WebSphere® software

Redbooks Paper

Craig Scott

WebSphere Application Server V6: 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`.

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 1).

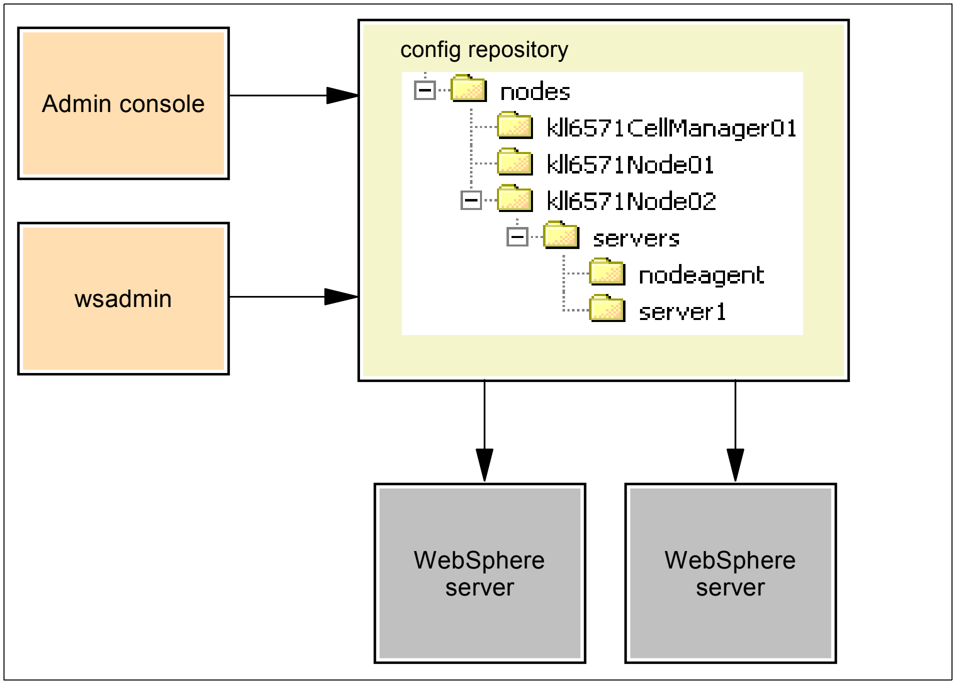


Figure 1 Overview of system administration

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 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.

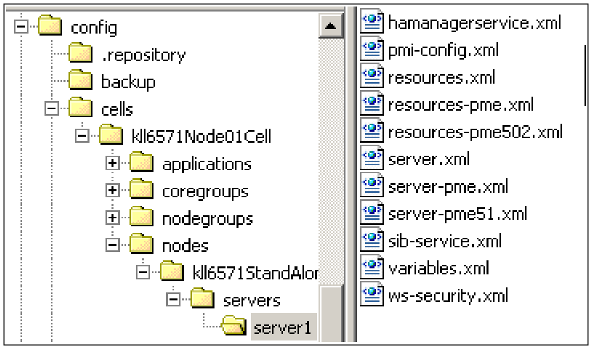


Figure 2 The config directories and files

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 3 on page 4 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.

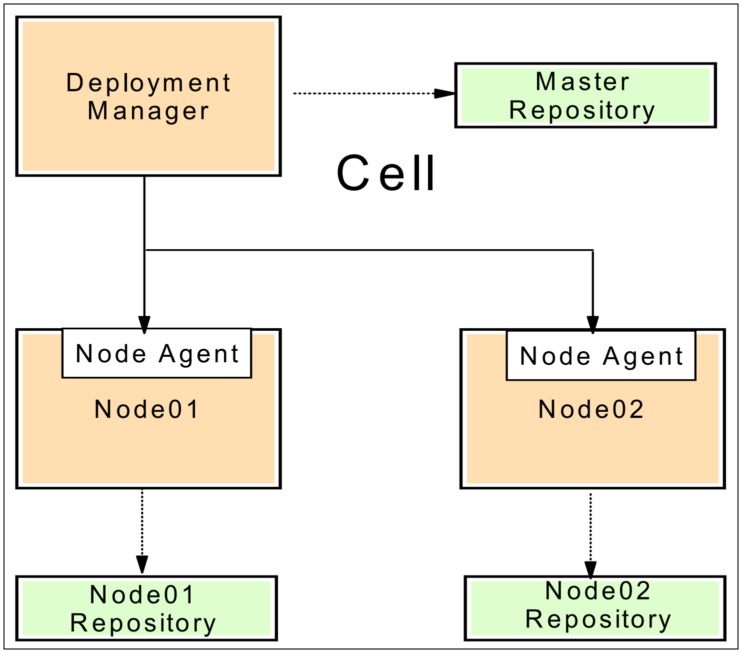


Figure 3 Replicated admin repository

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.bat|sh 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

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 shows the WebSphere Rapid Deployment components.

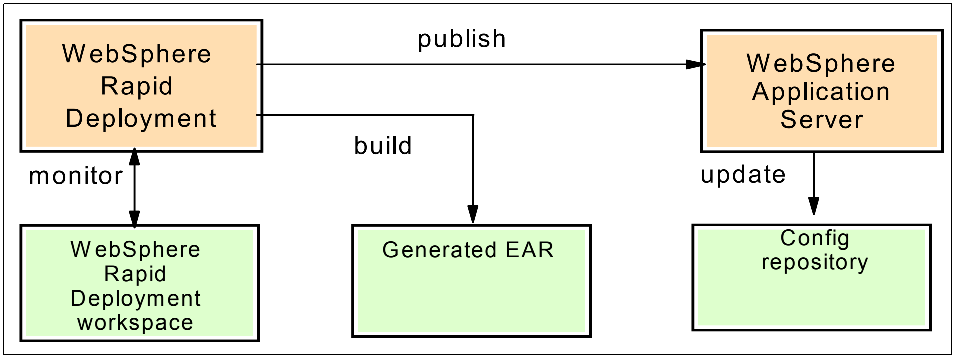

WebSphere Rapid Deployment
publish
WebSphere Application Server
build
monitor
update
WebSphere Rapid Deployment workspace
Generated EAR
Config repository

Figure 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.

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 48 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:SS
L=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

> **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

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 1.

Example 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 8.

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.

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.

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 13.

- 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 16.
- 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 19.
  - If you are unable to start an application server, go to “Problem: Unable to start a server process” on page 21.
  - If the deployment manager is not communicating with a node agent, go to “Problem: Unable to access a node agent” on page 23.
  - 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 24.
  - If you are unable to manage (start, stop, install) an application, go to “Problem: Unable to manage applications” on page 27.
- 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 29.
- 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 36.
- If enterprise applications no longer appear in the administrative console go to “Problem: enterprise applications missing” on page 37.
- 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 39
- If you are having problems with WebSphere Rapid Deployment, go to “Problem: WebSphere Rapid Deployment” on page 44.
- 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 48.

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 48.

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 8 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 accidently 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.

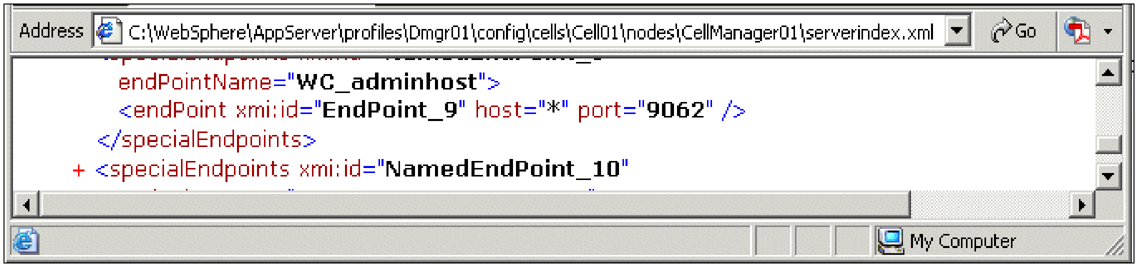
Review the WebSphere Application Server SystemOut.log to ensure that the administrative console has started. You will see messages similar to those in Example 2 on page 14, showing you that the administrative console has started and on which port the administrative console is listening.

Example 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>\serverindex.xml`



Address C:\WebSphere\AppServer\profiles\Dmgr01\config\cells\Cell01\nodes\CellManager01\serverindex.xml

```
      endPointName="WC_adminhost">
      <endPoint xmi:id="EndPoint_9" host="*" port="9062" />
    </specialEndpoints>
  + <specialEndpoints xmi:id="NamedEndPoint_10"
```

Figure 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 3.

Example 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 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 accidently.

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 39 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.

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 shows the host name and port number that wsadmin uses by default.

Example 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 5.

Example 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 kll6571 -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 6 shows a connection to a node agent and then issuing a **`stopServer`** command.

Example 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 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

Verify that the application server is running

You can use the **`netstat`** command to ensure that a process is listening on the port. Example 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 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 8.

Example 8 Using to 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 9

Example 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 39 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 13.

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

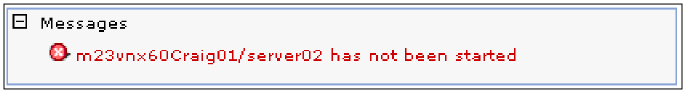
The following logs and output can be helpful in determining why you cannot stop an application server:

- Deployment manager SystemOut.log, SystemErr.log
- Node agent lSystemOut.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 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.



m23vnx60Craig01/server02 has not been started

Figure 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 10.

Example 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 11.

Example 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.

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 lSystemOut.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 12 shows the output from a **startServer** command that indicates why a server will not start.

Example 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 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 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 13   Output from the netstat command

```
TCP    9.42.171.145:8880      9.42.171.145:2698      TIME_WAIT
TCP    9.42.171.145:8880      9.42.171.145:2739      TIME_WAIT
```

Note: The **netstat** command is available on both UNIX and Windows.

Example 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 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 15 on page 23.

By checking the server logs, you can determine if the server has actually started.

Example 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

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.

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 6 on page 17.

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.

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 mange 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 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 16.

Example 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.
```

```
[7/12/05 14:56:22:843 EDT] 000000a6 NodeAgent     W   ADML0040E: The process timed out waiting
for server "webserver1" initialization: 1200 seconds
```

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 8, to gather further information about the source of a failure.

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 mange 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:

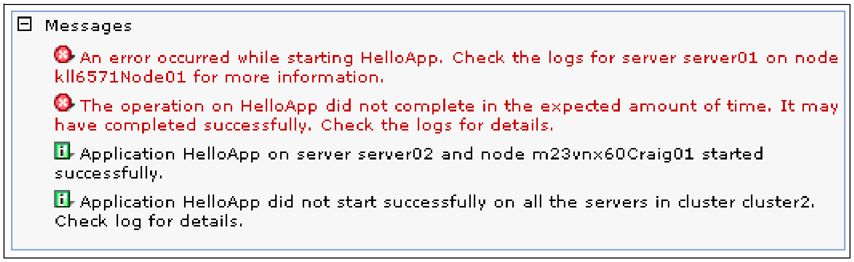
- Verify that there are no resource or classpath errors
- Verify that the application was installed correctly
- Verify that the configuration repository is not corrupted

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 7 on page 28. 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 7 on page 28.


⊟ Messages

An error occurred while starting HelloApp. Check the logs for server server01 on node kll6571Node01 for more information.

The operation on HelloApp did not complete in the expected amount of time. It may have completed successfully. Check the logs for details.

Application HelloApp on server server02 and node m23vnx60Craig01 started successfully.

Application HelloApp did not start successfully on all the servers in cluster cluster2. Check log for details.

Figure 7 Message from administrative console when application fails to start

This message also appears in the deployment manager SystemOut.log, as shown in Example 17.

Example 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=kl16571Cell01,version=6.0.1.2,name=ApplicationManager,mbea
nIdentifier=ApplicationManager,type=ApplicationManager,node=kl16571Node01,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 18.

Example 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.

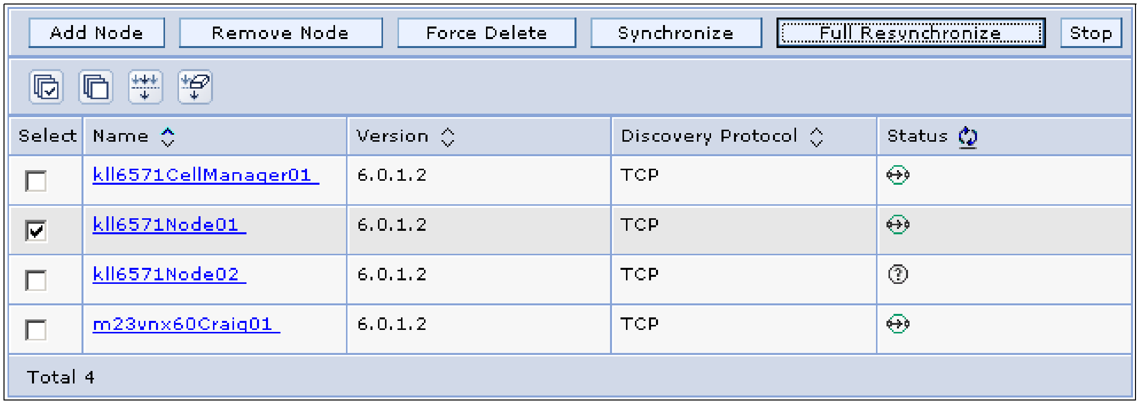
If this file becomes corrupt and the list of applications is lost, then the applications will not be started on the server in this node. When the server process starts, it thinks that the application is not installed and does not try and start it. If you try and start the application from the administrative console, you see the message that is shown in Figure 7 on page 28. The application server logs tell you that the application is not installed, as shown in Example 18 on page 28.

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 8.



Select	Name	Version	Discovery Protocol	Status
☐	kll6571CellManager01	6.0.1.2	TCP	⇔
☑	kll6571Node01	6.0.1.2	TCP	⇔
☐	kll6571Node02	6.0.1.2	TCP	?
☐	m23vnx60Craig01	6.0.1.2	TCP	⇔

Total 4

Figure 8 Full synchronization

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 19 shows that the connection from the addNode script to the deployment manager timed out before the work completed.

Example 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
           kll6571:8879.
ADMU0009I: Successfully connected to Deployment Manager Server: kll6571:8879
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: kll6571Cell01
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 20 on page 31.

Example 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
```

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.

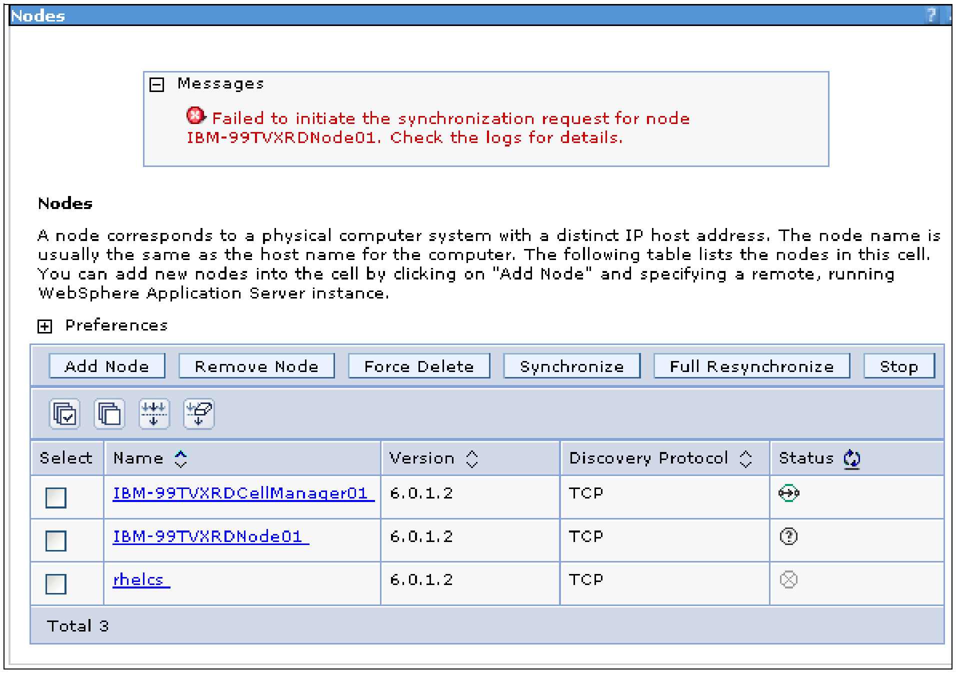
Data to collect

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 9.



⊟ Messages

Failed to initiate the synchronization request for node IBM-99TVXRDNode01. Check the logs for details.

Nodes

A node corresponds to a physical computer system with a distinct IP host address. The node name is usually the same as the host name for the computer. The following table lists the nodes in this cell. You can add new nodes into the cell by clicking on "Add Node" and specifying a remote, running WebSphere Application Server instance.

⊞ Preferences

Select	Name	Version	Discovery Protocol	Status
☐	IBM-99TVXRDCellManager01	6.0.1.2	TCP	↔
☐	IBM-99TVXRDNode01	6.0.1.2	TCP	?
☐	rhelcs	6.0.1.2	TCP	⊗

Total 3

Figure 9 Output from a failed node synchronization

The message shown in Example 21 appears in the SystemOut.log of the deployment manager.

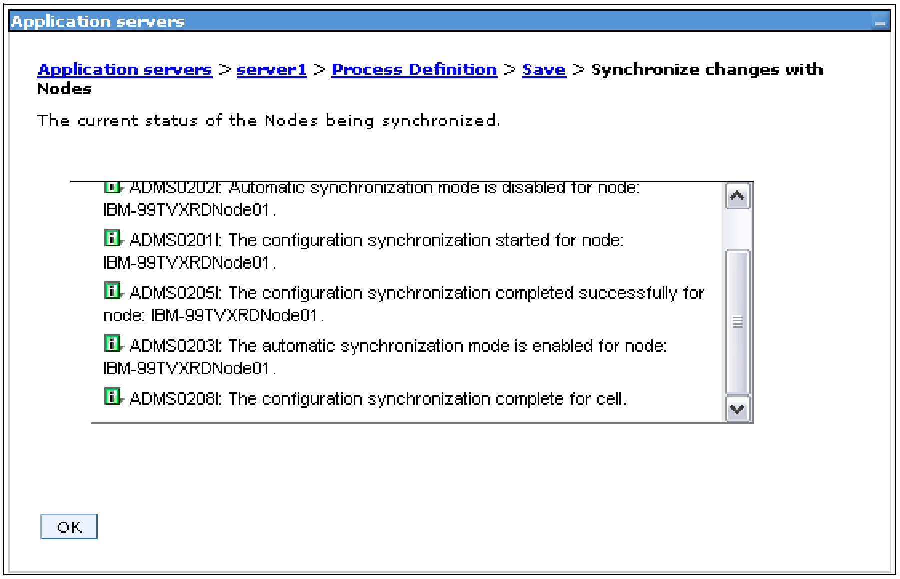
Example 21 Unable to synchronize a node

```
[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-99TVXRDCell01,version=6.0.1.2,name=nodeSync,mbeanIdentifier=
nodeSync,type=NodeSync,node=IBM-99TVXRDNode01,process=nodeagent.
javax.management.InstanceNotFoundException: MBeanServer cannot find MBean with ObjectName
WebSphere:platform=common,cell=IBM-99TVXRDCell01,version=6.0.1.2,name=nodeSync,mbeanIdentifier=
nodeSync,type=NodeSync,node=IBM-99TVXRDNode01,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 10.



Application servers > server1 > Process Definition > Save > Synchronize changes with Nodes

The current status of the Nodes being synchronized.

ADMS0202I: Automatic synchronization mode is disabled for node: IBM-99TVXRDNode01 .

ADMS0201I: The configuration synchronization started for node: IBM-99TVXRDNode01 .

ADMS0205I: The configuration synchronization completed successfully for node: IBM-99TVXRDNode01 .

ADMS0203I: The automatic synchronization mode is enabled for node: IBM-99TVXRDNode01 .

ADMS0208I: The configuration synchronization complete for cell.

Figure 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 22.

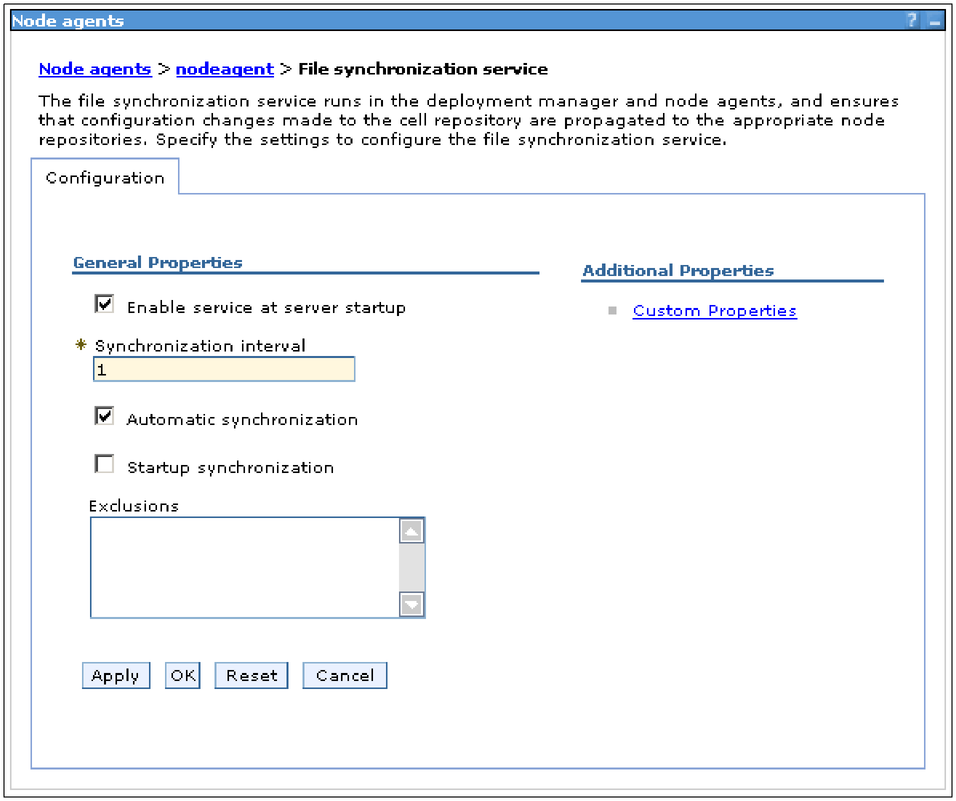
Example 22 Node agent fails to open server.xml

```
[7/3/05 16:11:13:902 EDT] 00000040 FileDocument  E   ADMR0104E: The system is unable to read
document cells/IBM-99TVXRDCell01/nodes/IBM-99TVXRDNode01/servers/server1/server.xml:
java.io.FileNotFoundException:
C:\IBM\WAS6\AppServer\profiles\AppSrv01\config\cells\IBM-99TVXRDCell01\nodes\IBM-99TVXRDNode01\
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 11 shows the file synchronization settings for the node agent. In this instance, automatic synchronization is enabled.



Node agents

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

Additional Properties

- Custom Properties

Figure 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 23. You need to pass it the host name and SOAP port of the deployment manager. You can also specify the **-trace** option.

Example 23 Running and tracing syncNode

```
[wasadmin@m23vnx60 bin]$ ./syncNode.sh kl16571 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 kl16571: 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 24.

Example 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/kl16571Cell01/nodes/m23vnx60Node01/servers/server02/server.xml:
java.io.FileNotFoundException:
/opt/IBM/WAS6/AppServer/profiles/Node01/config/cells/kl16571Cell01/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 25 on page 36 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.

Example 25   Benign FileNotFound exception

```
java.io.FileNotFoundException:
/opt/IBM/WAS6/AppServer/profiles/Node01/config/cells/kl16571Cell01/nodes/m23vnx60Node01/servers
/nodeagent/variables.xml (No such file or directory)
```

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.

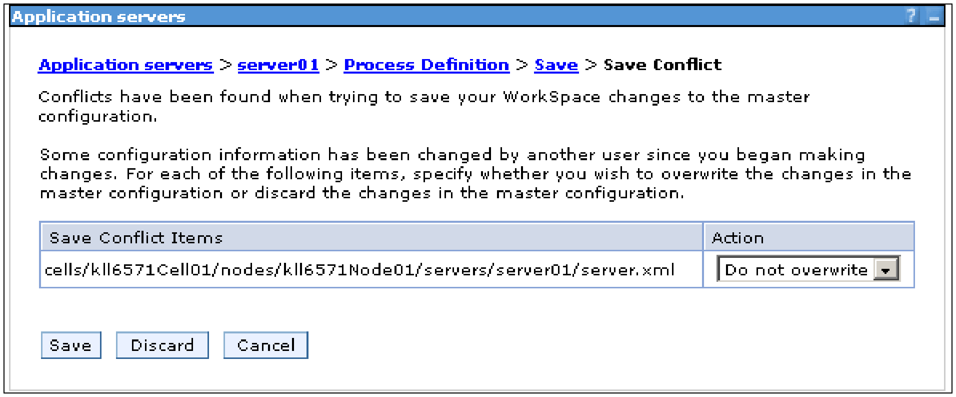
Data to collect

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 12.


Application servers
Application servers > server01 > Process Definition > Save > Save Conflict

Conflicts have been found when trying to save your WorkSpace changes to the master configuration.

Some configuration information has been changed by another user since you began making changes. For each of the following items, specify whether you wish to overwrite the changes in the master configuration or discard the changes in the master configuration.

Save Conflict Items	Action
cells/kll6571Cell01/nodes/kll6571Node01/servers/server01/server.xml	Do not overwrite

Figure 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 26.

Example 26 Overwriting configuration changes

```
[7/11/05 11:20:03:348 EDT] 00000037 FileRepositor A   ADMR0016I: User WTRNTDM/config modified
document cells/kll6571Node01Cell/nodes/kll6571StandAlone/servers/server1/server.xml.
[7/11/05 11:20:10:569 EDT] 0000003f ServletWrappe A   SRVE0242I:
[/secure/tiles/syncconflict.jsp]: Initialization successful.
[7/11/05 11:20:23:808 EDT] 00000038 FileRepositor W   ADMR0114W: The system is overwriting
document cells/kll6571Node01Cell/nodes/kll6571StandAlone/servers/server1/server.xml by request.
[7/11/05 11:20:23:848 EDT] 00000038 FileRepositor A   ADMR0016I: User WTRNTDM/wasadmin modified
document cells/kll6571Node01Cell/nodes/kll6571StandAlone/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.

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 27.

Example 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 28.

Example 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 29.

Example 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.

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 8 for details.

What to look for

WebSphere Application Server writes a warning message (Example 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 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 31:

```
traceString=ORBRas=all
```

Example 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 32.

Example 32 Key database files defined in security.xml

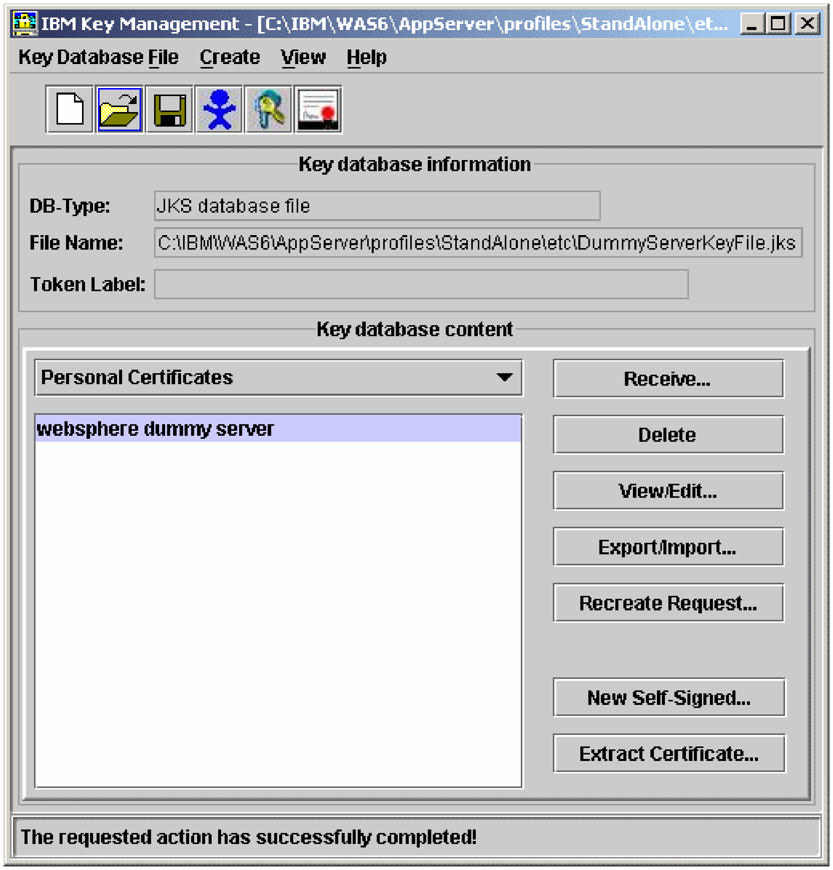
```
<setting xmi:id="SecureSocketLayer_1"
keyFileName="${USER_INSTALL_ROOT}/etc/DummyServerKeyFile.jks"
keyFilePassword="{xor}CDo9Hgw=" keyFileFormat="JKS"
trustFileName="${USER_INSTALL_ROOT}/etc/DummyServerTrustFile.jks"
trustFilePassword="{xor}CDo9Hgw=" trustFileFormat="JKS"
clientAuthentication="false" 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

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 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.



IBM Key Management - [C:\IBM\WAS6\AppServer\profiles\StandAlone\et...

Key database information

DB-Type: JKS database file

File Name: C:\IBM\WAS6\AppServer\profiles\StandAlone\etc\DummyServerKeyFile.jks

Token Label:

Key database content

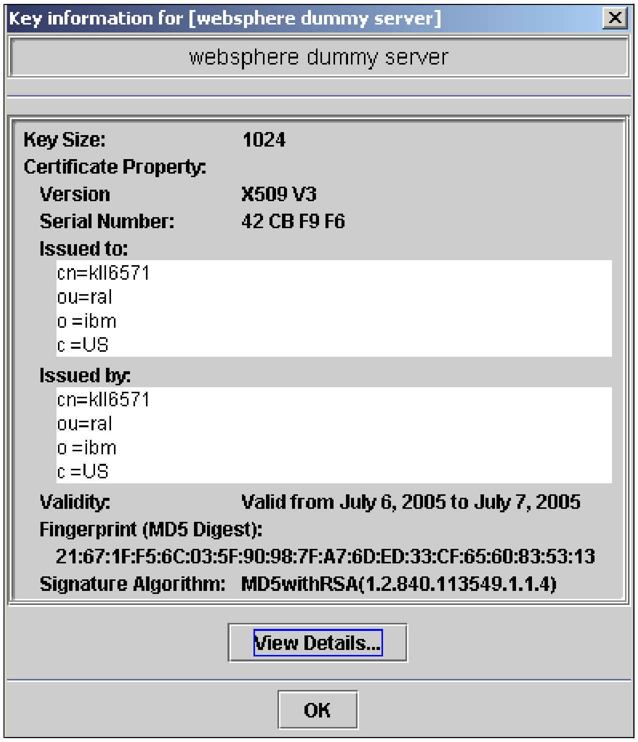
Personal Certificates

websphere dummy server

The requested action has successfully completed!

Figure 13 Open key database in the ikeyman utility

The Validity field also shows if the certificate has expired. In Figure 14 on page 42, the certificate is only valid up to July 7, 2005. Click **OK** to close the window.



Key Size: 1024

Certificate Property:

Version X509 V3

Serial Number: 42 CB F9 F6

Issued to:

cn=kll6571
ou=ral
o =ibm
c =US

Issued by:

cn=kll6571
ou=ral
o =ibm
c =US

Validity: Valid from July 6, 2005 to July 7, 2005

Fingerprint (MD5 Digest):

21:67:1F:F5:6C:03:5F:90:98:7F:A7:6D:ED:33:CF:65:60:83:53:13

Signature Algorithm: MD5withRSA(1.2.840.113549.1.1.4)

Figure 14 Expired certificate details

Action: Request a new key from a certificate authority

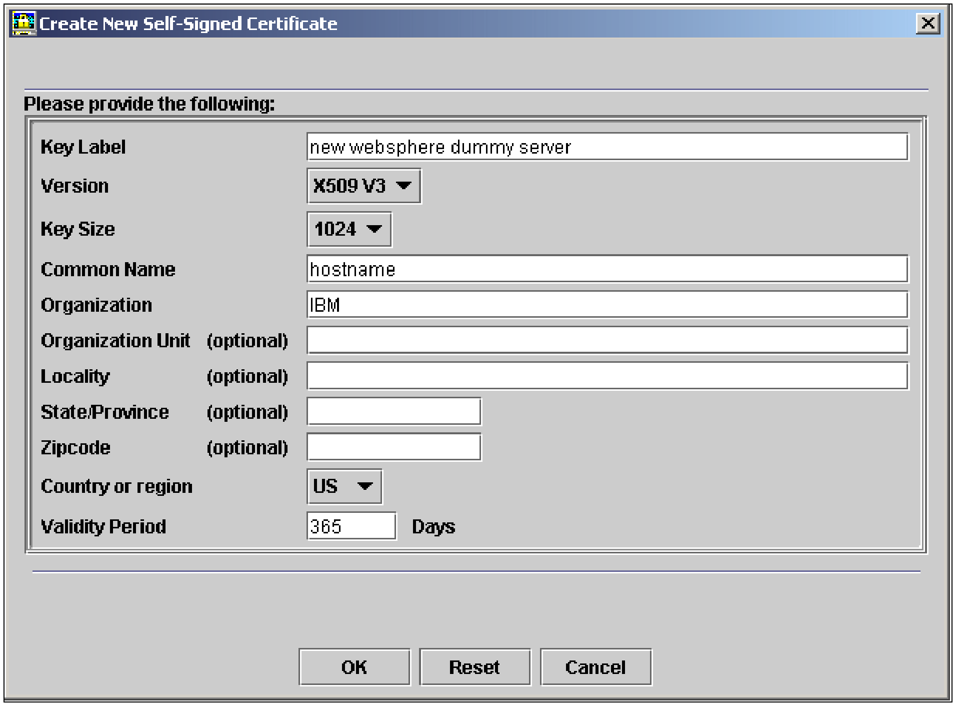
If the certificate has expired, you can resolve this problem by creating a new valid certificate in the existing key database file.

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 13 on page 41.



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

Figure 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`

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 48.

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 mange 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 **wrd-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>/.wrdconfig.xml
```

Example 33 shows the contents of this file.

Example 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 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 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: -noupdate -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 35 on page 47.

Example 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 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 36.

Example 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.

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

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.

This document created or updated on September 30, 2005.

Send us your comments in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HZ8 Building 662, P.O. Box 12195
  Research Triangle Park, NC 27709-2195 U.S.A.

Trademarks

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

eserver®
eserver®
Redbooks (logo) ™
IBM®
Redbooks™
SecureWay®
WebSphere®

The following terms are trademarks of other companies:

Java, JSP, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, 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.

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