process thread runs every  $x$  seconds to invalidate any invalid sessions, where  $x$  is determined based on the `session timeout interval` that is specified in the session manager properties. For the default value of 30 minutes,  $x$  is around 300 seconds. In this case, it could take up to 5 minutes (300 seconds) beyond the timeout threshold of 30 minutes for a particular session to become invalidated.

## Database persistent session

If session data is lost when the application server restarts or is not shared across the cluster, you might have a problem with the database used to persist sessions or with its configuration.

To review the database session persistent values using the administrative console, do the following:

1. In the session management settings, under the Additional Properties list, click **Distributed environment settings**.
2. If you are using a database for persistent sessions, the Database option is selected.
3. Click the **Database** link to check the data source definition.
4. Check the JNDI name of the data source.
5. Ensure that the correct user ID and password that are required to access the database are specified.
6. The data source should be non-JTA, that is non-XA enabled.
7. With DB2, for row sizes other than 4 KB, make sure that the specified row size matches the DB2 page size. Make sure the tablespace name is specified correctly.
8. Check the SystemOut.log file for appropriate database error messages.

**Note:** You should check these settings against the properties of an existing data source. The session manager does not create a session database automatically for you.

For more information, see:

- ▶ *Session manager user ID settings for database persistence*  
<http://www.ibm.com/support/docview.wss?uid=swg21203466>
- ▶ *Creating a DB2 table for session persistence - WebSphere V6*  
<http://www.ibm.com/support/docview.wss?uid=swg21199312>

## **Look at available online support**

If none of these steps fixes your problem, the following resources might be helpful:

- ▶ *Managing HTTP sessions: Resources for learning*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprs\\_r4ln.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rprs_r4ln.html)
- ▶ *Best practices for using HTTP Sessions*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/cprs\\_sesm.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/cprs_sesm.html)
- ▶ *Tuning session management*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/cprs\\_sesm.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/cprs_sesm.html)
- ▶ *State replication in the Web tier*  
<http://www.ibm.com/developerworks/java/library/j-jtp07294.html>

For current information available from IBM Support on known issues and resolutions that are related to session management, see:

<http://www.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCMPDS&rankprofile=8&dc=DB520+D800+D900+DA900+DA800&dtm>

If these steps do not resolve your problem, go to the next section for information about gathering the MustGather documentation for session management problems.



## 5.4 The next step

The symptoms and problem areas included in this paper are some that you are more likely to experience. However, there are other things that can go wrong, or the cause of the problem might be related to a component other than the Web container.

If, after going through this process, you still have an undiagnosed problem, it is recommended that you go back to *Approach to Problem Determination in WebSphere Application Server V6* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

Review the problem classifications to see if there are any other components that might be causing the problem.

If you feel sure that you have a Web container related problem, there are things you can do before contacting IBM support. First, you should review the documentation that you have gathered for errors related to the Web container that were not addressed in this paper and search support sites for information or fixes. For hints and tips, technotes, and fixes that are related to Web container problems, see:

<http://www.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCMPDF>

If the problem is performance related, the following references are useful for configuring the JSP engine for optimal performance:

- *JSP engine configuration parameters*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rweb\\_jspengine.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rweb_jspengine.html)

- *JSP class file generation*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cweb\\_jspclassfiles.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cweb_jspclassfiles.html)

- *Disabling JavaServer Pages run-time compilation*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rweb\\_jspdis.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rweb_jspdis.html)

- *Packages and directories for generated .java and .class files*

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cweb\\_javapkg.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cweb_javapkg.html)

If this search does not provide any information relevant to your problem, it might be time to contact IBM support. The MustGather documents for Web container problems are:

- ▶ HTTP status code 404, “NOT FOUND” in WebSphere Application Server V6.0  
<http://www.ibm.com/support/docview.wss?uid=swg21193551>
- ▶ *MustGather: Servlet engine and Web container errors on WebSphere Application Server V6.0*  
<http://www.ibm.com/support/docview.wss?uid=swg21193538>
- ▶ *MustGather: JavaServer Pages JSP exceptions for V6.0*  
<http://www.ibm.com/support/docview.wss?uid=swg21193099>
- ▶ *MustGather: Sessions and session management problems in V6.0*  
<http://www.ibm.com/support/docview.wss?uid=swg21192604>
- ▶ *MustGather: i18n (Internationalization) / Double Byte Character Set (DBCS)*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21141732>

Be sure to note all of the diagnostic work that you have done so far to minimize the amount of time that it takes IBM Support to assist you in resolving your problem.

You can find detailed information about contacting IBM support for issues that are related to WebSphere Application Server V6.0 in *Approach to Problem Determination in WebSphere Application Server V6* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

## JCA connection problem determination

IBM WebSphere Application Server V6 provides implementation and support for the new J2EE Connector Architecture (JCA) specification V1.5 as part of the J2EE 1.4 platform. This paper discusses the most common problem areas that are associated with JCA connections in WebSphere Application Server V6. Issues that are caused by JCA components or JCA connection configuration errors can appear as one or more of the following initial symptoms:

- ▶ A JDBC call returns incorrect data to the application
- ▶ An application cannot connect to or access a database or EIS
- ▶ WebSphere error messages with the prefixes DSRA, WSCL, J2CA, WTRN, CONM, or SQLException or with database error codes.

**Important:** We recommend that you start your problem determination process by reading *Approach to Problem Determination in WebSphere Application Server V6* at <http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>.

## 6.1 Introduction

The JCA specification provides a standard mechanism that allows modern J2EE applications to connect and use heterogeneous resources from various existing enterprise information systems (EIS) as well as modern relational database systems. Based on the JCA, WebSphere provides client applications all system services regarding connection, transaction, and security management on behalf of the resource managers.

JCA involves four main components:

- ▶ Application server
- ▶ Application component
- ▶ Resource adapter
- ▶ EIS

It also specifies a requirement for packaging and deployment facilities for a resource adapter to plug into an application server. Figure 6-1 illustrates the contracts or relationships between these four components.

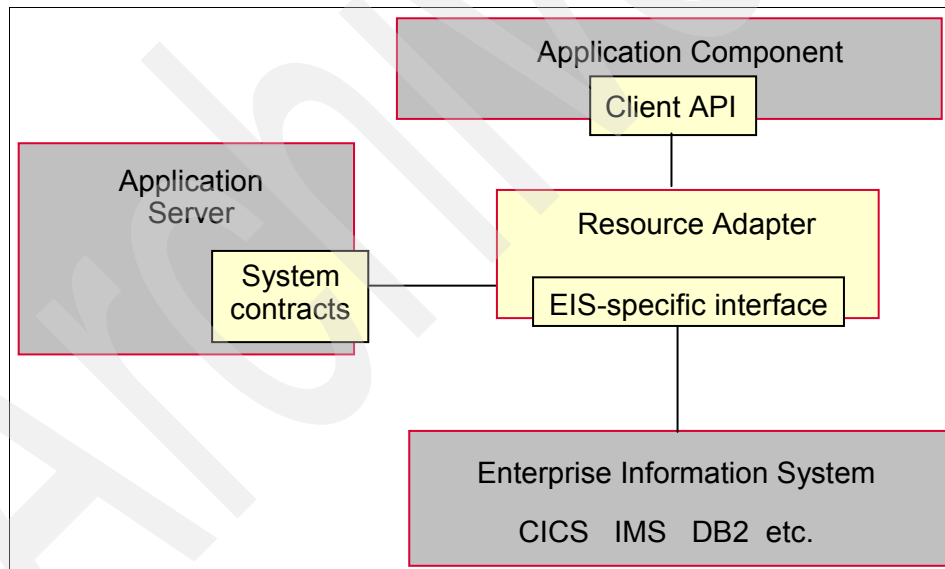


Figure 6-1 JCA overview

Based on the JCA specification, an EIS vendor can develop a standard resource adapter (RA) for its EIS to plug into any application sever that supports JCA. A resource adapter runs within the address space of an application server, while the EIS itself runs in a separate address space. For example, a DB2 database

(EIS) and WebSphere Application Server each run in a separate machine. An application component is able to access the EIS through the resource adapter.

The relationships between the four major components can be described as follows:

- ▶ The contracts between the application component and the resource adapter are provided through some form of client API. The client API can either be specific to a particular type of resource adapter, for example JDBC for relational data base, or a standard common client interface (CCI). The JCA recommends, but does not require, that a resource adapter implement the CCI. WebSphere Application Server V6 provides a relational resource adapter (RRA) that has an implementation for both the CCI and the traditional JDBC interfaces.
- ▶ The resource adapter and application server implement system contracts to provide the common mechanisms for connection, transaction, and security management.
- ▶ The contracts between the resource adapter and the EIS are specific to each underlying EIS. Thus, JCA does not impose any requirement on this proprietary relationship. For example, an RRA for a relational database accesses the resources through a JDBC driver that is supported by the database, or a resource adapter for an SAP R/3 systems accesses the business objects and functions through SAP's Business Application Programming Interface and Remote Function Call (BAPI/RFC).

### 6.1.1 JCA technical overview

The JCA allows applications to access the EIS in both managed and non-managed environments.

In a managed environment, the application components run within the address space of the application server. For example, an application component can be a servlet, or it can be an EJB that runs within a Web container or an EJB container of the WebSphere Application Server. The servlet or the EJB accesses the EIS through a resource adapter that is plugged into WebSphere Application Server. Figure 6-2 on page 214 shows this architecture.

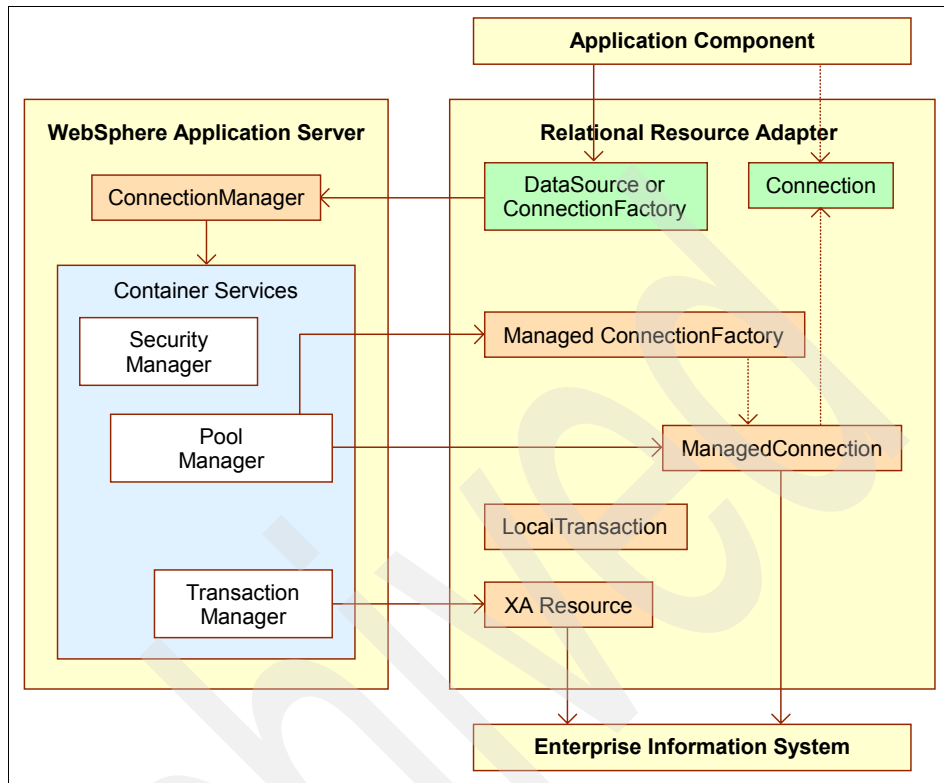


Figure 6-2 Managed application scenario

The application component acquires a ConnectionFactory through JNDI. The ConnectionFactory is used to get a Connection that provides access to the EIS. The ConnectionFactory and Connection interfaces are part of the CCI (from Java package javax.resource.cci) that are implemented by the resource adapter. In addition to the CCI, the WebSphere RRA provides implementation for the existing JDBC interfaces javax.sql.DataSource and javax.sql.Connection to allow existing applications to fit in the new JCA structure without changing application client code.

The remaining interfaces that are discussed in this section are specified in javax.resource.spi with the implementation that is provided by either WebSphere Application Server or the resource adapter.

The ConnectionFactory delegates the connection allocation request to the ConnectionManager (implemented by WebSphere). Through the ConnectionManager, WebSphere can provide applications with many system services such as security, transaction, pooling etc. The ConnectionManager

looks up the connection pool maintained by WebSphere to see if a free `ManagedConnection` (implemented by the resource adapter) is available. If there is none, the `ManagedConnectionFactory` (implemented by the resource adapter) will be asked to create a new `ManagedConnection` to add to the pool. In either case, WebSphere Application Server receives a `ManagedConnection` to create a `Connection` as its application-level handle to return to client. The returned `Connection` is just a logical handle that represents the physical connection within the underlying EIS.

The resource adapter also provides implementation for the `XAResource` interface to support transactions that involve multiple resource managers and the `LocalTransaction` interface to support transactions internal to the resource manager.

In a non-managed environment, application clients run outside the address space of the application server. The application programming model for the client code is the same as in a managed scenario. However, applications use the resource adapter library directly. The `ConnectionFactory` still delegates the task of allocating a `Connection` to the `ConnectionManager`, but it involves the resource adapter's default implementation for the `ConnectionManager` instead of the application server's.

This paper addresses problems that are experienced during connection to enterprise information systems or databases using JCA.

Users with the following initial symptoms might be experiencing a JCA-related problem:

- ▶ Symptom: A JDBC call returns incorrect data
- ▶ Symptom: Failure to connect to a new data source
- ▶ Symptom: Failure to connect to an existing data source
- ▶ Symptom: Failure to access a resource through JDBC
- ▶ Symptom: Failure to access a non-relational resource

Such symptoms can be observed in the WebSphere Application Server JVM logs in error messages with any of the following prefixes: WTRN, J2CA, WSCL, or DSRA.

This paper helps you find the cause of the problem if the cause is in one of the following areas:

- ▶ Application
- ▶ Configuration and tuning
- ▶ WebSphere components and resource adapters that implement the JCA system contracts

## 6.2 Work the problem

If you are experiencing problems during the execution of an application or task that impact the operation, the initial symptoms you encounter can help you determine what software components might be involved. An initial symptom is just an event that allows you to immediately identify a problem in a general and broad sense without getting into much detailed analysis. Thus, our problem determination process begins with a list of initial symptoms that can identify a JCA connection problem. The initial symptoms serve as the starting point for the subsequent analysis.

In the analysis of each symptom, this problem determination (PD) process guides you through the following tasks:

- ▶ Collecting diagnostic files
- ▶ Looking for specific warnings, errors, or other status information that is produced by the system
- ▶ Evaluating the error messages to see if they are related to your problem
- ▶ Assessing and applying a resolution as appropriate. Finding the root cause of the problem is the goal of the problem determination process, but suggesting a solution is not. However, during the analysis, we occasionally suggest solutions when they naturally follow from the analysis.

The problem determination process puts emphasis on actions or tasks. Performing a task produces a result, causes an event to happen, or gives a symptom for you to see what was happening. The outcome, event, or symptom in turn leads to another action or task. This process repeats until a root cause for the problem is found.

In the flow diagrams in this paper, a rectangular box with rounded corners represents an action that you perform and an oval-shaped element represents the detailed analysis of a problem area. (This detailed analysis is simply a series of actions that you need to perform for the problem area.) The transition between the elements in the flow diagram is qualified and labeled with the symptom (or outcome or event) that you receive. The diamond-shaped element is a branch that leads to more than one possible transition. When you have identified the problem area, you can proceed to the corresponding subsection of “Analyzing problem areas” on page 227 to continue the process.



Figure 6-3 shows the initial symptoms that you might experience when you have a JCA connection problem.

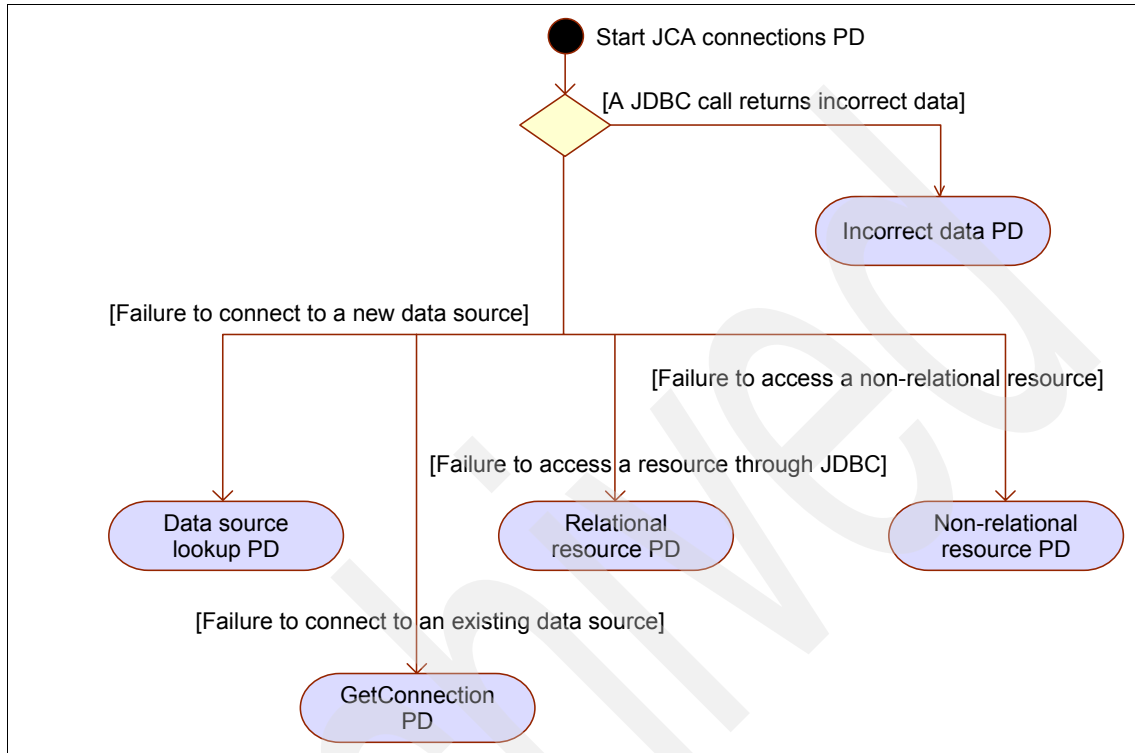


Figure 6-3 JCA connections - initial symptom analysis

### 6.2.1 Symptom: A JDBC call returns incorrect data

With this symptom, your application is running normally, but you receive incorrect data. For example, your database maintains a list of product inventory, but some product quantities or prices that are reported from your application are not in line with your expectation.

#### Rule out a database or JDBC driver problem

If you suspect data is being incorrectly returned to the application, try running the same query using the database's native client tools (such as Oracle SQL\*Plus, Sybase isql, Informix® dbaccess, DB2 Command Line Processor, or SQL Server Query Analyzer). Alternatively, try to run the same query in a stand-alone JDBC program using the JDBC driver directly. If the data returned from the client tool is the same as the data that your application returned when running in WebSphere

Application Server, then it is likely that the database has a problem. In this case, you need to engage the database vendor for further help.

If the data in the database is correct, try to write a stand-alone JDBC program that runs the same query to see if the same symptom occurs. If the data returned from the stand-alone JDBC program is the same as the data that your application returned, then problem is with the JDBC driver. In this case, you need to contact the JDBC vendor for further help.

**Tips:** To create a stand-alone JDBC program to run outside of WebSphere Application Server, you can:

- ▶ Execute `<WAS_install_root>/bin/setupCmdLine.bat` on Windows or `<WAS_install_root>/bin/setupCmdLine.sh` on UNIX to set up the `JAVA_HOME` environment variable.
- ▶ Set the `CLASSPATH` environment variable to include `<WAS_install_root>/lib/j2ee.jar` and the jar file(s) for the JDBC driver. (See the examples for the JAR file names that are associated with the JDBC drivers included in this paper).
- ▶ If you are using a type 2 JDBC driver, then you might also need to set other required environment variables. For example, if you are using Oracle OCI driver, you must set `ORACLE_HOME` and `LIBPATH`.
- ▶ Your JDBC program must use the same data source implementation class as the one used by WebSphere connection manager.

The following examples show how to connect to the database for various platforms.

#### *Example 6-1 Oracle JDBC Thin*

---

```
import java.sql.*;
import javax.sql.*;
import oracle.jdbc.driver.*;
import oracle.jdbc.pool.OracleConnectionPoolDataSource;
.....
OracleConnectionPoolDataSource ds = new OracleConnectionPoolDataSource();
ds.setUser("scott");
ds.setPassword("tiger");
ds.setDriverType("thin");/
ds.setURL("jdbc:oracle:thin:@localhost:1521:swanlake");
Connection con = ods.getConnection();
```

---

#### *Example 6-2 Oracle OCI*

---

```
import java.sql.*;
import javax.sql.*;
import oracle.jdbc.pool.OracleConnectionPoolDataSource;
...
OracleConnectionPoolDataSource ds = new OracleConnectionPoolDataSource();
ds.setUser("scott");
ds.setPassword("tiger");
ds.setDriverType("oci");
ds.setURL("jdbc:oracle:oci8@tnsnames");
// tnsnames is the connect alias defined in your Oracle's network
// client file $ORACLE_HOME/network/admin/tnsnames.ora
Connection con = ds.getConnection();
```

---

#### *Example 6-3 DB2 Legacy CLI*

---

```
import java.sql.*;
import javax.sql.*;
import COM.ibm.db2.jdbc.*;
...
DB2ConnectionPoolDataSource ds = new DB2ConnectionPoolDataSource();
ds.setDatabaseName("sample");
ds.setUser("db2admin");
ds.setPassword("db2admin");
PooledConnection pCon = ds.getPooledConnection();
Connection con = pCon.getConnection();
```

---

#### *Example 6-4 DB2 Universal JDBC (JCC)*

---

```
import java.sql.*;
import javax.sql.*;
import com.ibm.db2.jcc.*;
...
DB2ConnectionPoolDataSource ds = new DB2ConnectionPoolDataSource();
ds.setUser("db2admin");
ds.setPassword("db2admin");
ds.setDatabaseName("sample");
ds.setServerName("9.27.40.128");
ds.setPortNumber(50000);
ds.setDriverType(4);
PooledConnection pCon = ds.getPooledConnection();
Connection con = pCon.getConnection();
```

---

#### *Example 6-5 Informix JDBC*

---

```
import java.sql.*;
import javax.sql.*;
import com.informix.jdbcx.*;
...
IfxConnectionPoolDataSource ds = new IfxConnectionPoolDataSource();
ds.setIfxIFXHOST("perfdobo.rchland.ibm.com");
ds.setPortNumber(1526);
ds.setUser("tgpham");
ds.setPassword("sn0wlce");
ds.setServerName("ol_perfdobo");
ds.setDatabaseName("tgpham");
PooledConnection pCon = ds.getPooledConnection();
Connection con = pCon.getConnection();
```

---

#### *Example 6-6 Sybase jConnect*

---

```
import java.sql.*;
import javax.sql.*;
import com.sybase.jdbc2.jdbc.*;
...
SybConnectionPoolDataSource ds = new SybConnectionPoolDataSource();
ds.setDatabaseName("test");
ds.setUser("test01");
ds.setPassword("test01");
ds.setServerName("mySybaseServerHostname");
ds.setPortNumber(4100);
PooledConnection pCon = ds.getPooledConnection();
Connection con = pCon.getConnection();
```

---

#### *Example 6-7 SQL Server*

---

```
import java.sql.*;
import javax.sql.*;
// For WebSphere embedded Connect JDBC
import com.ibm.websphere.jdbcx.sqlserver.*;
// For DataDirect Connect JDBC
// import com.ddtek.jdbcx.sqlserver.*;
// For SequeLink JDBC
// import com.ddtek.jdbcx.*;
SQLServerDataSource ds = new SQLServerDataSource();
ds.setDatabaseName("sample");
ds.setServerName("localhost");
ds.setPortNumber(1433);
ds.setUser("dbuser1");
ds.setPassword("dbpwd1");
ds.setSelectMethod("cursor");
Connection con = ds.getConnection();
```

---

## What to look for

If you have determined that the problem is not caused by the database or JDBC driver, you can continue the analysis by looking for the situation in which the error occurs. The best way to do this is to perform a number of runs through the application, both from a fresh restart and repeated runs within the same application session.

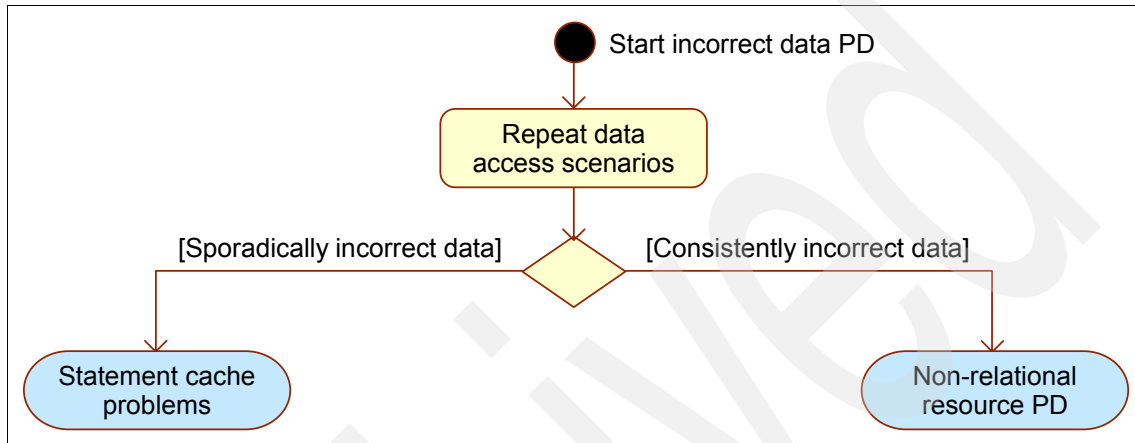


Figure 6-4 Incorrect data problem determination

As shown in Figure 6-4, you might be in one of the following two situations:

- Sporadically incorrect data, which means that the data that is returned to you is correct sometimes but not always.

If your application uses a prepared statement or callable statement, you can experience a problem with WebSphere's statement cache.

A prepared statement is a precompiled SQL statement that is stored in a prepared statement object for parameterized queries. This object is used to run the given SQL statement efficiently multiple times. A callable statement is used to invoke stored procedures.

In general, the more prepared statements and callable statements that your application has, the larger the cache should be. Be aware, however, that specifying a larger statement cache size than needed wastes application memory and does not improve performance.

Determine the value for your cache size by adding the number of uniquely prepared statements and callable statements (as determined by the SQL string, concurrency, and the scroll type) for each application that uses this data source on a particular server. This value is the maximum number of possible prepared statements and callable statements that are cached on a given connection over the life of the server.

You can try to disable the statement cache by setting Statement cache size to 0. This setting can be found in the administrative console by selecting **Resources** → **JDBC Providers** → *<JDBC\_provider>* → **Data sources** → *<data\_source>* → **WebSphere Application Server connection properties**.

You should call IBM Technical Support to help you with this scenario.

- ▶ Consistently incorrect data is when the data that is returned to you is always incorrect. In this case, it is highly possible that your application is connected to the wrong database or resource manager. In this case, you might have a problem with your environment or configuration. Proceed to “Configuration problems” on page 228 for further analysis.

## 6.2.2 Symptom: Failure to connect to a new data source

When you start a new working session in your application, the application first needs to look for a data source that provides the data. Any error at this early stage can be broadly identified as a failure on new data source.

### Data to collect

All relevant error messages and exceptions needed for our analysis are available in SystemOut.log and SystemErr.log.

### What to look for

To proceed for further analysis, you need to run a tool called TestConnection service. WebSphere Application Server provides a test connection service for testing connections to the data sources that you configure for database access. This test connection service can be activated in three different ways:

- ▶ Through the administrative console
- ▶ Using the wsadmin tool
- ▶ With a Java stand-alone program

For more information, see *Test connection service* in the WebSphere Information Center at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.base.doc/info/aes/ae/cdat\\_testcon.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.base.doc/info/aes/ae/cdat_testcon.html)

Depending on the messages or exceptions produced by TestConnection, you might need to perform the additional actions as shown in Figure 6-5.

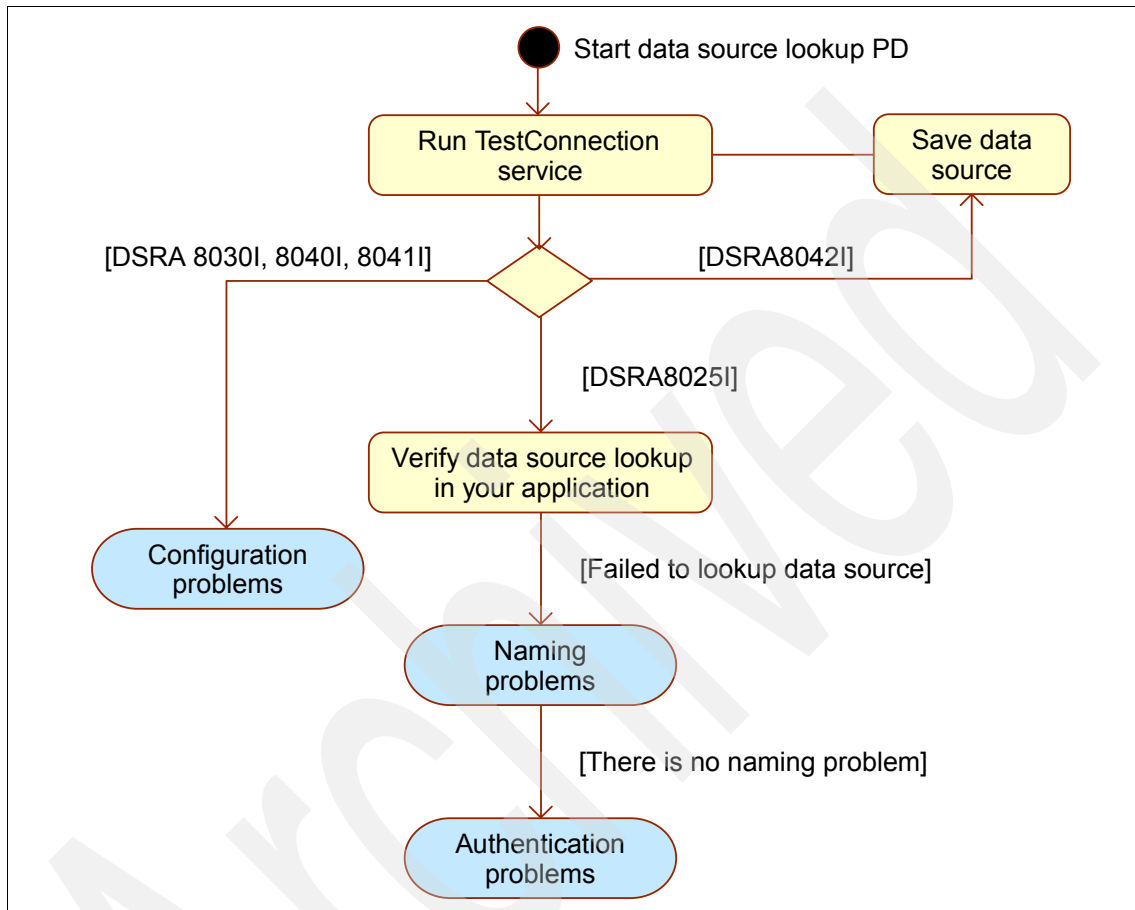


Figure 6-5 Data source lookup problem determination

Testing the connection results in one of the following outcomes:

► Message DSRA8042I

This message indicates that the data source does not exist.

If you created the data source but have not saved the configuration, the test connection will not find the data source. You need to save the data source and then re-run the TestConnection service.

- Messages DSRA8030I, DSRA8040I, or DSRA8041I

These messages indicate that your connection is either failing or having problems. In this case, the TestConnection service produces additional DSRA messages in the range of 8000 to 8499 in the SystemOut.log.

Proceed to “Configuration problems” on page 228 for further analysis.

- Message DSRA8025I

This message indicates that the TestConnection service can connect to the data source successfully. There is nothing wrong with the data source configuration. So, the next step is to verify the data source lookup code in your application. One possible way to perform this verification is to run the application through a debugger with a breakpoint set after the lookup() method call on your java.naming.Context object. Another way is to create and run a small program with the data source lookup code cut and pasted from your application.

If the data source lookup is successful, the new data source is OK, and you should go to “The next step” on page 249.

If the data source lookup is not successful, you might have a naming or authentication problem. To determine if you have a naming problem, see “Naming problems” on page 234.

If there is no problem with naming, see “Authentication problems” on page 235 to determine if you have a problem authenticating with the database.

If you reach this point without identifying the problem, go to “The next step” on page 249.

### 6.2.3 Symptom: Failure to connect to an existing data source

In the situation where a current session with the application has been working or a new session is started with no errors on naming or authentication, you can assume that the application was successful finding the data source (or connection factory). A failure to establish connectivity at this stage is broadly identified as a failure to get a new connection to an existing data source.

#### Data to collect

All relevant error messages and exceptions that are needed for our analysis are available in SystemOut.log and SystemErr.log.

#### What to look for

You need to evaluate the situation when there is failure on getting a connection. One possible way to do this is to run the application through a debugger with a



breakpoint set after the `getConnection()` method call on your `javax.resource.cci.ConnectionFactory` or `javax.sql.DataSource` object. Another way is to create and run a small program with the getting connection code cut and pasted from your application. You can run this test case many times and observe the outcomes as shown in Figure 6-6.

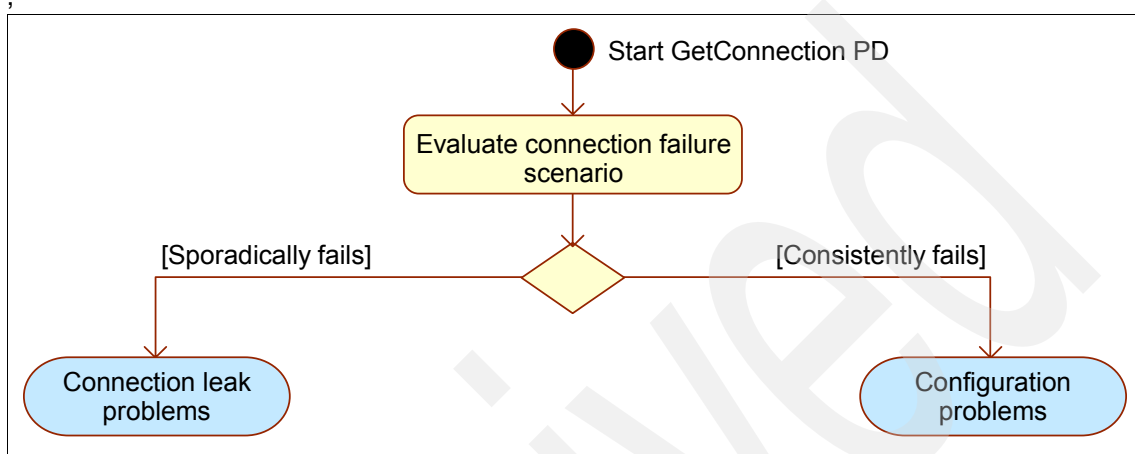


Figure 6-6 Problem determination path for failed connections

Examining the situation where there is a failure getting a new connection can lead you to one of the following two cases:

- ▶ Connection consistently fails, which means that you might have a problem with your application configuration. Proceed to “Configuration problems” on page 228 for further analysis.
- ▶ Connection fails sporadically, which means that it is very likely that you have a connection leak. Proceed to “Connection leak problems” on page 236 for further analysis.

If none of these symptoms apply, go to “The next step” on page 249.

## 6.2.4 Symptom: Failure to access a resource through JDBC

When an application uses JDBC to access a relational database and the access fails, you most likely receive error messages with prefixes WTRN (transaction problem) or DSRA (data source resource adapter).

### Data to collect

All relevant error messages and exceptions that are needed for our analysis are available in `SystemOut.log` and `SystemErr.log`.

## What to look for

Examine the exceptions in SystemOut.log and SystemErr.log for more information, as shown in Figure 6-7.

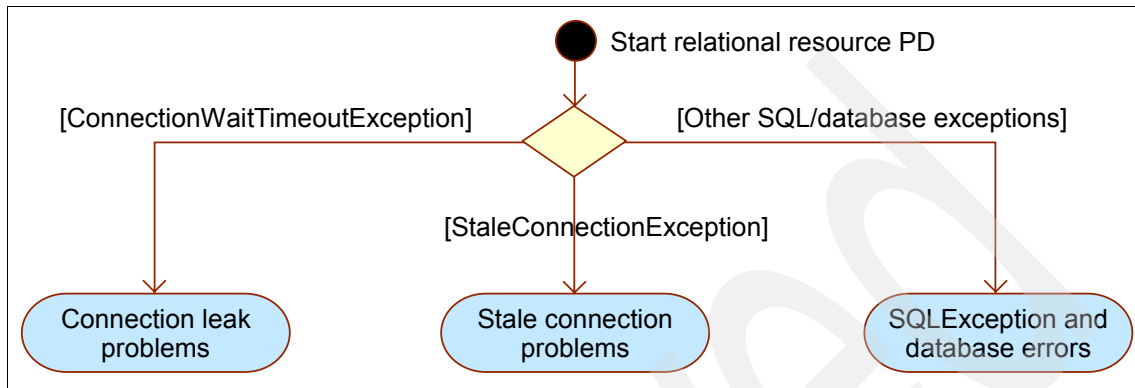


Figure 6-7 Relational resource problem determination

Depending on what you find, take one of the following actions:

- ▶ **ConnectionWaitTimeoutException**  
Proceed to “Connection leak problems” on page 236 for further analysis.
- ▶ **StaleConnectionException**  
Proceed to “Stale connection problems” on page 243 for further analysis.
- ▶ **Other SQL/database exceptions**  
Proceed to “SQLException and database errors” on page 245 for further analysis.

If none of these symptoms apply, go to “The next step” on page 249.

### 6.2.5 Symptom: Failure to access a non-relational resource

When your application uses non-relational resource adapters, such as CICS® ECI, Siebel, or SAP, the failure to access a resource usually involves the XAResource that manages transactions or connection factory that creates connections to the resource.

#### Data to collect

All relevant error messages and exceptions that are needed for our analysis are available in SystemOut.log and SystemErr.log.

## What to look for

In most situations, you receive error messages with prefixes WTRN (transaction problem) or J2CA (general JCA connector problem). The errors can be classified into scenarios as shown in Figure 6-8.

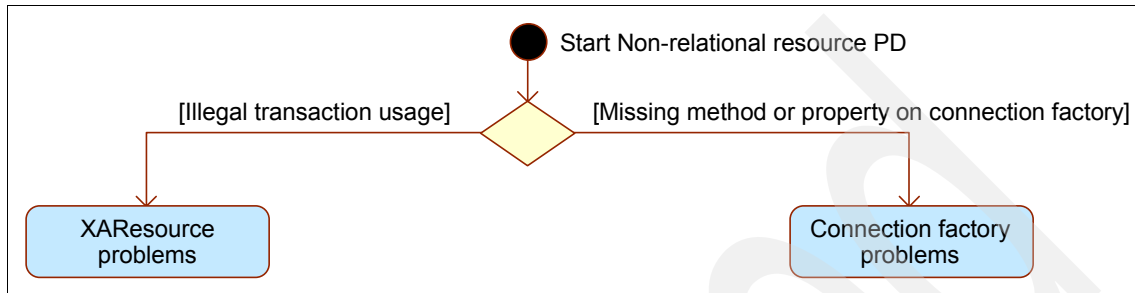


Figure 6-8 Non-relational resource problem determination

Depending on what you find, do one of the following:

- ▶ Illegal transaction usage  
Proceed to “XAResource problems” on page 246 for further analysis.
- ▶ Missing method or property on connection factory  
Proceed to “Connection factory problems” on page 247 for further analysis.

If none of these symptoms apply, go to “The next step” on page 249.

## 6.3 Analyzing problem areas

The symptom analysis from the previous section is designed to help you identify specific areas to explore that might be relevant to your problem. This section continues the process of symptom analysis but with more emphasis on the components that might be at the root of the problem. For each problem area, you perform a series of actions to help you determine the cause of the error and are given possible resolutions or advice on how to proceed.

If you reached this point without being able to identify possible problem areas, see “The next step” on page 249.

### 6.3.1 Configuration problems

Use the steps discussed in this section to diagnose configuration problems.

#### Review the configuration

The first step in addressing configuration problems is to ensure that you understand how the resource should be configured. If this is a new resource, review the following article in the WebSphere Information Center for information about how to create and configure a JDBC provider and data source:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.base.doc/info/aes/ae/tdat\\_tccrtprovds.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.base.doc/info/aes/ae/tdat_tccrtprovds.html)

#### Review the WebSphere Information Center for known issues

The next step is to check the WebSphere Information Center for information about common data source configuration problems:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rtrb\\_dsaccess.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rtrb_dsaccess.html)

#### Tip for Oracle users:

Oracle 10g is supported officially in WebSphere Application Server V6. If you are using the Oracle 10g JDBC driver, your data source must use the `com.ibm.websphere.rsadapter.Oracle10gDataStoreHelper` data store helper class.

Oracle 10g resolves the following known issues with the Oracle 8i or 9i JDBC drivers:

- ▶ XA with Serializable isolation level is now supported.
- ▶ Oracle lifted the 4 KB limitation on BLOB data type with Oracle Thin JDBC driver.
- ▶ Oracle now returns a connection which allows connections with auto commit mode set to true to start an XA transaction.

If the configuration appears to be correct and none of the known issues apply to your situation, the next step is to search the JVM logs for meaningful error messages or exceptions.

#### Examine the JVM logs for error messages and exceptions

Search SystemOut.log and SystemErr.log for DSRA messages ranging from 8000 through 8499. If you find messages in this range, it is still likely that your data source configuration needs correction.

The following WebSphere Information Center article contains information about these messages, including an explanation and user response:

<http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.messages.doc/doc/DSRA.html>

The various resolution options depends on the types of exceptions that you encounter. The most common exceptions that are related to data source's configuration are:

► **ClassNotFoundException**

The JVM throws a `ClassNotFoundException` when an application tries to load in a class using its string name in the following three methods but no definition for the class with the specified name could be found:

- `Class.forName()`
- `ClassLoader.findSystemClass()`
- `ClassLoader.loadClass()`

Verify that the class path and the JAR file(s) for the JDBC provider are correct and exist on the server where the application with the associated data source is running.

If the definition of the data source includes a WebSphere variable, make sure that the variable is defined and set at a correct scope (from the administrative console, go to **Environment** → **WebSphere Variables**).

Remember that directories, file names, symbolic links, and so forth on UNIX platforms are case sensitive. Make sure that you have defined the directory and file names exactly as you see them on the UNIX file system.

► **ClassCastException**

The JVM throws a `ClassCastException` to indicate that the code has attempted to cast an object to a subclass of which it is not an instance:

```
(targetClass)sourceObject;
```

`ClassCastException` occur whenever:

- The type (class) of `sourceObject` is not a subclass of `targetClass`
- The type (class) of `sourceObject` is in the class ancestry of class `targetClass`, but the class loader which loaded the class of `sourceObject` is different than the class loader that loaded class `targetClass`

When dealing with a data source configuration, the latter case is the most likely. WebSphere Application Server's runtime is a multi-class loader environment and when `targetClass` is visible to more than one class loader, more than one instance is deployed.

Check to make sure that there is only one copy of each JAR file that is required for the JDBC provider. Search the file system starting from the

<WAS\_install\_root> directory for duplication. Make sure the application's EAR file does not include the JDBC driver's JAR files.

► UnsatisfiedLinkError

The JVM throws UnsatisfiedLinkError when it cannot find the native library or when the JVM has already loaded the native library.

This exception only occurs when you are using type 2 JDBC drivers (for example, DB2 CLI Legacy, DB2 Universal Type 2, and Oracle OCI drivers). Type 2 JDBC drivers require that some binary code is loaded on each client machine (WebSphere Application Server machine). These native libraries are:

- \*.dll files on Windows platforms
- \*.so files on Solaris
- \*.sl files on HP-UX
- \*.a or \*.so files on AIX

For more information about different types of JDBC drivers, see the Sun™ Developer Network at:

<http://java.sun.com/products/jdbc/driverdesc.html>

If there is no value specified for the native library path in your data source's configuration, you might get the UnsatisfiedLinkError:

- On Windows platforms, set the system environment PATH to include the path where the \*.dll files can be loaded.
- On UNIX platforms, you might need to set up the user's profile to set the environment variables before starting the WebSphere Application Server processes.

Examine the native file name from the error message and confirm that it has the correct format and exists on the server.

Table 6-1 on page 230 shows the list of JDBC drivers that are supported by WebSphere Application Server V6 and their JAR files, dependent environment variables, and so forth.

Table 6-1 Driver type-to-JAR file mapping

Database	Driver	Type	Required Environment Variable(s)	Default Jar Files
1. Informix: ifxjdbc_g.jar and ifxjdbcx_g.jar (debug version provides JDBC trace) 2. SQL Server: spy-sl.jar provides JDBC trace 3. SQL Server: spy.jar provides JDBC trace 4. Oracle: ojdbc14_g.jar (debug version provides JDBC trace) 5. LD_LIBRARY_PATH for Solaris and Linux, LIBPATH for AIX, SHLIB_PATH for HP-UX and PATH for Windows.				

<b>IBM</b>				
DB2	DB2 CLI Legacy	2	DB2INSTANCE LD_LIBRARY_PATH <sup>5</sup>	db2java.zip
	DB2 Universal JDBC (or JCC)	2, 4	LD_LIBRARY_PATH <sup>5</sup>	db2jcc.jar db2jcc_license_cu.jar db2jcc_license_cisuz.jar
	OS/400® Toolbox	4		jt400.jar
	OS/400 Native	2		db2_classes.jar
Informix	Informix JDBC	4		ifxjdbc.jar and ifxjdbcx.jar <sup>1</sup>
Cloudscape™	Universal JDBC	4		db2jcc.jar, db2jcc_license_cu.jar and db2jcc_license_cisuz.jar
<b>DataDirect Technology</b>				
SQL Server	Sequelink 5.4	3		sljc.jar, spy-sl.ja <sup>2</sup>
	Connect JDBC	4		base.jar, sqlserver.jar, util.jar and spy.jar <sup>3</sup>
<b>Oracle Corp</b>				
Oracle	OCI	2	ORACLE_HOME LD_LIBRARY_PATH <sup>5</sup>	ojdbc14.jar <sup>4</sup>
	Thin	4		ojdbc14.jar <sup>4</sup>
<b>Microsoft Corp</b>				
SQL Server	SQL Server 2000	4		msbase mssqlserver.jar msutil.jar
<b>Sybase Corp</b>				
Sybase	jConnect	4		jconn2.jar
<ol style="list-style-type: none"> <li>1. Informix: ifxjdbc_g.jar and ifxjdbcx_g.jar (debug version provides JDBC trace)</li> <li>2. SQL Server: spy-sl.jar provides JDBC trace</li> <li>3. SQL Server: spy.jar provides JDBC trace</li> <li>4. Oracle: ojdbc14_g.jar (debug version provides JDBC trace)</li> <li>5. LD_LIBRARY_PATH for Solaris and Linux, LIBPATH for AIX, SHLIB_PATH for HP-UX and PATH for Windows.</li> </ol>				

If you have verified that the JDBC provider and data source configuration are correct and you still get the error connecting to the database, write a stand-alone JDBC program to see if you can connect to the back-end database using the same JDBC driver.

If the stand-alone JDBC program gives the same error, then the JDBC driver version might be incompatible with the back-end database version. If this is the case, you might need to re-install the JDBC driver and database client.

If the stand-alone JDBC program can connect to the back-end database, compare the settings of the environment variables (for example CLASSPATH, LIBPATH, and so forth) to make sure the settings from the successful session are the same as those settings in WebSphere Application Server.

► `java.sql.SQLException: invalid arguments in call`

This exception occurs when the database requires a user ID and password for a connection but the data source does not have the custom properties for user and password defined or there is no J2C authentication alias specified for the component-managed authentication alias (used only when the application resource reference is using `res-auth = Application`).

To specify the user ID and password using a J2C authentication alias, you first need to create the alias (if it does not exist). To create new J2C authentication alias using the administrative console:

- a. Navigate to **JDBC providers** → **<provider>** → **Data sources** → **<datasource>**.
- b. In the Related Items section, select **J2EE Connector Architecture (J2C) authentication data entries**.
- c. Click **New** to create a new entry.

When the alias is created, you need to select it in the component-managed authentication alias field in the data source.



### Tips when using an XA connection:

#### ► Oracle

Oracle databases must be configured with the Oracle JVM. See the following Technote for one method that you can use to configure an Oracle database with the Oracle JVM:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21046276>

As an alternative, refer to the Oracle documentation for instructions on configuring the database with the Oracle JVM using scripts.

#### ► SQL Server

The Microsoft Distributed Transaction Coordinator service must be running on the server where SQL Server is running.

When using DataDirect Connect JDBC or WebSphere embedded Connect JDBC Driver, a set of stored procedures must be installed on the master database. See the *DataDirect Connect for JDBC User's Guide and Reference*, that is available from DataDirect Technologies at URL:

<http://www.datadirect.com/techres/jdbcproddoc/index.ssp>

In particular, see *Installing Stored Procedures for JTA Instructions* in Chapter 6, *The Microsoft SQL Server Driver*.

#### ► DB2

If the DB2 server is on OS/390®, you must configure the Sync Point Manager for multi-site updates from DB2 Connect™ to DB2 OS/390. See technote:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21114514>

#### ► Informix

If your application requires an XA connection to an Informix database, then you should use Informix JDBC 3.0 because there are known problems with an XA connection using Informix JDBC 2.2.x. See the following Technotes:

- *XA transactions are not working properly with Informix Dynamic Server*

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21204793>

- *setTransactionIsolation on a connection does not work for XA transactions in Informix 9.4 and Informix 10.0*

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21210281>

## 6.3.2 Naming problems

This section discusses the situation where you see `NamingException` when the application accesses a data source. Proceed with the following steps to diagnose a naming problem.

### Ensure that the application uses resource references

Applications are required to use a resource reference to access a data source or connection factory. For information about this, see the WebSphere Information Center item *Looking up data sources with resource references for relational access* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/cdat\\_datsorres.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/cdat_datsorres.html)

### Ensure that an indirect JNDI name is used

When defining a connection factory or data source, a JNDI name by which the connection factory or data source can be looked up by a component is identified. Preferably an indirect name with the `java:comp/env` prefix should be used (and must be used in future releases). An indirect name makes the resource reference data associated with the application available to the connection management runtime. This enhances resource management through the use of the `res-auth`, `res-isolation-level`, `res-sharing-scope`, and `res-resolution-control` settings.

Though you can still use a direct JNDI name (for example `jdbc/myDataSource`), this naming method is deprecated in WebSphere Application Server V6. WebSphere Application Server assigns default values to the resource reference data when you use this method. You see the warning message `J2CA0294W` logged in the JVM's `SystemOut.log` to document the defaults, as shown in Example 6-8.

#### *Example 6-8 Warning message J2CA0294W when using direct JNDI name*

---

```
J2CA0294W: Deprecated usage of direct JNDI lookup of resource jdbc/IOPEntity.
The following default values are used: [Resource-ref settings]
    res-auth:                1 (APPLICATION)
    res-isolation-level:      0 (TRANSACTION_NONE)
    res-sharing-scope:        true (SHAREABLE)
    loginConfigurationName:   null
    loginConfigProperties:    null
[Other attributes]
    res-resolution-control:   999 (undefined)
isCMP1_x:                    false (not CMP1.x)
isJMS:                        false (not JMS)
```

---

The application should be changed to use an indirect JNDI name (for example, `java:comp/env/jdbc/myDataSource`) instead of the direct JNDI name

(jdbc/myDataSource), and a resource reference should be created. See the WebSphere Information Center item *Connection factory JNDI name tips* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rdat\\_jnditips.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rdat_jnditips.html)

It is possible to suppress the J2CA0294W warning message from the SystemOut.log by setting the logging level to:

```
*=info:com.ibm.ejs.j2c.ConnectionFactoryBuilderImpl=severe
```

A setting of severe for that class suppresses the logging of warning and lower level messages. However, this should be considered as an interim solution until the application can be updated.

For more information, see the WebSphere Information Center section *Naming* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/welc6tech\\_nam.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/welc6tech_nam.html)

### 6.3.3 Authentication problems

J2C connector authentication data entries are used by resource adapters and JDBC data sources. Authentication is the process that identifies the caller. If authentication fails, it is likely that you have incorrectly configured the user's identity.

For more information, see the *Troubleshooting security configuration* section in the WebSphere Information Center at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec\\_trouble.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/tsec_trouble.html)

To verify the required configuration, do the following steps.

#### **Check user registry configuration**

First, you need to ensure the following:

- ▶ The user name and password that are specified in the J2C authentication alias are correct and can be used to access the database.
- ▶ The user registry that you defined the user and user ID in is the same registry that is specified in the WebSphere global security settings.

## Check the J2C authentication data entry

When configuring resource adapters or data sources, the administrator specifies the alias for the J2C authentication data entry to use to access the resource. In the General properties of the data source, verify that you have a J2C authentication data entry specified in the Component-managed authentication alias field.

## Check the application

Any client running in the same cell that can look up a resource in the JNDI namespace can obtain connections without explicitly providing authentication data on the `getConnection()` call. If the component's `res-auth` setting is `Application`, authentication is taken from the component-managed authentication alias defined on the connection factory. With `res-auth` set to `Container`, authentication is taken from the login configuration defined on the component's resource-reference. It is important to note that J2C authentication alias is defined per cell. An enterprise bean or servlet in one application server cannot look up a resource in another server process in a different cell, because the alias would not be resolved.

If the application explicitly provides a user ID and password on the `getConnection()` when obtaining the connection (that is `getConnection(user,password)` and the component's `res-auth` setting is set to `Application`, the authentication data that is used is the user ID and password that is passed in the `getConnection()` method. You need to ensure that the specified user ID and password can be used to access the database.

## Enable traces

If the configuration appears to be correct and you continue to have authentication problems, you should enable `WAS.j2c` and `RRA` traces, recreate the problem, and then proceed to "The next step" on page 249.

### 6.3.4 Connection leak problems

A connection leak occurs when the application uses a connection but it never explicitly calls the `close()` method on the connection object to return the connection back to the connection pool for reuse. Example 6-9 demonstrates the proper closing of connection after usage.

#### *Example 6-9 Closing a connection*

---

```
Connection pooledCon = null;
DataSource ds = null;
InitialContext ic = null;
try {
    ds = (DataSource)ic.lookup("java:comp/env/jdbc/myDataSource");
```

```

        pooledCon = ds.getConnection("username", "password");
        // Processing Code goes here
    } catch (Exception ignored) {
        // catch JNDI or JDBC exceptions here
    } finally {
        // If not calling close, it is considered connection leak
        if(pooledCon != null)
            pooledCon.close();
    }
}

```

---

You might have a connection leak if your application receives exceptions such as `com.ibm.websphere.ce.cm.ConnectionWaitTimeoutException` or `com.ibm.websphere.ce.j2c.ConnectionWaitTimeoutException` when attempting to access a data source or JCA-compliant resource adapter, respectively.

In Example 6-10, the application receives a `ConnectionWaitTimeoutException` after waiting for 1800030 milliseconds for a connection, but no free connection from the pool is available. If the connection pool has a high value for the connection timeout (for example, 1800 seconds), you might not see the `ConnectionWaitTimeoutException` immediately.

#### *Example 6-10 ConnectionWaitTimeoutException*

---

```

[7/12/05 14:04:58:328 AST] 3ad6086e ConnectionMan E J2CA0020E: The
Connection Pool Manager could not allocate a Managed Connection:
com.ibm.websphere.ce.j2c.ConnectionWaitTimeoutException: Connection
not available, Timed out waiting for 1800030
at java.lang.Throwable.<init>(Throwable.java:195)
at java.lang.Exception.<init>(Exception.java:41)
at
javax.resource.ResourceException.<init>(ResourceException.java:73)
at
javax.resource.spi.ResourceAllocationException.<init>(ResourceAllocationException.java:55)
at
com.ibm.websphere.ce.j2c.ConnectionWaitTimeoutException.<init>(ConnectionWaitTimeoutException.java:38)
at
com.ibm.ejs.j2c.poolmanager.FreePool.createOrWaitForConnection(FreePool.java:1100)
at
com.ibm.ejs.j2c.poolmanager.PoolManager.reserve(PoolManager.java:1747)

```

---

Before deciding this problem is a connection leak, you should rule out any performance tuning problems. When you have ruled out these and if the problem still exists, you have to do further analysis.

## Review your current connection pool use

Examining the connection pool settings can reveal some possible causes for `ConnectionWaitTimeoutException` due to the pool over use. The following are possibilities:

- ▶ The maximum number of connections for a given pool is set too low.  

The demand for concurrent use of connections is greater than the configured maximum value for the connection pool. One indication of this problem is that you receive these exceptions regularly, but your process use is not high. This exception indicates that there are too few connections available to keep the threads in the server busy.
- ▶ Connection timeout is set too low.  

Current demand for connections is high enough such that sometimes there is not an available connection for short periods of time. If your connection wait timeout value is too low, you might timeout shortly before a user returns a connection back to the pool. Adjusting the connection wait time can give you some relief. One indication of this problem is that you use close to the maximum number of connections for an extended period and are receiving this error regularly.
- ▶ You are not closing some connections, or you are returning connections back to the pool at a very slow rate.  

This situation can happen when using unshareable connections, when you forget to close connections, or you close them long after you are finished using them, thus keeping the connection from returning to the pool for reuse. The pool soon becomes empty, and all applications get `ConnectionWaitTimeoutExceptions`. One indication of this problem is that you run out of connections in the connection pool and you receive this error on most requests.
- ▶ You are driving more load than the server or back-end system has resources to handle.  

In this case, you must determine which resources you need more of and upgrade configurations or hardware to address the need. One indication of this problem is that the application or database server process is nearly 100% busy.

## Tune the connection pool settings

Before continuing, review the WebSphere Information Center item *Connection pool settings* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.express.doc/info/exp/ae/udat\\_conpoolset.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.express.doc/info/exp/ae/udat_conpoolset.html)

You can adjust the following parameters for a data source:

- **Maximum Connections**

Specifies the maximum number of `ManagedConnections` that can be created in the pool.

`ManagedConnections` represent the physical connection to the back-end resource. When the maximum number is reached, no new `ManagedConnections` are created, and the requester waits until a `ManagedConnection` currently in use is returned to the pool or a `ConnectionWaitTimeoutException` is thrown. The default value is 10, allowing the number of `ManagedConnections` to grow to 10. If the maximum connections is changed to 0, the number of `ManagedConnections` can grow infinitely, and `Connection Timeout` will not be used.

For example, if `Maximum Connections` is set to 5 and there are 5 `ManagedConnections` in use, the pool manager waits for a managed connection to become free for `Connection Timeout` seconds. A `ConnectionWaitTimeoutException` is thrown if a connection is not returned to the pool in time.

- **Connection Timeout**

Specifies the interval, in seconds, after which a connection request times out and a `ConnectionWaitTimeoutException` is thrown.

This value indicates the number of seconds that a request for a connection waits when there are no connections available in the free pool and no new connections can be created, usually because the maximum value of connections in the particular connection pool has been reached. For example, if `Connection Timeout` is set to 300, and the maximum number of connections are all in use, the pool manager waits for 300 seconds for a physical connection to become available.

If a physical connection is not available within this time, the pool manager initiates a `ConnectionWaitTimeoutException`. It usually does not make sense to retry the `getConnection()` method. If a longer wait time is required, you should increase the `Connection Timeout` setting value. If a `ConnectionWaitTimeoutException` is caught by the application, the administrator should review the expected connection pool usage of the application and tune the connection pool and database accordingly.

If the `Connection Timeout` is set to 0, the pool manager waits as long as necessary until a connection becomes available. This happens when the application completes a transaction and returns a connection to the pool, or when the number of connections falls below the value of `Maximum Connections`, allowing a new physical connection to be created.

If `Maximum Connections` is set to 0, which enables an infinite number of physical connections, then the `Connection Timeout` value is ignored.

## Enable connection leak trace facility

If you have ruled out performance tuning problems and the problem still exists, you need to enable the connection leak trace logic for further analysis.

After adjusting your connection pool settings, if it takes longer to receive the `ConnectionWaitTimeoutException`, then there is a high possibility that your application has a connection leak. The WebSphere Application Server V6 connection manager provides a diagnostic feature called the connection leak trace logic that gathers information about which application methods could potentially lead to leaking connections (not closing connection) or holding on to connections longer than expected (for example, long running queries).

To enable the connection leak trace logic, set the log detail level to `ConnLeakLogic=finest`.

**Note:** The WAS.j2c trace includes the `ConnLeakLogic` trace. If you enable WAS.j2c trace, then you do not have to enable `ConnLeakLogic`.

When you enable the `ConnLeakLogic` trace, for every time interval (the default is 10 seconds), WebSphere connection manager checks how long a connection has been in use and prints the stack trace to trace log. Currently, the default time interval is unchangeable. If you have a need to change the default value, contact IBM technical support to obtain an iFix that allows you to add a custom property at the data source.

### *What to look for in the trace*

When a connection is leaked or held longer than the trace time interval (10 seconds), the trace contains the string `Connection Leak Logic Information` followed by a stack trace with all methods involved up to the `getConnection()`, including the application method that acquired the connection, as shown in Example 6-11.

#### *Example 6-11 Connection Leak Logic Information with stack trace*

---

**Connection Leak Logic Information:**

```
MCWrapper id 16089ef2 Managed connection
com.ibm.ws.rsadapter.spi.WSRdbManagedConnectionImpl@6e909ef2
State:STATE_ACTIVE_INUSE Thread Id: 15861ecd Thread Name:
Servlet.Engine.Transports : 1
  Start time inuse Wed Nov 03 08:04:54 CST 2004 Time inuse 20 (seconds)
  Last allocation time Wed Nov 03 08:04:54 CST 2004
    getConnection stack trace information:
      at
com.ibm.ejs.j2c.ConnectionManager.allocateConnection(ConnectionManager.java:565
)
```



```

    at
com.ibm.ws.rsadapter.jdbc.WSJdbcDataSource.getConnection(WSJdbcDataSource.java:
215)
    at
com.ibm.ws.rsadapter.jdbc.WSJdbcDataSource.getConnection(WSJdbcDataSource.java:
306)
    at SnoopServlet.doGet(SnoopServlet.java:130)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:740)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
    at
com.ibm.ws.webcontainer.servlet.StrictServletInstance.doService(StrictServletIn
stance.java:110)
    at
com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet._service(StrictLifecycle
Servlet.java:174)
    at
com.ibm.ws.webcontainer.servlet.IdleServletState.service(StrictLifecycleServlet
.java:313)
    at
com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet.service(StrictLifecycleS
ervlet.java:116)
    at
com.ibm.ws.webcontainer.servlet.ServletInstance.service(ServletInstance.java:28
3)
    at
com.ibm.ws.webcontainer.servlet.ValidServletReferenceState.dispatch(ValidServletReferenceState.java:42)
    at
com.ibm.ws.webcontainer.servlet.ServletInstanceReference.dispatch(ServletInstan
ceReference.java:40)
    at
com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.handleWebAppDispatch(Web
AppRequestDispatcher.java:1059)
    at
com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.dispatch(WebAppRequestDi
spatcher.java:588)
    at
com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.forward(WebAppRequestDis
patcher.java:206)
    at com.ibm.ws.webcontainer.srt.WebAppInvoker.doForward(WebAppInvoker.java:80)
    at
com.ibm.ws.webcontainer.srt.WebAppInvoker.handleInvocationHook(WebAppInvoker.ja
va:214)
    at
com.ibm.ws.webcontainer.cache.invocation.CachedInvocation.handleInvocation(Cach
edInvocation.java:71)
    at
com.ibm.ws.webcontainer.srp.ServletRequestProcessor.dispatchByURI(ServletReques
tProcessor.java:182)

```

```
at
com.ibm.ws.webcontainer.oselister.OSEListenerDispatcher.service(OSEListener.j
ava:334)
at
com.ibm.ws.webcontainer.http.HttpConnection.handleRequest(HttpConnection.java:5
6)
at
com.ibm.ws.http.HttpConnection.readAndHandleRequest(HttpConnection.java:615)
at com.ibm.ws.http.HttpConnection.run(HttpConnection.java:439)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:672)
```

---

After enabling the connection leak trace logic, let the application run and monitor the trace. Search the trace file for string `Connection Leak Logic Information`. If you see it, there are connections that have been in use for more than 10 seconds. You can analyze the stack trace for the suspected application methods. In Example 6-11 on page 240, the `doGet()` method of the `SnoopServlet.java` is the suspected source for leaking connections.

### Check shareable connection settings

WebSphere Application Server supports both *unshareable* and *shareable* connections. An unshareable connection is not shared with other components in the application. The component that uses this connection has full control of this connection. On the other hand, a shareable connection can be shared with other components within the same transaction, as long as each `getConnection()` request has the same connection properties.

If you use shareable connections, incorrect settings can cause a connection leak. So, ensure that the following connection properties are the same:

- ▶ JNDI name. While not actually a connection property, this requirement simply means that you can only share connections from the same data source in the same server.
- ▶ Resource authentication.
- ▶ In relational databases:
  - Isolation level (corresponds to access intent policies applied to CMP beans)
  - Readonly
  - Catalog
  - TypeMap

To enable connection sharing for resource adapters within the same transaction, the following connection properties must be the same:

- ▶ JNDI name. While not actually a connection property, this requirement simply means that you can only share connections from the same resource adapter in the same server.
- ▶ Resource authentication.

## Resolution

After the source for leaking connections has been identified, you should ask your application developer to review the source and fix the problem. Some other possible solutions to correct the `ConnectionWaitTimeoutException` are:

- ▶ Modify your application to use fewer connections.
- ▶ Check your application to make sure the connections are properly closed.

WebSphere Application Server establishes a queuing network that is comprised of the Web server, Web container, EJB container, Object Request Broker (ORB), and data source components. You should review and adjust these queues to improve WebSphere performance. See the following for more information:

- ▶ *IBM WebSphere Application Server - Performance Tuning: Adjusting WebSphere Application Server System Queues*  
<http://www.redbooks.ibm.com/abstracts/tips0244.html>
- ▶ *IBM WebSphere Application Server - Performance Tuning: Determining Optimum Queue Sizes*  
<http://www.redbooks.ibm.com/abstracts/tips0245.html>

Also, review the following articles in the WebSphere Information Center:

- ▶ *Tuning the application serving environment*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/tprf\\_tuneprf.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/tprf_tuneprf.html)
- ▶ *Tuning parameters for data access resources*  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rdat\\_datobjtune.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rdat_datobjtune.html)

### 6.3.5 Stale connection problems

You have a stale connection problem when you receive the exception `com.ibm.websphere.ce.cm.StaleConnectionException`. This exception indicates that the connection currently held is no longer valid.

When a connection to a database becomes stale, any operation that is performed on that connection receives an `SQLException` from the JDBC driver. `SQLException` is a generic exception. It contains `SQLState` and `SQLCode` values that you can use to determine the meaning of the exception. The meaning of the `SQLState` and `SQLCode` varies depending on the database vendor. The WebSphere RRAs maintain the mapping of `SQLStates` and `SQLCodes` to `StaleConnectionExceptions` for each of the supported relational databases. When the connection pooling runtime catches an `SQLException`, it checks to see if this `SQLException` is considered a `StaleConnection` exception for the database server in use.

Before proceeding, see the *Stale connections* item in the WebSphere Information Center for information about how to detect stale connections and how to recover from stale connections:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.base.doc/info/aes/ae/cdat\\_stalecon.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.base.doc/info/aes/ae/cdat_stalecon.html)

To analyze the stale connection problems, proceed with the steps discussed in the following sections.

### **Check timeout settings**

Check with your database administrator or network system administrator to determine if there are timeout settings on the database or firewall. Timeouts on the database and firewall can close your connections and thus cause the `StaleConnectionException` to occur. You might need to disable your database or firewall timeouts or review the connection pool settings so that they are suited to your environment. For example, the connection pool aged timeout should be less than the firewall timeout, which should be less than the database timeout.

### **Determine if a specific query is getting the exception**

If you are getting `StaleConnectionException` when executing a certain query, it is likely that the query causes the JDBC driver to return the `SQLException` with an `SQLState` and `SQLCode` that are mapped to `StaleConnectionException`. The `SQLState` and `SQLCode` can help you determine the root cause of the problem.

### **Trace the problem**

A JDBC driver returns an `SQLException` when there is problem with the query that is executed on the connection. Sometimes the connections are unusable after the `SQLException` occurs, but the application server does not throw the `StaleConnectionException` because the `SQLState` and `SQLCode` are not the ones from the mapping list for `StaleConnectionException`.

If this is the case, you should try to recreate the problem with a simple test case and trace the problem with the `WAS.database`, `RRA`, and `WAS.j2c` trace options

enabled. The trace should help you identify the query that is causing the problem and the SQLException that is returned by JDBC driver.

### 6.3.6 SQLException and database errors

If you receive an SQLException (not caused by connection leaks or stale connections) or database-specific exceptions, perform the steps that are discussed in this section.

#### Check your database error code documentation

An SQLException is often accompanied with a database-specific error code. In this case, you should check the database product documentation for an explanation and actions to take on the error.

Example 6-12 shows an SQLException for a DB2 database.

---

#### *Example 6-12 SQLException caused by a database-specific error*

---

```
java.sql.SQLException: [IBM][CLI Driver] CLI0108E  Communication link
failure. SQLSTATE=40003DSRA0010E: SQL State = 40003, Error Code =
-99,999
```

---

You can check DB2's CLI messages in the DB2 Information Center at:

<http://publib.boulder.ibm.com/infocenter/db2help/topic/com.ibm.db2.udb.doc/core/r0climg.htm>

If you look up the message found in the SQLException, you find the following explanation:

CLI0108E Communication link failure.

Explanation: The connection between the driver and the data source failed during execution of this function.

User Response: Establish a new connection.

Based on the information from your database documentation, you can decide on a suitable course of action. As in this example, one of the possible causes is a communication link failure. In this case, you need to check your network connection and then re-run the application to re-establish the connection.

Example 6-13 shows another instance of a DB2-specific error.

---

#### *Example 6-13 SQL1040N*

---

```
COM.ibm.db2.jdbc.DB2Exception: [IBM][CLI Driver] SQL1040N  The maximum number
of applications is already connected to the database.  SQLSTATE=57030
```

---

You can check DB2's SQL messages in the DB2 Information Center at:

<http://publib.boulder.ibm.com/infocenter/db2help/topic/com.ibm.db2.udb.doc/core/rsql1000.htm>

Example 6-14 shows the explanation for this message.

*Example 6-14 Message SQL1040N explanation*

---

SQL1040N The maximum number of applications is already connected to the database.

Explanation: The number of applications connected to the database is equal to the maximum value defined in the configuration file for the database. The command cannot be processed.

User Response: Wait for other applications to disconnect from the database. If more applications are required to run concurrently, increase the value for maxappls. After all applications disconnect from the database and the database is restarted, the new value takes effect.

---

In this case, the DB2 parameters, maxappls and maxagents, are set too low for the DB2 database. When configuring the data source settings for the database, ensure that the DB2 maxappls setting is greater than the maximum number of connections for the data source. The DB2 maxagents setting should be the sum of maxappls for all databases.

### **Trace the problem**

When the suggested user response to the error code is not sufficient to resolve the problem, you might need to enable JDBC and database traces and engage the JDBC driver and database vendors to help you further determine the cause of the problem.

## **6.3.7 XAResource problems**

The resource adapter has implementation code for the XAResource interface that supports transactions involving multiple resource managers. If your application has an error relating to invalid use of a transaction, you receive `com.ibm.ws.Transaction.IllegalResourceIn2PCTransactionException` (Example 6-15). This exception indicates that the application has incorrectly attempted to enlist multiple 1PC (1-phase commit or local transaction) resources in a 2PC (two-phase commit or XA) operation.

*Example 6-15 XAResource transaction exception*

---

```
[20/01/05 12:41:35:920 EST] 00000036 RegisteredRes E   WTRN0062E: An
illegal attempt to use multiple resources that have only one-phase
capability has occurred within a global transaction.
[20/01/05 12:41:36:326 EST] 00000036 LocalTransact E   J2CA0030E: Method
```

```

enlist caught
com.ibm.ws.Transaction.IllegalResourceIn2PCTransactionException: Illegal
attempt to enlist multiple 1PC XAResources
    at
com.ibm.ws.Transaction.JTA.RegisteredResources.enlistResource(Registered
Resources.java:362)
    at
com.ibm.ws.Transaction.JTA.TransactionImpl.enlistResource(TransactionImp
l.java:2693)
    at
com.ibm.ws.Transaction.JTA.TranManagerSet.enlistOnePhase(TranManagerSet.
java:417)
    at
com.ibm.ejs.j2c.LocalTransactionWrapper.enlist(LocalTransactionWrapper.j
ava:514)
    at
com.ibm.ejs.j2c.ConnectionManager.initializeForUOW(ConnectionManager.jav
a:1250)
    at
com.ibm.ejs.j2c.ConnectionManager.involveMCInTran(ConnectionManager.java
:938)
    at
com.ibm.ejs.j2c.ConnectionManager.allocateConnection(ConnectionManager.j
ava:571)
    at
com.ibm.connector2.cics.CICSConnectionFactory.getConnection(CICSConnecti
onFactory.java:211)

```

---

If you have this error, check the application to make sure that you do not have a resource that supports only local transactions (1PC) included accidentally within a context of a 2PC XA transaction. This is illegal.

### 6.3.8 Connection factory problems

Problems on ManagedConnectionFactory that are related to missing methods, missing connection factory properties, or failure to instantiate an object usually produce error messages with prefix J2CA. Most problems of this type can be tracked to the resource adapter's deployment descriptor (ra.xml) and WebSphere Application Server's resource configuration file (resources.xml).

The ra.xml file is supplied by the resource adapter vendor and can be found in the following location:

```

<WAS_install_root>/profiles/<profile>/installedConnectors/<rarfile>/
META-INF/ra.xml

```

To find the resources.xml file, you need to be aware of the scope at which the resource adapter was defined (node, cell, server). For example, if you defined the resource adapter at the node level, the file is at:

```
<WAS_install_root>/profiles/<profile>/config/cells/<cell>/nodes/<node>/resources.xml
```

Proceed with the following actions.

## Check ra.xml

If the error message indicates that a missing method or property is the problem, check for a mismatch between the resource adapter's XML definition in the ra.xml file and the actual Java class that is provided for the JavaBean's implementation. If there is a mismatch, you need to provide this failure notification to the resource adapter provider.

The error shown in Example 6-16 is caused because the Siebel-managed connection factory implementation class does not provide the code for the method set UserName.

### *Example 6-16 Connection factory problem: missing method*

---

```
[2/22/05 15:19:26:508 EST] 3c6d43cd J2CXAResource W J2CA0008W: Class com.ibm.wps.wpai.jca.siebel.SiebelManagedConnFactory used by resource siebel did not contain method set UserName. Processing continued.
```

---

If the connection factory fails to instantiate an instance, as shown in Example 6-17, you might have an invalid definition in the ra.xml file that causes the class loader to fail to load the ManagedConnectionFactory class.

### *Example 6-17 Connection factory problem: missing class*

---

```
[6/28/05 10:20:27:216 HKT] 3b0010ba ConnectionFac E J2CA0009E: An exception occurred while trying to instantiate the ManagedConnectionFactory class com.ibm.ws.rsadapter.spi.WSManagedConnectionFactoryImpl used by resource
```

---

## Check resources.xml

It is also possible that the resources.xml file might be corrupted. You should verify that resources.xml contains the correct entry for the resource adapter. For example:

```
<resources.j2c:J2CResourceAdapter xmi:id="J2CResourceAdapter_1125056353109" name="IMS Connector for Java" archivePath="${CONNECTOR_INSTALL_ROOT}/jca15_ims9102.rar">
```

If not, or if it looks incorrect, delete and re-create the resource adapter using the administrative console (**Resource** → **Resource Adapters**).



## 6.4 The next step

The symptoms and problem areas included in this paper are some that you are more likely to experience. However, there are other things that can go wrong, or the cause of the problem might be related to something other than JCA components.

If, after going through this process, you still have an undiagnosed problem, it is recommended that you go back to *Approach to Problem Determination in WebSphere Application Server V6* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

Review the problem classifications to see if there are any other components that might be causing the problem.

If you feel sure you have a JCA connection problem, there are things you can do before contacting IBM support. First, review the documentation that you have gathered for errors that were not addressed in this paper and search support sites for information or fixes.

Next, collect all of the data that is outlined in the appropriate MustGather documents and raise a problem record with IBM as follows:

- ▶ MustGather: Database connections or connection pooling problems for all releases of V4.0, V5.0, and V6.0 at:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21145599>

- ▶ JVM logs
- ▶ FFDC logs
- ▶ A WebSphere Application Server trace with the following components enabled:
  - WAS.j2c
  - RRA
  - WAS.database

**Note:** The WAS.database trace group traces the JDBC calls for any of the supported JDBC drivers. You must use the debug jar file(s) for the associated JDBC driver if you enable WAS.database (see Table 6-1 on page 230).

You can use this documentation when you contact IBM support.



## Web server plug-in problem determination

This paper discusses techniques for diagnosing problems that are related to the Web server plug-in for WebSphere Application Server V6, including those resulting from configuration errors. The Web server plug-ins handle communications between a Web server and the Web container in an application server or servers. Symptoms of a problem that involve the plug-in include:

- ▶ Users cannot access an application through the Web server
- ▶ Load balancing and failover are not working properly
- ▶ Session data is being lost
- ▶ Slow or intermittent application response
- ▶ The Web server will not start after plug-in installation or configuration

The Web server plug-in works with six different Web servers. However, this paper concentrates on the IBM HTTP Server and its plug-in. It does not address Web server problems or problems that are related to the application server Web container.

**Important:** We recommend that you start your problem determination process by reading *Approach to Problem Determination in WebSphere Application Server V6* at <http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>.

## 7.1 Introduction

Figure 7-1 shows how the Web server plug-in interacts with both the Web server and the Web container in the application server. Note that the Web server and application server need not reside on the same physical server.

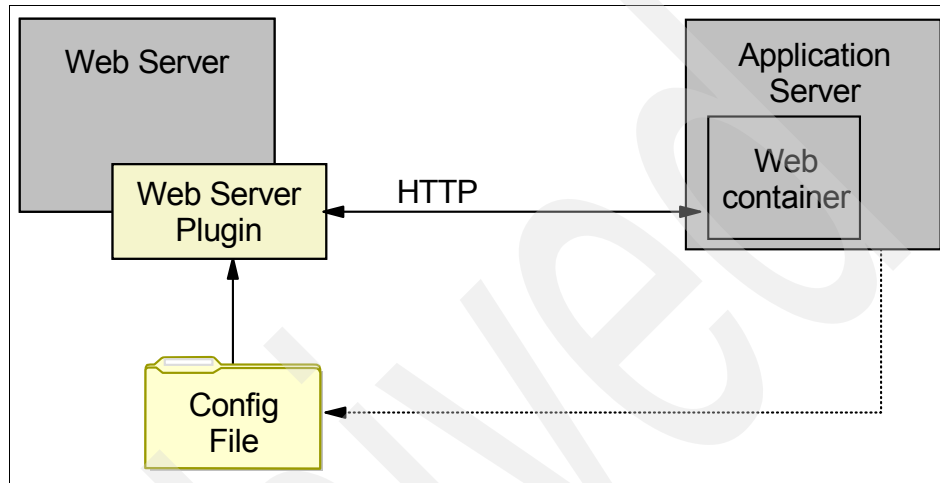


Figure 7-1 Web server plug-in interactions

As illustrated in Figure 7-1, the problem determination techniques in this paper focus on the Web server plug-in and its configuration file. Installing the plug-in requires that you change the Web server configuration and that you create the plug-in configuration file from using the WebSphere administrative tools. Thus, this paper also discusses these areas.

We look at the following symptoms that a user would see:

- ▶ Users cannot reach the application (page not found and page cannot be displayed).
- ▶ Session data is lost (the user is asked to log on every time and shopping cart data lost).
- ▶ Slow application response, intermittent availability.

We also look at the following symptoms that an administrator would see:

- ▶ Web server will not start.
- ▶ Applications are not distributed properly to servers in clustered environments.

All of the Web servers with which the Web server plug-in is compatible allow external modules to be loaded by the Web server at runtime; the Web server

plug-in is one of these modules. For example, the IBM HTTP Server uses two directives in its configuration file to load the plug-in module and to refer to the plug-in configuration file.

In Example 7-1, the first directive loads the module and the second directive tells the plug-in module where to find the plug-in configuration file. The plug-in configuration file is an XML document that tells the Web server how to handle requests for URLs that are sent to the WebSphere Application Server.

*Example 7-1 IBM HTTP Server plug-in directives*

---

```
LoadModule was_ap20_module "C:\IBM\WAS6\Plugins\bin\mod_was_ap20_http.dll"  
WebSpherePluginConfig "C:\IBM\WAS6\Plugins\config\webserver1\plugin-cfg.xml"
```

---

The plug-in intercepts all requests that are received by the Web server . It compares the host name and port of the incoming URL to those that are defined in the VirtualHostGroup, and the requested URI to the list of URIs that are defined in the UriGroup. If a match is found in both the VirtualHostGroup and UriGroup, the plug-in then looks at the Route to determine what ServerCluster to use. When the Route is determined, the plug-in sends the request to the WebSphere Application Server that is defined in the ServerCluster. In Example 7-2, a request for `http://server/snoop` is directed to WebSphere Application Server. All other requests are passed back to the Web server to handle.

*Example 7-2 Excerpt from the plug-in config file*

---

```
<VirtualHostGroup Name="default_host">  
  <VirtualHost Name="*:80"/>  
</VirtualHostGroup>  
<UriGroup Name="default_host_cluster1_URIs">  
  <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"  
    Name="/snoop/*"/>  
</UriGroup>  
<Route ServerCluster="cluster1"  
  UriGroup="default_host_cluster1_URIs"  
  VirtualHostGroup="default_host"/>
```

---

The plug-in configuration file is managed via the WebSphere administrative console. Whenever you make a change that affects the plug-in, you need to regenerate the plug-in. Changes that affect the plug-in configuration include:

- ▶ Changing the virtual hosts definitions
- ▶ Changing the HTTP transport type or port
- ▶ Creating or deleting a server
- ▶ Installing or uninstalling an enterprise application

**Tip:** You can set your server to generate and propagate the plug-in automatically. Refer to the WebSphere Information Center for further details.

## Load balancing

The Web server plug-in can perform load balancing across multiple applications servers in a cluster.

Figure 7-2 shows a basic load balanced configuration involving three servers. The Web server plug-in is installed on the Web server machine and directs requests to each application server in turn, either on a round-robin or random basis. This is only one of many topologies that can be used for load balancing.

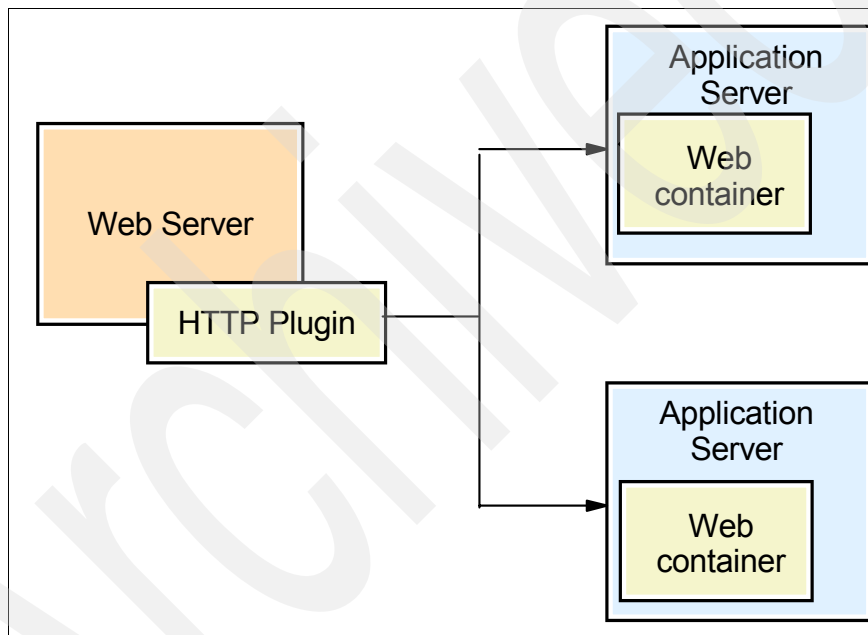


Figure 7-2 Load balancing

For more information, see *WebSphere Application Server V6 Scalability and Performance Handbook*, SG24-6392 at:

<http://publib-b.boulder.ibm.com/abstracts/sg246392.html>

In addition to defining the URLs that should be sent to the Web server plug-in, the plugin-cfg.xml file also describes the server clusters. Example 7-2 on page 253 shows the entries for the snoop URL and tells the plug-in that the snoop application should be handled by a UriGroup named default\_host\_cluster1\_URIs.

Example 7-3 highlights the route entry that shows applications from the UriGroup named default\_host\_cluster1\_URIs will be handled by a cluster named cluster1. The cluster named cluster1 has two separate servers. The first server is accessed via HTTP on port 9080 and HTTPS on port 9443, while the second server is accessed by HTTP on port 9081 and HTTPS on port 9444.

*Example 7-3 Clustering excerpts from the plug-in configuration file*

---

```
<Route ServerCluster="cluster1" UriGroup="default_host_cluster1_URIs"
  VirtualHostGroup="default_host"/>
...
<ServerCluster CloneSeparatorChange="false" LoadBalance="Round Robin" Name="cluster1"
PostSizeLimit="-1" RemoveSpecialHeaders="true" RetryInterval="60">
  <Server CloneID="10ig7jdv" ConnectTimeout="0" ExtendedHandshake="false"
LoadBalanceWeight="2" MaxConnections="-1" Name="k116571Node01_server1" ServerIOTimeout="0"
WaitForContinue="false">
    <Transport Hostname="k116571" Port="9080" Protocol="http"/>
    <Transport Hostname="k116571" Port="9443" Protocol="https">
      <Property Name="keyring" Value="C:\ibm\was6\plugins\etc\plugin-key.kdb"/>
      <Property Name="stashfile" Value="C:\ibm\was6\plugins\etc\plugin-key.sth"/>
    </Transport>
  </Server>
  <Server CloneID="10ig7jfq" ConnectTimeout="0" ExtendedHandshake="false"
LoadBalanceWeight="2" MaxConnections="-1" Name="k116571Node01_server2" ServerIOTimeout="0"
WaitForContinue="false">
    <Transport Hostname="k116571" Port="9081" Protocol="http"/>
    <Transport Hostname="k116571" Port="9444" Protocol="https">
      <Property Name="keyring" Value="c:\ibm\was6\plugins\etc\plugin-key.kdb"/>
      <Property Name="stashfile" Value="c:\ibm\was6\plugins\etc\plugin-key.sth"/>
    </Transport>
  </Server>
```

---

The result of this is that a request for the snoop application is directed to an application server, either to a server listening on port 9080 or another server listening on port 9081.

The plug-in monitors the requests that it sends to each server. If either server cannot be contacted, then that server is marked as down and taken out of the cluster until it is restored.

Load balancing is configured partly by settings under the Web server entry in the WebSphere administrative console and partly through updating the plugin-cfg.xml file directly. The WebSphere Information Center contains details on configuring the various settings, see *Clusters and workload management* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun\\_srvgrp.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun_srvgrp.html)

## Session affinity

A J2EE application supports the concept of sessions. A *session* is a way to maintain data about a user over subsequent requests. For example, a shopping cart application can maintain the list of items that you have selected to buy as session data.

Sessions are managed at the Web container level. However, the Web server plug-in has a role to play in maintaining *session affinity*. When a user connects to an application that uses sessions, such as a shopping cart application, the Web container starts a session and passes the user's session ID back to the browser, usually in a cookie.

You can maintain sessions across the servers in a cluster by a variety of methods (such as, by writing session data to a shared database after every request). If session affinity is used, the plug-in routes subsequent requests to the same application server that started the session. It does this by internally maintaining a list of servers and session IDs. Using session affinity provides the best application performance because you are eliminating the database write and read for each request. Session data is instead maintained in the application server's memory.

You do not have to do anything to the Web server plug-in to enable session affinity. If you have enabled session support in the application server and your plug-in configuration file contains the CloneID parameter as in the following example, then you have session affinity:

```
<Server CloneID="10ig7jdvd" ...>
```

The plug-in generation process adds the CloneID parameter by default. This is how the plug-in identifies each application server.

## 7.2 Work the problem

You begin the problem determination process by collecting the appropriate data that is required to diagnose the problem. We give you a list of all the documentation that might be required and how to collect it. If you have limited ability to recreate the problem, you might want to collect all the documentation at once, before starting the problem determination process. If you would rather only collect the relevant data as you move through the process, the data that you need is pointed out at each step.

Next, you go through a series of questions and actions that will help you define the high-level symptoms that you are experiencing. Each of these steps leads you to a more detailed procedure that is designed to take you through the



process of collecting and analyzing data to determine the most likely source of the problem.

And lastly, we provide guidance on the next step to take for resolution, whether it be a support site, contacting IBM, information about configuration, or some other suggestion as to how to proceed.

## 7.2.1 Collect the data

The logs and traces that will help you diagnose plug-in related problems include the following:

- ▶ Web server log files
- ▶ Web server plug-in log
- ▶ Plug-in trace (logged to the Web server plug-in log)
- ▶ WebSphere Application Server SystemOut and SystemErr logs
- ▶ WebSphere Application Server console messages
- ▶ Network protocol analysis (sometimes referred to as an iptrace)

If the problem is difficult to recreate or disruptive to business operations, see “The next step” on page 283 for a complete list of documentation to collect before continuing. In particular, you should review the MustGather documents for a complete list of documentation that is required by IBM support.

### Web server log files

The Web server plug-in supports five different Web servers running on various platforms. Most Web servers write two log files: an access log that contains details of all accesses to the Web server and an error log that contains details of any errors. The default location of the logs is as follows:

- ▶ IBM HTTP Server, Apache, and SunOne
  - Windows
    - Access log: <Web\_server\_home>\logs\access.log
    - Error log: <Web\_server\_home>\logs\error.log
  - UNIX
    - Access log: <Web\_server\_home>/logs/access\_log
    - Error log: <Web\_server\_home>/logs/error\_log
- ▶ Microsoft IIS
  - Windows
    - Access log: C:\WINNT\SYSTEM32\LogFiles\W3SVC1\<date>.log
    - Error log: Windows event log

► Domino Web server

The Domino Web server logs to a database. Refer to the Domino documentation for information about how to view these logs.

## Web server plug-in log

The plug-in also writes its own log, which can be found in the Web server plug-in install directory path. The log you are looking for is under another directory structure that includes the logical name of the Web server that you chose when you installed the plug-in.

You can find the location of the log file by first looking at the Web server configuration. This refers to the plug-in configuration file as shown in Example 7-1 on page 253. The plug-in configuration file then tells you where the log file is as shown in Example 7-4. This example also shows you where you set the amount of detail that is logged.

*Example 7-4 Location of plug-in log file*

---

```
<Log LogLevel="Error"
Name="c:\ibm\was6\plugins\logs\webserver1\http_plugin.log" />
```

---

The default LogLevel is Error, but you can set it to Trace to collect significantly more information. Should you need to raise this problem with IBM Support, they will request a plug-in trace.

## Plug-in trace

To get an effective trace, you need to enable as much logging as possible on the Web server. For example, you can set the logging level to debug to capture verbose output in the IBM HTTP Server by modifying the LogLevel directive in the configuration file as shown:

```
LogLevel debug
```

You need to restart the IBM HTTP Server for this change to take effect.

Enable trace logging in the Web server plug-in by setting the LogLevel directive in the plugin-cfg.xml file as shown:

```
<Log LogLevel="Trace"
Name="c:\ibm\was6\plugins\logs\webserver1\http_plugin.log" />
```

You do not need to restart the IBM HTTP Server for this change to take effect.

**Tip:** The plug-in trace generates significant amounts of data. Make your test as specific as possible and run it in isolation to reduce the number of lines that are generated.

## WebSphere Application Server logs

You also need to look at the application server log files to determine if your application is actually being called by the plug-in but failing at the application server. You need to check the log files for the application server where the application is running. The log files are:

```
<WAS_install_root>/profiles/<profile>/logs/<server>/SystemOut.log  
<WAS_install_root>/profiles/<profile>/logs/<server>/SystemErr.log
```

**Note:** Applications often do some logging of their own to application specific log files.

## Network trace

In rare cases, you might need to use a network protocol analyzer that will allow you to capture an iptrace. This might help you to determine where the problem lies. WebSphere Application Server does not supply such a tool, but there are third party tools available (for example, Ethereal that is available from <http://www.ethereal.com/>).

### 7.2.2 Analyze the high-level symptoms

Let's start working the problem by checking the following criteria in the order listed to define the high-level symptoms:

- ▶ You are not getting a response from the Web server.

Check to see if the Web server has started by looking for the process or by accessing the top level of the URL via a Web browser. For example:

```
http://localhost/
```

If you fail to get the Web server welcome page, check the operating system to see if the process for the Web server is active. If not, try starting the Web server manually.

If the Web server will not start, go to "Problem: Web server will not start" on page 261.

- ▶ The Web server has started, but you cannot access the application through it. For example, using the following URL to access the snoop servlet does not work:

```
http://Web_server/snoop
```

Check to see if the application can be accessed via the Web container directly. For example:

`http://Application_server:WC_port/snoop`

**To access the application directly through the Web container:**

1. Find the port for the Web container:
  - a. In the WebSphere administrative console, select **Servers** → **Application Servers**.
  - b. Click the server name.
  - c. Under the Communications section, expand Ports.
  - d. Use the port number listed for WC\_defaulthost.
2. Use the port number to access the resource from a browser.

For example, if the port is 9080, the URL is:

`http://localhost:9080/snoop`

If the application can be accessed directly through the Web container but not through the Web server, go to “Problem: Failure between the Web server and plug-in” on page 263.

- ▶ The application relies on sessions, but these sessions appear to be getting lost between requests.

For example, if you are prompted to log in every time you access the application or saved data such as the items in a shopping cart are being lost, the application might be losing session data.

If you are experiencing this problem, go to “Problem: Sessions are being lost” on page 272.

- ▶ The application works sometimes but fails at others, and you have a clustered environment.

For example, two out of every three requests works, but the third times out.

If you are experiencing this problem, go to “Problem: The application works intermittently” on page 277.

- ▶ The application load is not being evenly distributed in a clustered environment.

For example, you might be seeing 80% CPU on one server in a cluster and very low usage on the other servers.

If you are experiencing this problem, go to “Problem: Application load is not being evenly distributed” on page 280.

If you are having a problem with the Web server plug-in that is not covered in this list, see “The next step” on page 283).

## 7.3 Analyzing problem areas

Your analysis of the data you gathered will most likely lead you to one of the following problem areas. If not, see “The next step” on page 283.

### 7.3.1 Problem: Web server will not start

If you have recently installed the plug-in, or regenerated the plug-in configuration and the Web server will not start, the problem could be related to the plug-in.

#### Data to collect

The following logs are required to determine why the Web server will not start:

- ▶ Operating system messages
- ▶ Web server logs
- ▶ Web server plug-in log

You should collect these logs and copy them to a place where you can view them. This is important because the original logs might be overwritten during the debugging process.

In some instances, the error message that identifies the problem might not appear in any logs. In this case, you need to start the Web server process from the command line and observe the messages that are written to the console directly.

#### What to look for

Browse through the logs for messages that indicate why the Web server did not start. In the event that there are no failure messages, try running the executable to start the Web server directly to see the message.

Example 7-5 shows how you would see the message that is generated by the IBM HTTP Server when the plugin-cfg.xml file is missing.

*Example 7-5 Starting the HTTP Server from the command line*

---

```
C:\IBM\HTTP\bin>apache
ws_common: websphereUpdateConfig: Failed to stat plugin config file for
C:\IBM\WAS6\Plugins\config\webserver1\plugin-cfg.xml
```

---

In a UNIX environment, you normally see any relevant messages on the command line when trying to start the Web server. Example 7-6 shows the messages you see on UNIX if the plug-in module is in some way corrupt.

*Example 7-6 Messages indicating a corrupt plug-in module on UNIX*

---

```
Syntax error on line 844 of /opt/IBMIHS/conf/httpd.conf:
Cannot load /opt/IBM/WAS6/Plugins/bin/mod_was_ap20_http.so into server:
/opt/IBM/WAS6/Plugins/bin/mod_was_ap20_http.so: undefined symbol: ap_palloc
```

---

In Windows, you rarely see a useful error message when starting the Web server from the Windows services panel. You need to look for the log files that are written by the Web server or the plug-in. A message similar to that shown in Example 7-6 appears in the http\_plugin.log file in a Windows environment.

When the Web server plug-in stops the Web server from starting, the error messages are generally quite specific and accurate. When you have resolved the issue in the message, you will be able to start the Web server.

If the error message refers to a file, ensure that the file exists at the path shown in the message and that it is not corrupt. Try doing this as the user who runs the Web server to ensure that the problem is not related to file or directory permissions.

If the message refers to a problem with the plugin-cfg.xml file, you could try regenerating the plug-in from the WebSphere administrative console.

The plug-in is only tested with certain releases of the Web servers. If you have old or unsupported versions of the plug-in, Web server or GSKit, then you might need to upgrade that component. The following URL links to the software requirements page for WebSphere Application Server:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/welcome\\_base.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/welcome_base.html)

**Tip:** You can run the GSKit version command to get detailed version information. The command will be in the <gskit\_install>/bin directory.

For example:

```
C:\Program Files\IBM\gsk7\bin\gsk7ver
```

If you do not see any messages similar to those shown in the examples in this section, then it is possible that the plug-in is not causing your Web server to fail. In this case, you should review the symptoms that led you to this conclusion.

### 7.3.2 Problem: Failure between the Web server and plug-in

If you have determined that the Web server is not sending HTTP requests to the WebSphere Application Server and you suspect the plug-in to be the cause of the problem, this section can help you determine the cause.

The most obvious symptom is the ability to reach the application directly through the Web container but not through the Web server.

The Web server error log will also contain a message that indicates that the file could not be found, as follows:

```
[Tue Jun 28 15:54:43 2005] [error] [client 127.0.0.1] File does not exist:  
C:/IBM/HTTP/htdocs/en_US/snoop
```

#### Data to collect

The following logs are required to determine why your application is not responding:

- ▶ Web server logs
- ▶ Web server plug-in log
- ▶ Plug-in trace (advanced debugging)
- ▶ Network trace (for advanced debugging)
- ▶ Plug-in configuration file

You should collect these logs and copy them to a place where you can view them. This action is important because the original logs might be overwritten during the debugging process.

#### What to look for

The first thing to check is that the plugin-cfg.xml file that the Web server is referencing contains the entries for the application that is not working. The Web server configuration will have an entry that points to the plugin-cfg.xml file that it is using:

```
WebSpherePluginConfig /opt/WAS6/Plugins/config/web1/plugin-cfg.xml
```

Open this file either in a text editor or Web browser, and check to see if the context root of the application that you are testing appears in the file. An example of this is shown in Example 7-2 on page 253.

#### ***Verify you are using the correct configuration file***

If the application entry is missing, ensure that the configuration file the Web server is using is the same file that you generated. When you generate the Web server plug-in either through the WebSphere administrative console or the **GenPluginCfg.sh/bat** command, the output tells you the exact location of the

generated file. Figure 7-3 shows the location of the new plug-in file in the file system.

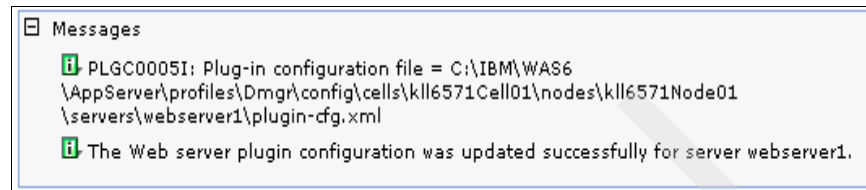


Figure 7-3 Location of generated plug-in file

You can compare this path to that in the Web server's configuration to ensure that they match:

WebSpherePluginConfig "C:\IBM\WAS6\Plugins\config\webserver1\plugin-cfg.xml"

In some configurations, the plug-in is propagated automatically to the correct locations on the local or remote host. The plug-in configuration file can only be propagated automatically if the following conditions are met:

- ▶ The remote Web server is IBM HTTP Server V6
- ▶ The plug-in configuration service is enabled
- ▶ Either
  - A WebSphere node agent must be running on the Web server machine
  - The IBM HTTP Server must be running on the Web server machine and an administrative password has been created
- ▶ The automatic propagation field must be checked as shown in Figure 7-4 on page 265



[Web servers](#) > [webserver1](#) > **Plug-in properties**

Configure Web server plug-in properties. The plug-in is used to pass HTTP requests from a Web server to WebSphere Application Servers.

**Runtime** Configuration

---

### Plug-in properties

- \* Plug-in installation location  
c:\ibm\was6\plugins
- \* Plug-in configuration file name  
plugin-cfg.xml [View](#)
- ☒ Automatically generate the plug-in configuration file
- ☒ Automatically propagate plugin configuration file
- ☐ Ignore DNS failures during webserver startup
- Refresh configuration interval  
60 seconds

**Plug-in logging:**

- \* Log file name  
c:\ibm\was6\plugins\logs\webserver1\ht
- Log level  
Error

### Additional Properties

- [Request and Response](#)
- [Caching](#)
- [Request Routing](#)
- [Custom Properties](#)

[Apply](#) [OK](#) [Reset](#) [Cancel](#)

Figure 7-4 Plug-in configuration properties

For more information about automatic propagation of the plug-in, see *Web server plug-in properties* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/uwsv\\_plugin\\_props.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/uwsv_plugin_props.html)

If the correct configuration file is in use, the next step is to try regenerating the Web server plug-in.

### **Verify that the plug-in configuration file is generated properly**

If you have tried regenerating the plug-in, but the application's context root still does not appear in the generated plugin-cfg.xml file, you could be generating the configuration file incorrectly. The configuration included in a generated plugin-cfg.xml file depends on the method that you use to generate the file.

If you generate the Web server plug-in using the **GenPluginCfg.sh/bat** command line tool without parameters, every application and application server in the topology is generated into the configuration file. However, you can narrow the topology that is included in the configuration by using parameters. For example, if you generate the Web server plug-in using **GenPluginCfg** and specify a Web server name, or generate the plug-in through the administrative console, only those applications and application servers that are mapped to the Web server are generated into the configuration file.

The problem might be simply that you have not mapped the application or application server to the Web server. Use the administrative console to perform this mapping.

### ***Check for Virtual Host and WebGroup not found error***

If the Web server plug-in configuration appears to be correct, but the application is still not working, determine a Virtual Host / WebGroup not found message is being returned to the browser. If so, this error is covered in *WebSphere Application Server V6: Web Container Problem Determination* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4058.pdf>

### **Further analysis: Trace the request**

At this point, you have confirmed that everything is configured correctly and as far as you can tell, the Web server plug-in should be sending your request to a WebSphere Application Server to process. You have verified that the plug-in configuration is correct and that the Web server is accepting the request. However, you are not getting any response from the application.

The next step is to trace a particular HTTP request through the topology to determine where the request is failing. You have to take a plug-in trace (see “Plug-in trace” on page 258).

After you have enabled the trace, send an HTTP request through the system, and check each log in turn to see where the transaction fails. Figure 7-5 on page 267 shows the path a HTTP request takes.

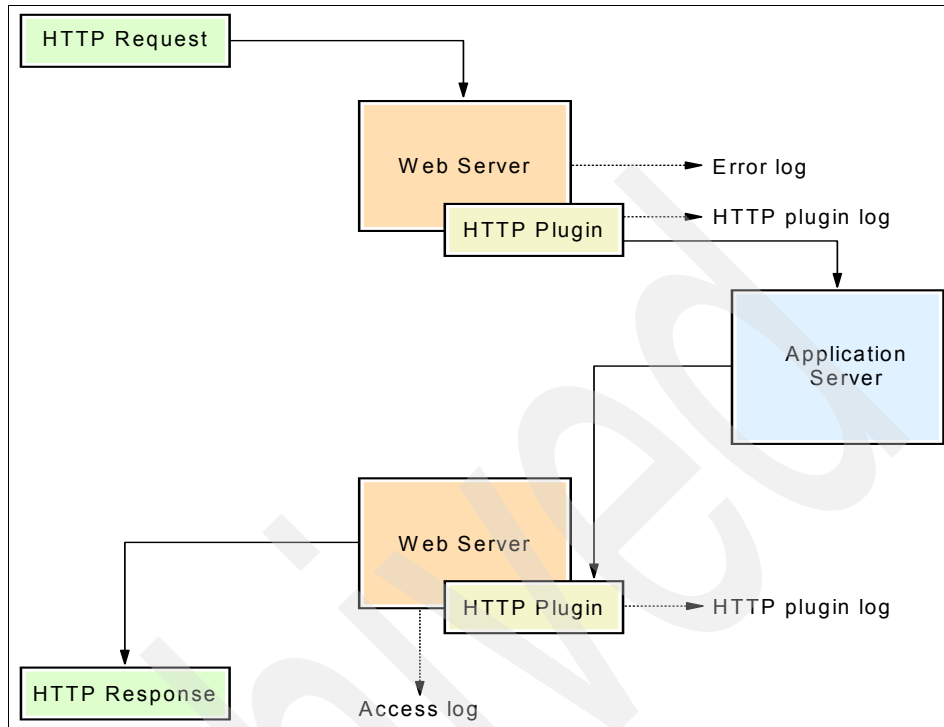


Figure 7-5 Following an HTTP request

Following a given request through that path shows you where the problems lie. Example 7-9 on page 268 shows the log entries that are generated by a successful request to the snoop servlet. These log entries show the successful transition of the request through each component.

The first entries written to http\_plugin.log file show the plug-in building the structures it uses to parse URLs. If all is well, you can see it building the structures for your URL. Example 7-7 shows excerpts from the trace log for the snoop servlet.

Example 7-7 Plug-in trace initial entries

```

[Thu Jun 23 14:35:46 2005] 0000097c 00000ae4 - TRACE: ws_uri: uriCreate:
Creating uri
[Thu Jun 23 14:35:46 2005] 0000097c 00000ae4 - TRACE: ws_uri: uriSetName:
Setting the name /snoop/* with score 7
[Thu Jun 23 14:35:46 2005] 0000097c 00000ae4 - TRACE: ws_uri:
uriSetAffinityURL: Setting the affinity cookie jsessionid
[Thu Jun 23 14:35:46 2005] 0000097c 00000ae4 - TRACE: ws_uri:
uriSetAffinityCookie: Setting the affinity cookie JSESSIONID
  
```

```
[Thu Jun 23 14:35:46 2005] 0000097c 00000ae4 - TRACE: ws_uri_group:  
uriGroupAddUri: Adding uri /snoop/* to front of list
```

---

After the plug-in has parsed the configuration file and is successfully initialized, it writes entries that include the build version and other data about the plug-in, as shown in Example 7-8.

*Example 7-8 Plug-in initialization messages*

---

```
-----System Information-----  
[Thu Jun 23 14:35:46 2005] 0000097c 00000ae4 - PLUGIN: Bld version: 6.0.0  
[Thu Jun 23 14:35:46 2005] 0000097c 00000ae4 - PLUGIN: Bld date: Oct 31 2004,  
11:15:26  
[Thu Jun 23 14:35:46 2005] 0000097c 00000ae4 - PLUGIN: Webserver:  
IBM_HTTP_Server/6.0 Apache/2.0.47 (Win32)  
[Thu Jun 23 14:35:46 2005] 0000097c 00000ae4 - PLUGIN: Hostname = KLL6571  
[Thu Jun 23 14:35:46 2005] 0000097c 00000ae4 - PLUGIN: OS version 5.0, build  
2195, 'Service Pack 4'  
[Thu Jun 23 14:35:46 2005] 0000097c 00000ae4 - PLUGIN:  
-----
```

---

The entries in Example 7-9 show the successful resolution of the URL that was requested by the plug-in. It shows the processing of the request through the selection of a server from the cluster through setting up the transport. It also shows that the request has been forwarded to an application server. Shortly after that, it shows the response coming back from the application server with a HTTP return code of 200, that is, successful. The time stamps show that the application server took seven seconds to process the request.

*Example 7-9 Web server plug-in request trace entries for snoop*

---

```
[Thu Jun 23 14:35:56 2005] 000007bc 000007d4 - TRACE: ws_common:  
websphereShouldHandleRequest: trying to match a route for: vhost='localhost';  
uri='/snoop'  
...  
[Thu Jun 23 14:35:56 2005] 000007bc 000007d4 - TRACE: ws_common:  
websphereUriMatch: Found a match '/snoop' to '/snoop' in UriGroup:  
default_host_cluster1_URIs with score 6  
...  
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: ws_common:  
websphereExecute: Executing the transaction with the app server  
...  
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: lib_htrrequest:  
htrrequestWrite: Writing the request:  
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: GET /snoop HTTP/1.1  
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: User-Agent: Wget/1.9  
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: Host: localhost  
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: Accept: */*
```

```

[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: Connection: Keep-Alive
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: $WSIS: false
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: $WSSC: http
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: $WSPR: HTTP/1.0
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: $WSRA: 127.0.0.1
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: $WSRH: 127.0.0.1
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: $WSSN: localhost
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: $WSSP: 80
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: Surrogate-Capability:
WS-ESI="ESI/1.0+"
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: lib_htrequest:
htrequestWrite: Writing the request content
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: ws_common:
websphereExecute: Wrote the request; reading the response
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: lib_htresponse:
htresponseRead: Reading the response: 5397bc
...
[Thu Jun 23 14:35:57 2005] 000007bc 000007d4 - TRACE: lib_htresponse:
htresponseRead: Reading the response: 5397bc
[Thu Jun 23 14:36:04 2005] 000007bc 000007d4 - TRACE: HTTP/1.1 200 OK
[Thu Jun 23 14:36:04 2005] 000007bc 000007d4 - TRACE: Content-Type:
text/html; charset=ISO-8859-1
[Thu Jun 23 14:36:04 2005] 000007bc 000007d4 - TRACE: Content-Language:
en-US
[Thu Jun 23 14:36:04 2005] 000007bc 000007d4 - TRACE: Content-Length: 16166
[Thu Jun 23 14:36:04 2005] 000007bc 000007d4 - TRACE: lib_htresponse:
htresponseSetContentLength: Setting the content length |16166|

```

---

Immediately after this, the Web server sends the request back to the browser and then updates the HTTP access log to show that the transaction completed successfully, as shown in Example 7-10.

*Example 7-10 Web server access log entry for snoop*

---

```
127.0.0.1 - - [23/Jun/2005:14:35:56 -0400] "GET /snoop HTTP/1.0" 200 16166
```

---

The Web server writes to the access log at the end of processing.

Following your failing HTTP request through these log files shows you where the problem lies.

Example 7-11 on page 270 shows the messages that are written by the plug-in when it marks a server down. These messages appear even when you are not tracing requests. The message includes the error that is returned by the operating system, that is err=10061, which means connection refused. You should check your operating system documentation for a list of these error codes.

#### *Example 7-11 Plug-in messages when a server goes down*

---

```
[Thu Jul 07 13:53:20 2005] 00000d6c 000010b8 - ERROR: ws_common:
websphereGetStream: Failed to connect to app server on host 'k116571', OS
err=10061
[Thu Jul 07 13:53:20 2005] 00000d6c 000010b8 - ERROR: ws_common:
websphereExecute: Failed to create the stream
[Thu Jul 07 13:53:20 2005] 00000d6c 000010b8 - ERROR: ws_server:
serverSetFailoverStatus: Marking k116571Node01_server01 down
[Thu Jul 07 13:53:20 2005] 00000d6c 000010b8 - ERROR: ws_common:
websphereHandleRequest: Failed to execute the transaction to
'k116571Node01_server01' on host 'k116571'; will try another one
```

---

If the Web server plug-in trace shows it building the request and sending it to an application server, but does not show a reply coming back, then the problem is likely to be in the network between the Web server and the application server.

Example 7-12 shows the plug-in writing the request to a server and then waiting on the response. In this instance, the plug-in has been configured so that it times out waiting for a response after 10 seconds. The example shows the plug-in timing out the read and marking the server down.

#### *Example 7-12 Marking a server down after time out*

---

```
[Thu Jul 07 15:09:48 2005] 00000ed4 00000ffc - TRACE: lib_htrequest:
htrequestWrite: Writing the request content
[Thu Jul 07 15:09:48 2005] 00000ed4 00000ffc - TRACE: ws_common:
websphereExecute: Wrote the request; reading the response
[Thu Jul 07 15:09:48 2005] 00000ed4 00000ffc - TRACE: lib_htresponse:
htresponseRead: Reading the response: 4cf1504
[Thu Jul 07 15:09:58 2005] 00000ed4 00000ffc - TRACE: lib_htresponse:
htresponseSetError: Setting the error |1|
[Thu Jul 07 15:09:58 2005] 00000ed4 00000ffc - ERROR: ws_common:
websphereExecute: Failed to read from a new stream; App Server may have gone
down during read
```

---

In some instances, the plug-in does not detect that a server has gone down or hung after successfully establishing a connection and writing a request. This circumstance generally occurs when one server is Windows-based and the other is UNIX-based.

Example 7-13 shows a request that responds normally. The plug-in writes the request and then waits for the response. The response arrives seven seconds later.

*Example 7-13 A request that responds*

---

```
[Thu Jul 07 14:27:00 2005] 00000a20 00000f80 - TRACE: lib_htrequest:
htrequestWrite: Writing the request content
[Thu Jul 07 14:27:00 2005] 00000a20 00000f80 - TRACE: ws_common:
websphereExecute: Wrote the request; reading the response
[Thu Jul 07 14:27:00 2005] 00000a20 00000f80 - TRACE: lib_htresponse:
htresponseRead: Reading the response: 5397cc
[Thu Jul 07 14:27:07 2005] 00000a20 00000f80 - TRACE:      HTTP/1.1 200 OK
```

---

Example 7-14 shows where the plug-in writes the request and then waits for the response. Almost five minutes later, another request arrives and the plug-in begins parsing it. There was no response from the original request.

*Example 7-14 A request that never responds*

---

```
[Thu Jul 07 14:22:20 2005] 00000a20 00000f88 - TRACE: ws_common:
websphereExecute: Wrote the request; reading the response
[Thu Jul 07 14:22:20 2005] 00000a20 00000f88 - TRACE: lib_htresponse:
htresponseRead: Reading the response: 4cf1504
[Thu Jul 07 14:27:00 2005] 00000a20 00000f80 - TRACE: lib_util:
parseHostHeader: Host: 'localhost', port 80
```

---

At this point, you could use a network protocol analyzer to determine what is happening to the network packets after they leave the Web server. This is also called an *iptrace*. Network protocol analyzers are quite technical tools and generate a lot of output because there is a lot of network traffic, even on a seemingly idle network. If you are not confident about capturing and interpreting this output, you should consider engaging a network specialist to assist you.

Most problems that are perceived to be the fault of the plug-in are in fact configuration problems or network problems. Aside from the load balancing features, the plug-in is quite a simple component that simply takes a request, parses it for a match and then forwards matching requests to an application server.

If you have been through these diagnostic steps and not found the source of your problem, it is time to contact IBM Support (see “The next step” on page 283).

### 7.3.3 Problem: Sessions are being lost

If session data appears to be getting lost and you have a clustered environment, the plug-in is a potential source of the problem.

#### Data to collect

You should collect the following data:

- ▶ Plug-in trace
- ▶ Web server plug-in log

You should collect the log and copy it to a place where you can view it. This is important because the original log might be overwritten during the debugging process.

#### What to look for

The Web server plug-in trace shows you the processing of session affinity in detail. The sample applications that are provided with WebSphere Application Server allow you to test session affinity. The following URL displays a simple JSP demonstrating that WebSphere Application Server is alive and sets up a session:

`http://servername/HelloHTML.jsp`

The excerpt from the plug-in trace shown in Example 7-15 shows the session that is created. It shows the plug-in looking for the various cookies that it can use for session management and then using a round-robin load balancing algorithm because no session could be found.

#### *Example 7-15 Creating a session*

---

```
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: ws_common:
websphereWriteRequestReadResponse: Enter
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: ws_common:
websphereHandleSessionAffinity: Checking for session affinity
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: ws_common:
websphereHandleSessionAffinity: Checking the SSL session id
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: lib_htrequest:
htrequestGetCookieValue: Looking for cookie: 'SSLJSESSION'
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: lib_htrequest:
htrequestGetCookieValue: No cookie found for: 'SSLJSESSION'
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: ws_common:
websphereHandleSessionAffinity: Checking the cookie affinity: JSESSIONID
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: lib_htrequest:
htrequestGetCookieValue: Looking for cookie: 'JSESSIONID'
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: lib_htrequest:
htrequestGetCookieValue: No cookie found for: 'JSESSIONID'
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: ws_common:
websphereHandleSessionAffinity: Checking the url rewrite affinity: jsessionid
```



```

[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: ws_common:
websphereParseSessionID: Parsing session id from '/HelloHTML.jsp'
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: ws_common:
websphereParseSessionID: Failed to parse session id
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: ws_common:
websphereHandleSessionAffinity: Bypassing check for partitionID cookie
affinity. No stored partition table.
[Mon Jun 27 14:48:33 2005] 00000798 00000e00 - TRACE: ws_server_group:
serverGroupNextRoundRobinServer: Round Robin load balancing
...
[Mon Jun 27 14:51:36 2005] 00000798 00000e18 - TRACE: ws_server_group:
serverGroupIncrementConnectionCount: Server k1l6571Node01_server2 picked,
pendingConnectionCount 1 totalConnectionsCount 5.

```

---

Example 7-16 shows a second request that is processed. This time, there is a session ID, and so the request is routed to the same server as that in Example 7-15 on page 272.

#### *Example 7-16 Processing session affinity*

```

[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_common:
websphereHandleSessionAffinity: Checking for session affinity
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_common:
websphereHandleSessionAffinity: Checking the SSL session id
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: lib_htrequest:
htrequestGetCookieValue: Looking for cookie: 'SSLJSESSION'
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: lib_htrequest:
htrequestGetCookieValue: No cookie found for: 'SSLJSESSION'
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_common:
websphereHandleSessionAffinity: Checking the cookie affinity: JSESSIONID
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: lib_htrequest:
htrequestGetCookieValue: Looking for cookie: 'JSESSIONID'
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: lib_htrequest:
htrequestGetCookieValue: name='JSESSIONID',
value='00004RRFkLCKLGWVw-37Cd-mEN7:10ig7jfq'
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_common:
websphereParseCloneID: Parsing clone ids from
'00004RRFkLCKLGWVw-37Cd-mEN7:10ig7jfq'
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_common:
websphereParseCloneID: Adding clone id '10ig7jfq'
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_common:
websphereParseCloneID: Returning list of clone ids
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_server_group:
serverGroupFindClone: Looking for clone
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_server_group:
serverGroupGetFirstPrimaryServer: getting the first primary server
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_server_group:
serverGroupFindClone: Comparing curCloneID '10ig7jfq' to server clone id
'10ig7jdvd'

```

```

[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_server_group:
serverGroupGetNextPrimaryServer: getting the next primary server
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_server_group:
serverGroupFindClone: Comparing curCloneID '10ig7jfqi' to server clone id
'10ig7jfqi'
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_server_group:
serverGroupFindClone: Match for clone 'k116571Node01_server2'
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_server:
serverHasReachedMaxConnections: currentConnectionsCount 0, maxConnectionsCount
-1.
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - STATS: ws_server_group:
serverGroupCheckServerStatus: Checking status of k116571Node01_server2,
ignoreWeights 1, markedDown 0, retryNow 0, wlbAllows -1
reachedMaxConnectionsLimit 0
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_server:
serverHasReachedMaxConnections: currentConnectionsCount 0, maxConnectionsCount
-1.
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_server_group:
serverGroupIncrementConnectionCount: Server k116571Node01_server2 picked,
pendingConnectionCount 1 totalConnectionsCount 7.
[Mon Jun 27 14:51:51 2005] 00000798 00000e00 - TRACE: ws_common:
websphereHandleSessionAffinity: Setting server to k116571Node01_server2

```

---

Reviewing a plug-in trace log shows whether session affinity is working correctly and whether the plug-in is routing sessions correctly to the servers.

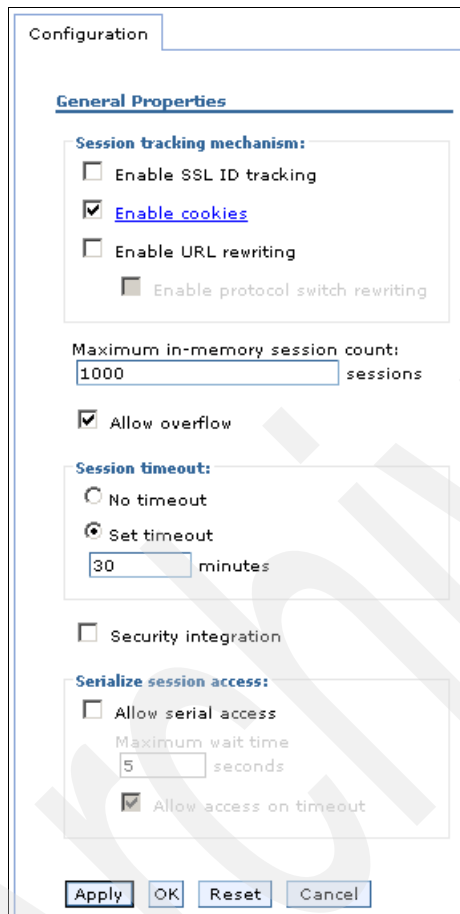
If it appears that session affinity is not working, check the WebSphere Application Server configuration to ensure that the session management parameters are set correctly. Also, check the plugin-cfg.xml file to ensure that the CloneID parameter is set.

Figure 7-4 on page 265 shows the administrative console page for the WebSphere Application Server session configuration parameters. You get to this page by choosing **Application servers** → **servername** → **Web container settings** → **Session management**.

There are three different session tracking mechanisms:

- ▶ Enable SSL ID tracking, which can only be used when securing your applications with SSL.
- ▶ Enable cookies, which is the most common and generally accepted method of tracking sessions.
- ▶ Enable URL rewriting, which is only rarely used and has a high performance impact because every URL must be modified to include the session ID.

To enable sessions, you choose one or more of these session tracking mechanisms. The other parameters are for setting how long your sessions will last and other session management purposes.



The image shows a 'Configuration' dialog box for WebSphere Application Server session settings. It has a 'Configuration' tab at the top. Below it is the 'General Properties' section. Under 'Session tracking mechanism:', there are four checkboxes: 'Enable SSL ID tracking' (unchecked), 'Enable cookies' (checked), 'Enable URL rewriting' (unchecked), and 'Enable protocol switch rewriting' (disabled). Below this is a text field for 'Maximum in-memory session count:' set to '1000' sessions, and a checked 'Allow overflow' checkbox. The 'Session timeout:' section has two radio buttons: 'No timeout' (unchecked) and 'Set timeout' (checked), with a text field for '30' minutes. Below that is an unchecked 'Security integration' checkbox. The 'Serialize session access:' section has an unchecked 'Allow serial access' checkbox, a 'Maximum wait time' text field set to '5' seconds, and a checked 'Allow access on timeout' checkbox. At the bottom are 'Apply', 'OK', 'Reset', and 'Cancel' buttons.

Figure 7-6 WebSphere Application Server session configuration

You will only see errors or problems related to sessions with a plug-in trace.

Example 7-17 on page 276 shows a similar sequence of processing a request to that shown in Example 7-16 on page 273 except that the chosen server is down.

The plug-in then chooses another server in the cluster but the session on the server that is down is lost.

*Example 7-17 Session lost after application server goes down*

---

```
[Thu Jul 07 15:55:59 2005] 00000bcc 000010c0 - TRACE: ws_server_group:
serverGroupFindClone: Match for clone 'm23vnx60Craig01_server02'
...
[Thu Jul 07 15:55:59 2005] 00000bcc 000010c0 - TRACE: ws_server_group:
lockedServerGroupUseServer: Server m23vnx60Craig01_server02 picked, weight 0.
[Thu Jul 07 15:55:59 2005] 00000bcc 000010c0 - TRACE: ws_common:
websphereFindTransport: Finding the transport
[Thu Jul 07 15:55:59 2005] 00000bcc 000010c0 - TRACE: ws_common:
websphereFindTransport: Setting the transport(case 2): m23vnx60 on port 19082
[Thu Jul 07 15:55:59 2005] 00000bcc 000010c0 - TRACE: ws_common:
websphereExecute: Executing the transaction with the app server
[Thu Jul 07 15:55:59 2005] 00000bcc 000010c0 - TRACE: ws_common:
websphereGetStream: Getting the stream to the app server
[Thu Jul 07 15:55:59 2005] 00000bcc 000010c0 - TRACE: ws_transport:
transportStreamDequeue: Checking for existing stream from the queue
[Thu Jul 07 15:56:00 2005] 00000bcc 000010c0 - ERROR: ws_common:
websphereGetStream: Failed to connect to app server on host 'm23vnx60', OS
err=10061
[Thu Jul 07 15:56:00 2005] 00000bcc 000010c0 - TRACE: ws_common:
websphereGetStream: socket 6628 closed - failed to connect
[Thu Jul 07 15:56:00 2005] 00000bcc 000010c0 - ERROR: ws_common:
websphereExecute: Failed to create the stream
[Thu Jul 07 15:56:00 2005] 00000bcc 000010c0 - ERROR: ws_server:
serverSetFailoverStatus: Marking m23vnx60Craig01_server02 down
[Thu Jul 07 15:56:00 2005] 00000bcc 000010c0 - STATS: ws_server:
serverSetFailoverStatus: Server m23vnx60Craig01_server02 : pendingConnections 0
failedConnections 1 affinityConnections 1 totalConnections 0.
[Thu Jul 07 15:56:00 2005] 00000bcc 000010c0 - ERROR: ws_common:
websphereHandleRequest: Failed to execute the transaction to
'm23vnx60Craig01_server02' on host 'm23vnx60'; will try another one
...
[Thu Jul 07 15:56:00 2005] 00000bcc 000010c0 - TRACE: ws_server_group:
lockedServerGroupUseServer: Server k1l6571Node01_server01 picked, weight 0.
[Thu Jul 07 15:56:00 2005] 00000bcc 000010c0 - TRACE: ws_common:
websphereFindTransport: Finding the transport
[Thu Jul 07 15:56:00 2005] 00000bcc 000010c0 - TRACE: ws_common:
websphereFindTransport: Setting the transport(case 2): k1l6571 on port 9082
```

---

If session affinity appears to be working from the plug-in's point of view, then it is possible that the application is at fault. Review the application with the developers to ensure that they are handling sessions correctly.

Example 7-18 shows the entries from the plug-in trace when the application requires session affinity via cookies but the browser does not allow cookies. The plug-in is not able to maintain session affinity as it has no way of saving the session ID.

*Example 7-18 Session failure as cookies are not allowed*

---

```
[Thu Jul 07 15:59:04 2005] 00000978 00000ff4 - TRACE: ws_common:
websphereHandleSessionAffinity: Checking for session affinity
[Thu Jul 07 15:59:04 2005] 00000978 00000ff4 - TRACE: ws_common:
websphereHandleSessionAffinity: Checking the SSL session id
[Thu Jul 07 15:59:04 2005] 00000978 00000ff4 - TRACE: lib_htrequest:
htrequestGetCookieValue: Looking for cookie: 'SSLJSESSION'
[Thu Jul 07 15:59:04 2005] 00000978 00000ff4 - TRACE: lib_htrequest:
htrequestGetCookieValue: No cookie found for: 'SSLJSESSION'
[Thu Jul 07 15:59:04 2005] 00000978 00000ff4 - TRACE: ws_common:
websphereHandleSessionAffinity: Checking the cookie affinity: JSESSIONID
[Thu Jul 07 15:59:04 2005] 00000978 00000ff4 - TRACE: lib_htrequest:
htrequestGetCookieValue: Looking for cookie: 'JSESSIONID'
[Thu Jul 07 15:59:04 2005] 00000978 00000ff4 - TRACE: lib_htrequest:
htrequestGetCookieValue: No cookie found for: 'JSESSIONID'
[Thu Jul 07 15:59:04 2005] 00000978 00000ff4 - TRACE: ws_common:
websphereHandleSessionAffinity: Checking the url rewrite affinity: jsessionid
[Thu Jul 07 15:59:04 2005] 00000978 00000ff4 - TRACE: ws_common:
websphereParseSessionID: Parsing session id from '/HelloHTML.jsp'
[Thu Jul 07 15:59:04 2005] 00000978 00000ff4 - TRACE: ws_common:
websphereParseSessionID: Failed to parse session id
```

---

The WebSphere Information Center contains information about managing sessions in WebSphere Application Server clusters. In particular, see *Clusters and workload management* at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun\\_srvgrp.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/crun_srvgrp.html)

### 7.3.4 Problem: The application works intermittently

In the event that users are having regular but inconsistent failures accessing the application and you have a clustered environment, you might have a plug-in problem.

#### Data to collect

The following logs are required to determine why your application is not responding:

- Web server logs

- ▶ Web server plug-in log
- ▶ WebSphere Application Server logs

If you cannot identify the cause of the problem from these logs, you need to take a plug-in trace. Trace entries are also written to the Web server plug-in log.

You should collect these logs and copy them to a place where you can view them. This action is important because the original logs might be overwritten during the debugging process.

## What to look for

Intermittent failures in applications can be caused by the plug-in not detecting that a WebSphere Application Server has failed and thus continuing to route requests to that application server.

If the application server is down, the plug-in will not be able to establish a TCP/IP connection and will not be able to send the request. Under these conditions, the plug-in marks the server as down.

If the application server is hung, it can still accept requests but cannot respond to them. Under these conditions, the application server is not marked down until the operating system times out the connection. Depending on the operating system, this action could take up to 10 minutes. At some point after the plug-in has marked the server down, it checks again and might well find that the server is accepting requests again. In this case, the server is marked as up, at least until the next time out.

If the application server is not hung but running slowly, to the point where all TCP/IP connections have been used up in the application server, the plug-in is not able to establish a connection and marks the server as down.

A failure in an application will not be detected by the plug-in because the application server is accepting requests and responding to them. This is the way the plug-in is designed to work.

**Note:** With all of these conditions, you would not get a response when trying to access the application directly on the server, for example:

`http://servername:9080/snoop`

In a clustered environment, you need to try and access each server in the cluster in turn to determine which is hung. For further information about dealing with application server hangs, refer to *Approach to Problem Determination in WebSphere Application Server V6* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

The cause of the problems that you are experiencing in this section is related to WebSphere Application Server not responding as you expect and so you should be resolving that issue. However, you can minimize the impact of a problem in WebSphere Application Server by tuning the parameters that control how the plug-in responds to time-out conditions. You can use the following parameters to tune the plug-in:

- ▶ **RetryInterval**

The `RetryInterval` parameter sets how long the plug-in waits to check a server for availability after marking that server down.

- ▶ **ServerIOTimeout**

The `ServerIOTimeout` attribute of a server element enables the plug-in to set a time out value, in seconds, for sending requests to and reading responses from the application server. If a value is not set for the `ServerIOTimeout`, the plug-in uses blocked I/O to write the request to and read the response from the application server until the TCP connection times out. For example:

```
<Server Name="server1" ServerIOTimeout=300>
```

In this case, the plug-in waits 300 seconds (5 minutes) before timing out the TCP connection. Setting the `ServerIOTimeout` attribute to a reasonable value enables the plug-in to time out the connection sooner and transfers requests to another application server when possible.

When selecting a value for this attribute, remember that sometimes it might take a couple of minutes for an application server to process a request. Setting the value of the `ServerIOTimeout` attribute too low could cause the plug-in to send a false server error response to the client.

**Note:** The `ServerIOTimeout` attribute is ignored for a plug-in that is running on a Solaris platform.

- ▶ **ExtendedHandshake**

You should use the `ExtendedHandshake` parameter when there is a proxy firewall between the Web server plug-in and an application server. It forces the plug-in to do more extensive checking to determine if an application server should be marked as down.

- ▶ **MaxConnections**

The maximum number of concurrent connections that can be sent to an application server can be used to stop the plug-in from sending requests to a server that might be running slow.

► ConnectTimeout

The ConnectTimeout attribute of a Server element enables the plug-in to perform non-blocking connections with the application server. Non-blocking connections are beneficial when the plug-in is unable to contact the destination to determine if the port is available or unavailable.

If no ConnectTimeout value is specified, the plug-in performs a blocking connect in which the plug-in sits until an operating system times out (as long as two minutes depending on the platform) and allows the plug-in to mark the server unavailable. A value of zero (0) causes the plug-in to perform a blocking connect. A value greater than zero (0) specifies the number of seconds you want the plug-in to wait for a successful connection. If a connection does not occur after that time interval, the plug-in marks the server down and fails over to one of the other servers defined in the cluster.

Where possible, use the administrative console to set these properties for a given Web server definition. Any manual changes that you make to the plug-in configuration file for a given Web server are overridden whenever the file is regenerated. You can update the RetryInterval parameter in the administrative console by navigating to **Web servers** → **webserver** → **Plug-in properties** → **Request routing** and set the RetryInterval as shown in Figure 7-7.

A screenshot of a web form titled "Retry interval". It features a text input field containing the number "60" and a label "seconds" to its right. The entire form is enclosed in a thin black border.

Figure 7-7 Setting the RetryInterval parameter

All the other parameters must be set manually in the plugin-cfg.xml file. Thus, every time you regenerate the plug-in, you must edit the generated file to add these parameters.

For details on these and other plug-in properties, see *plugin-cfg.xml* file in the WebSphere Information Center at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/rwsv\\_plugincfg.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/rwsv_plugincfg.html)

### 7.3.5 Problem: Application load is not being evenly distributed

Uneven load distribution across servers in a cluster might be attributed to a plug-in problem.

#### Data to collect

Collect the following data:



- ▶ Plug-in trace
- ▶ Web server plug-in log

The plug-in trace data is written to the Web server plug-in log. You should collect the log and copy it to a place where you can view it. This action is important because the original log might be overwritten during the debugging process.

You need to analyze the trace to determine how many of each request was sent to each application server in the cluster.

## What to look for

Each time the plug-in chooses an application server to which it sends a request, it writes a message in the trace log similar to the following:

```
TRACE: ws_server_group: serverGroupIncrementConnectionCount: Server servername
picked, pendingConnectionCount count totalConnectionsCount count.
```

You can manually count the number of times that the request is sent to each server in the cluster. Alternatively, you might write a simple script to parse this log and extract the data for you. This data shows exactly how many requests are being routed to each server in the cluster. If the distribution is not even, you should check the relative weight of each server in the cluster as defined in the plugin-cfg.xml file by the LoadBalanceWeight parameter:

```
<Server CloneID=cloneid ConnectTimeout=0 ExtendedHandshake=false
LoadBalanceWeight=3 MaxConnections=-1 Name=servername ServerIOTimeout=0
WaitForContinue=false>
```

Whenever the plug-in sends a request to an application server, it decrements a counter that it maintains for each server in a cluster. It starts with each server's LoadBalanceWeight. When a server's counter reaches zero, the plug-in stops sending requests to that server. When all server's counters reach zero, it resets them back to their starting value and continues. The default value for LoadBalanceWeight is 2.

For example, consider 2 servers (server1 and server2). Server1 has a LoadBalanceWeight of 2 and server2 has a LoadBalanceWeight of 4. Table 7-1 shows how the requests are sent to each server.

Table 7-1 Effect of differing load balancer weights

Request #	Sent to	server1 counter	server2 counter
-	-	2	4
1	server1	1	4
2	server2	1	3

Request #	Sent to	server1 counter	server2 counter
3	server1	0	3
4	server2	0	2
5	server2	0	1
6	server2	0	0
	<b>counters reset:</b>	2	4
7	server1	1	4

The ratio of requests that are sent to servers is simply the ratio of the relative weights. So, in Table 7-1, two requests were sent to server1 and four requests were sent to server2, for a ratio of 1:2. This is equivalent to the relative weights, 2:4.

Compare the number of requests that are sent to each server in the cluster to the ratio of the load balance weight settings.

Other parameters might also influence load balancing. For example, if your applications are running slowly on one application server, then that server might reach its MaxConnections limit. So, the plug-in stops sending requests to it. This is clearly marked in the plug-in trace.

Session affinity, as discussed in “Problem: Sessions are being lost” on page 272, could also be contributing to uneven load balancing. Again, you can see messages that the plug-in is choosing a particular server to maintain session affinity.

If the number of requests that are sent to each server matches the ratio of the load balance weight settings, then plug-in load balancing is working correctly. The uneven CPU usage must be caused by some other factor, such as a rogue process on the server or a problem in the application.

If they do not match, then some other factor might be influencing the load balancing algorithm. For example, you might be reaching MaxConnections on one server, or a server might be marked down by the plug-in.

If you need more intelligent routing based on application server load or response time, you need to use another load balancing product, such as WebSphere Application Server Edge Component.

## 7.4 The next step

The symptoms and problem areas included in this paper are some that you are more likely to experience. However, there are other things that can go wrong, or the cause of the problem might be related to a component other than the Web server plug-in.

If, after going through this process, you still have an undiagnosed problem, it is recommended that you go back to *Approach to Problem Determination in WebSphere Application Server V6* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

Review *Classify the problem and determine the root cause* to see if there are any other components that might be causing the problem.

If you feel sure that you have a plug-in related problem, there are things you can do before contacting IBM support. First, you should review the documentation that you have gathered for errors related to the plug-in that were not addressed in this paper and search support sites for information or fixes.

Next, you should collect all of the data that is outlined in the MustGather document for Web server plug-in problems and raise a problem record with IBM.

- ▶ MustGather for general problems with the Web server plug-in:  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21174894>
- ▶ MustGather for problems generating the plug-in configuration file:  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21199421>
- ▶ MustGather for problems installing the Web server plug-in:  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21199343>

Be sure to spell out all of the diagnostic work that you have done so far to minimize the amount of time it takes IBM Support to assist you in resolving your problem.



## Application server crash problem determination

This paper discusses application server crashes that are related to the Java Virtual Machine (JVM) or Just In Time (JIT) compilers in WebSphere Application Server V6. A problem in these components is normally fatal to the application server and can cause it to crash.

High-level symptoms of issues with these components that are most often reported by users are:

- ▶ The application appears to be hung or not available, and the application is not responding to incoming requests.
- ▶ The performance of the application is degrading.

If the application server process is gone from the system or if the process ID is constantly changing (the application server is being restarted), you are probably experiencing an application server crash.

**Important:** We recommend that you start your problem determination process by reading *Approach to Problem Determination in WebSphere Application Server V6* at <http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>.

## 8.1 Introduction

The JVM is an interpretive computing engine that is responsible for running the bytecode in a compiled Java program. The JVM translates the Java bytecodes into the native instructions of the host machine. The application server, being a Java process, requires a JVM in order to run and to support the Java applications that are running on it. JVM settings are part of an application server configuration.

The JVM provides the following:

- ▶ Class loader

As the name indicates, a class loader loads and verifies the classes. Multiple class loaders are involved in loading the required libraries for an application to run. Each class must be loaded by a classloader.

- ▶ Garbage collection

Garbage collection takes care of memory management for the entire application server. It searches memory to reclaim space from program segments or inactive data.

- ▶ Execution management

Manages the bookkeeping work for all the Java threads.

- ▶ Execution engine

Interprets the Java methods.

All JVMs use the just in time (JIT) compiler to compile heavily used Java bytecode into native instructions during server runtime to enhance performance.

JVM and JIT are platform specific. These components make use of the functionality that is provided by the operating system for enhancing WebSphere Application Server performance.

## 8.2 Work the problem

You begin the problem determination process by collecting the appropriate data that is required to diagnose the problem. We first take you through a high-level analysis of your symptoms to determine if you are truly experiencing an application server crash. If so, we provide a list of all the documentation that might be required and how to collect it. You then go through the process of analyzing the data to determine the most likely source of the problem.

And lastly, we provide guidance on the next step to take for resolution, whether it be a support site, contacting IBM, information about configuration, or some other suggestion as to how to proceed.

**Solaris and HP-UX users:**

- ▶ If the application server is running on Sun Solaris, go directly to “Sun Solaris” on page 300.
- ▶ If the application server is running on HP-UX, go directly to “HP-UX” on page 300.

## 8.2.1 High-level symptom analysis

Symptoms caused by an application server crash fall into the following basic categories:

- ▶ Application server stops responding

An application server that does not respond might be hung or the process might have ended. Users see hung applications or are not able to access new applications.

You should check to see if the application server process is running to determine if you are experiencing a crash. To do this, you need to know the process ID of the application server. You can find the process ID in the server name.pid file in:

```
<WAS_install_root>/profiles/<profile>/logs/<server>
```

Open the `<server_name>.pid` file in a text editor. The four-digit number is the process ID. You can then use the appropriate operating system command to check to see if the process is running. If it is not running, the problem is a crash.

If the application server is still running, the problem you have is most likely a hang situation. For advice on approaching hang problems, refer to *Approach to Problem Determination in WebSphere Application Server V6* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

- ▶ Performance degradation

An application server can suffer performance degradation when an application server is repeatedly crashing and being restarted automatically. A quick way to tell if this is the case is to monitor the application server process ID. If it changes over time, the application server is probably crashing and being restarted.

If either is the case, proceed to “Data to collect” on page 288.

Other problems can be caused by the JVM or JIT but are not covered in this paper. Review the following to make sure that your problem does not fit into the following categories:

- ▶ If CPU activity is low but the application server has not terminated, you most likely have a hang or deadlock situation. The following are possibilities:
  - Deadlock caused by JIT generated code (for not releasing the monitor).
  - Deadlock caused by system locks (JIT and JVM internal Locks).
- ▶ If CPU activity is high and the application server is using the cycles, you most likely have a loop. The following are possibilities:
  - Looping in JIT compiled code.
  - Looping in JIT compiler code (while compiling a method).

If either of these is the case, then you need to go back to *Approach to Problem Determination in WebSphere Application Server V6* at <http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf> for general guidance on these types of problems. However, note that some of the techniques in this paper might be helpful in narrowing down the failing code and in gathering documentation.

## 8.2.2 Data to collect

Diagnosing JVM and JIT problems can involve analyzing information from the following basic sources:

- ▶ Javacore files, also known as javadump files or thread dump files
- ▶ Process dumps, also known as crash dump, core file, or user dumps
- ▶ Process (native) stdout log

If the problem is difficult to recreate or disruptive to business operations, see “The next step” on page 299 for a complete list of documentation to collect before continuing.

### Javacore files

A javacore is a text file that is created by an application server during a failure. Javacore is specific to the IBM JDK. Javacore files contain diagnostic information that is related to the JVM and a Java application captured at a point during execution. For example, the information can be about the operating system, the application environment, threads, native stack, locks, and memory. The exact contents are dependent on the platform where the application server is running. By default, a javacore occurs when the JVM terminates unexpectedly. A javacore can also be triggered by sending specific signals to the JVM.



The JVM checks each of the following locations for existence and write-permission and stores the javacore in the first one available. Note that you must have enough free disk space (possibly up to 2.5 MB) for the javacore file to be written correctly.

- ▶ The location specified by the IBM\_JAVACOREDIR environment variable if set.
- ▶ `<WAS_install_root>/profiles/<profile>`.
- ▶ The location that is specified by the TMPDIR environment variable, if it is set.
- ▶ The /tmp directory or on Windows the location that is specified by the TEMP environment variable, if it is set.
- ▶ Windows only: If the javacore cannot be stored in any of the above, it is put to STDERR.

### Working with environment variables

Environment variables for the server process are defined by the administrator as name/value pairs. You can manage environment variables using the following navigation path in the administrative console: **Servers** → **Application Servers** → `<server>` → **Java and Process Management** → **Environment Entries**.

See the Technote at:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21162255>

Environment variable settings can also be seen in the Environment Variables section of the javacore.

### Process dumps

Process dumps are a complete dump of your computer virtual memory and can, therefore, be quite large. For example, if you have 4 GB of memory on your server, the dump size will also be in the GB range.

Note that you will not be analyzing the process dump but will simply note its existence. In the event that you have to call IBM support, the process dump is required as part of the documentation.

If a process dump exists, you can find it at the following location:

► On Windows

Windows has an embedded function for collecting data from processes that crash. The dumps are put into a file called *user.dmp* and are called *user dumps* for this reason.

To ensure these dumps are enabled and to find the location where they are stored:

- a. Go to **Start** → **Run**, and type `drwtsn32`.
- b. Look for the Crash Dump field to find the location of the dump file and make sure that the Create Crash Dump File option is selected.  
  
If it is a Windows XP machine, then set the Crash Dump Type to NT4 Full Compatible in the dialog box.
- c. Click **OK**. Enabling these settings is a one time process.

► On AIX/Linux

Process dumps are put into a file that is called *core* and are called *core file* for this reason. These core files are located in the `<WAS_install_root>/bin` directory or the `<operating_system_root>/tmp` directory.

### Process (native) stdout log

Native code running in a WebSphere Application Server process can write data to the process logs (also called native logs). Native code is non-Java code typically found in files with `.dll`, `.exe`, and `.so` extensions. The process logs are named `native_stdout.log` and `native_stderr.log`. They are located in the `<WAS_install_root>/profiles/<profile>/logs/<server>` directory.

## 8.2.3 Analyze the data

The following sections describe what information you should look for. The documentation is listed in the order of importance, so do one at a time.

### Is there a javacore file?

If a javacore file exists, check the application server to see if it is still running by looking at the list of process running in the task list. If the server is running, then the application has requested the thread dump (a call to `printStackTrace`), and it is not a true failure.

If the application server stopped after receiving the javacore, go to “Analyze the javacore file” on page 291.

## Is there a process dump?

If you did not get a javacore file, check to see if a process dump (core file) exists. If it does, it is still likely that you have had an application server crash.

The first action you should take is to upgrade the JDK version and test again to see if the problem still exists. If it does, go to “The next step” on page 299 for a list of documentation to gather before calling IBM support.

### Upgrading the JDK

The IBM JDK has a service refresh two or three times a year. During this time, many JIT and JVM fixes are integrated and new features could be incorporated. Thus, it is advisable to use the latest service refresh before going any further with problem determination.

To determine what level you have, open a command window and issue the **java -fullversion** command.

For information about upgrading the JDK, see:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27004980>

JDK updates are only available in refresh packs (for example, 6.0.2), not in fix packs (for example, 6.0.1.3). When a JDK update is available separate from a fix pack, the information is included at this Web site also.

## Was a stopServer command issued?

If you do not have a javacore or process dump, you probably have not had an application server crash. Look in native\_stdout.log to see if the application server is being stopped with a command. If you see a command to stop the server, investigate why the command is being issued. If you see no indications in the log that the application server was deliberately stopped, go to “The next step” on page 299 for a list of documentation to gather before calling IBM support.

## 8.2.4 Analyze the javacore file

To begin the analysis of a javacore file, follow these steps:

1. Look into the TITLE tag in the javacore file to find the signal information.

Signals -1,0, OUTFOMEMORY and SIGNONE are the memory signals (see Example 8-1 on page 292). If you see one of these signals, go to “Out of memory error” on page 298.

Example 8-1 Javacore signal information

```
NULL
-----
OSECTION      TITLE subcomponent dump routine
NULL          =====
1TISIGINFO    signal 0 received
1TIDATETIME   Date:                2005/05/10 at 08:55:41
1TIFILENAME   Javacore filename:
/app/WebSphere/wpsIn/AppServer/javacore22490.1115729741.txt
NULL
-----
OSECTION      XHPI subcomponent dump routine
NULL          =====
1XHETIME      Tue May 10 08:55:41 2005
1XHSIGRECV    SIGNONE received at 0x0 in <unknown>. Processing terminated.
1XHFULLVERSION J2RE 1.3.1 IBM AIX build ca131-20040517
```

Signals 10 and 11 indicate an application server crash. In Example 8-2, a signal 11 (SIGSEGV) occurred and caused the crash.

Example 8-2 Signal 11

```
OSECTION      TITLE subcomponent dump routine
NULL          =====
1TISIGINFO    signal 11 received
1TIDATETIME   Date:                2005/01/24 at 18:56:08
1TIFILENAME   Javacore filename:   F:\tmp\javacore.20050124.185608.1108.txt
NULL
```

2. Look for the CIJAVAVERSION tag to check the JDK version. Example 8-3 shows a typical entry.

Example 8-3 Checking JDK version

```
1CIJAVAVERSION J2RE 1.4.2 IBM Windows 32 build cn142-20041202
```

3. Look for the phrase *Fault module* in the first few lines of the javacore file (Example 8-4). This section gives the failing module (library) name and location.

Example 8-4 Javacore showing “Fault Module”

```
SECTION      TITLE subcomponent dump routine
NULL          =====
1TISIGINFO    signal 11 received
1TIDATETIME   Date:                2005/01/24 at 18:56:08
1TIFILENAME   Javacore filename:   F:\tmp\javacore.20050124.185608.1108.txt
NULL
-----
```

OSECTION	XHPI subcomponent dump routine
NULL	=====
1XHEXCPCODE	Exception code: C0000005 Access Violation
1XHEXCADDRESS	Fault address: 100D15B0 01:000D05B0
1XHEXCPCMODULE	Fault module: D:\IBM_142\jre\bin\classic\jvm.dll

---

If the fault module is:

– The JVM module:

- Windows: JVM.dll
- AIX: libjvm.a
- Linux: libjvm.so

Upgrade the JDK (see “Upgrading the JDK” on page 291) and retry to see if the problem goes away. If the problem still exists after the upgrade, check for a stack overflow problem (see “Stack overflow” on page 295).

If both of these steps fail to resolve the problem, go to “The next step” on page 299.

– The JIT module:

- Windows: JITC.dll
- AIX: libjitc.a
- Linux: libjitc.so

Upgrade the JDK (see “Upgrading the JDK” on page 291) and retry to see if the problem goes away. If the problem still exists after the upgrade, go to “Finding a workaround for JIT problems” on page 293.

– A WebSphere MQ module, see the following link for fix pack and interim fix information:

<https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=wsmqcsd>

– A DB2 module, see the following link for fixpack information:

<http://www-306.ibm.com/software/data/db2/udb/support/downloadv8.html>

## 8.2.5 Finding a workaround for JIT problems

You might not be able to fix a JIT-related problem. So, the key is to find a workaround that is acceptable while you report the problem to IBM and get a solution. The best way to find a workaround is to determine the method on which the failure is occurring and have the JIT compiler skip this method. An alternative is to completely disable JIT, though this action can have performance implications.

When you have a workaround, go to “The next step” on page 299”. Meanwhile, you can use the workaround that you have identified.

## Skip the failing method

Search the javacore for the phrase `Current Thread Details`. The method listed at the top is the failing method (Example 8-5).

### *Example 8-5 Find the current thread and failing method*

---

```
1XMCURTHDINFO  Current Thread Details
NULL          -----
3XMTHREADINFO "Servlet.Engine.Transports : 2" (TID:0x10759F18, sys_thread_t:0x39CCF340,
state:R, native ID:0x8F4) prio=5
4XESTACKTRACE at
com.ibm.workplace.wcm.app.ui.portlet.widget.HTMLPopupMenuButtonRenderer.render(Unknown Source)
4XESTACKTRACE at
com.ibm.workplace.wcm.app.ui.portlet.widget.HTMLPopupMenuButtonRenderer.render(Unknown Source)
4XESTACKTRACE at
com.ibm.psw.wcl.core.ARendererFactory.performRender(ARendererFactory.java(Compiled Code))
```

---

When you have identified the failing method, set the `JITC_COMPILEOPT` environment variable to skip the method that is causing the crash in the thread (Example 8-6). See “Working with environment variables” on page 289.

### *Example 8-6 Skip the failing method*

---

```
JITC_COMPILEOPT=SKIP{
com/ibm/workplace/wcm/app/ui/portlet/widget/HTMLPopupMenuButtonRenderer}{render}
```

---

It might be helpful to turn on the JIT compiling trace to provide additional documentation for IBM support and to verify that you have chosen the correct method to skip. The compiling trace logs every method that is compiled. The last method in this trace before a crash caused by the JIT is the method that should be skipped. The compiling trace is directed to the `native_stderr.log` for the server.

You can combine turning on the compiling trace (`COMPILING`) with skipping the method in the same setting, as shown in Example 8-7.

### *Example 8-7 Turn on the compiling trace*

---

```
JITC_COMPILEOPT=COMPILING:SKIP{
com/ibm/workplace/wcm/app/ui/portlet/widget/renderer/HTMLPopupMenuButtonRenderer}{render}
```

---

Note that `JITC_COMPILEOPT` options uses a semicolon (;) as a separator on Windows operating system and a colon (:) as a separator on UNIX systems.

To define an environment entry in the application server process definitions for Example 8-7 on page 294, use the following values:

- ▶ Name: JITC\_COMPILEOPT
- ▶ Value: `COMPILING:SKIP{  
com/ibm/workplace/wcm/app/ui/portlet/widget/renderer/  
HTMLPopupMenuButtonRenderer}{render}`

### Disable JIT

An alternative to skipping the failing method is to simply disable JIT. However, keep in mind that this action can cause performance problems and might not be a viable option.

To disable JIT from the administrative console:

1. Select **Servers** → **Application Servers**.
2. Select the server name.
3. Under Server Infrastructure, expand Java and Process Management. Click **Process Definition**.
4. Under Additional Properties, click **Java Virtual Machine**.
5. Select **Disable JIT**.
6. Restart the application server.

## 8.3 Analyzing problem areas

Your analysis of the javacore has most likely led you to a workaround or to one of the areas that are discussed in this section. If not, go to “The next step” on page 299.

### 8.3.1 Stack overflow

The JVM allocates a Java and native stack for each thread that is created by the application. If either of these stacks become exhausted, a stack overflow error occurs. A Java stack overflow can occur for the following reasons:

- ▶ The stack is not large enough to handle the request. Setting the following JVM parameter increases the Java stack size to 1 MB:

```
-Xoss1m
```

The default is 400 KB.

- ▶ The JIT compiler is causing a recursion in the application code.
- ▶ There is a recursion in the application code

A native stack overflow can occur for the following reasons:

- The stack is not large enough to handle the request. Setting the following JVM parameter increases the native stack size to 1 MB:

`-Xss1m`

The default is 400 KB.

- A recursion in native code (JDK or JNI code).

**Setting JVM parameters:** To set a JVM parameter, use the *Generic JVM arguments* field for the application server. You can find this at **Servers** → **Application Servers** → **<server>** → **Java and Process Management** → **Process Definition** → **Java Virtual Machine**.

To identify a crash that is caused by a stack overflow in a javacore file, look for a signal 11 and a current thread, similar to that shown in Example 8-8.

#### Example 8-8 Identifying a stack overflow

---

```
1XMCURTHDINFO  Current Thread Details
NULL          -----
3XMTTHREADINFO  "Servlet.Engine.Transports : 1" (TID:0x30AED438, sys_thread_t:0x7DCB59A0,
state:R, native ID:0x4553) prio=5: pending=java.lang.StackOverflowError
4XESTACKTRACE   at
org.apache.xpath.XPathContext.popSubContextList(XPathContext.java:949)
4XESTACKTRACE   at
org.apache.xpath.axes.PredicatedNodeTest.executePredicates(PredicatedNodeTest.java(Compiled
Code))
4XESTACKTRACE   at
org.apache.xpath.axes.PredicatedNodeTest.acceptNode(PredicatedNodeTest.java(Compiled Code))
4XESTACKTRACE   at org.apache.xpath.axes.AxesWalker.nextNode(AxesWalker.java(Compiled
Code))
4XESTACKTRACE   at
org.apache.xpath.axes.WalkingIterator.nextNode(WalkingIterator.java(Compiled Code))
4XESTACKTRACE   at
org.apache.xpath.axes.NodeSequence.nextNode(NodeSequence.java(Compiled Code))
4XESTACKTRACE   at org.apache.xpath.objects.XNodeSet.compare(XNodeSet.java(Compiled
Code))
4XESTACKTRACE   at org.apache.xpath.objects.XNodeSet.greaterThan(XNodeSet.java:667)
4XESTACKTRACE   at org.apache.xpath.operations.Gt.operate(Gt.java:45)
4XESTACKTRACE   at org.apache.xpath.operations.Operation.execute(Operation.java(Compiled
Code))
4XESTACKTRACE   at org.apache.xpath.Expression.bool(Expression.java(Compiled Code))
4XESTACKTRACE   at org.apache.xpath.operations.And.bool(And.java:70)
4XESTACKTRACE   at org.apache.xpath.operations.And.bool(And.java:70)
4XESTACKTRACE   at org.apache.xpath.operations.And.bool(And.java:70)
4XESTACKTRACE   at org.apache.xpath.XPath.bool(XPath.java(Compiled Code))
4XESTACKTRACE   at
```



```
org.apache.xalan.templates.ElemChoose.execute(ElemChoose.java(Compiled Code))
4XESTACKTRACE          at
org.apache.xalan.transformer.TransformerImpl.executeChildTemplates(TransformerImpl.java(Compiled Code))
4XESTACKTRACE          at
org.apache.xalan.transformer.TransformerImpl.transformToRTF(TransformerImpl.java(Compiled Code))
4XESTACKTRACE          at
org.apache.xalan.transformer.TransformerImpl.transformToRTF(TransformerImpl.java(Compiled Code))
4XESTACKTRACE          at
org.apache.xalan.templates.ElemVariable.getValue(ElemVariable.java(Compiled Code))
4XESTACKTRACE          at
org.apache.xalan.templates.ElemVariable.execute(ElemVariable.java(Compiled Code))
4XESTACKTRACE          at
org.apache.xalan.transformer.TransformerImpl.executeChildTemplates(TransformerImpl.java(Compiled Code))
4XESTACKTRACE          at
org.apache.xalan.templates.ElemTemplate.execute(ElemTemplate.java:394)
4XESTACKTRACE          at
org.apache.xalan.templates.ElemCallTemplate.execute(ElemCallTemplate.java(Compiled Code))
4XESTACKTRACE          at
org.apache.xalan.transformer.TransformerImpl.executeChildTemplates(TransformerImpl.java(Compiled Code))
4XESTACKTRACE          at org.apache.xalan.templates.ElemIf.execute(ElemIf.java(Compiled Code))
4XESTACKTRACE          at
org.apache.xalan.transformer.TransformerImpl.executeChildTemplates(TransformerImpl.java(Compiled Code))
4XESTACKTRACE          at
org.apache.xalan.templates.ElemTemplate.execute(ElemTemplate.java:394)
4XESTACKTRACE          at
org.apache.xalan.templates.ElemCallTemplate.execute(ElemCallTemplate.java(Compiled Code))
4XESTACKTRACE          at
org.apache.xalan.transformer.TransformerImpl.executeChildTemplates(TransformerImpl.java(Compiled Code))
4XESTACKTRACE          at org.apache.xalan.templates.ElemIf.execute(ElemIf.java(Compiled Code))
4XEMORENOTSHOWN      ... (more frames not shown)
```

---

You can tell that this crash was caused by a stack overflow because of the following clues in the trace:

- The following entry:

```
pending=java.lang.StackOverflowError
```

- The stack appears to loop in the following code:

```
org.apache.xalan.templates.ElemCallTemplate.execute(ElemCallTemplate.java(Compiled Code))
```

```
org.apache.xalan.transformer.TransformerImpl.executeChildTemplates(TransformerImpl.java(Compiled Code))
```

If the current thread contains `pending=java.lang.StackOverflowError` but the Java stack does not appear to be in a loop, the stack overflow is probably thrown from the native stack.

To resolve a crash that is caused by a stack overflow, do the following:

1. In the example above, try increasing the Java stack (`-Xoss1m`). For stack overflows that are caused by native stack exhaustion, use `-Xss1m`. If the server still crashes with an identical current thread, then the stack overflow is caused by a recursion in either the JIT or the application code.
2. Upgrade the JDK. If the stack overflow is caused by JIT, upgrading the JDK might resolve the issue.
3. Skip the methods that appear to be looping (see “Skip the failing method” on page 294). In the example above, we would try skipping the following:

```
org.apache.xalan.templates.ElemCallTemplate.execute  
org.apache.xalan.transformer.TransformerImpl.executeChildTemplates
```

4. If these actions do not resolve the problem, then the problem is most likely a recursion in the application code. The owners of the code should be asked to review their code for the cause of the loop.

### 8.3.2 Out of memory error

During compilation of a method, JIT uses native memory as its work buffer. It also uses native memory to store the compiled code address. This memory is in use as long as those methods are in use by the application. When the classes from the application are garbage collected, this memory is released to the operating system. If the available memory is too small and methods are getting compiled in large numbers, there is a chance that the application will receive an `OutOfMemory` fatal error. In this case, you might need to increase the physical memory of the system.

For information about how to address memory problems, see *Approach to Problem Determination in WebSphere Application Server V6* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

## 8.4 The next step

The symptoms and problem areas included in this paper are some that you are more likely to experience. However, there are other things that can go wrong, or the cause of the problem might be related to a component other than the JVM or JIT.

If, after going through this process, you still have an undiagnosed problem, it is recommended that you go back to *Approach to Problem Determination in WebSphere Application Server V6* at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

Review the problem classifications to see if there are any other components that might be causing the problem.

If you feel sure you have a JVM or JIT related problem, there are things you can do before contacting IBM support. First, review the documentation that you have gathered for errors related to the problem that were not addressed in this paper, and search support sites for information or fixes.

The following diagnostic guides could be of use:

- ▶ *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2 Diagnostics Guide*, SC34-6358:

<http://www-106.ibm.com/developerworks/java/jdk/diagnosis>

- ▶ *Introduction to IBM JVM for Linux JIT diagnostics*

<http://www-128.ibm.com/developerworks/eserver/library/es-JITDiag.html>

Next, collect all of the data that is outlined in the MustGather documentation for JVM and JIT problems and raise a problem record with IBM. Be sure to spell out all of the diagnostic work that you have done so far to minimize the time it takes IBM Support to assist you in resolving your problem.

The following URL lists the MustGather documents for JVM and JIT related problems:

<http://www-1.ibm.com/support/search.wss?rs=180&tc=SSEQTP&tc1=SSCYP8L&q=MustGatherDocument>

## 8.4.1 Sun Solaris

IBM does not supply a software developer kit or runtime environment for the Sun Solaris platform. However, IBM does make strategic products, such as the WebSphere Application Server, for this platform. WebSphere Application Server contains an embedded copy of the Sun Solaris JVM along with some IBM add-ons, such as security, XML, and ORB packages. The WebSphere Application Server Solaris SDK is, therefore, a hybrid of Sun and IBM products. However, the core JVM and JIT are Sun Solaris. Thus, this book is not appropriate for diagnosis on Sun Solaris.

IBM does service the Sun Solaris SDK, but only when it is an embedded part of IBM middleware, for example, WebSphere Application Server. If you get a Java problem on Solaris as a result of using an IBM middleware product, then follow these steps:

- ▶ Check the following URL for the latest operating system patches. (You should be always on the latest patches.)

<http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/J2SE>

- ▶ MustGather information for crashes on Solaris

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21049530>

## 8.4.2 HP-UX

IBM does not supply a software developer kit or runtime environment for HP platforms. However, IBM does make strategic products, such as the WebSphere Application Server, for this platform.

In this case, WebSphere Application Server contains an embedded copy of the HP JVM alongside some IBM add-ons, such as security packages. The WebSphere Application Server HP SDK is, therefore, a hybrid of HP and IBM products. However, the core JVM and JIT are HP software. Thus, this book is not appropriate for diagnosis on HP platforms.

IBM does service the HP SDK, but only when it is an embedded part of IBM middleware (for example, WebSphere Application Server). If you get a Java problem on an HP platform as a result of using an IBM middleware product, check the following URL for latest OS patches:

<http://www.hp.com/products1/unix/java/patches/index.html>

## Default messaging provider problem determination

This paper describes some of the problems that can be encountered when using the IBM WebSphere Application Server V6 messaging component. Typically, this type of failure would manifest itself as an application that fails to start, an application hang, or an application that fails to receive an expected message. There might or might not be an accompanying error message from the application itself.

Although WebSphere Application Server V6 can use messaging mechanisms familiar to users of WebSphere Application Server V4 and V5 (IBM WebSphere MQ provider and generic message providers), the default messaging provider included with V6 is very different to its predecessors. This paper confines itself to investigating messaging problems that are specific to the default messaging provider that is included with WebSphere Application Server.

**Important:** We recommend that you start your problem determination process by reading *Approach to Problem Determination in WebSphere Application Server V6*, REDP-4073, at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

## 9.1 Introduction

WebSphere Application Server V6 messaging is a general term for a group of components that provide the messaging functionality for applications. WebSphere Application Server provides a default messaging provider, as well as support for IBM WebSphere MQ or generic messaging providers.

The *default messaging provider* is the Java Message Service (JMS) API implementation for messaging (connection factories, JMS destinations, and so on). The concrete destinations (queues and topic spaces) behind the default messaging provider interface are implemented in a *service integration bus*. Similarly, the WebSphere MQ JMS provider is the JMS API implementation with WebSphere MQ (with queue managers, for example) implementing the real destinations for the JMS interface.

A service integration bus consists of one or more bus members. A bus member can be an application server or a cluster. Each bus member will have one (or possibly more in the case of clusters) *messaging engine* that manages connections to the bus and messages. A service integration bus can connect to other service integration buses and to WebSphere MQ.

Figure 9-1 on page 303 shows a simplified schematic of some of the components that make up the default messaging provider function. Although this is a simplified schematic, it allows us to see the many components involved: the service integration bus, messaging engines, links to remote messaging providers, data sources, destinations, queue points, and more.

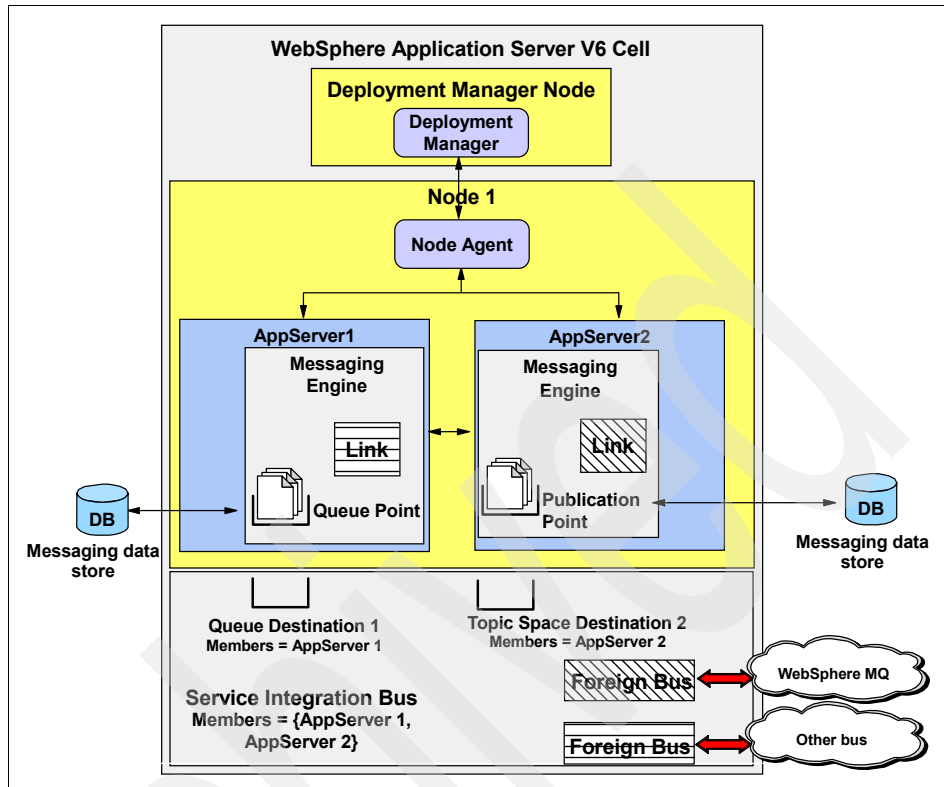


Figure 9-1 Default messaging provider structure

To gain a common understanding of the areas where potential problems can occur and before attempting to isolate and identify a messaging problem, it is useful to start with an overview of the various components using the diagram in Figure 9-1. These components include the following items:

- ▶ The service integration bus (referred to simply as the *bus*) is the primary architectural component on which the WebSphere Application Server V6 message functionality is based. It is used by the default messaging provider.
- ▶ The bus can hold references to other service integration buses or to WebSphere MQ. These are *foreign bus* definitions.
- ▶ *JMS-administered objects* encapsulate the information that is required within a JMS application to connect to the JMS provider and to access the *destinations* that are defined by the JMS provider. JMS-administered objects refer to objects on the bus.

- ▶ A destination defined on the bus can be a queue, a topic space, or a foreign bus destination. Each destination has a *reliability* property that defines a quality of service. The JMS interface can specify persistent or non-persistent for the messages sent, and the JMS-administered objects can map this JMS persistence setting to a preferred reliability in the bus.
- ▶ Destinations can have an associated *mediation*. A mediation allows the processing of messages at the destination before delivery to the consumer.
- ▶ Access to the bus is managed by the *messaging engine*. The messaging engine runs in an application server. When a cluster of application servers is added as a member of the bus, a single messaging engine is automatically created and associated with the application server cluster, regardless of the number of application servers defined as members of the cluster. At run time, this messaging engine is activated within a single application server within the cluster. The application server that is chosen to host the messaging engine will be the first cluster member to start.
- ▶ A queue is assigned to one bus member. The messaging engine in the bus member hosts the message point for the queue, known as a *queue point*. The queue point is the location where messages for the queue are stored and processed. If the bus member has more than one messaging engine, the queue is partitioned across the messaging engines. Each messaging engine has a separate queue point for the queue and handles a share of the messages arriving at the destination.
- ▶ Each messaging engine has its own data store where it stores persistent messages. Non-persistent messages can be spilled to the data store if the messaging engine decides to do so (for example, to reduce the memory footprint of messages).
- ▶ A bus can be connected to other service integration buses or to WebSphere MQ. These connections are made by a *link* definition on a messaging engine within the bus, with the type of link depending on the target. A messaging engine can also be configured to emulate the WebSphere Application Server V5 embedded messaging provider to allow V5 embedded messaging clients to connect to the bus as though they were connected to the V5 embedded provider.

As you can see, there are many different components and many opportunities for configuration errors. These might be with the local components, such as the messaging engine or data store, or with WebSphere MQ.

For a comprehensive description of the messaging features, terminology, and topologies available, see Chapters 10 and 11 in the IBM Redbook *WebSphere Application Server V6 System Management and Configuration Handbook*, SG24-6451.



### 9.1.1 Problem categories

Messaging problems fall broadly into the following categories:

- ▶ Messaging engine startup problems
- ▶ Message flow problems, including interfaces to foreign buses and client applications
- ▶ Application configuration and resource problems

**Note:** Although this paper does not specifically address performance problems related to messaging, you should be aware that references to large message objects must be maintained by many components because they are delivered around the network. This can have severe implications for the Java Virtual Machine (JVM). If you suspect messaging applications are causing memory problems, this is something to consider.

## 9.2 Work the problem

You begin the problem determination (PD) process by collecting the appropriate data that is required to diagnose the problem. We first provide a list of all the documentation that might be required.

Next, you go through a series of questions and actions that will help you define the high-level symptoms that you are experiencing. Each of these steps leads to a more detailed procedure that is designed to take you through the process of analyzing data to determine the most likely source of the problem. While it is impossible to enumerate every possible failure scenario, we describe various use cases that have occurred during testing and real customer use as examples for these categories. We also provide a strategy for analyzing and resolving these problems.

And lastly, we provide guidance on the next step to take for resolution, whether it be a support site, contacting IBM, information about configuration, or some other suggestion as to how to proceed.

Before going further with the investigation, it is worth asking some basic questions about the nature of the problem in an attempt to isolate why the problem is occurring. Basic questions include:

- ▶ What is the error message received or the scenario that leads you to believe you have a messaging problem? Any error or warning messages with a prefix of CWSI or CESJ can be an indicator of the problem. If your application has a message-driven bean that uses WebSphere Application Server V5 JMS resources, look for the prefixes MSGS and WMSG.

For a complete list of messaging message prefixes, see:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/welc\\_ref\\_trb\\_msg.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/welc_ref_trb_msg.html)

You can find explanations of individual messages in the WebSphere Information Center. When searching for message prefixes, enclose the prefix in double quotation marks.

- ▶ What changes have been made to the application, configuration, WebSphere Application Server maintenance level, operating system, network, or hardware?
- ▶ What are the implications of backing out any changes that were made to verify whether that the change precipitated the problem?
- ▶ Do the basic functions of the application work, or is this some new scenario (for example, some code that has been added or a newly tested function)?
- ▶ Have you tested the IVT applications and the samples to ensure that they work OK? This is extremely important, because it establishes the basic functionality of the application server.
- ▶ Have you validated the WebSphere Application Server configuration using **\$AdminConfig validate**? Although this is not infallible, it does provide a check of the configuration changes that you might have made. Configuration problems can also be viewed from the administrative console (**Troubleshooting** → **Configuration Problems**).
- ▶ Has there been a change to the data being used? A commonly encountered problem is where a new data format (long messages, object messages, or null data) is used for the first time and this precipitates an error in the processing of that data.

Having established the type of change that might have precipitated the problem, it might be expedient to back out the change to allow the system to continue running.

### 9.2.1 Collect the data

There is a minimum set of documentation that is required for investigating any problem. This information should be gathered before starting to examine any messaging problems. Then, depending on the type and the complexity of the problem, additional documentation might be required. For our initial investigations, the minimum set of documentation that you should gather is:

- ▶ SystemOut.log
- ▶ SystemErr.log
- ▶ Exception logs in FFDC directory
- ▶ Exceptions received by any client applications

You can find information about finding and reviewing logs and traces in *WebSphere Application Server V6: Diagnostic Data*, REDP-4085, at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4085.pdf>

If the problem is difficult to recreate or disruptive to business operations, see “The next step” on page 350 for a complete list of documentation to collect before continuing. In particular, you should review the MustGather document for service integration technologies for a complete list of documentation that is required by IBM support.

## 9.2.2 Analyze the high-level symptoms

At this point, the problem could be any of the categories that we have discussed. The following analysis strategy should help you define the problem.

Begin by examining the documentation that you have gathered. Look for any error codes, exception codes, or stack traces. Use these as search arguments in the appropriate IBM support sites and the WebSphere Information Center.

It is also a useful technique to use an Internet search engine to search the IBM Web site for matching symptoms and message codes:

<http://www.ibm.com/software/webservers/appserv/was/support/>

Often, the explanatory notes in the WebSphere Information Center will be sufficient to resolve a problem.

### Ensure that the messaging engine has started

The first step in diagnosing any messaging problem is to ensure that all the messaging engines on the bus are active. To check the status of the messaging engines from the administrative console:

1. Select **Service integration** → **Buses**.
2. Click the bus name to open the details page.
3. Click **Bus members**.
4. Click the bus member.

This shows a list of messaging engines and their status, as shown in Figure 9-2 on page 308.

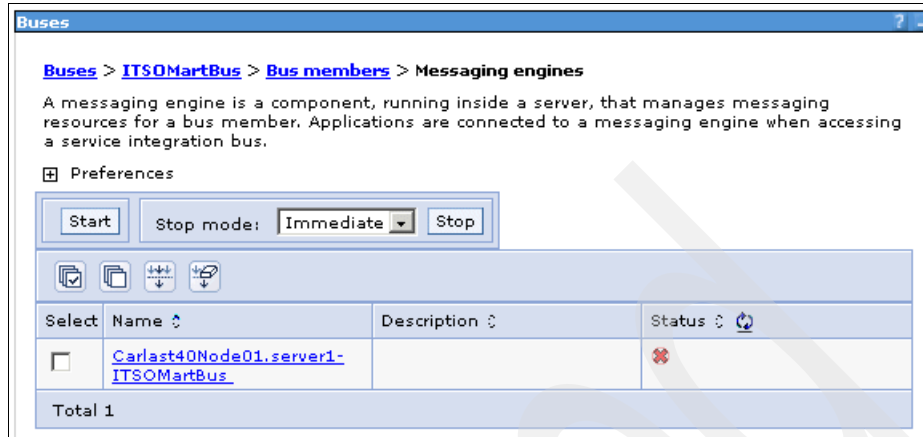


Figure 9-2 Checking the messaging engine status

If a messaging engine is stopped, go to “Messaging engine startup problems” on page 310.

This should always be the first area to isolate and eliminate from your investigations. It might be that you iterate through this process quickly for every problem simply to ensure that the underlying messaging components have started correctly and are active.

## Check the message flow to the destination

If the application has been running successfully and then fails, it would appear to rule out a startup problem. So, you should continue by investigating message delivery and JMS application problems.

The next thing to do is see if there is a message available for the application to consume. This can be done easily by using the administrative console and looking at the runtime values for the queue. To do this, select **Service integration** → **Buses** → **<busname>** → **Destination** → **<destinationname>** → **Queue Point** → **<qname>**.

Figure 9-3 on page 309 shows an example.

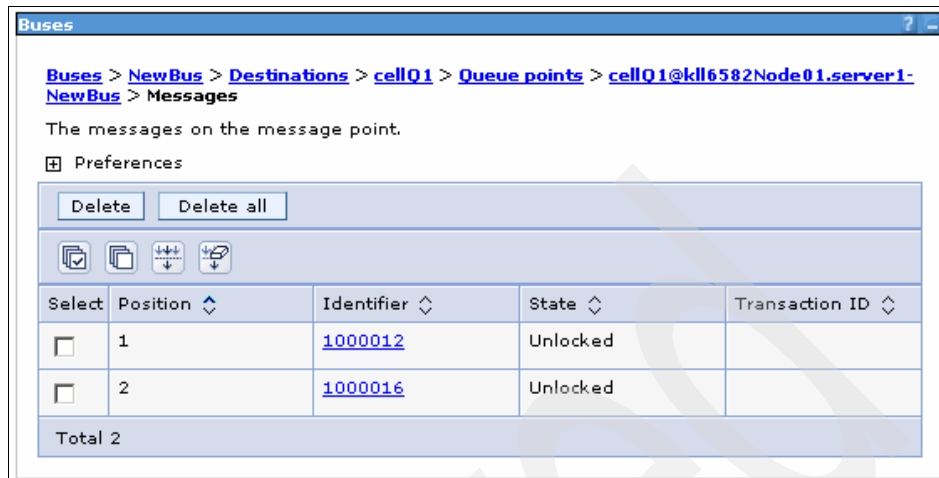


Figure 9-3 Viewing messages on the queue

If the messaging engine started correctly and there are messages on the destination queue point that you believe your application should be processing, go to “Message flow problems” on page 324.

If there are no messages, continue to the next section.

## Ensure that the application is producing messages

If you have read through thus far, then you have now confirmed that the messaging engine has started but that there are no messages available for processing on the destination queue point.

At this point, you could still be examining a message flow problem. Why are there none of the expected messages on the destination? This could be a problem with the message producer application. Has the message producer previously delivered messages? You have checked the messaging engine and queues for the message consumer in an earlier step. Now, it is time to perform the same checks for the message producer.

Assuming that the message producer’s messaging engine is working correctly, you should next ensure that the application that is able to access the default messaging provider and the resources have been configured properly. For information about this, go to “Application configuration and resource problems” on page 344.

Even assuming the message producer has been executing correctly up to now, this does not rule out an application problem, for example when processing a

specific message or data type. You should next look at the possibility of a message flow problem as described in “Message flow problems” on page 324.

### 9.2.3 What to do if your symptom is not listed here

As stated previously, this paper cannot cover all possible scenarios that might occur. It is quite possible that your problem does not fall neatly into one of the categories that we have mentioned. If you do not see your symptom listed here, go to “The next step” on page 350 for more information about how to proceed.

## 9.3 Analyzing problem areas

This section explores the following problem areas:

- ▶ Messaging engine startup problems
- ▶ Message flow problems
- ▶ Application configuration and resource problems

### 9.3.1 Messaging engine startup problems

This type of error is not usually apparent at startup. Few applications will fail to start due to being unable to connect to the bus. Only message-driven bean (MDB) applications attempt to connect to the bus during startup, and they will start successfully as long as the bus is defined, even if the messaging engine they are connecting to is not available yet. This is by design, because applications start before the messaging engine in the server startup sequence. MDBs will fail to start if there is no ActivationSpec for them, or if the bus they are configured to connect to does not exist.

EJB applications generally do not connect to the bus until a client connects to them and tells them to. At this point, they will get a JMS exception on connect, but the application will have started and probably will not stop due to the failure to connect.

#### **Becoming familiar with a normal startup**

To evaluate a messaging engine startup problem, it is advisable to familiarize yourself with how a normal startup appears in the log files. Example 9-1 on page 311 is an extract from a SystemOut.log file that shows a normal messaging engine startup.

**Note:** In the following examples, the date and time that is displayed in the log file output has been replaced by the sequence [...]. This has been done for brevity. In some examples, the messaging engine name has been replaced by mename.

#### Example 9-1 Messaging engine startup

```

:
[...] 0000000a SibMessage I [:] CWSIU0000I: Release: WAS601.SIB Level: o0518.09
:
:
[...] 0000000a ResourceMgrIm I WSVR0049I: Binding _k1l6582Node01.server1-TestSIBus as
jdbc/com.ibm.ws.sib/k1l6582Node01.server1-TestSIBus
:
:
[...] 0000000a TCPChannel A TCPC0001I: TCP Channel SIB_TCP_JFAP is listening on host *
(IPv4) port 7276.
[...] 0000000a WSChannelFram A CHFW0019I: The Transport Channel Service has started chain
InboundBasicMessaging.
[...] 0000000a TCPChannel A TCPC0001I: TCP Channel SIB_TCP_JFAP_SSL is listening on host *
(IPv4) port 7286.
[...] 0000000a WSChannelFram A CHFW0019I: The Transport Channel Service has started chain
InboundSecureMessaging.
[...] 0000000a SibMessage A [:] CWSIC2001I: Messaging connections are being accepted.
[...] 0000000f SibMessage I [TestSIBus:k1l6582Node01.server1-TestSIBus] CWSID0016I:
Messaging engine k1l6582Node01.server1-TestSIBus is in state Joined.
[...] 00000014 SibMessage I [TestSIBus:k1l6582Node01.server1-TestSIBus] CWSID0016I:
Messaging engine k1l6582Node01.server1-TestSIBus is in state Starting.
:
:
[...] 00000014 InternalGener I DSRA8203I: Database product name : DBMS:db2j
[...] 00000014 InternalGener I DSRA8204I: Database product version : 5.1.60.17
[...] 00000014 InternalGener I DSRA8205I: JDBC driver name : Cloudscape Embedded JDBC Driver
[...] 00000014 InternalGener I DSRA8206I: JDBC driver version : 5.1.60.17
[...] 0000000a WsServerImpl A WSVR0001I: Server server1 open for e-business
:
:
[...] 00000014 SibMessage I [TestSIBus:k1l6582Node01.server1-TestSIBus] CWSIS1538I: The
messaging engine is attempting to obtain an exclusive lock on the data store.
[...] 00000033 SibMessage I [TestSIBus:k1l6582Node01.server1-TestSIBus] CWSIS1537I: The
messaging engine has acquired an exclusive lock on the data store.
[...] 00000014 SibMessage I [TestSIBus:k1l6582Node01.server1-TestSIBus] CWSID0016I:
Messaging engine k1l6582Node01.server1-TestSIBus is in state Started.

```

Becoming familiar with your own system's normal startup messages will help you to recognize irregularities in the log files.

Note that messaging engines frequently only get to state Started after the “Open for e-business” message in the SystemOut.log.

### **What to look for**

If the messaging engine is stopped, check the log for the following message:

Message CWSIS0002E

If you find this message go to “Symptom: CWSIS0002E (messaging engine exception)” on page 312.

Otherwise, the messaging engine has not started, but there does not appear to be a relevant message code in the SystemOut.log. In this case, go to “The next step” on page 350.

### **Symptom: CWSIS0002E (messaging engine exception)**

Exceptions received during messaging engine startup are reported using CWSIS0002E. This message might contain the information that tells you the cause of the problem, or you might have to scan back through the log to find more information.

Message CWSIS0002E is often followed by message CWSID0027I, which indicates that a serious error has occurred while restarting the messaging engine, and message CWSID00016I, which indicates that the messaging engine is in a stopped state. These messages are generic in nature and are not indications of the underlying cause.

To begin, focus on the information that is contained in CWSIS0002E. The following are examples of common configuration problems that can cause the messaging engine to fail during startup:

- ▶ JNDI name error when accessing the messaging data store.
- ▶ Invalid authentication alias information.
- ▶ Invalid database tables or tables with invalid data. For example, tables for another messaging engine exist in the data store after the messaging engine has been deleted and recreated.

For information about the messaging data store, see “Example: Incorrect data store for the messaging engine” on page 318.

#### ***Example: JNDI error for the messaging data store***

In Example 9-2 on page 313, the original error is contained in the CWSIS0002E message. In this case, it is a configuration error that has occurred while processing the JNDI name of the data source. The message itself contains an



embedded exception code, CWSIS1524E, and looking at the data source name, you can see the error in the name.

Note that you also see CWSID0035E. CWSID0035E is an internal message indicating which Java class and method experienced the problem. This is not directly helpful to you. However, it might be helpful to the IBM service team in the event that you need to contact them.

*Example 9-2 Messaging engine startup failure: JNDI error*

---

```
[...] 00000010 SibMessage    I    [mename] CWSID0016I: Messaging engine <mename> is in state
Joined.
[...] 00000014 SibMessage    I    [mename] CWSID0016I: Messaging engine <mename> is in state
Starting.
[...] 00000014 SibMessage    E    [mename] CWSIS0002E: The messaging engine encountered an
exception while starting. Exception: com.ibm.ws.sib.msgstore.MessageStoreRuntimeException:
CWSIS1524E: Data source, jdbc/com.ibm.ws.sib/xxxxx, not found.
[...] 00000014 SibMessage    E    [mename] CWSID0035E: Messaging engine <mename> cannot be
started; detected error reported during com.ibm.ws.sib.msgstore.impl.MessageStoreImpl start()
[...] 00000014 SibMessage    E    [mename] CWSID0027I: Messaging engine <mename> cannot be
restarted because a serious error has been reported.
[...] 00000014 SibMessage    I    [mename] CWSID0016I: Messaging engine <mename> is in state
Stopped.
[...] 00000014 SibMessage    I    [mename] CWSID0016I: Messaging engine <mename> is in state
Joined.
```

---

In this specific example, the problem was caused because an incorrect value was specified in the data source JNDI name of the definition. This type of error most likely occurs when you are setting up an application server or cluster as a bus member for the first time.

You can verify or alter the setting from the administrative console by doing the following:

1. Select **Service integration** → **Buses**.
2. Click the bus name.
3. Under Additional Properties, click **Messaging engines**.
4. Click the messaging engine name.
5. Under Additional Properties, click **Data store**.



Figure 9-4 Messaging engine data store properties

The JNDI name specified here has to match the JNDI name that is specified for the data store. For an example of finding the data source for the messaging data store, see “Example: Authentication error for the messaging data store” on page 314.

### **Example: Authentication error for the messaging data store**

A common startup problem is caused when a failure to access the messaging data store occurs due to an authentication problem. In the extract from the SystemOut.log shown in Example 9-3, you can see the CWSIS0002E message. The message indicates an authentication error has occurred when trying to access the messaging data store. Note that the J2CA0046E and SQL30082N messages provide supporting information.

#### **Example 9-3 Messaging engine startup failure: authentication error**

```
[...] 0000002f SibMessage I [NewBus:k116582Node03.server1-NewBus] CWSID0016I: Messaging
engine k116582Node03.server1-NewBus is in state Starting.
[...] 0000002f FreePool E J2CA0046E: Method createManagedConnectionWithMCWrapper caught
an exception during creation of the ManagedConnection for resource jdbc/JDBCDataSource,
throwing ResourceAllocationException. Original exception: com.ibm.ws.exception.WsException:
DSRA8100E: Unable to get a PooledConnection from the DataSource. with SQL State : 08001 SQL
Code : -30082
:
```

```

:
Caused by: java.sql.SQLException: SQL30082N Attempt to establish connection failed with
security reason "24" ("USERNAME AND/OR PASSWORD INVALID"). SQLSTATE=08001
DSRA0010E: SQL State = 08001, Error Code = -30,082DSRA0010E: SQL State = 08001, Error Code =
-30,082
... 23 more
Next Linked Exception:
java.sql.SQLException: SQL30082N Attempt to establish connection failed with security reason
"24" ("USERNAME AND/OR PASSWORD INVALID"). SQLSTATE=08001
DSRA0010E: SQL State = 08001, Error Code = -30,082DSRA0010E: SQL State = 08001, Error Code =
-30,082
:
:
[...] 0000002f SibMessage E [NewBus:k116582Node03.server1-NewBus] CWSIS0002E: The
messaging engine encountered an exception while starting. Exception:
com.ibm.ws.sib.msgstore.PersistenceException: CWSIS1501E: The data source has produced an
unexpected exception: java.sql.SQLException: SQL30082N Attempt to establish connection failed
with security reason "24" ("USERNAME AND/OR PASSWORD INVALID"). SQLSTATE=08001
DSRA0010E: SQL State = 08001, Error Code = -30,082DSRA0010E: SQL State = 08001, Error Code =
-30,082
[...] 0000002f FreePool E J2CA0046E: Method createManagedConnectionWithMCWrapper caught
an exception during creation of the ManagedConnection for resource jdbc/JDBCDataSource,
throwing ResourceAllocationException. Original exception: com.ibm.ws.exception.WsException:
DSRA8100E: Unable to get a PooledConnection from the DataSource. with SQL State : 08001 SQL
Code : -30082
:
:
Caused by: java.sql.SQLException: SQL30082N Attempt to establish connection failed with
security reason "24" ("USERNAME AND/OR PASSWORD INVALID"). SQLSTATE=08001
DSRA0010E: SQL State = 08001, Error Code = -30,082DSRA0010E: SQL State = 08001, Error Code =
-30,082
... 23 more
Next Linked Exception:
java.sql.SQLException: SQL30082N Attempt to establish connection failed with security reason
"24" ("USERNAME AND/OR PASSWORD INVALID"). SQLSTATE=08001
DSRA0010E: SQL State = 08001, Error Code = -30,082DSRA0010E: SQL State = 08001, Error Code =
-30,082
:
:
[...] 0000002f SibMessage E [NewBus:k116582Node03.server1-NewBus] CWSID0035E: Messaging
engine k116582Node03.server1-NewBus cannot be started; detected error reported during
com.ibm.ws.sib.msgstore.impl.MessageStoreImpl start()
[...] 0000002f SibMessage E [NewBus:k116582Node03.server1-NewBus] CWSID0027I: Messaging
engine k116582Node03.server1-NewBus cannot be restarted because a serious error has been
reported.
[...] 0000002f SibMessage I [NewBus:k116582Node03.server1-NewBus] CWSID0016I: Messaging
engine k116582Node03.server1-NewBus is in state Stopped.

```

---

The solution to this problem is to verify the component managed authentication alias. This alias can be specified for the data source definition, or for the messaging engine definition.

**Note:** The recommended (and more secure) method of specifying the authentication alias is to do it in the messaging engine data store definition. Be aware, that if you specify the authentication alias in this way, testing the data source connection in the administrative console will fail.

If the authentication alias is specified in both places (data source and messaging engine), the alias specified on the messaging engine will be used.

If specified at the messaging engine, you can find the authentication alias specification by performing the following steps in the administrative console:

1. Select **Service integration** → **Buses**.
2. Click the bus name.
3. Under Additional Properties, click **Messaging engines**.
4. Click the messaging engine name.
5. Under Additional Properties, click **Data store**.

If specified at the data source, you can find it by performing the following steps:

1. Select **Resources** → **JDBC providers**.
2. Select the proper scope.
3. Select the JDBC provider from the list.
4. Under Additional Properties, click **Data sources**.
5. Select the data source name.

The authentication alias must contain a valid user name and password combination that has appropriate access privileges for the database. Refer to Section 11.8.4 of *WebSphere Application Server V6 System Management and Configuration Handbook*, SG24-6451.

**Tip:** Defining the data source for the data store at the wrong scope is a common error.

The problem generally lies with an administrator that does not notice that the default scope for creating resources in the administrative console is set to the deployment manager node level (Network Deployment only). While defining the data source at the node level will work, using the deployment manager node will not. The node where the application server is defined needs to be selected.

In general, we recommend that, for a server bus member, you define the data source at the server scope, and for a cluster, that you define it at the cluster scope.

In Figure 9-5 on page 318, you can see an incorrectly specified authentication alias for this database. The resolution for this problem is to specify a correct authentication alias.

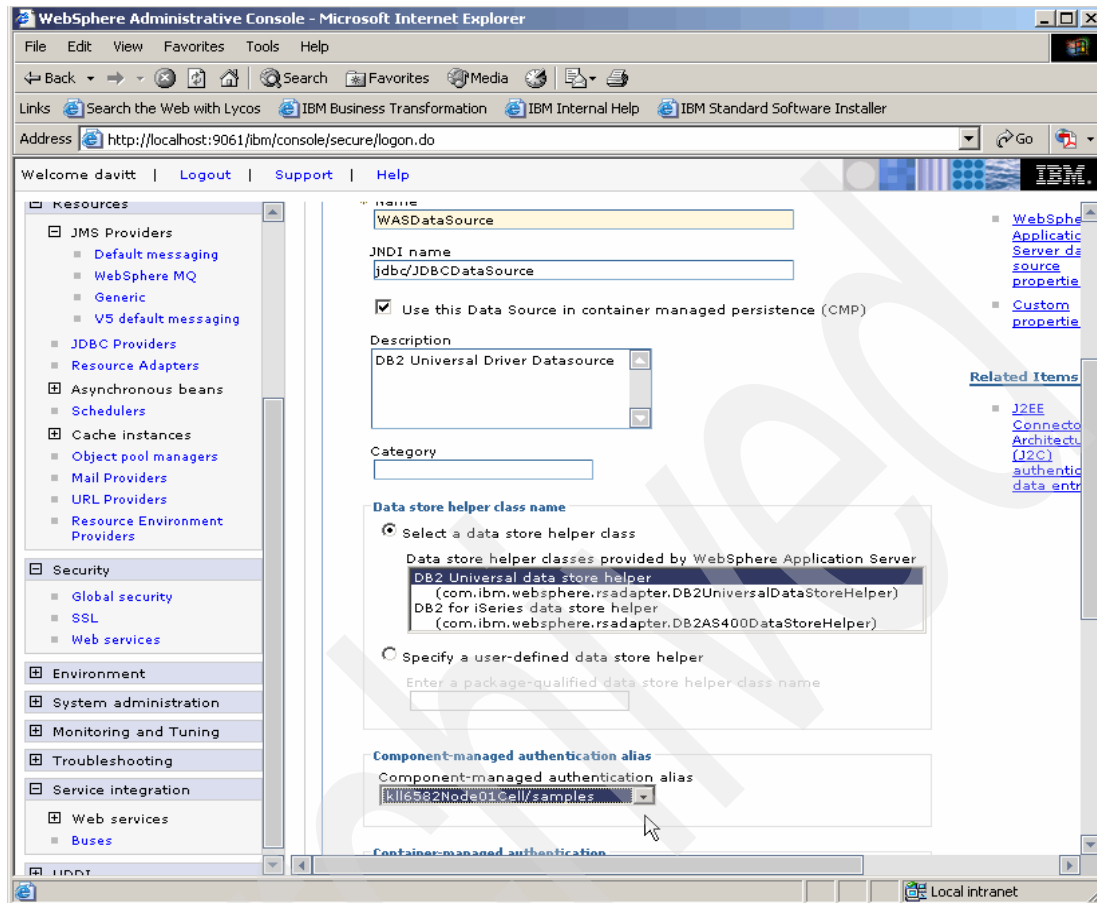


Figure 9-5 Specifying the authentication alias for the messaging data store

### **Example: Incorrect data store for the messaging engine**

In this example the messaging engine again fails during start up with message code CWSIS0002E. However, in this case, the information that is included in the message does not make it clear what happened. Looking at the extract from SystemOut.log, as shown in Example 9-4 on page 319, you can see other errors that appear to precipitate that message code.

Immediately after attempting to take an exclusive lock on the data store you can see the CWSIS1535E message code.

*Example 9-4 Messaging engine startup failure: incorrect data store*

---

```
[...] 0000002d InternalGener I   DSRA8203I: Database product name : DB2/NT
[...] 0000002d InternalGener I   DSRA8204I: Database product version : SQL08020
[...] 0000002d InternalGener I   DSRA8205I: JDBC driver name : IBM DB2 JDBC Universal Driver
Architecture
[...] 0000002d InternalGener I   DSRA8206I: JDBC driver version : 2.3.63
[...] 0000002d WSRdbDataSour I   DSRA8208I: JDBC driver type : 2
[...] 0000002d SibMessage I   [NewBus:k116582Node01.server1-NewBus] CWSIS1538I: The
messaging engine is attempting to obtain an exclusive lock on the data store.
[...] 0000002e SibMessage I   [NewBus:k116582Node01.server1-NewBus] CWSIS1535E: The
messaging engine's unique id does not match that found in the data store.
ME_UUID=884177EB370F543C, ME_UUID(DB)=7E9CB093BBD8794E
[...] 0000002f SibMessage I   [NewBus:k116582Node01.server1-NewBus] CWSIS1519E: Messaging
engine k116582Node01.server1-NewBus cannot obtain the lock on its data store, which ensures it
has exclusive access to the data.
[...] 0000002d SibMessage E   [NewBus:k116582Node01.server1-NewBus] CWSIS0002E: The
messaging engine encountered an exception while starting. Exception:
com.ibm.ws.sib.msgstore.PersistenceException: CWSIS1501E: The data source has produced an
unexpected exception: com.ibm.ws.sib.msgstore.persistence.DatasourceWrapperStateException: New
connections cannot be provided because the persistence layer has been stopped
[...] 0000002d SibMessage E   [NewBus:k116582Node01.server1-NewBus] CWSID0035E: Messaging
engine k116582Node01.server1-NewBus cannot be started; detected error reported during
com.ibm.ws.sib.msgstore.impl.MessageStoreImpl start()
[...] 0000002d SibMessage E   [NewBus:k116582Node01.server1-NewBus] CWSID0027I: Messaging
engine k116582Node01.server1-NewBus cannot be restarted because a serious error has been
reported.
[...] 0000002d SibMessage I   [NewBus:k116582Node01.server1-NewBus] CWSID0016I: Messaging
engine k116582Node01.server1-NewBus is in state Stopped.
[...] 0000002d SibMessage I   [NewBus:k116582Node01.server1-NewBus] CWSID0016I: Messaging
engine k116582Node01.server1-NewBus is in state Joined.
[...] 0000002d SibMessage E   [NewBus:k116582Node01.server1-NewBus] CWSID0039E:
HAManager-initiated activation has failed, messaging engine k116582Node01.server1-NewBus will
be disabled.
```

---

Each messaging object has a unique identifier associated with it that is called a *UUID*. When a messaging engine is created and first accesses the database, the messaging engine registers its UUID in the database SIBOWNER table. When a database has been marked in such a way, it cannot be used by another messaging engine.

The error shown in Example 9-4 was caused by the WebSphere administrator modifying the database name in the messaging data store data source to a name that is already in use by another messaging engine. In this case, the solution is to correct the data source configuration.

Other common scenarios for this type of error include:

- ▶ An administrator deletes an application server or removes it from the bus, and then tries to use the same database when adding a new server to the bus.
- ▶ An administrator creates a bus, deletes it, and then creates a new bus with the same name and configuration, especially if using the default data store (IBM Cloudscape). This often occurs during testing or initial setup.

The solution for these cases is to drop and recreate the database (or tables). If you are using Cloudscape, a quick fix for this is to stop the server, delete the Cloudscape database folder found at `<WAS_install_root>/profiles/<profile_name>/databases/com.ibm.ws.sib/<messaging_engine_name>`, and then restart the server.

You should only do this if you do not need to keep any persistent messages currently stored in the data store.

### ***Example: messaging data store not available J2CS0046E***

This example shows what happens when you start an application server and the database that is associated with the messaging engine data source is not available. The error that is reported by the messaging engine is CWSIS0002E. However, from the abbreviated extract from SystemOut.log shown in Example 9-5, you can see there are multiple messages that are reported from the J2C component.

#### ***Example 9-5 Messaging engine startup failure: data store is not available***

---

```
[7/8/05 10:49:21:283 EDT] 00000028 SibMessage I [NewBus:k116582Node03.server1-NewBus]
CWSID0016I: Messaging engine k116582Node03.server1-NewBus is in state Joined.
[7/8/05 10:49:21:623 EDT] 0000002a SibMessage I [NewBus:k116582Node03.server1-NewBus]
CWSID0016I: Messaging engine k116582Node03.server1-NewBus is in state Starting.
[7/8/05 10:49:26:390 EDT] 0000002a FreePool E J2CA0046E: Method
createManagedConnectionWithMCWrapper caught an exception during creation of the
ManagedConnection for resource jdbc/JDBCDataSource, throwing ResourceAllocationException.
Original exception: com.ibm.ws.exception.WsException: DSRA8100E: Unable to get a
PooledConnection from the DataSource. with SQL State : 57019 SQL Code : -1032
:
:
Caused by: java.sql.SQLException: SQL1032N No start database manager command was issued.
SQLSTATE=57019
DSRA0010E: SQL State = 57019, Error Code = -1,032DSRA0010E: SQL State = 57019, Error Code =
-1,032
... 28 more
Next Linked Exception:
java.sql.SQLException: SQL1032N No start database manager command was issued. SQLSTATE=57019
DSRA0010E: SQL State = 57019, Error Code = -1,032DSRA0010E: SQL State = 57019, Error Code =
-1,032
:
```



```

:
[7/8/05 10:49:26:631 EDT] 0000002a SibMessage      E   [NewBus:k116582Node03.server1-NewBus]
CWSIS0002E: The messaging engine encountered an exception while starting. Exception:
com.ibm.ws.sib.msgstore.PersistenceException: CWSIS1501E: The data source has produced an
unexpected exception: java.sql.SQLException: SQL1032N  No start database manager command was
issued.  SQLSTATE=57019
DSRA0010E: SQL State = 57019, Error Code = -1,032DSRA0010E: SQL State = 57019, Error Code =
-1,032
[7/8/05 10:49:26:651 EDT] 0000002a FreePool          E   J2CA0046E: Method
createManagedConnectionWithMCWrapper caught an exception during creation of the
ManagedConnection for resource jdbc/JDBCDataSource, throwing ResourceAllocationException.
Original exception: com.ibm.ws.exception.WsException: DSRA8100E: Unable to get a
PooledConnection from the DataSource. with SQL State : 57019 SQL Code : -1032
:
:
Caused by: java.sql.SQLException: SQL1032N  No start database manager command was issued.
SQLSTATE=57019
DSRA0010E: SQL State = 57019, Error Code = -1,032DSRA0010E: SQL State = 57019, Error Code =
-1,032

```

---

To resolve this problem:

1. Ensure that the database is started correctly.
2. Attempt to restart the messaging engine.
3. If the messaging engine will not start, stop and restart the application server.
4. Test connectivity to the messaging data store database. (Note that this will not work if the database is secured and the authentication alias was defined at the messaging engine data store level.)

To test using the administrative console:

- a. Under Resources, select **JDBC providers** and click the JDBC provider name.
- b. Under Additional Properties, select **Data sources**.
- c. Select the box to the left of the data source.
- d. Select **Test Connectivity**.

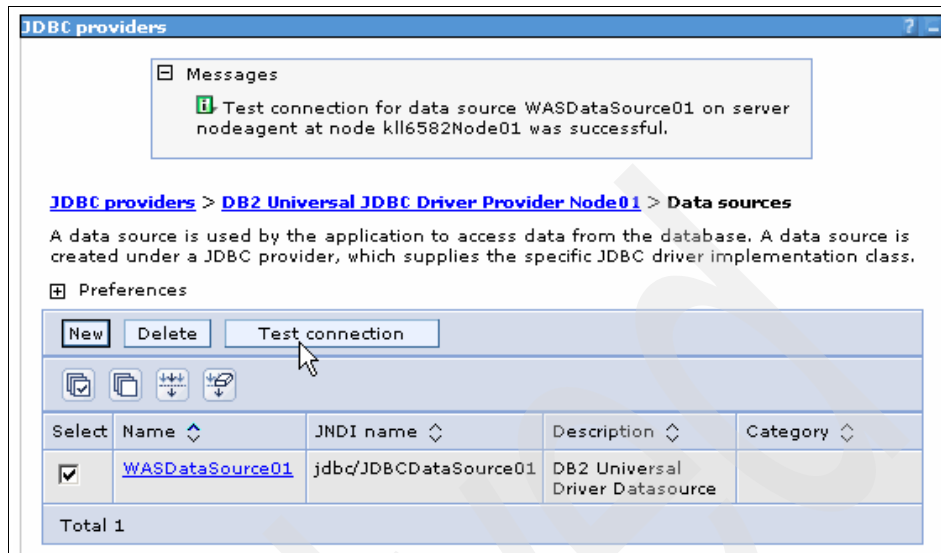


Figure 9-6 Test the data source connection

### Example: Data store tables missing using DB2 for z/OS

It is possible to configure WebSphere Application Server on a distributed system to use IBM DB2 for z/OS® as the messaging data store. If so, you might see the message shown in Example 9-6 during the first attempt to access the data store.

#### Example 9-6 Messaging engine startup failure: DB2 for z/OS data store tables missing

Exception: com.ibm.ws.sib.msgstore.PersistenceException: **CWSIS1501E**: The data source has produced an unexpected exception:java.lang.IllegalStateException: **CWSIS1523E**: Dynamic allocation of database objects in DB2 for z/OS is not allowed.  
com.ibm.ws.sib.utils.ras.SibMessage

When using a DB2 for z/OS messaging data store, the tables are not created dynamically. These tables must be created manually before starting the messaging engine.

The rsibDDLgenerator utility assists in resolving this problem by producing the syntactically correct DDL statements. You can find information about this utility in the WebSphere Information Center:

[http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.pmc.zseries.doc/ref/rjm0630\\_.html](http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.pmc.zseries.doc/ref/rjm0630_.html)

You can redirect the output from this command into a file to submit at to DB2.

Finally, you must ensure that you deselect **Create tables** when specifying the messaging engine data store, as shown in Figure 9-7.

**Buses > NewBus > Messaging engines > kll6582Node01.server1-NewBu > Data store**

The persistent store for messages and other state managed by the messaging engine.

Configuration

**General Properties**

UUID  
BEABDDF5867A4983

\* Data source JNDI name  
jdbc/JDBCDataSource01

Schema name  
ITSO\_SIB

Authentication alias  
(none)

☐ Create tables

Apply OK Reset Cancel

**Related Items**

- J2EE Connector Architecture (J2C) authentication data entries

Figure 9-7 Deselect the Create tables field

### Example: problem is not clear, message CWSID0003E

In Example 9-7, another error has occurred with starting the messaging engine.

#### Example 9-7 Messaging engine startup failure: no clear indicator of problem

```
[...] 0000001b SibMessage I [mename] CWSID0016I: Messaging engine <mename> is in state Joined.
[...] 0000001c SibMessage I [mename] CWSID0016I: Messaging engine <mename> is in state Starting.
[...] 0000001c SibMessage I [mename] CWSIS9999E: Attempting to obtain an exclusive lock on the data
store.
[...] 00000034 SibMessage I [mename] CWSIS9999E: Obtained an exclusive lock on the data store
[...] 0000001c SibMessage E [mename] CWSID0003E: An internal error occurred; reason: Messaging engine
<mename> cannot be started; detected error reported during com.ibm.ws.sib.msgstore.impl.MessageStoreImpl
start()
[...] 0000001c SibMessage E [mename] CWSID0027I: Messaging engine <mename> cannot be restarted
because a serious error has been reported.
[...] 0000001c SibMessage I [mename] CWSID0016I: Messaging engine <mename> is in state Stopped.
[...] 0000001c SibMessage I [mename] CWSID0016I: Messaging engine <mename> is in state Joined.
```

At first sight, this appears to be similar to the error seen in “Example: JNDI error for the messaging data store” on page 312. However, looking prior to that message in the example log, you can also see a message CWSID0003E with further details that indicates a problem with a Java class, as follows:

```
com.ibm.ws.sib.msgstore.impl.MessageStoreImpl start()
```

In this example, there is no clear indication of the exact nature of the problem. So, at this point, it is appropriate to gather suitable documentation and contact the IBM service organization. To determine the documentation that you need to collect, see “The next step” on page 350. Where possible, consider obtaining trace information that might be useful to the service team (see Table 9-2 on page 352).

### 9.3.2 Message flow problems

A variety of things can cause message flow problems. In fact, a failure in any component will stop the flow of messages. This section addresses the following problem symptoms:

- ▶ Symptom: Message consumer fails, possible CWSIA0144E
- ▶ Problems with messages flowing between the bus and WebSphere MQ:
  - Symptom: Messages disappear when using an MQ link
  - Symptom: CWSIC3098I receiving msgs from WebSphere MQ
  - Symptom: Messages not sent across MQ link
- ▶ Symptom: Unrecoverable error from data source DSRA0080E
- ▶ Problems with message-driven beans:
  - Symptom: A message-driven bean has not started
  - Symptom: MDB is not receiving messages
  - Symptom: MDB is causing CNTR0020E message
  - Symptom: An MDB fails and is invoked in an infinite loop
  - Symptom: Message not restored to queue after MDB failure
- ▶ Problems with mediation:
  - Symptom: Mediation of a destination not working
  - Symptom: Mediation fails with CWSIZ0002E, messages disappear
  - Symptom: Mediation implemented but messages disappear

#### **Symptom: Message consumer fails, possible CWSIA0144E**

A common error that can occur is in multi-messaging engine environment where a message consumer appears to be failing to receive messages. This can occur when the messaging engine where the queue has its queue point has stopped. Figure 9-8 on page 325 illustrates a message producer and a message

consumer that are connected to different messaging engines but that use the same destination.

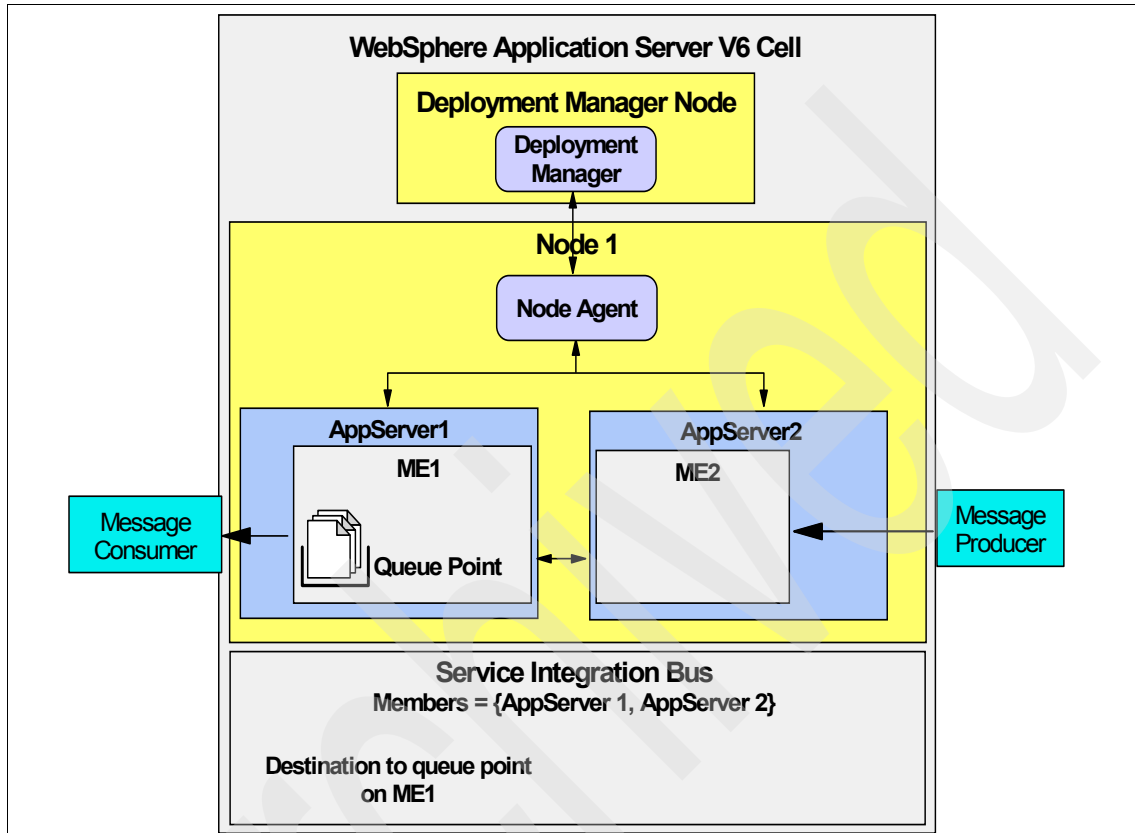


Figure 9-8 Multiple messaging engines using the same destination

In this problem scenario, the message consumer attempts to get messages from a destination and gets a **JMSEException** code (Example 9-8). A message producer attached to a second messaging engine sending messages to the same destination appears unaffected.

Example 9-8 shows the exception code caught by the application.

#### Example 9-8 JMSEException when a consumer tries to get a message

JMSEException caught

Stack Trace:

```
javax.jms.JMSEException: CWSIA0144E: An exception was received during the call to the method
createBrowserSession: com.ibm.websphere.sib.exception.SIResourceException: CWSIC8007E: An
exception was caught from the remote server with Probe Id 3-023-0003. Exception: CWSIP0002E: An
internal messaging error occurred in com.ibm.ws.sib.processor.impl.BrowserSessionImpl, 10,
```

```
com.ibm.ws.sib.msgstore.MessageStoreException: CWSIP0532E: A timeout occurred while remotely
browsing destination cellQ2...
    at
com.ibm.ws.sib.api.jms.impl.JmsQueueBrowserImpl.instantiateBrowser(JmsQueueBrowserImpl.java:575
)
    at com.ibm.ws.sib.api.jms.impl.JmsQueueBrowserImpl.<init>(JmsQueueBrowserImpl.java:183)
    at com.ibm.ws.sib.api.jms.impl.JmsSessionImpl.createBrowser(JmsSessionImpl.java:1489)
    at com.ibm.ws.sib.api.jms.impl.JmsSessionImpl.createBrowser(JmsSessionImpl.java:1439)
```

Example 9-9 shows the stack trace seen from the same error.

#### Example 9-9 Stack trace

```
[...] 00000012 E UOW=null source=com.ibm.ws.sib.utils.ras.SibMessage org=IBM prod=WebSphere
component=Application Server thread=[JFAP TCP Channel : 6]
[:] CWSIC1010E: An internal error occurred. A protocol error has occurred. Unexpected data was
received from the server (kl16582.itso.ral.ibm.com:7276 - BootstrapBasicMessaging). The data ID
was 223
(0xDF).
```

In this case, the messaging engine that holds the queue point for the destination has stopped. This should be relatively easy to see in the administrative console by looking at the status of the messaging engine associated with the failing engine server, as shown in Figure 9-9.

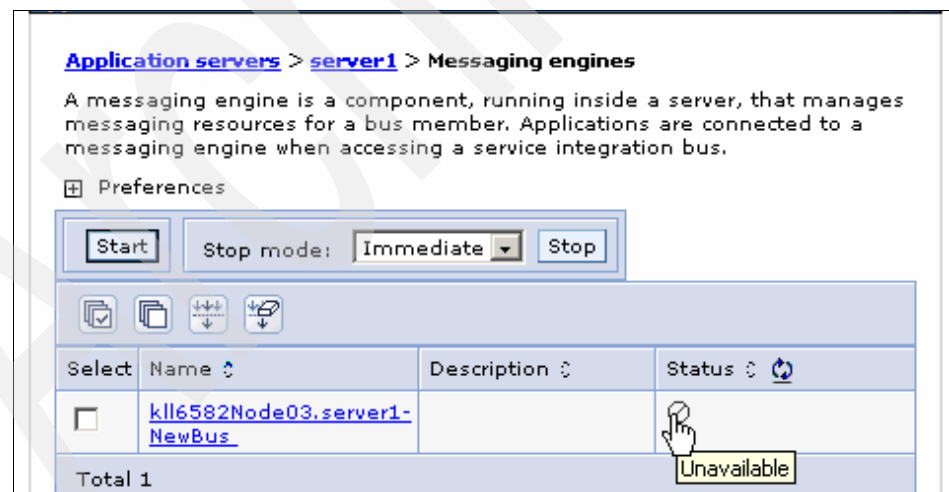


Figure 9-9 Check the status of the messaging engine

Fortunately, the resolution for this problem is relatively easy. Ensure that the messaging engine that is holding the queue point for the failing destination has been started.

## Symptom: Messages disappear when using an MQ link

In this problem scenario, you have defined a connection to WebSphere MQ as a foreign bus. You have two application servers and two messaging engines and a foreign bus and MQLink to connect to WebSphere MQ. You are expecting messages to flow to and from the MQ queue manager but this is not happening.

See Figure 9-10 as an example of the topology.

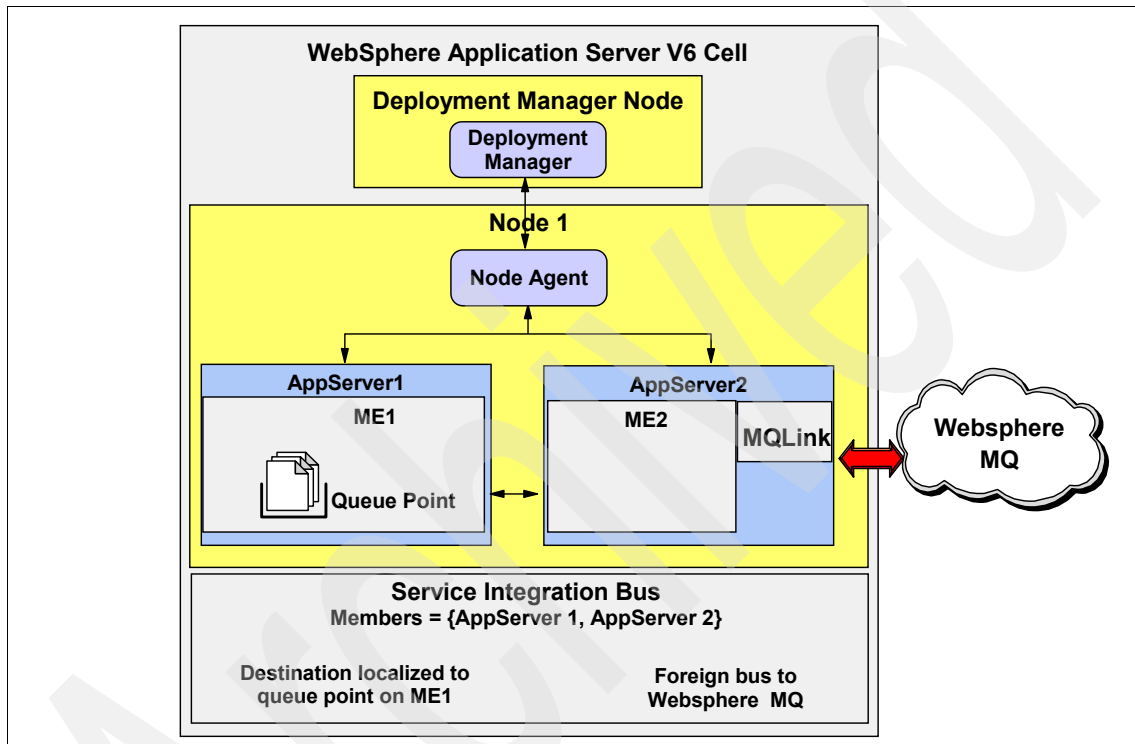


Figure 9-10 Topology of bus connection to WebSphere MQ

**Note:** Be careful when defining the WebSphere MQ link definitions. The Queue manager name field can be confusing. This defines a virtual name given to the current bus so that the remote queue manager can address messages to it.

In this scenario, the basic cause of the problem is that ME1 was restarted. The message producer sends messages using WebSphere MQ to the application on Appserver1. The message producer, WebSphere MQ, and the bus appear to be working fine. However, the messages are not arriving at ME1. Using the

administrative console, you view the SYSTEM.Exception.Queue, but there are no messages. They have apparently disappeared.

The solution to this problem is to stop and restart AppServer2. When this occurs, any messages that are waiting to be transmitted to the target system are sent. After restarting the AppServer2, you can view the destination queue point using the administrative console. Looking at Figure 9-11, you can see that there is a considerable discrepancy between the message time stamp and the current messaging arrival time. This discrepancy was caused while the messages were waiting for AppServer2 to be restarted.

The screenshot shows the WebSphere MQ administrative console interface. At the top, a breadcrumb trail reads: **Buses** > **NewBus** > **Destinations** > **SIB.N3.Q1** > **Queue points** > **SIB.N3.Q1@kl16582Node03.server1-NewBus** > **Messages** > **5000021**. Below this, it says "The messages on the message point." and "Runtime".

The main section is titled "JMS Message properties" and contains the following fields:

- Identifier: 5000021
- State: Unlocked (dropdown menu)
- Transaction ID: (empty field)

Below this is the "Run-time message properties" section with the following fields:

- Message type: JMS
- Approximate length: 2936
- Time stamp: Jul 7, 2005 2:42:21 PM
- Message wait time: 918431
- Current messaging engine arrival time: Jul 7, 2005 2:50:23 PM
- Redelivered count: (empty field)

Figure 9-11 Checking the time stamp of a message

This problem occurs in the early versions of the code and should be fixed in a future version.

### Symptom: CWSIC3098I receiving msgs from WebSphere MQ

In this problem scenario, messages are not being delivered to the application server from WebSphere MQ. No error is seen from the WebSphere MQ side.



However, a message code is seen in the application server SystemOut.log (Example 9-10).

*Example 9-10 Errors receiving messages from WebSphere MQ*

---

```
[...] 00000018 SibMessage   I   [:] CWSIC3098I: While receiving message from queue manager
QM_MQ53_QMGR down WebSphere MQ link MyMQLink one or more messages were put to the exception
destination.
[...] 00000018 SibMessage   W   [:] CWSIP0291W: An attempt to send a message to exception
destination of REMOTE.SIB.Q on messaging engine k116582Node02.server1-NewBus failed.
```

---

In this case, the remote queue manager is attempting to send messages to a destination called REMOTE.SIB.Q. Unfortunately, no such destination exists on the bus, and the messages are routed to the SYSTEM.Exception.Destination queue for this messaging engine.

The resolution is to either correct the WebSphere MQ definitions to reference a valid destination on the bus, or to create a destination and queue point that match that which is used by WebSphere MQ. Another option is to create an alias destination in the bus with that name to redirect the messages to the proper bus destination.

### **Symptom: Messages not sent across MQ link**

In a topology similar to that shown in Figure 9-10 on page 327, problems can arise when sending messages from a bus to a WebSphere MQ provider. One such case would be when the network connection between the two systems is lost.

Unfortunately, there are no messages in the SystemOut.log to indicate this event, and the WebSphere administrative console does not show any change of status for the MQ link sender channel (see Figure 9-12 on page 330).

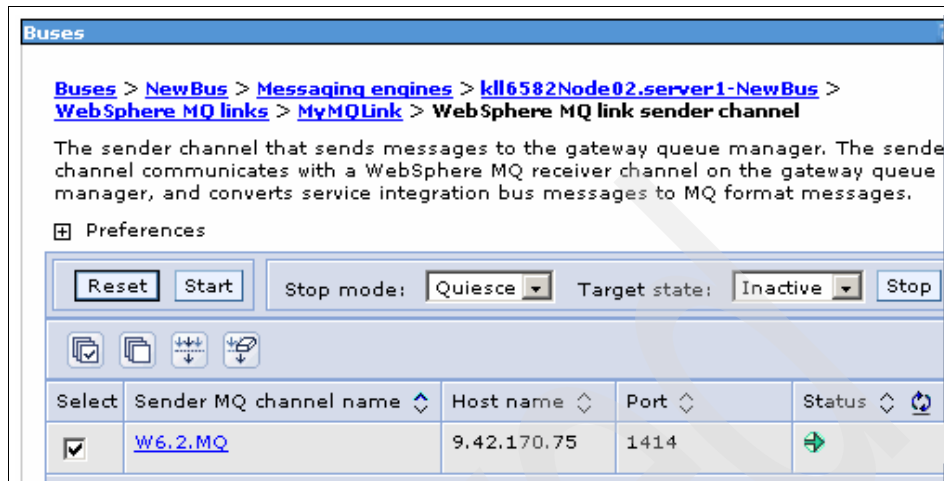


Figure 9-12 Sender MQ channel status

It is possible for you to check the link from WebSphere MQ by viewing the status of the MQ channel.

The resolution from the WebSphere Application Server perspective is to:

1. Correct the network or link error that is causing the problem (this might involve obtaining TCP/IP documentation to see what causes the network to fail).
2. Stop and start the messaging engine that uses the MQ link (it is not sufficient to stop and start the sender channel or the MQ link).

### Symptom: Unrecoverable error from data source (DSRA0080E)

The messaging engines might use one of several different database providers for the storage of message data. This dependence on the database can cause a problem if there is an unrecoverable error from the database. For example:

- ▶ Network access error
- ▶ Catastrophic media error
- ▶ Accidental database closure

Example 9-11 on page 331 shows an extract from SystemOut.log that shows such an error.

Initially, the error DSRA0080E is reported from the message data store component followed by J2CA0056I. Each of these messages is repeated several times. This sequence is then followed by message codes CWSIS1546I and

CWSIS1538I, which indicate that the lock on the database has been lost and repeated attempts are being made to recover the lock.

Finally the application server JVM is ended to avoid any possible data loss. See message HMGR0130I. The application server is terminated in a orderly way.

---

*Example 9-11 Message data store errors*

---

```
:
:
[...] 00000038 WSRdbManagedC W   DSRA0080E: An exception was received by the Data Store
Adapter. See original exception message: com.ibm.db2.jcc.b.SqlException: invalid operation:
connection closed
    at com.ibm.db2.jcc.b.o.yb(o.java:3433)
    at com.ibm.db2.jcc.b.o.getAutoCommit(o.java:1006)
    at com.ibm.db2.jcc.b.wb.getAutoCommit(wb.java:154)
    at
com.ibm.ws.rsadapter.spi.WSRdbManagedConnectionImpl.introspectSelf(WSRdbManagedConnectionImpl.j
ava:1108)
    at com.ibm.ws.rsadapter.FFDCLogger.introspect(FFDCLogger.java:169)
    at com.ibm.ws.rsadapter.jdbc.WSJdbcConnection.introspectSelf(WSJdbcConnection.java:1511)
    at com.ibm.ws.rsadapter.jdbc.WSJdbcObject.introspectSelf(WSJdbcObject.java:355)
    at
com.ibm.ws.ffdc.IntrospectionLevelMember.getNextMembers(IntrospectionLevelMember.java:451)
    at com.ibm.ws.ffdc.IntrospectionLevel.getNextLevel(IntrospectionLevel.java:181)
    at com.ibm.ws.ffdc.ObjectIntrospectorImpl.dumpContents(ObjectIntrospectorImpl.java:67)
    at com.ibm.ws.ffdc.ObjectIntrospectorImpl.dumpContents(ObjectIntrospectorImpl.java:51)
    at com.ibm.ws.ffdc.IncidentStreamImpl.introspectAndWrite(IncidentStreamImpl.java:427)
    at com.ibm.ws.ffdc.IncidentStreamImpl.introspectAndWriteLine(IncidentStreamImpl.java:663)
    at com.ibm.ws.ffdc.DiagnosticEngine.dumpObjectAndStack(DiagnosticEngine.java:311)
    at com.ibm.ws.ffdc.DiagnosticEngine.processIncident(DiagnosticEngine.java:152)
    at com.ibm.ws.ffdc.FFDCFilter.filterEngine(FFDCFilter.java(Compiled Code))
    at com.ibm.ws.ffdc.FFDCFilter.processException(FFDCFilter.java(Inlined Compiled Code))
    at
com.ibm.ws.rsadapter.jdbc.WSJdbcPreparedStatement.executeQuery(WSJdbcPreparedStatement.java(Com
piled Code))
    at
com.ibm.ws.sib.msgstore.persistence.impl.MEOwnerTable.readOwningME(MEOwnerTable.java(Compiled
Code))
    at
com.ibm.ws.sib.msgstore.persistence.lock.DBLockingThread.waitAndRefreshLock(DBLockingThread.jav
a(Compiled Code))
    at
com.ibm.ws.sib.msgstore.persistence.lock.DBLockingThread.run(DBLockingThread.java(Compiled
Code))
.
:
[...] 00000038 ConnectionEve A   J2CA0056I: The Connection Manager received a fatal connection
error from the Resource Adaptor for resource jdbc/JDBCDataSource. The exception which was
```

```

received is com.ibm.websphere.ce.cm.StaleConnectionException: A communication error has been
detected. Communication protocol being used: UWReply.fill(). Communication API being used:
recvBuff(). Location where the error was detected: ReasonCode=71. Communication function
detecting the error: *. Protocol specific error codes(s) TCP/IP SOCKETS
[...] 00000038 MCWrapper      E   J2CA0081E: Method cleanup failed while trying to execute
method cleanup on ManagedConnection WSRdbManagedConnectionImpl@5915eba0 from resource No longer
available. Caught exception: com.ibm.ws.exception.WsException: DSRA1130E: A fatal connection
error occurred on another connection while this connection was active. This connection cannot
be reset to a usable state.
    at
com.ibm.ws.rsadapter.exceptions.DataStoreAdapterException.<init>(DataStoreAdapterException.java
:226)
    at
com.ibm.ws.rsadapter.exceptions.DataStoreAdapterException.<init>(DataStoreAdapterException.java
:177)
    at com.ibm.ws.rsadapter.AdapterUtil.createDataStoreAdapterException(AdapterUtil.java:232)
    at
com.ibm.ws.rsadapter.spi.WSRdbManagedConnectionImpl.cleanup(WSRdbManagedConnectionImpl.java:301
2)
    at com.ibm.ejs.j2c.MCWrapper.cleanup(MCWrapper.java:1343)
    at com.ibm.ejs.j2c.poolmanager.FreePool.cleanupAndDestroyMCWrapper(FreePool.java:627)
    at
com.ibm.ejs.j2c.poolmanager.FreePool.removeCleanupAndDestroyAllFreeConnections(FreePool.java:18
17)
    at com.ibm.ejs.j2c.poolmanager.PoolManager.fatalErrorNotification(PoolManager.java:1082)
    at com.ibm.ejs.j2c.MCWrapper.connectionErrorOccurred(MCWrapper.java:1981)
    at
com.ibm.ejs.j2c.ConnectionEventListener.connectionErrorOccurred(ConnectionEventListener.java:31
3)
    at
com.ibm.ws.rsadapter.spi.WSRdbManagedConnectionImpl.processConnectionErrorOccurredEvent(WSRdbMa
nagedConnectionImpl.java:1989)
    at
com.ibm.ws.rsadapter.jdbc.WSJdbcConnection.fireConnectionErrorEvent(WSJdbcConnection.java:1253)
    at com.ibm.ws.rsadapter.jdbc.WSJdbcUtil.mapException(WSJdbcUtil.java:874)
    at
com.ibm.ws.rsadapter.jdbc.WSJdbcPreparedStatement.executeQuery(WSJdbcPreparedStatement.java(Com
piled Code))
    at
com.ibm.ws.sib.msgstore.persistence.impl.MEOwnerTable.readOwningME(MEOwnerTable.java(Compiled
Code))
    at
com.ibm.ws.sib.msgstore.persistence.lock.DBLockingThread.waitAndRefreshLock(DBLockingThread.jav
a(Compiled Code))
    at
com.ibm.ws.sib.msgstore.persistence.lock.DBLockingThread.run(DBLockingThread.java(Compiled
Code))
:
:

```

```
[...] 00000037 SibMessage I [NewBus:k116582Node03.server1-NewBus] CWSIS1546I: The
messaging engine, ME_UUID=7E9CB093BBD8794E, INC_UUID=0ee96bb71181170a, has lost an existing
lock or failed to gain an initial lock on the data store.
[...] 00000037 SibMessage I [NewBus:k116582Node03.server1-NewBus] CWSIS1546I: The
messaging engine, ME_UUID=7E9CB093BBD8794E, INC_UUID=0ee96bb71181170a, has lost an existing
lock or failed to gain an initial lock on the data store.
[...] 00000037 SibMessage I [NewBus:k116582Node03.server1-NewBus] CWSIS1538I: The
messaging engine, ME_UUID=7E9CB093BBD8794E, INC_UUID=0ee96bb71181170a, is attempting to obtain
an exclusive lock on the data store.
:
:
[...] 00000037 SibMessage I [NewBus:k116582Node03.server1-NewBus] CWSIS1538I: The
messaging engine, ME_UUID=7E9CB093BBD8794E, INC_UUID=0ee96bb71181170a, is attempting to obtain
an exclusive lock on the data store.
[...] 00000032 SibMessage E [NewBus:k116582Node03.server1-NewBus] CWSID0046E: Messaging
engine k116582Node03.server1-NewBus detected an error and cannot continue to run in this
server.
[...] 00000032 HAGroupImpl I HMGR0130I: The local member of group
WSAF_SIB_BUS=NewBus,WSAF_SIB_MESSAGING_ENGINE=k116582Node03.server1-NewBus,type=WSAF_SIB has
indicated that is it not alive. The JVM will be terminated.
[...] 00000032 SystemOut 0 Panic:component requested panic from isAlive
:
:
[...] 000009fa ServerCollabo A WSVR0024I: Server server1 stopped
```

---

Naturally, such errors should be rare, and the resolution is to investigate and correct the error with the database.

## Problems with message-driven beans

Message-driven beans (MDBs) are EJBs that act as message consumers. When a message producer is putting messages on a queue, but those messages are not taken off the queue by the MDB, you will see the messages collect on the queue.

### ***Symptom: A message-driven bean has not started***

When an MDB does not start, verify that the configuration is correct. For a sample of how to set up and configure an MDB, see:

[http://www.ibm.com/developerworks/websphere/techjournal/0504\\_reinitz/0504\\_reinitz.html](http://www.ibm.com/developerworks/websphere/techjournal/0504_reinitz/0504_reinitz.html)

One indication that the MDB is correctly configured and that the messaging engine has started is the message CWSIV0764I in the SystemOut.log.

*Example 9-12 MDB startup messages in SystemOut.log*

---

```
[...] 00000033 SibMessage I [NewBus:k116582Node03.server1-NewBus] CWSIS1537I: The
messaging engine, ME_UUID=7E9CB093BBD8794E, INC_UUID=52e3ef4111668436, has acquired an
exclusive lock on the data store.
[...] 0000002e SibMessage I [NewBus:k116582Node03.server1-NewBus] CWSIP0212I: messaging
engine k116582Node03.server1-NewBus on bus NewBus is starting to reconcile the WCCM destination
and link configuration.
[...] 0000002e SibMessage I [NewBus:k116582Node03.server1-NewBus] CWSIP0213I: messaging
engine k116582Node03.server1-NewBus on bus NewBus has finished reconciling the WCCM destination
and link configuration.
[...] 0000002e SibMessage I [NewBus:k116582Node03.server1-NewBus] CWSIV0764I: A consumer
has been created for a message-driven bean against destination [destinationname] on bus NewBus
following the activation of messaging engine k116582Node03.server1-NewBus.
```

---

Note that the CWSIV0764I message is only seen if the messaging engine starts after the application that contains the MDB (there will have been a corresponding CWSIV0759W when the application started). If the messaging engine is started before the application, you will not see the informational message even if there are no configuration problems.

***Symptom: MDB is not receiving messages***

By default, no SystemOut.log messages are produced when an MDB handles a received message. With this in mind, it is advisable that you add some form of logging to an application so that you can recognize that messages are being processed successfully.

If the MDB fails to handle a message, the messaging engine attempts to redeliver the message based on the destination properties of the queue. These can be set or displayed by selecting **Service Integration** → **Buses**. Click the bus name to display its properties. Then, click **Destinations** to display a list of queue and topic destinations. Finally, click the queue destination. See Figure 9-13 on page 335.

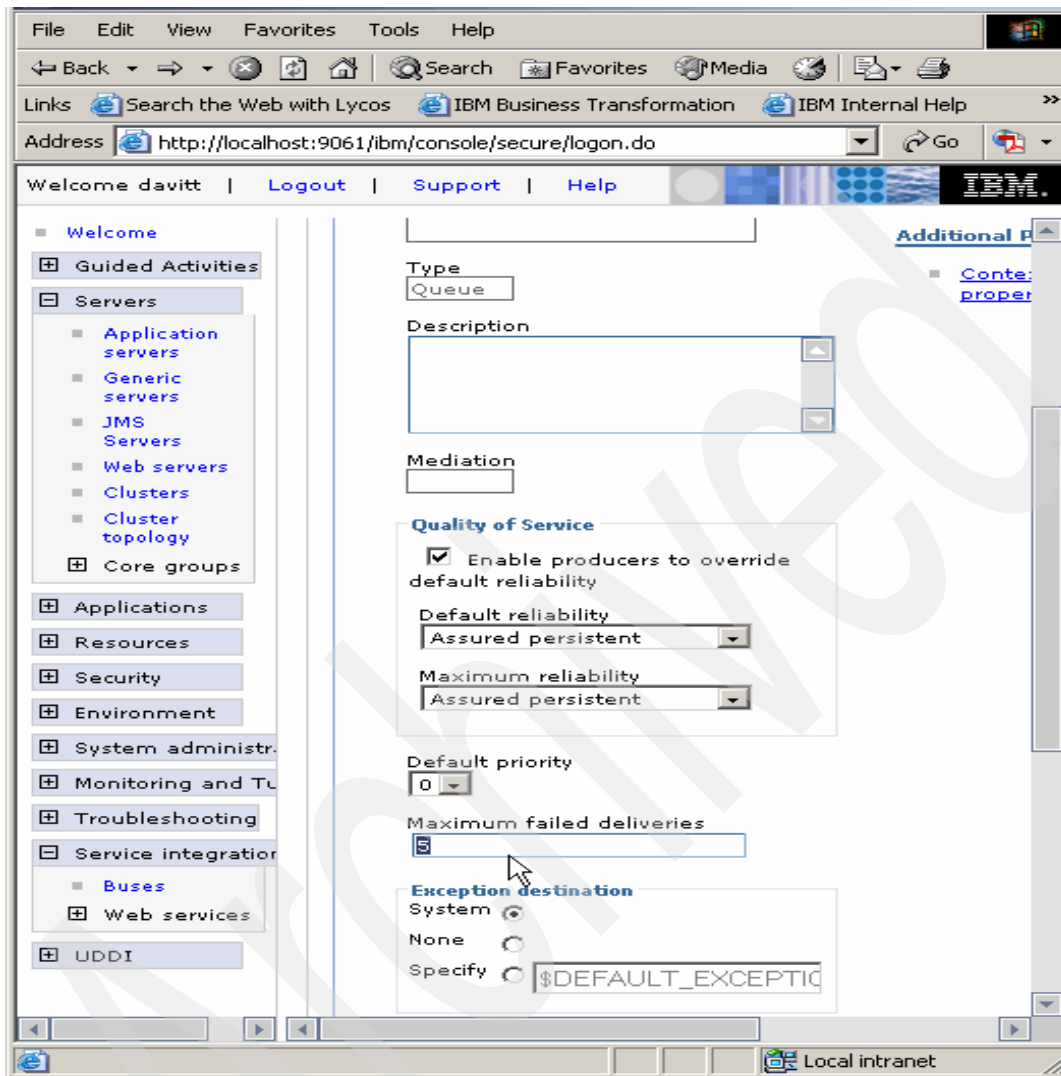


Figure 9-13 Queue destination properties

If the message cannot be processed, it is redelivered based on the *maximum failed deliveries* value.

If this value is exceeded, the messaging engine attempts to deliver the message to the *exception destination* address. By default, this address uses a destination that is generic for that messaging engine. The format of the exception destination is as follows:

SYSTEM.Exception.Destination.<messaging\_engine\_name>

If required, you can define your own exception destination specific for the queue that is processing the MDB messages.

The resolution in this case is to make sure the destination definition is correct. If it is, check the exception destination queue in case the message has been delivered there.

***Symptom: MDB is causing CNTR0020E message***

When an MDB experiences an unexpected exception code this is recognized by the application server. The exception is captured and a message code added to SystemOut.log (see Example 9-13).

***Example 9-13 MDB exception***

---

```
[...] 00000066 ExceptionUtil E    CNTR0020E: EJB threw an unexpected (non-declared) exception
during invocation of method "onMessage" on bean
"BeanId(PackageReceivedEAR#PackageReceived.jar#PackageReceived, null)". Exception data:
java.lang.RuntimeException: MDB error
    at receiver.PackageReceivedBean.onMessage(Unknown Source)
    at
com.ibm.ejs.container.MessageEndpointHandler.invokeMdbMethod(MessageEndpointHandler.java:990)
    at com.ibm.ejs.container.MessageEndpointHandler.invoke(MessageEndpointHandler.java:723)
    at $Proxy0.onMessage(Unknown Source)
    at
com.ibm.ws.sib.api.jmsra.impl.JmsJcaEndpointInvokerImpl.invokeEndpoint(JmsJcaEndpointInvokerImp
l.java:201)
    at com.ibm.ws.sib.ra.inbound.impl.SibRaDispatcher.dispatch(SibRaDispatcher.java:544)
    at
com.ibm.ws.sib.ra.inbound.impl.SibRaSingleProcessListener$SibRaWork.run(SibRaSingleProcessListe
ner.java:403)
    at com.ibm.ejs.j2c.work.WorkProxy.run(WorkProxy.java:434)
    at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1455)
```

---

It is the responsibility of the MDB application to handle exceptions. The solution in this case is for the MDB to gracefully handle exceptions.

***Symptom: An MDB fails and is invoked in an infinite loop***

An exception destination is used to handle messages that cannot be delivered to their intended bus destination. In this scenario, you have defined that no exception destination be associated with the bus destination (see Figure 9-14 on page 337).



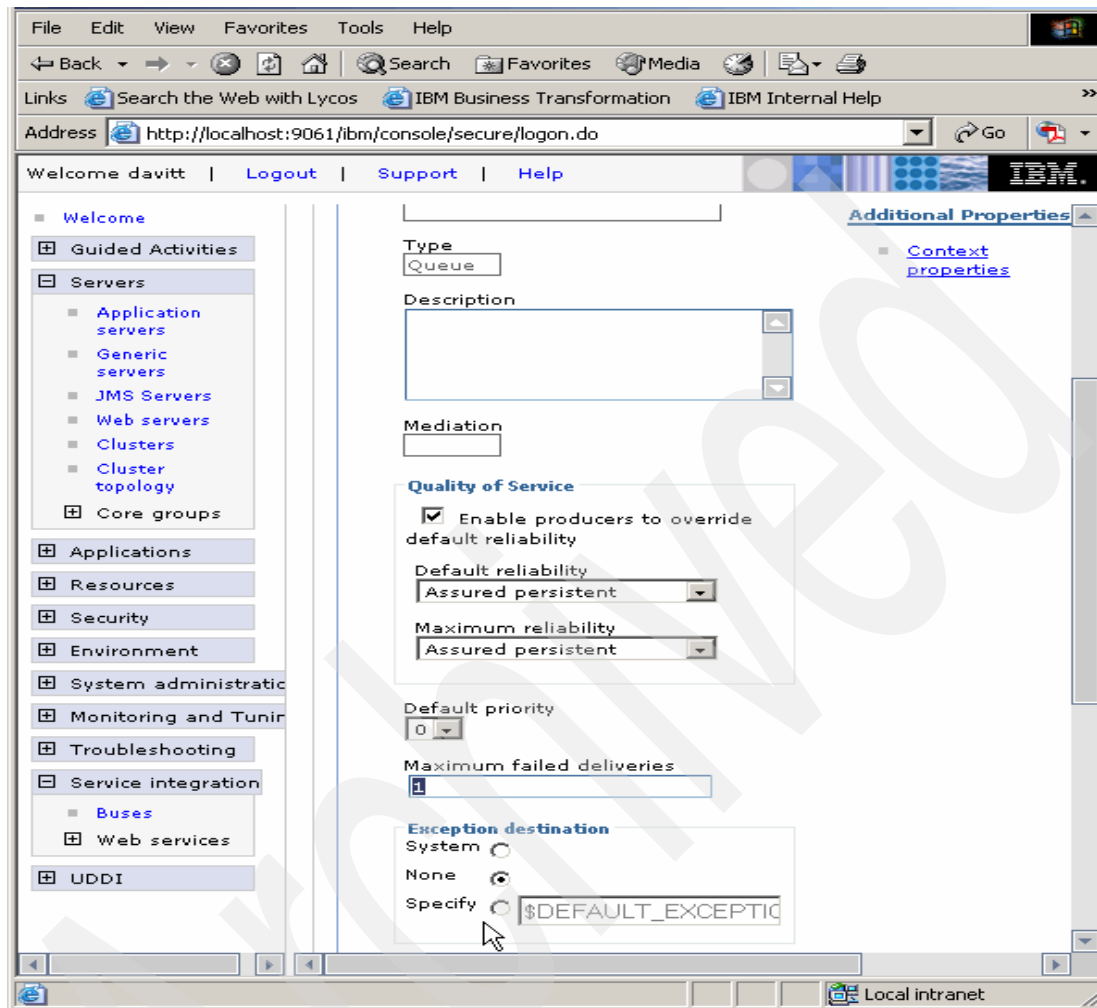


Figure 9-14 Exception destination is set to None

In many customer environments, defining an MDB on a queue with no exception destination is intentional. It is done to ensure that the message order is preserved for delivery to the MDB from the queue (if one message cannot be processed, they do not want any following messages to be processed).

When no exception destination is used and a message cannot be delivered (for example, the MDB fails), the *Maximum failed deliveries* limit specified for the destination is ignored, and the MDB is repeatedly driven to consume the message. This situation continues until either the message is removed from the destination (for example, by an administrator using the WebSphere

administrative console), or the MDB is able to handle the message. The problem cannot be corrected by setting *Discard messages*.

Be sure that the MDB is performing as expected. Also consider modifying the modify the destination properties to use another exception destination mechanism.

***Symptom: Message not restored to queue after MDB failure***

The MDB transaction-type, set in the EJB module deployment descriptor, can be container-managed or bean-managed. A bean-managed transaction is responsible for committing or rolling back the transaction. It does this by the use of the UserTransaction interface. If the MDB is bean managed and does not correctly cope with transactions and exceptions, messages can end up “lost.”

Refer to Chapter 10 of *WebSphere Application Server V6 System Management and Configuration Handbook*, SG24-6451, for details.

**Problems with mediation**

You can configure mediations in two ways:

- ▶ A single mediation handler associated with a destination
- ▶ A mediation list associated with a destination

If you have implemented a mediation list (a sequence of mediations), you should ensure that:

- ▶ They are called in the correct sequence.
- ▶ Each mediation returns correctly.

A single mediation or a mediation handler list can be associated with multiple destinations. You should ensure that only those destinations that are required are associated with the mediation. Otherwise, the mediation handler might fail to recognize the message format for that particular destination. Finally, each of the mediations might require its own diagnostic output to ensure that they are working correctly.

Figure 9-15 on page 339 illustrates how mediations are associated with destinations.

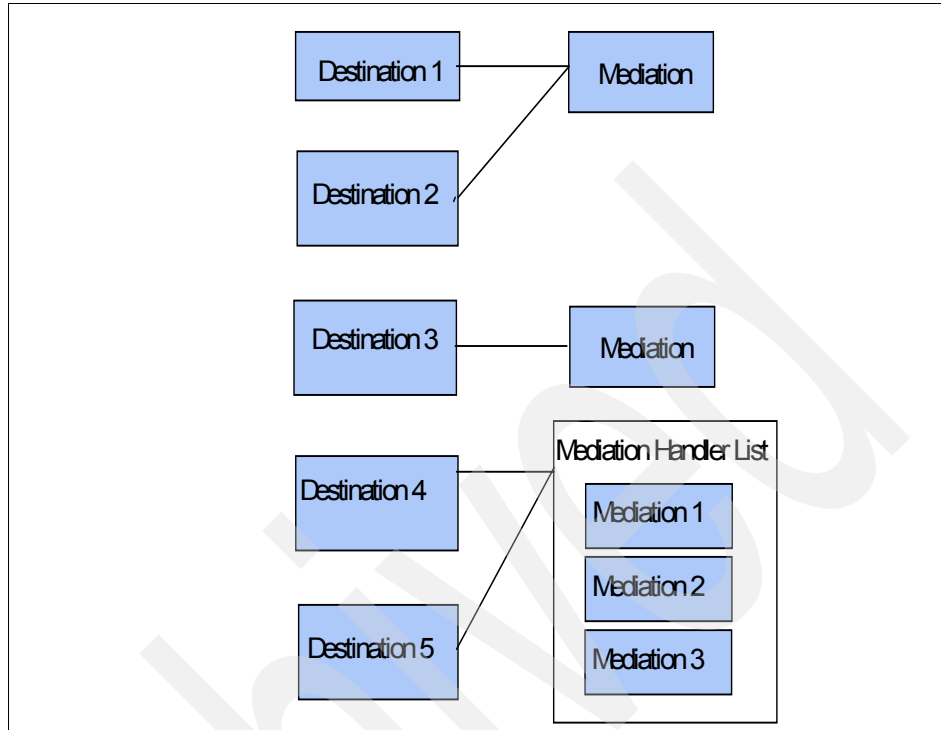


Figure 9-15 Mediation

**Symptom: Mediation of a destination not working**

Mediation is implemented using a stateless EJB. Before a mediation will work properly, the infrastructure must be set up correctly. To check the mediation EJB status from the administrative console, select **Applications** → **Enterprise applications** and ensure that any mediation EJBs are started correctly.

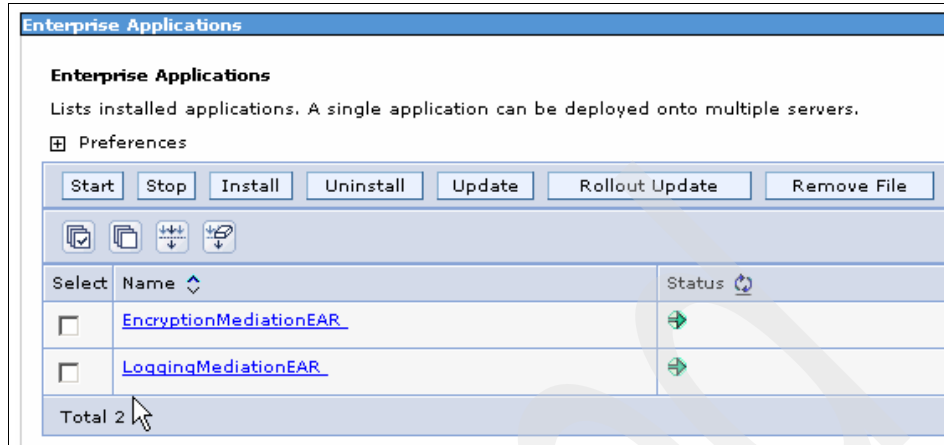


Figure 9-16 Check the mediation application

If the mediations are started correctly you must ensure they have started on the correct node. This is a common error. Examine the SystemOut.log for the node where the mediation is implemented.

You should see messages WSVR0200I and WSVR0221I for each mediation that is started. See Example 9-14.

Example 9-14 Log showing mediations starting

```
[...] 0000001d ApplicationMg A WSVR0200I: Starting application: LoggingMediationEAR
[...] 0000001e ApplicationMg A WSVR0200I: Starting application: EncryptionMediationEAR
[...] 0000001e EJBContainerI I WSVR0207I: Preparing to start EJB jar: EncryptionMediation.jar
[...] 0000001d EJBContainerI I WSVR0207I: Preparing to start EJB jar: LoggingMediation.jar
[...] 0000001e EJBContainerI I WSVR0037I: Starting EJB jar: EncryptionMediation.jar
[...] 0000001d EJBContainerI I WSVR0037I: Starting EJB jar: LoggingMediation.jar
[...] 0000001e ApplicationMg A WSVR0221I: Application started: EncryptionMediationEAR
[...] 0000001d ApplicationMg A WSVR0221I: Application started: LoggingMediationEAR
```

If these do not appear, then you should verify that the mediation is configured to the correct node.

When installing the mediation EJB with the administrative console:

1. Select **Applications** → **Enterprise applications**.
2. Select the **Install** option.
3. Browse and select the correct **Local or Remote file system** for the mediation ear file.
4. Continue to **Map modules to servers**.

5. Ensure that **Select Module name** is selected.
6. Also ensure that in the **Clusters and Servers** drop down the correct node name and server have been selected.
7. Apply the correct options for both. See Figure 9-17 for details.
8. Continue and **Finish** the installation of the EJB.

**Install New Application**

Specify options for installing enterprise applications and modules.

**Step 1**  
Select installation options

**→ Step 2: Map modules to servers**

**Step 3**  
Provide options to perform the EJB Deploy

**Step 4**  
Provide JNDI Names for Beans

**Map modules to servers**

Specify targets such as application servers or clusters of application servers where you want to install the application. Modules can be installed on the same application server or dispersed among several application servers. Specify the Web servers as targets that will serve as routers for requests to this application. The plugin-cfg.xml for each Web server is generated based on the applications which are routed through the Web server.

**Clusters and Servers:**

WebSphere:cell=kll6582Cell01,node=kll6582Node01,server=server1  
 WebSphere:cell=kll6582Cell01,node=kll6582Node02,server=server1  
 WebSphere:cell=kll6582Cell01,node=kll6582Node03,server=server1

**Apply**

Select	Module	URI	Server
<input checked="" type="checkbox"/>	EncryptionMediation	EncryptionMediation.jar,META-INF/ejb-jar.xml	WebSphere:cell=kll6582Cell01,node=kll6582Node01,server=server1

Figure 9-17 Map module to server

After the EJB has been correctly defined, start the EJB from the **Applications** → **Enterprise applications** panel and verify in the SystemOut.log of the target application server that it has started correctly. Refer again to Example 9-14 on page 340.

**Note:** A common error made when implementing mediations is incorrectly specifying the mediation handler list name when defining the mediation in the administrative console. The mediation handler list is defined in the deployment descriptor for the EJB and must be typed in when you create the mediation.

The following messages can be seen when in the log when the names do not match:

CWSIP0655E: The local mediation point for destination firstDestination failed to start because an error occurred. Mediation MyFirstMediation was not found. The error is CWSIZ0002E: The mediation named MyFirstMediation that is attached to destination firstDestination is defined to use mediation handler list RouterHandler. However this handler list does not exist.

### **Symptom: Mediation fails with CWSIZ0002E, messages disappear**

Messages can apparently disappear for a variety of reasons. One possibility is that a mediation has failed to start correctly. Typically, this can be seen in the SystemOut.log as CWSIZ0002E message codes. See Example 9-15.

#### *Example 9-15 A mediation fails to start*

---

```
[...] 0000002f SibMessage    I    [busname] CWSID0016I: Messaging engine
k116582Node02.server1-NewBus is in state Started.
[...] 0000002f SibMessage    E    [busname] CWSIZ0002E: The mediation named compressionMediation
that is attached to destination [DestinationName] is defined to use mediation handler list
[mediationListName]. However this handler list does not exist.
[...] 0000002f SibMessage    I    [busname] CWSIZ0052I: Mediation handler lists defined in the
server are [].
```

---

In this scenario, the mediation was defined correctly, and the destination was correctly modified to be mediated. However, the mediation handler was not correctly loaded. Messages directed to the destination will not be delivered but will wait at the mediation point that holds messages until they have been mediated. You can confirm this by viewing the mediation point.

In this example, destination cellQ2 is mediated by handler compressionMediation. However, messages sent to that destination are not arriving as expected. You would now suspect messages should be held prior to mediation at the mediation point. You can verify that in the administrative console by doing the following.

1. Select **Buses** and click the bus name.
2. Under the Additional Properties section, select **Destinations**.
3. Select the mediated destination in this example **cellQ2**.
4. Select **Mediation point**.
5. Select the mediation point name.
6. Select the **Runtime** tab.
7. Select **Messages**.

Figure 9-18 on page 343 shows a list of any messages awaiting mediation before possible delivery to the destination.

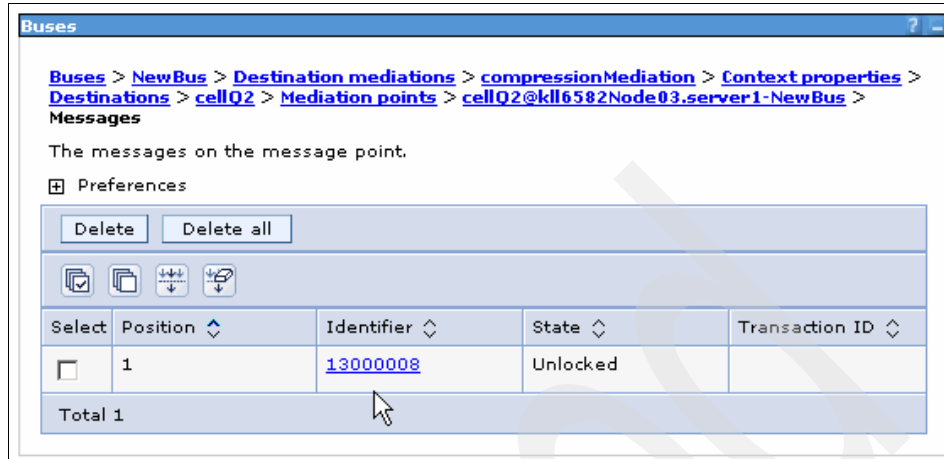


Figure 9-18 Messages waiting for mediation

This scenario is only one in which messages seem to disappear. There are others which you should also investigate, such as the following:

- ▶ The message has expired.
- ▶ The mediation might have discarded the message (see “Symptom: Mediation implemented but messages disappear” on page 343).
- ▶ The messaging engine might have discarded the message based on its reliability.
- ▶ The message might have been directed to an exception destination.

### **Symptom: Mediation implemented but messages disappear**

Mediation depends on user written code and sometimes puts the reliability of message delivery outside the control of the messaging engine. If the mediation has been correctly implemented and started, it is quite possible that the mediation code processes and removes a message from the target destination.

The mediation code is implemented using the MediationHandler interface that uses a handle method.

#### *Example 9-16 Mediation handler that discards a message*

```
public class LoggingMediation implements MediationHandler {
    /* (non-Javadoc)
     * @see
     com.ibm.websphere.sib.mediation.handler.MediationHandler#handle(javax.xml.rpc.handler.MessageCo
     ntext)
     */
    public boolean handle(MessageContext arg0) throws MessageContextException {
```

```

:
:
:
    return false; // FALSE indicates that the message should be discarded!!
}

```

---

By default, there are no messages written to `SystemOut.log` to indicate that the message has been discarded. You should carefully implement your mediation handlers to output appropriate diagnostic messages when it returns false.

### 9.3.3 Application configuration and resource problems

In this section, we look at some of the errors that can happen when an application attempts to use the default messaging provider. These errors include configuration errors and resource problems as follows:

- ▶ “Symptom: V5 application gets `JMSEException MQJMS2005`” on page 344
- ▶ “Symptom: `ServiceUnavailableException` on bus connect” on page 346
- ▶ “Symptom: `JMSEException` on bus connect, msg `CWSIT0006E`” on page 347
- ▶ “Symptom: Bus connect fails, message `CWSIT0019E`” on page 348
- ▶ “Symptom: JMS application returns `MQJMS2007`” on page 349

#### Symptom: V5 application gets `JMSEException MQJMS2005`

It is common to find configuration problems during the migration of WebSphere Application Server V5 and WebSphere MQ JMS V5.3 applications to WebSphere Application Server V6 messaging. An existing WebSphere MQ JMS client application would be configured to use the WebSphere MQ V5.3 queue manager name, channel name, host name, and port number to access the JMS objects.

Example 9-17 shows an application that is migrated to V6 and has experienced an exception `MQJMS2005`.

#### *Example 9-17 JMSEException MQJMS2005*

---

```

javax.jms.JMSEException: MQJMS2005: failed to create MQQueueManager for 'hostname:servername'
    at com.ibm.mq.jms.services.ConfigEnvironment.newException(ConfigEnvironment.java:569)
    at com.ibm.mq.jms.MQConnection.createQM(MQConnection.java:2311)
    at com.ibm.mq.jms.MQConnection.createQMNonXA(MQConnection.java:1739)
    at com.ibm.mq.jms.MQQueueConnection.<init>(MQQueueConnection.java:144)
    at com.ibm.mq.jms.MQQueueConnection.<init>(MQQueueConnection.java:54)
    at com.ibm.mq.jms.MQQueueConnectionFactory.createQueueConnection(MQQueueConnectionFactory.java:106)
    at JMSPutClient.main(JMSPutClient.java:79)

```

---



To connect correctly to the bus, the application should present correct values to the application server. Table 9-1 shows the connection parameters to check and where you can find them.

*Table 9-1 Connection parameters*

Property	Location
Queue manager name	WebSphere MQ Client link
Host name / IP address	Network configuration
Port number	SIB_MQ_ENDPOINT_ADDRESS or SIB_MQ_ENDPOINT_SECURE_ADDRESS
Channel name	WebSphere MQ Client link
Queue / Topic name (destination)	Messaging engine / endpoint definition

These values can be configured in the connection factory, queue, and topic objects that are looked up through JNDI. The JNDI values or those used by the application code should be cross-referenced with those that are used by the application server.

When the JMS client successfully connects to the application server, then message CWSIC3704I is displayed in the SystemOut.log file as shown in Example 9-18.

*Example 9-18 Client successfully connects to the application server*

---

```
[...] 00000034 SibMessage    I    [:] CWSIC3704I: A WebSphere MQ client application has
connected from host 127.0.0.1:1164 on transport chain InboundBasicMQLink.
```

---

When a client is connected, then you can confirm this by viewing the connections:

1. Select **Service Integration** → **Buses** and click the bus name.
2. Click **Messaging Engines** and then select the engine name.
3. Click **WebSphere MQ client links**, then the link name.
4. Click **Connections**.

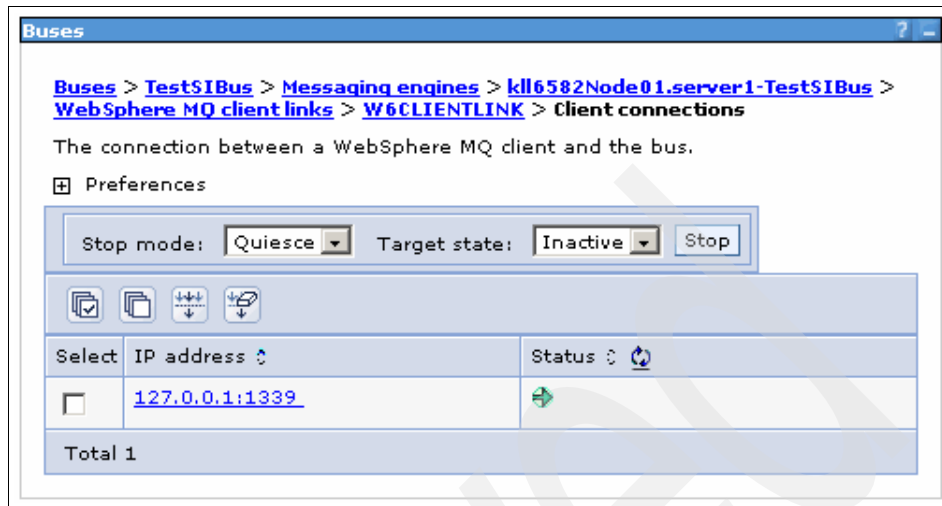


Figure 9-19 Client connections

**Note:** WebSphere MQ client links are intended to be used as a temporary migration path. Longer term, all migrated applications should have their JMS resources migrated to V6 default messaging provider JMS resources. This means that you need to delete the V5 embedded messaging provider JMS resources and create, in their place, V6 default messaging provider JMS resources with the same JNDI names.

### Symptom: ServiceUnavailableException on bus connect

Possibly one of the most common errors when starting client applications in the client container is that the application servers have not been started. Naturally, there will not be a message in the application server log. So, the application stack trace must be investigated. These can be very lengthy, but fortunately, the most appropriate error usually appears at the start of the stack trace.

Example 9-19 shows that the first exception code displayed is `ServiceUnavailableException`, and the accompanying text indicates the actual node and port number that the application was attempting to use.

#### Example 9-19 ServiceUnavailableException

##### ServiceUnavailableException caught

Stack Trace:

```
javax.naming.ServiceUnavailableException: A communication failure occurred while attempting to
obtain an initial context with the provider URL: "corbaloc:iiop:kll6582.itso.ra1.ibm.com".
Make sure that any bootstrap address information in the URL is correct and that the target name
server is running. A bootstrap address with no port specification defaults to port 2809.
Possible causes other than an incorrect bootstrap address or unavailable name server include
```

```
the network environment and workstation network configuration. Root exception is
org.omg.CORBA.TRANSIENT: java.net.ConnectException: Connection refused:
connect:host=9.42.171.161,port=2809 vmcid: IBM minor code: E02 completed: No
    at
com.ibm.CORBA.transport.TransportConnectionBase.connect(TransportConnectionBase.java:443)
    at com.ibm.ws.orbimpl.transport.WSTransport.getConnection(WSTransport.java:437)
```

---

You need to check the following:

- ▶ The node address and port are correct.
- ▶ The bootstrap server are running on that specific URL and port.

### **Symptom: JMSEException on bus connect, msg CWSIT0006E**

Another common example is a JMSEException while trying to connect to the bus. This can be caused by a stopped messaging engine, or more commonly, a wrong or misspelled bus name (case-sensitive) in the connection factory.

Example 9-20 shows an extract from an application stack trace showing such a problem. In this case, the messaging engine is stopped.

#### *Example 9-20 JMSEException: Messaging engine stopped*

---

```
JMSEException caught
Stack Trace:
javax.jms.JMSEException: CWSIA0241E: An exception was received during the call to the method
JmsManagedConnectionFactoryImpl.createConnection:
com.ibm.websphere.sib.exception.SIResourceException: CWSIT0006E: It is not possible to contact
a messaging engine in bus TestSIBus..
    at
com.ibm.ws.sib.api.jms.impl.JmsManagedConnectionFactoryImpl.createConnection(JmsManagedConnecti
onFactoryImpl.java:225)
    at
com.ibm.ws.sib.api.jms.impl.JmsManagedConnectionFactoryImpl.createConnection(JmsManagedConnecti
onFactoryImpl.java:148)
```

---

You can see the generalized JMSEException code CWSIA0241E and the CWSIT0006E message, which indicates that when the connection was made, there was no messaging engine active for the specified bus. To see why the messaging engine was stopped, you need to check the SystemOut.log for applicable messages. Examine the log file starting at the time period when the client experienced the problem and search backward.

*Example 9-21 SystemOut.log messages for a stopped messaging engine*

---

```
:  
[...] 00000033 SibMessage I [:] CWSID0016I: Messaging engine [mename] is in state Stopping.  
[...] 00000033 SibMessage I [:] CWSID0016I: Messaging engine [mename] is in state Stopped.  
:
```

---

Example 9-22 shows another example of an application failing to connect to a bus. This shows a stack trace from an application that has failed to create a connection to a named bus. As before, you see the **JMSEException** message, **CWSIA0241E** and **CWSIT0006E**.

*Example 9-22 Bus connect fails, msg CWSIT0006E*

---

```
JMSEException caught  
Stack Trace:  
javax.jms.JMSEException: CWSIA0241E: An exception was received during the call to the method  
JmsManagedConnectionFactoryImpl.createConnection:  
com.ibm.websphere.sib.exception.SIResourceException: CWSIT0006E: It is not possible to contact  
a messaging engine in bus AnyBus..  
at  
com.ibm.ws.sib.api.jms.impl.JmsManagedConnectionFactoryImpl.createConnection(JmsManagedConnecti  
onFactoryImpl.java:225)
```

---

Again, review the `SystemOut.log` file for messages related to the external bus. Example 9-23 illustrates a problem with the bus.

*Example 9-23 Bus connect fails, msg CWSIT0018W*

---

```
[...] 0000001a SibMessage W [BusName] CWSIT0018W: It is not possible to create the  
inter-bus connection AnyBus.Link to messaging engine AnyBus.ME in bus AnyBus, on host  
nn.nn.nn.nn port nnn using protocol BootstrapBasicMessaging.
```

---

If you were expecting the application to continue running on any available messaging engine, look at the JMS connection factory definitions to ensure the appropriate target name, target type, and proximity values have been matched.

### **Symptom: Bus connect fails, message CWSIT0019E**

There are many other reasons for the an application being unable to connect to the bus, for example:

**CWSIT0019E**: No suitable messaging engine is available in bus

If you see this message, make sure the appropriate messaging engines have started correctly. Look for the following message in `SystemOut.log` before the exception occurred:

**CWSID0016I**: Messaging engine [mename] is in state Started.

If the messaging engines have started, then you could have a configuration problem.

In the JMS connection factory, you can define a list of provider endpoints that govern which servers the application will attempt to connect to. If you have defined these endpoints, ensure that the messaging engines for these servers have all started and that the port and transport chains defined in the endpoint list are correct. Also, ensure that the values for the Connection proximity (bus, cluster, host or server) and the Target groups and Target significance settings are appropriate. For example, if the Target significance value is Required and no suitable messaging engine could be located, then this would cause a connection failure.

For more information about configuring the JMS connection factory and controlling messaging engine selection, refer to Chapter 10 of *WebSphere Application Server V6 System Management and Configuration Handbook*, SG24-6451.

If this solution does not resolved the problem, go to “The next step” on page 350.

### 9.3.4 Product errors

As with any large and complex software application, there is the possibility of products errors. The following are examples that arose during the writing of this paper. If you believe that you have discovered others, go to “The next step” on page 350.

#### **Symptom: JMS application returns MQJMS2007**

In Example 9-24, a JMS application is sending a message via the WebSphere MQ client link. During testing, all appears well. However, the application suddenly produces a stack trace with MQJMS2007.

*Example 9-24 Stack trace with message MQJMS2007*

---

```
javax.jms.JMSException: MQJMS2007: failed to send message to MQ queue
    at com.ibm.mq.jms.services.ConfigEnvironment.newException(ConfigEnvironm
ent.java:553)
    at com.ibm.mq.jms.MQMessageProducer.sendInternal(MQMessageProducer.java:
1598)
    at com.ibm.mq.jms.MQMessageProducer.send(MQMessageProducer.java:1022)
    at com.ibm.mq.jms.MQMessageProducer.send(MQMessageProducer.java:1056)
    at JMSPutClient.main(JMSPutClient.java:94)
```

---

By further investigation of the JMS application, it was noticed that then problem only seems to occur when sending a JMS TextMessage greater than or equal to

32079 bytes in size. The WebSphere MQ client link configuration is checked and all appears correct.

Ensure that you are using the latest version of the WebSphere MQ client code by checking the jar files that are used by the JMS application and by verifying that they are at the current service level. See the following link for additional information:

<http://www.ibm.com/software/integration/mqfamily/support/summary>

Having established what we believe to be a product defect, we refer to “The next step” on page 350 for what to do next.

**Note:** This problem has been resolved in the 6.0.2 refresh pack.

### Message CWSID0032W

One minor problem found during the writing of this book was the message code CWSID0032W during startup. See Example 9-25.

#### Example 9-25 CWSID0032W during startup

---

```
SibMessage      E   [busname] CWSID0032W: An inconsistency in the WCCM document sib-engines.xml
was detected; reason uuid=B77F76AAB6EC0BC35414B268 targetUuid=B77F76AAB6EC0BC35414B268
```

---

The error did not appear to cause any problems to the runtime and was resolved by the installation of refresh pack 6.0.2.

## 9.4 The next step

The symptoms and problem areas included in this paper are some that you are more likely to experience. However, there are other things that can go wrong, or the cause of the problem might be related to a component other than the default messaging provider.

If, after going through this process, you still have an undiagnosed problem, we recommend that you go back to *Approach to Problem Determination in WebSphere Application Server V6*, REDP-4073, at:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>

Review the “Classify the problem and determine the root cause” section to see if there are any other components that might be causing the problem.

If you feel sure you have a messaging problem, there are things you can do before contacting IBM support. First, you should review the documentation you

have gathered for errors related to the problem that were not addressed in this paper and search support sites for information or fixes.

The WebSphere Information Center contains problem determination information for troubleshooting service integration technologies. Take the time to review this information to see if there are any tips that will help you. You can find this guide at:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.pmc.nd.doc/ref/rjk\\_prob0.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.pmc.nd.doc/ref/rjk_prob0.html)

To assist the service organization it is useful to gather appropriate documentation. As we have mentioned, there are some basics such as SystemOut.log, SystemErr.log, server logs, and the exception logs from the FFDC directory. From your own analysis, you might have determined that traces would also be required. Before doing so, remember the following:

- ▶ Make the documentation relevant. Ensure that the documentation is recent and that it shows the problem.
- ▶ Provide a clear and concise description of the problem. By now, you have investigated the problem yourself, and you have a good understanding of how the problem occurs.
- ▶ Is the problem easily recreated? If possible, describe the steps that are needed to reproduce the problem or provide a small test case to demonstrate the problem. It is easier to diagnose and fix a problem with a suitable small test case that does not require a large configuration (for example using databases or network configurations). The service organization is aware that this is not always possible.
- ▶ Can you gather a suitable trace for the problem? Because you have done some preliminary analysis, you might have an understanding of which component you believe is causing the problem. See Table 9-2 on page 352 for a listing of the trace strings that might be appropriate for the components you believe are responsible.

For a list of documentation required to report a problem to IBM support, see *Mustgather: Service Integration Technology*, at:

<http://www.ibm.com/support/docview.wss?uid=swg21199330>

You can also find general information about what documentation to gather in the *Troubleshooting Guide for WebSphere Application Server* at:

<http://www.ibm.com/support/docview.wss?rs=180&context=SSEQTP&uid=swg27005324>

If you believe you have identified a messaging component area where the problem exists but have not identified the particular problem, you could attempt

to identify it further or assist the support organization by providing an appropriate trace. To do this, you need to set one or more trace strings.

*Table 9-2 Component trace strings*

Component	Trace string or strings
Data store	SIBMessageStore=all
MQ link	SIBCommunications=all=enabled SIBMfp=all=enabled SIBMqFapChannel=all=enabled
Security	SIBSecurity=all=enabled
Mediation	SIBMessageTrace=all SIBMediations*=all=enabled
JMS applications	SIBJms*=all=enabled
Bus message routing This can include client connections, ME to ME connections, and interbus connections.	SIBTrm=all=enabled

If your problem relates to the interoperability with WebSphere MQ, it might be necessary to gather trace for that system.

For instructions about gathering traces for WebSphere MQ systems, refer to the platform-specific *Administration Guide* at:

<http://www.ibm.com/software/integration/mqfamily/library/manualsa/>



# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 356. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *WebSphere Application Server V6: System Management and Configuration Guide*, SG24-6451-00
- ▶ *WebSphere Application Server V6 Scalability and Performance Handbook*, SG24-6392
- ▶ *WebSphere Application Server V6 Migration Guide*, SG24-6369
- ▶ *WebSphere Application Server V6 Security Handbook*, SG24-6316

## Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ *Approach to Problem Determination in WebSphere Application Server V6:*  
<http://www.redbooks.ibm.com/redpapers/pdfs/redp4073.pdf>
- ▶ *WebSphere Application Server V6: Diagnostic Data:*  
<http://www.redbooks.ibm.com/redpapers/pdfs/redp4085.pdf>
- ▶ *WebSphere Application Server V6: Installation Problem Determination:*  
<http://www.redbooks.ibm.com/redpapers/pdfs/redp4068.pdf>
- ▶ *WebSphere Application Server V6: System Management Problem Determination:*  
<http://www.redbooks.ibm.com/redpapers/pdfs/redp4067.pdf>
- ▶ *WebSphere Application Server V6: Web Container Problem Determination:*  
<http://www.redbooks.ibm.com/redpapers/pdfs/redp4058.pdf>
- ▶ *WebSphere Application Server V6: JCA Connection Problem Determination:*  
<http://www.redbooks.ibm.com/redpapers/pdfs/redp4080.pdf>

- ▶ *WebSphere Application Server V6: Web Server Plug-in Problem Determination:*  
<http://www.redbooks.ibm.com/redpapers/pdfs/redp4045.pdf>
- ▶ *WebSphere Application Server V6: Application Server Crash Problem Determination:*  
<http://www.redbooks.ibm.com/redpapers/pdfs/redp4059.pdf>
- ▶ *WebSphere Application Server V6: Default Messaging Provider Problem Determination:*  
<http://www.redbooks.ibm.com/redpapers/pdfs/redp4076.pdf>
- ▶ *Comment lines from Ruth Willenborg: Selecting WebSphere performance tools:*  
[http://www-128.ibm.com/developerworks/websphere/techjournal/0410\\_col\\_willenborg/0410\\_col\\_willenborg.html](http://www-128.ibm.com/developerworks/websphere/techjournal/0410_col_willenborg/0410_col_willenborg.html)
- ▶ *Customizing SimpleFileServlet to disable file serving*  
<http://www.ibm.com/support/docview.wss?uid=swg21116838>
- ▶ *Globalize your On Demand Business:*  
<http://www-306.ibm.com/software/globalization/j2ee/encoding.jsp>
- ▶ *HeapRoots Tool for Analyzing JVM Memory Usage:*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg24005757>
- ▶ *How to read heapdump generated in .phd format:*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21190476>
- ▶ *IBM Developer Kit and Runtime Environment, Java 2 Technology Edition, Version 1.4.2 Diagnostics Guide, SC34-6358-01:*  
<http://www-106.ibm.com/developerworks/java/jdk/diagnosis>
- ▶ *IBM Education Assistant*  
<http://www-306.ibm.com/software/info/education/assistant>
- ▶ *IBM Research Web site*  
<http://www.research.ibm.com>
- ▶ *IBM Software Support Handbook*  
<http://techsupport.services.ibm.com/guides/handbook.html>
- ▶ *IBM Support and Downloads site:*  
<http://www.ibm.com/support>
- ▶ *IBM Training Courses*  
<http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=page&contentID=a0000048>

- ▶ *IBM WebSphere Application Server - Performance Tuning: Adjusting WebSphere Application Server System Queues:*  
<http://www.redbooks.ibm.com/abstracts/tips0244.html>
- ▶ *IBM WebSphere Application Server - Performance Tuning: Determining Optimum Queue Sizes:*  
<http://www.redbooks.ibm.com/abstracts/tips0245.html>
- ▶ IBM WebSphere Training and Certification  
<http://www-306.ibm.com/software/info1/websphere/index.jsp?tab=education/index>
- ▶ *Identifying memory leaks with the WebSphere Studio Profiler and the JDK Heapdump utility*  
[http://www-128.ibm.com/developerworks/websphere/library/techarticles/0411\\_persichetti/0411\\_persichetti.html](http://www-128.ibm.com/developerworks/websphere/library/techarticles/0411_persichetti/0411_persichetti.html)
- ▶ *Introduction to IBM JVM for Linux JIT diagnostics*  
<http://www-128.ibm.com/developerworks/eserver/library/es-JITDiag.html>
- ▶ Java Servlet Specification Version 2.4 at  
<http://jcp.org/aboutJava/communityprocess/final/jsr154/index.html>
- ▶ JavaServer Pages Specification Version2.0:  
<http://jcp.org/aboutJava/communityprocess/final/jsr152/index.html>
- ▶ *Performance Management Guide article Performance Overview of the Virtual Memory Manager (VMM):*  
[http://publib16.boulder.ibm.com/pseries/en\\_US/aixbman/prftungd/resmgmt2.htm](http://publib16.boulder.ibm.com/pseries/en_US/aixbman/prftungd/resmgmt2.htm)
- ▶ *Recommended reading list: Software testing:*  
[http://www-128.ibm.com/developerworks/websphere/library/techarticles/0402\\_polozoff/0402\\_polozoff.html](http://www-128.ibm.com/developerworks/websphere/library/techarticles/0402_polozoff/0402_polozoff.html)
- ▶ Supported assembly tools in WebSphere Application Server V6:  
[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/catk\\_assemblytools.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/catk_assemblytools.html)
- ▶ *State replication in the Web tier*  
<http://www.ibm.com/developerworks/java/library/j-jtp07294.html>
- ▶ *Steps to getting support for WebSphere Application Server*  
[http://www-106.ibm.com/developerworks/websphere/support/appserver\\_support.html](http://www-106.ibm.com/developerworks/websphere/support/appserver_support.html)
- ▶ *Submitting Diagnostic Information to IBM Technical Support for Problem Determination*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21153852>

- ▶ *The Truth About Garbage Collection*  
[http://java.sun.com/docs/books/performance/1st\\_edition/html/JAppGC.fm.html](http://java.sun.com/docs/books/performance/1st_edition/html/JAppGC.fm.html)
- ▶ *Using HeapAnalyzer to analyze Java heap usage and detect possible Java heap leak:*  
<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg21190608>
- ▶ WebSphere Application Server system requirements  
<http://www-306.ibm.com/software/webservers/appserv/doc/latest/prereq.html>
- ▶ WebSphere Information Center  
<http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp>
- ▶ *WebSphere developerWorks*  
<http://www-130.ibm.com/developerworks/websphere>

## How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)

# Index

## Symbols

\$AdminConfig validate 306

## Numerics

100% CPU utilization 32, 47

1PC 246

2PC 246

## A

accept-language header 199

access log 269

access.log 90

access\_log 90

activity.log 89

addNode 31, 108, 144

command fails 143

adminconsole.ear 129

administrative console 138

adminconsole application 128

can't access 37, 115, 127

save conflicts 37, 115, 150

URL 127

analysis strategy 29–30

Apache JMeter 9

APAR 4–6, 11, 28, 72

application

accessing directly through the Web container  
174, 260

adminconsole 128

assembly 31

automatic deployment 121

can't connect to resource 32, 63

can't manage 141

change control 13

deployment 35

document root 167, 175

doesn't respond 32, 43, 45–46

intermittent failure 45, 260, 277

missing from the administrative console 37,  
115, 151

packaging 35

performance 32

problem determination 31–32

testing 7, 10

unexpected results 32, 49

URL specification 186

application server

check status 174

check the process 287

crash 21, 32, 46

fails to start 109, 111

hang 46

restart 285

stops responding 287

application server crash 285, 292

Application Server Toolkit 36

asynchronous bean thread pool 22

authentication 43, 235

auto commit mode 228

autoRequestEncoding 200

autoResponseEncoding 200

average response time 19

## B

baseline configuration 12

BLOB 228

blocking connect 280

bootstrap server 347

bus connect failure 348

## C

cache 221

caching 43

Caching Proxy 44

callable statement 221

Catalog 242

CCI 214

certificate 129

create a self-signed 156

expired 115, 153, 155

request a new key 156

certificate authority 156

change team 72

character encoding 50, 171, 196–197, 199–200

online support 201

- class loader 286
- Class.forName() 229
- ClassCastException 52, 229
- classloader
  - problem determination 52
- ClassLoader.findSystemClass() 229
- ClassLoader.loadClass() 229
- ClassNotFoundException 52, 229
- CLASSPATH environment variable 218
- classpath error 141
- client API 213
- client.encoding.override 200
- CloneID 256, 274
- Cloudscape Universal JDBC 231
- cluster 14, 56
- collector tool 23, 31, 87
- com.ibm.ssl.trustStore 140
- com.ibm.websphere.ce.cm.ConnectionWaitTime-outException 237
- com.ibm.websphere.ce.cm.StaleConnectionExcep-tion 243
- com.ibm.websphere.ce.j2c.ConnectionWaitTime-outException 237
- com.ibm.websphere.rsadapt-er.Oracle10gDataStoreHelper 228
- com.ibm.ws.Transac-tion.IllegalResourceIn2PCTransactionException 246
- com.ibm.ws.webcontainer.httpsession 173
- common client interface (CCI) 213
- compiling trace 294
- component-managed authentication 232, 236
- config directory 117
- configuration archive file 11
- Connect JDBC 231
- Connection 214–215
- connection leak 236, 240, 242
- connection leak trace 240
- connection pool 215, 238
- connection pool size 19
- connection properties 222
- connection proximity 349
- connection refused 269
- Connection Timeout 238–239
- ConnectionFactory 214–215
- ConnectionManager 214–215
- ConnectionWaitTimeoutException 27, 226, 237–240, 243
- ConnectTimeout 280

- ConnLeakLogic trace 240
- contentType 197–198
- context root 188–189
- cookie domain 205
- cookie maximum age 205
- cookie path 205
- cookie prompt 205
- cookies 204–205, 256, 274, 277
- core file 288, 290
- core group 14
- crash dump 288
- Crash Dump Type 290
- current thread 294, 296, 298
- custom profile error 112

## D

- data replication service
  - problem determination 58
- data source 222
  - failure to connect 215, 222, 224
  - lookup 224
  - test connection 222
- database connection
  - problem determination 64
- database error 245
- database persistence 53
- DB2 233, 246
- DB2 CLI Legacy 219, 230–231
- DB2 Command Line Processor 217
- DB2 Universal JDBC (JCC) 219, 231
- DB2 Universal Type 2 230
- DBCS 171, 195
  - problem determination 61
- deadlock 288
- default messaging provider 302
  - problem determination 64
- DefaultApplication 189
- DefaultCoreGroup 14
- deployment manager
  - can't start 42
- destination 65, 303, 308, 338
- diagnostic data 21–23, 26, 29, 77, 88
- diagnostic tools 29
- digital certificates 129, 133
- document root 167, 175
- dummy key database file 157
- dumpNameSpace 31, 65
- dumps

- javacore 288
- process 290–291
- dynamic cache
  - problem determination 54

## E

- Edge components
  - problem determination 44
- EJB 213
- EJB client 66
- EJB container
  - problem determination 51
- ejbdeploy 31
- Electronic Service Request (ESR) tool 75
- Enable Log 82
- Enable protocol switch rewriting 206
- Enable SSL ID tracking 206
- Enable URL rewriting 206
- encoding 195, 197–199
  - ISO-8859-1 197
- encoding passwords 120
- encoding.properties 199–200
- enterprise information systems (EIS) 212–213
- Enterprise JavaBeans (EJB) 32
- environment entries 289
- environment variable 289, 294
- error.log 90
- error\_log 90
- event type 81, 86
- exception 32
  - com.ibm.websphere.ce.cm.ConnectionWaitTimeoutException 237
  - com.ibm.websphere.ce.cm.StaleConnectionException 243
  - com.ibm.websphere.ce.j2c.ConnectionWaitTimeoutException 237
  - com.ibm.ws.Transaction.IllegalResourceIn2PCTransactionException 246
  - ConnectionWaitTimeoutException 226
  - java.sql.SQLException
    - invalid arguments in call 232
  - ServiceUnavailableException 346
  - StaleConnectionException 226
  - Web module 50
- exception destination 335, 343
- ExceptionFileMaximumAge 88
- execution engine 286

- ExtendedHandshake 279

## F

- failed to compile 180
- failed to find resource 175
- failover 14–15, 19, 45, 56, 251, 280
- favicon.ico 178
- federate 37, 115
- federation 112
- FFDC 87, 306, 351
- ffdcRun.properties 88
- file path length error 112
- file serving enablement 191
- fileServingEnabled 191
- filetransfer.ear 129
- firewall 136, 140, 244
- First Failure Data Capture (FFDC) 87
- fix pack 5–6
- foreign bus 327
- foreign destination 304
- Format print statements 79
- forward() 185
- functional testing 8

## G

- garbage characters 195
- garbage collection 286
- generic JMS provider 65
- GenPluginCfg 263, 266
- getConnection() 225, 236, 239–240
- getParameter() 199
- getWriter() 198
- global security 120
- gsk7ver 262
- GSKit 262

## H

- hang 21, 32, 46, 287–288
  - application server 134, 278, 285
  - connection 215
  - install 97
  - installation wizard 103
  - killing the process 134
  - node agent 134
  - problem determination 46
- HeapRoots 29
- Hewlett-Packard 300

- high availability 14–15, 19
- high availability manager 14–15
  - problem determination 57
- historical log file 79
- historyInfo 6
- host alias
  - verify 177
- host name error 112
- htpasswd 139
- HTTP error
  - 404 127, 168, 170, 172, 175, 177
  - 500 168, 171, 179, 181
- HTTP header 197
- HTTP inbound channel 166
- HTTP return code
  - 200 268
- HTTP transport 45
- HTTP transport type 253
- http\_plugin.log 267
- HTTPS 255
- HttpServlet 183
- HttpServletResponse 183–184
- HttpServletResponse.sendError() 184
- HttpServletResponse.sendRedirect() 184
- HttpSession 181–182

## I

- IBM HTTP administration server 139
- IBM HTTP Server 140, 251, 258, 264
  - admin password 139
  - fails to start from the administrative console 139
  - logs 90
  - plug-in directives 253
  - problem determination 43
- IBM HTTP Server V6 138–139
- IBM Key Management tool (ikeyman) 154
- IBM Service Logs 89
- IBM Support 70
- IBM Support Assistant 28
- IBM\_JAVACOREDIR 289
- IBMTrackerDebug 203
- ibm-web-ext.xml 191, 193
- illegal transaction usage 227
- ikeyman 31, 155
- IllegalStateException 181–182, 184–185
- incorrect data 217, 221–222
- Informix 233
- Informix dbaccess 217

- Informix JDBC 220, 231
- installation
  - overview 98
  - problem determination 31, 34, 97
- installation wizard 34
  - hang 34
  - won't start or fails 101
- Installed Application Output 79
- internationalization 195
  - problem determination 61
- invalid session object 181
- iptrace 271
- ISO-8859-1 197–198, 200
- isolation level 242
- isolation strategy 30
- IVT 111
  - failure 34, 108
- IVT applications 306
- ivtClient.log 90

## J

- J2C authentication data entry 236
- J2C connection manager
  - trace 85
- J2EE Connector Architecture (JCA) 211
- Java Connector Architecture (JCA) 32
- Java heap size 147
- Java Logging API 17
- Java Naming and Directory Interface (JNDI) 65
- Java Software Development Kit (SDK) 5
- Java stack 295, 298
- Java Virtual Machine (JVM) 285
- java.io.OutputStream 196
- java.io.PrintWriter 196
- java.lang.IllegalStateException 184
  - Cannot forward as Output Stream or Writer has already been obtained 185
  - Header already sent 185
  - Response already committed 183
- java.lang.OutOfMemoryError 48
- java.naming.Context 224
- java.sql.SQLException
  - invalid arguments in call 232
- java.util.logging 17
- Java2WSDL 31, 67
- javacore 288–291
- jadadump 288
- javax.resource.cci 214



- javax.resource.cci.ConnectionFactory 225
- javax.resource.spi 214
- javax.sql.Connection 214
- javax.sql.DataSource 214, 225
- JCA
  - overview 213
- JCA connection
  - initial symptoms 217
  - problem determination 211
- JCA connection manager 64
  - problem determination 63
- JCA connection manager pools 63
- JCA resource adapter 63
- JDBC database connection 63
- JDBC driver 217
  - type 2 218, 230
- JDBC program
  - standalone 218
- JDK
  - check the installed version 292
  - determine installed version 291
  - upgrade 291
- JFunc 8
- JIT
  - disable 295
  - loop 288
  - module name 293
  - overview 286
  - trace 294
  - work-around 293
- JITC.dll 293
- JITC\_COMPILEOPT 294
- JMS administered objects 303
- JMS connection 63
- JMS connection factory 65
- JMSEException 325, 347–349
  - MQJMS2005 344
- JMX clients
  - problem determination 40
- JNDI naming
  - problem determination 65
- JSP 45, 50, 54
  - attributes 194
  - code errors 180
  - compilation 50
  - processing error 179
  - reloading 171
  - runtime reloading 193
- JSP processor

- normal start 179
- JSR-047 17
- JUnit 8
- Just In Time (JIT) 285
- Just-In-Time (JIT) compiler
  - problem determination 49
- JVM 102, 305, 331
  - memory 19
  - module name 293
  - overview 286
- JVM logs 78
- JVM.dll 293

## K

- key database file 120, 154, 157

## L

- launchpad 34, 97–98
  - won't start or fails 101
- level 1 team 71
- level 2 team 71
- level 3 team 72
- libjitc.a 293
- libjitc.so 293
- libjvm.a 293
- libjvm.so 293
- LIBPATH 218
- link 304
- Load Balancer 44
- load balancing 15, 45, 56, 251, 254–255, 260, 280, 282
  - round robin 272
- load testing 9
- LoadBalanceWeight 281
- local transaction 246
- LocalTransaction 215
- Log Analyzer 89
- log.txt 90
- LogLevel 91
- LogLevel directive 258
- logs
  - access log 90
  - activity.log 89
  - application installation and configuration 107
  - default server creation 107
  - error log 90
  - federation 107
  - IBM HTTP Server 90

- installation 90, 101
- IVT 109
- ivtClient.log 90
- JVM 16, 78
- launchpad 101
- log.txt 90
- management scripts 92, 124
- native 88, 290
- native\_stderr.log 88
- native\_stdout.log 88
- pctlog.txt 90
- process 88, 290
- profile creation 104
- profile management 92, 124
- service log 89
- stdout 290
- stopServer 92
- system management 92
- Tivoli Access Manager 106
- transaction 14
- Web server 257
- Web server plug-in 90, 258
- WebSphere Application Server 122
- WebSphere Rapid Deployment 125
- Windows service creation 107
- loop 298
- lsol 136

## M

- managed environment 213
- ManagedConnection 215
- ManagedConnectionFactory 215, 247–248
- ManagedConnections 239
- management script logs 124
- managing Web servers 138
- master repository 117
- maxagents 246
- maxappls 246
- MaxConnections 279, 282
- Maximum Connections 239
- maximum failed deliveries 335
- Maximum File Size 83, 89
- maximum number of connections 238–239
- Maximum Number of Historical Log Files 79, 83
- Maximum Size 79
- MBeans 117, 124–125, 144
- mediation 304, 338
  - messages disappear 342–343

- not working 339
- mediation list 338
- MediationHandler interface 343
- memory 21
- memory problem 305
- memory signal 291
- memory-to-memory sessions 53
- message
  - CLI0108E 245
  - CNTR0020E 336
  - CWSIA0144E 324
  - CWSIA0241E 347–348
  - CWSIC3098I 328
  - CWSIC3704I 345
  - CWSID00016I 312
  - CWSID0003E 323
  - CWSID0027I 312
  - CWSID0032W 350
  - CWSID0035E 313
  - CWSIP0655E 341
  - CWSIS0002E 312, 314, 318, 320
  - CWSIS1524E 313
  - CWSIS1535E 319
  - CWSIS1538I 331
  - CWSIS1546I 330
  - CWSIT0006E 347–348
  - CWSIT0019E 348
  - CWSIV0764I 334
  - CWSIZ0002E 341–342
  - DSRA0080E 330
  - DSRA8000-8499 228
  - DSRA8025I 224
  - DSRA8030I 224
  - DSRA8040I 224
  - DSRA8041I 224
  - DSRA8042I 223
  - HMGR0130I 331
  - J2CA0046E 314
  - J2CA0056I 330
  - J2CA0294W 234–235
  - J2CS0046E 320
  - JSPG0036E 175
  - MQJMS2007 349
  - SESNxxxxE 173
  - SESNxxxxW 173
  - SQL1040N 245
  - SQL30082N 314
  - SRVE0017W 177–178
  - system 16

- WSVR0605W 22
- message consumer fails 324
- message engine startup problems 310
- message flow 308
- message flow problem 305, 310, 324
- message prefix
  - application deployment 35
  - data replication service 58
  - data source resource adapter 225
  - dynamic cache 54
  - EJB 51
  - high availability manager 57
  - installation 34
  - IVT 34
  - JCA connection 63–64, 211, 215
  - JCA connector 227
  - JNDI 65
  - JSF 165
  - JSP 165
  - list 33
  - management utility 110
  - message-driven beans 65
  - messaging 306
  - migration 35
  - ORB 66
  - PMEs 59
  - profile creation 34
  - security 39
  - server runtime 110
  - service integration bus 64
  - session 53
  - session manager 173
  - transaction 55, 225, 227
  - transport channel service 62
  - Web container 50, 165
  - Web server plug-in 45
  - Web services 67
- message-driven bean 333
  - infinite loop 336
  - message not restored after failure 338
  - not receiving messages 334
  - not started 333
- messages disappear 342–343
- messaging
  - problem determination 64
- messaging data store 319
  - authentication error 314
  - data store tables missing 322
  - invalid authentication 312
  - invalid database tables 312
  - JNDI name error 312
  - not available 320
  - sibDDLgenerator 322
  - SIBOWNER table 319
  - test connectivity 321
  - unrecoverable error 330
- messaging engine 302, 304
  - invalid messaging data store 318
  - normal startup 310
  - viewing the status 307
- messaging engine exception 312
- messaging engine startup problem 305, 310
- method
  - skip JIT compile 298
  - skipping JIT compile 294
- method name 86
- migration
  - problem determination 31, 35
- missing static resources 171
- monitor for problems 15
- MQ channel 330
- MQ link 329
- MLink 327
- MustGather 23, 72, 85
  - 100% CPU 47
  - application deployment 36
  - Application Server Toolkit 36
  - classloader 52
  - deployment manager start and stop 42
  - dynamic cache 55
  - EJB container 52
  - EJB workload management 57
  - enhanced EAR file 36
  - hang 47
  - high availability 58
  - IBM HTTP Server 44
  - installation 113
  - internationalization 62
  - Java Transaction Service (JTS) 56
  - JMX client 41
  - JNDI 66
  - JVM and JIT problems 299
  - migration 35
  - node agent start and stop 42
  - ORB 67
  - out of memory 48
  - persistence manager 52
  - PMI 41

- read first 72
- security 40
- session management 54
- system management 38, 162
- UDDI 69
- Web container 210
- Web server plug-in 283
- Web services 69
- ws\_ant 38
- Mustgather 351

## N

- NamingException 234
- native code 88
- native logs 88, 290
- native stack 295–296, 298
- native\_stderr.log 88, 290
- native\_stdout.log 88, 290
- netstat 111, 132, 136
- Network protocol analyzer 271
- NoClassDefFoundError 52
- node
  - can't add a node 143
- node agent
  - can't access 137
  - can't start 42
- non-managed environment 213, 215
- non-relational resource
  - fail to connect 226
- non-root user 152–153
- number of requests 19

## O

- Oracle 217, 228, 233
- Oracle JDBC Thin 218
- Oracle OCI driver 218–219, 230–231
- Oracle Thin driver 228, 231
- ORACLE\_HOME 218
- ORB 22
  - problem determination 66
- OS/400 Native 231
- OS/400 Toolbox 231
- out of memory 32
  - problem determination 47
- out of memory error 298

## P

- page directive 197
- page not found 172
- pageEncoding 197–198
- paging 19
- pctlog.txt 90
- performance 32, 54, 62, 285–287
  - monitoring 32
  - problem determination 69
  - testing 8
- Performance Monitoring Infrastructure (PMI) 41
- persistent session 207
- personalized Web content 53
- plugin-cfg.xml 91, 254–255, 261–263, 265, 280
- PME
  - problem determination 59
- PMI 15
  - problem determination 41
- PMR 7, 29, 71–73, 88
  - creating 74
- pool manager 239
- pooling 214
- port conflict 109, 111
- precompiled SQL statement 221
- prepared statement 221
- prerequisites 7
- print() 78–79
- println() 78–79
- printStackTrace() 78, 290
- problem entry team 71
- problem log 26–27
- problem symptom 25
- process dump 289–291
- process ID 46, 134, 136, 285, 287
- process logs 88, 290
- profile 119
  - creation failure 34, 97
  - custom 112
  - default server 103
  - errors during creation 105, 112
- Profile creation wizard 111–112
- profile management logs 124
- PropFilePasswordEncoder.bat 120

## Q

- queue manager name 327
- queue point 302, 304, 309, 324, 326
- queues 304

## R

- ra.xml 247–248
- Readonly 242
- recursion 295, 298
- Redbooks Web site 356
  - Contact us xiv
- refresh pack 5–6, 291
- regression test 6
- reliability 304, 343
- reload interval 193
- reloadEnabled 194
- reloadingEnabled 193
- removeNode 145
- Repeat Time 79
- repository
  - file permissions 153
- repository corruption 142
- request encoding 197, 199
- RequestDispatcher 184
- RequestDispatcher.forward() 184
- res-auth 234, 236
- res-isolation-level 234
- resource adapter 212–213, 215
- resources.xml 247–248
- response encoding 198–199
- response generation problem 183
- response.getOutputStream() 185
- response.getWriter() 185
- res-resolution-control 234
- res-sharing-scope 234
- RetryInterval 279
- revert to safe conditions 24
- RMI 120
- Route 253
- RRA trace 236, 244

## S

- safe operational procedures 12–13
- SAP 213
- sas.client.props 120, 140
- save conflicts 37, 150
- scalability testing 9
- scripting tools
  - problem determination 37
- SDK 5–6, 11
- security 214
  - administrative console application roles 151
  - problem determination 31, 39

- roles needed to access a resource 12, 189
- turning off manually 129
- verifying the configuration 133
- security.xml 154
- sender channel 330
- sendRedirect() 185
- Sequelink 5.4 231
- serve servlets by classname 203
- server name.pid 46
- server process
  - can't access 37
  - can't start 32, 42, 135
  - can't stop 133
- server.xml 118, 123
- ServerCluster 253
- serverindex.xml 111, 117, 128, 143
- ServerIOTimeout 279
- serverJMXHost 159
- serverJMXPort 159
- server-side classes 167
- serverStatus 111, 174
- service integration bus 302–303
- service log 89
- ServiceUnavailableException 346
- servlet 45, 50, 54, 213
- servlet mappings 189
- servlet processing 167
- ServletResponse.reset() 184
- ServletResponse.resetBuffer() 184
- session
  - persistent 207
- session affinity 256, 272, 274, 277, 282
- session data 251–252, 256
- session ID 181, 206, 273
- session management 50, 166–167, 202–203
  - problem determination 52
  - search for known problems 185
- session management tracking 206
- session manager 173, 181
- session storage 53
- session timeout 53
- session timeout interval 207
- session tracking 274
- sessions 256
  - invalid object 181
  - lost 45, 53, 171, 260, 272
  - timeout 181
- setBufferSize() 184
- setCharacterEncoding() 199

- setCharacterEncoding(String enc) 197
- setContentType() 198
- setupCmdLine.bat 218
- severity 73
- shareable connection 242
- Show application print statements 79
- showlog 89
- sibDDLgenerator 322
- signals 291
- slow application response 252
- snoop servlet 45, 254, 259, 267
- SOAP 120
- SOAP connection timeout 145
- SOAP timeout parameter 137
- soap.client.props 120, 137
- SQL Server 220, 231, 233
- SQL Server 2000 231
- SQL Server Query Analyzer 217
- SQLCode 244
- SQLException 244–245
- SQLState 244
- SSL 39, 43, 115, 120, 129, 133, 140, 157, 188
- SSL ID tracking 206, 274
- SSLV3TIMEOUT 206
- stack overflow 295–296, 298
- stack trace 30, 326, 348–349
- StackOverflowError 298
- stale connections 243
- StaleConnectionException 226, 244
- startManager 32
- startNode 32
- startServer 32, 110–111, 135
- startupTraceSpecification 124
- statement cache size 221
- static content processing 167
- static resource
  - file location 190
  - not displayed 190
- STDERR 289
- stopServer 124, 131–133, 291
- stopServer.log 92
- stored procedures 221
- Struts 199, 206
- Sun Solaris 300
- Support Assistant 28
- Sybase isql 217
- Sybase jConnect 220
- symptom database 89
- synchronization 117–119

- automatic 148
- fails 145
- from the command line 149
- message indicating it worked 146
- syncNode 119
- system documentation 19–21
- system management
  - overview 116
  - problem determination 31, 37, 115
  - security 119
- System.Err 78
- SYSTEM.Exception.Destination 329, 335
- SYSTEM.Exception.Queue 328
- System.Out 78
- SystemErr 78–80
- SystemOut 78–80, 85

## T

- target significance 349
- technotes 28
- telnet 132
- template path error 112
- test environment 10
  - synchronize with production 13
- TestConnection 222, 224
- thread dump 288, 290
- thread ID 80, 85
- thread monitor 22
- thread monitor for hang detection 22
- thread pool size 19
- ThreadAnalyzer 29
- Tivoli Monitoring for Transaction Performance (TMTP) 19
- Tivoli Performance Advisor 8
- Tivoli Performance Viewer 8, 18, 41
  - problem determination 41
- topic spaces 304
- topology diagram 20
- trace 82–83, 85, 123
  - compiling 294
  - connection leak 240
  - ConnLeakLogic trace 240
  - J2C connection manager 85
  - JCA connection 236
  - messaging component trace strings 352
  - network 259
  - ORB service 154
  - RRA 236, 244

- stale connections 244
  - system management 123
  - WAS.database 244, 249
  - WAS.j2c 236, 240, 244
  - Web server plug-in 91, 258, 266, 272, 281
  - wsadmin 92
- trace components and groups 83, 85
- trace events 85
- Trace Output 82
- Trace Output Format 83
- transaction 214–215
- transaction logs 14
- transaction manager 14–15
  - problem determination 55
- transaction recovery 55
- transaction rollback 55
- transaction timeout 55
- transport chain 166
- transport channel service
  - problem determination 62
- TypeMap 242

## U

- unit testing 8
- UnsatisfiedLinkError 52, 230
- unshareable connection 242
- UriGroup 253–254
- URL pattern 189
- URL rewriting 206, 274
- user dump 288, 290
- user registry 119, 235
- user.dmp 290
- UserTransaction 338
- UTF-8 200
- UUID 319

## V

- verbose output 91
- version
  - checking 6
- versionInfo 6
- virtual host 186–187, 253
- Virtual Host / WebGroup not found 266
- Virtual Host/WebGroup Not Found 178
- VirtualHostGroup 253

## W

- WAS.database trace 244, 249
- WAS.j2c trace 236, 240, 244
- WASenv.jar 87
- wasprofile 31, 92, 104, 106
- WC\_adminhost 128
- WC\_defaulthost 187, 260
- WC\_defaulthost\_secure 187
- Web application deployment descriptor 167
- Web Archive (WAR) 167
- Web browser
  - installation requirements 110
- Web container 22
  - accessing applications directly 260
  - finding the HTTP port 260
  - online support 175
  - overview 166
  - problem determination 50, 165
  - successful start 172
- Web container inbound chain 167
- Web container transport chain 188
- Web deployment descriptor 191, 194
- Web module
  - can't start 50
  - location 176
  - successful start 172
- Web module reloading 193
- Web page
  - incorrect information 171
- Web resource
  - can't access 50
- Web server 115
  - can't manage 138
  - can't start 45, 252, 261
  - fails to start from the administrative console 140
  - no response 259
- Web server plug-in 43, 167
  - automatic propagation 264–265
  - build version 268
  - doesn't pass requests to the Web server 263
  - generation 265
  - load balancing 254–255
  - logs 90
  - overview 252
  - problem determination 45, 251
  - regenerate 189, 194
  - server marked "down" 269, 276, 278, 280
  - timeout 270
  - trace 91

Web service 32, 54  
    problem determination 67  
web.xml 167, 191, 194  
WebGroup/Virtual Host has not been defined 177  
WEB-INF 167, 191  
WebSphere Application Server Support site 28  
WebSphere developerWorks 28  
WebSphere Information Center 28  
WebSphere MQ 302, 304, 324, 327–330, 344, 352  
WebSphere MQ client link 345, 349–350  
WebSphere MQ JMS provider 65  
WebSphere MQ provider 329  
WebSphere Rapid Deployment 36, 115, 121, 151,  
158  
    verify the configuration 158  
WebSphere Rapid Deployment (WRD) 93  
WebSphere RRA 214  
WebSpherePluginConfig 263–264  
welcome file list 189  
Windows services manager 136  
WLM 14–15, 56  
workload management (WLM)  
    problem determination 56  
WORKSPACE 93  
WORKSPACE variable 158  
wsadmin 116, 124  
    can't access server 130  
    output 92  
    SOAP port 130  
wsadmin.properties 92, 130–131  
wsadmin.traceout 92, 124  
WSDL2Java 67  
wstemp directory 153

## **X**

XA with Serializable isolation level 228  
XAResource 215, 246  
-Xoss 298





## WebSphere Application Server V6 Problem Determination for Distributed Platforms

(0.5" spine)  
0.475" <-> 0.875"  
250 <-> 459 pages







**Redbooks**

# WebSphere Application Server V6

## Problem Determination for Distributed Platforms

**Learn how to approach the problem**

**Collect and analyze the data**

**Find a resolution**

This IBM Redbook is intended to help customers through the process of identifying and resolving problems in WebSphere Application Server V6 on distributed platforms. Previously published as a series of IBM Redpapers, each chapter addresses a specific aspect of the problem determination process. Links to the original papers are provided throughout the book.

The first paper, *Approach to Problem Determination*, should be the first stop for IBM WebSphere Application Server system managers. It provides information that is useful in preparing for and preventing problems. It then details what actions to take when a problem occurs and helps you define the problem. Once the primary symptoms are identified, it helps you narrow down the problem to terms that are similar to those used by the IBM support team, and leads you to more specific information that can help you pinpoint the problem.

Subsequent papers address selected topics that have been identified as some of the more common problem types that customers usually address.

### INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

### BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**  
[ibm.com/redbooks](http://ibm.com/redbooks)