IBM

IBM Systems - iSeries

e-business and Web serving
WebSphere Application Server - Express Version 5.1
Troubleshooting

Version 5 Release 4

IBM

IBM Systems - iSeries

e-business and Web serving
WebSphere Application Server - Express Version 5.1
Troubleshooting

Version 5 Release 4

> **Note**
>
> Before using this information and the product it supports, be sure to read the information in "Notices," on page 31.

Contents

Troubleshooting

WebSphere$^{(R)}$ Application Server - Express offers several methods you can use to troubleshoot problems. Which method you use depends on the nature of the problem. Generally, you use a combination of methods to determine the cause of a problem and then decide on an appropriate method for its resolution. Whether you are a beginner or experienced user, this problem determination section leads you to resources and techniques to help you identify and respond to problems.

Support services for IBM$^{(R)}$ WebSphere Application Server - Express for iSeries are provided under the usual terms and conditions for iSeries software products. Support services include program services, voice support, and consulting services. For more information, use the online information provided at http://www.IBM.com/eserver/iseries/support/

or contact your local IBM representative. Specify program number 5722-E51.

For information on using the iSeries System Debugger to debug your applications, see the iSeries System Debugger topic.

"Troubleshoot by topic"
This topic describes several general areas where problems may occur, and what resources you should use to determine the cause of the problem for Web resources such as servlets, Java$^{(TM)}$Server Pages $^{(TM)}$ (JSP $^{(TM)}$), and HTML files.

Detect hung threads in J2EE applications
This topic describes hung threads and how to adjust and configure the hang detection policy.

"Resources for monitoring the WebSphere Application Server - Express environment" on page 8
This topic describes the resources you can use to monitor the WebSphere Application Server - Express environment.

"WebSphere Application Server log files" on page 17
This topic describes the log files created by the WebSphere Application Server - Express components. The topic also includes information on how to configure the log files.

"Use the WebSphere Application Server - Express trace service" on page 25
This topic describes how to enable and configure the WebSphere Application Server - Express trace service. The topic also includes information on interpreting the contents of a trace file.

Add logging and tracing to your application
This topic describes how to add logging and tracing to your applications using the JRas framework.

WebSphere Application Server - Express system topology
This topic describes the topology of the WebSphere Application Server - Express environment and how your application fits into this topology.

Troubleshoot by topic

Errors occurring within the application server fall within one of the categories shown below. Select the link that most closely matches the area of the problem to see the appropriate resources to consult for determining what is causing the error.

"Troubleshoot: WebSphere Application Server - Express startup" on page 3
This topic describes the resources available to determine what may be causing a problem when you attempt to start the application server or administrative console.

"Troubleshoot: WebSphere Application Server - Express administration and administrative console" on page 3
This topic describes the resources available to determine what may be causing a problem when you attempt to use the administrative console or the administrative scripting (wsadmin) tool.

"Troubleshoot: Enterprise applications" on page 8
This topic describes the resources available to determine what may be causing a problem when you attempt to run your enterprise application within the application server environment.

HTTP plug-in component
This topic describes the resources available to determine what may be causing a problem with the HTTP plug-in component - the component that sends requests from your HTTP server.

HTTP sessions
This topic describes the resources available to determine what may be causing a problem when you run HTTP sessions.

"Troubleshoot: Installation"
This topic describes the resources available to determine what may be causing a problem when you install WebSphere Application Server - Express for iSeries.

"Troubleshooting: Naming service" on page 6
This topic describes the resources available to determine what may be causing a problem when you attempt to use the naming service to lookup application server resources such as data sources.

"Troubleshoot: Security" on page 4
This topic describes the resources available to determine what may be causing a problem when using security.

Starting applications
This topic describes the resources available to determine what may be causing a problem when you start an application.

"Troubleshoot: Universal Description, Discovery, and Integration (UDDI), Web Service, and SOAP" on page 8
This topic describes the resources available to determine what may be causing a problem when you attempt to deploy and run applications that use the application server Web Services, UDDI, or SOAP.

"Troubleshoot: Web resources" on page 6
This topic describes the resources available to determine what may be causing a problem when an application server fails to serve Web resources such as servlets, JavaServer Pages (JSP), and HTML files.

Troubleshoot: Installation

If a problem occurs when you attempt to install WebSphere Application Server - Express for iSeries or a new resource, the first thing you should do is consult the installation documentation for the product. See Install WebSphere Application Server - Express for iSeries in the *Installation* topic for more information.

For the resources available for determining what the installation problem might be, see the list below:

- If an error occurs during installation, a message will be entered in the job log. To get more information about the message, use the Display Message Description (DSPMSGD) command from an CL command line.
- If an error occurs when you are installing or configure a resource, typically an error page or message containing error information is displayed. Click the **Details** button to view more information on the possible cause of the problem.
- Check the application server standard output and standard error log files. For information on the application server log files and where they are located, see "WebSphere Application Server log files" on page 17.
- Read the release notes:
  - WebSphere Application Server - Express V5.1.0.X Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp51.html)
  - WebSphere Application Server - Express V5.1.1 Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp511.html)

Troubleshoot: WebSphere Application Server - Express startup

If a problem occurs when you attempt to start the WebSphere Application Server - Express product, the first thing you should do is consult the installation documentation. See the installing WebSphere Application Server - Express for iSeries topic for more information.

These resources are available to help you determine what the startup problem might be:
- Check the job log of the failing application server for errors. For information on viewing job logs, see "Resources for monitoring the WebSphere Application Server - Express environment" on page 8.
- Check the application server log files for errors. For information on the application server log files and where they are located, see "WebSphere Application Server log files" on page 17.
- Read the release notes:
  - WebSphere Application Server - Express V5.1.0.X Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp51.html)
  - WebSphere Application Server - Express V5.1.1 Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp511.html)

Troubleshoot: WebSphere Application Server - Express administration and administrative console

Use the following resources to determine the cause of the problem when attempting to use the administrative console or the WebSphere Application Server admin scripting tool (wsadmin):

- The server process must be running in order to use the administrative console. For wsadmin, many functions require a connection to the server as well. Verify that the server is running and ready to receive administrative requests. For more information on how to verify that the server is running, see the verify the installation topic.

  If the server has not started and you are using wsadmin for an administrative function that does not require the server to be started, ensure that you have specified -conntype NONE when invoking wsadmin.

- Use the TCP/IP ping command to test that the hostname is reachable from the system where the browser or wsadmin program is being used. If you are able to ping the hostname, this indicates that there are no firewall or connectivity issues.

- If the host where the application server is running is remote, ensure that hostname in the browser URL for the console is correct, or the -host *hostname* option of the wsadmin command is being used to direct wsadmin to the correct server.

- If you are using the wsadmin tool, see the Start and stop applications with wsadmin topic. Check the wsadmin log files located in the logs directory of your instance for errors.

- Check the log files for server errors. For information on the log files and where they are located, see "WebSphere Application Server log files" on page 17.

- Read the release notes:
  - WebSphere Application Server - Express V5.1.0.X Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp51.html)
  - WebSphere Application Server - Express V5.1.1 Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp511.html)

Troubleshoot: Security

Use these resources to determine the cause of problems that occur when using the application server security.

- Check the application server standard output and standard error log files. See "WebSphere Application Server log files" on page 17 for more information.

- When troubleshooting security-related problems, consider the following:

  Does the problem occur when security is disabled?
  The problem may be a result of the enablement of security. More troubleshooting is necessary to ensure the problem is security related.

  For more information, see Troubleshoot: Enabling security.

  Did security appear to initialize properly?
  The following sequence of messages generated in the SystemOut.log indicate normal code initialization of an application server. This varies based on the configuration, but the message are similar:

```
SASRas        A JSAS0001I: Security configuration initialized.
SASRas        A JSAS0002I: Authentication protocol: CSIV2/IBM
SASRas        A JSAS0003I: Authentication mechanism: SWAM
SASRas        A JSAS0004I: Principle name: BIRKt20/pbirk
SASRas        A JSAS0005I: SecurityCurrent registered.
SASRas        A JSAS0006I: Security connection interceptor initialized.
SASRas        A JSAS0007I: Client request interceptor registered.
SASRas        A JSAS0008I: Server request interceptor registered.
SASRas        A JSAS0009I: IOR interceptor registered.

NameServerIMP I NMSV0720I: Do Security service listener registration.
```

```
SecurityCompo A SECJ0242A: Security service is starting
UserRegistryI A SECJ0136I:
        Custom Registry:com.IBM.ws.security.registry.nt.NTLocalDomainRegistryIm
        has been initialized
SecurityCompo A SECJ0202A: Admin application initialized successfully
SecurityCompo A SECJ0203A: Naming application initialized successfully
SecurityCompo A SECJ0204A: Rolebased authorizer initialized successfully
SecurityCompo A SECJ0205A: Security Admin mBean registered successfully
SecurityCompo A SECJ0243A: Security service started successfully
SecurityCompo A SECJ0210A: Security enabled true
```

Is there a stack trace or exception printed in the SystemOut.log?

The stack trace will log any code incorrectly initialized, failing components, and the failing class.

Is this a distributed security problem or a local security problem?

– If the problem is local, the code involved does not make a remote method invocation, then troubleshooting is isolated to a single process. It is important to know when a problem is local or distributed since the behavior of the Object Request Broker (ORB), among other components, is different between the two.

– Once a remote method invocation takes place, a different security code path is entered. When you know the problem involves two or more servers, check the log files of all servers involved. If possible, make sure the timestamps on all machines match as closely as possible to identify request and reply pairs from two different processes easier.

Is the problem related to authentication or authorization?

Most security problems fall under one of these two categories. Authentication is the process of determining who the caller is. Authorization is the process of validating that the caller has the proper authority to invoke the requested method. When authentications fails, typically this is related to either the authentication protocol, authentication mechanism, or user registry. When authorization fails, this is usually related to the application bindings from assembly or deployment and to the identity of the caller who is accessing the method and the roles required by the method.

Does the problem seem to be related SSL?

The Secure Socket Layer (SSL) is a separate layer of security. Troubleshooting SSL is different than troubleshooting authentication and authorization problems. SSL errors are often caused by incorrect configurations. Each keystore used by a client must contain the certificate of the Certificate Authority (CA) that signed the certificate used by the server. During mutual authentication, the server requires the client to present a certificate for authorization. Each server keystore must contain the certificate of the CA that signed the certificate presented by the client. Another common error are configurations where the client and the server do not have common configured SSL cipher suites.

Is the problem related to Java 2 Security?

If Java 2 Security is enabled, deployers and administrators are required to make sure that all applications are granted required permissions, otherwise, applications may fail to run.

- Read the release notes:

  – WebSphere Application Server - Express V5.1.0.X Release Notes

  (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp51.html)

  – WebSphere Application Server - Express V5.1.1 Release Notes

  (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp511.html)

Troubleshooting: Naming service

The name service is a J2EE service which publishes and provides access to resources such as connection pools. If you have problems in accessing a resource which otherwise appears to be active, the naming service might be involved. Use these resources to determine the cause of the problem:

- Check the log files of the server for errors. Messages starting with NMSV are related to the naming service. For information on the logs files and where they are located, see "WebSphere Application Server log files" on page 17.
- With the application server running, use the dumpNameSpace script to view the resources bound in to the name space for the application server. If the object a client needs to access does not appear in the name space, use the administrative console to verify that:
  - The server hosting the target resource is started.
  - The Web module hosting the target resource is running.
  - The JNDI name of the target resource is correct and up-to-date.
- If you see an exception that appears to be CORBA related, where CORBA appears as part of the exception name, look for a naming-services-specific CORBA minor code, in the exception stack for information on the cause of the problem.
- Read the release notes:
  - WebSphere Application Server - Express V5.1.0.X Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp51.html)
  - WebSphere Application Server - Express V5.1.1 Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp511.html)

Troubleshoot: Web resources

Use the resources listed below to determine the cause of problems that occur when you are unable to access Web resources:

Unable to serve any Web resources

- See the installation of WebSphere Application Server - Express for iSeries topic. The topic will guide you through the system verification and serving of a Web resource.
- If you are using HTTP Server (powered by Apache), generate a trace on your HTTP Server instance and look for errors. See "Use the WebSphere Application Server - Express trace service" on page 25for more information.
- Web server plug-in trace.

  The plug-in log records HTTP server-side processing and servlet request routing between the HTTP Server and WebSphere Application Server - Express. (The plug-in connects the HTTP Server and WebSphere Application Server - Express together.)

  Manage this log by editing the application server plugin-cfg.xml file. The plugin-cfg.xml file contains a tag near the beginning named Log. Log has two attributes: LogLevel and Name.
  - LogLevel specifies the amount and type of information that is logged to a file. Valid values are Trace, Warn, and Error.
  - Name specifies the location and name of the file where logging information is written.
- Read the release notes:
  - WebSphere Application Server - Express V5.1.0.X Release Notes

(http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/
express/docs/relnotesexp51.html)

– WebSphere Application Server - Express V5.1.1 Release Notes

(http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/
express/docs/relnotesexp511.html)

Unable to serve a particular Web resource

- Check the application server standard output and standard error log files for errors. For information on the application server log files and where they are located, see "WebSphere Application Server log files" on page 17.
- Read the release notes:
  - WebSphere Application Server - Express V5.1.0.X Release Notes

(http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/
express/docs/relnotesexp51.html)
  - WebSphere Application Server - Express V5.1.1 Release Notes

(http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/
express/docs/relnotesexp511.html)

Servlet or JSP displays an error or Java exception instead of the expected output

- Check the application server standard output and standard error log files for errors. For information on the application server log files and where they are located, see "WebSphere Application Server log files" on page 17.
- Add logging and tracing to your application with the JRas framework. See "Use the WebSphere Application Server - Express trace service" on page 25 for more information.
- Add output to your servlet, JSP, or Java code in the form of System.out.println() statements to aid in debugging. The output from any System.out.println() statements is written to the standard output log file for the application server in which the code is running. For information on the application server log files and where they are located, see "WebSphere Application Server log files" on page 17.
- If the failure occurs within a JSP, use the keepgenerated attribute to keep the generated servlet source file to inspect for errors.
- Read the release notes:
  - WebSphere Application Server - Express V5.1.0.X Release Notes

(http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/
express/docs/relnotesexp51.html)
  - WebSphere Application Server - Express V5.1.1 Release Notes

(http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/
express/docs/relnotesexp511.html)

Troubleshoot: Enterprise applications

Use the resources listed below to determine the cause of problems that occur when running your application with the application server:

- Check the application server standard output and standard error log files for errors. For information on the application server log files and where they are located, see "WebSphere Application Server log files" on page 17.

- Add logging and tracing to your application with the JRas framework. See "Use the WebSphere Application Server - Express trace service" on page 25 for more information.

- Add output to your Java code in the form of System.out.println() statements to aid in debugging. The output from any System.out.println() statements is written to the standard output log file for the application server in which the code is running. For information on the application server log files and where they are located, see "WebSphere Application Server log files" on page 17.

- Read the release notes:
  - WebSphere Application Server - Express V5.1.0.X Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp51.html)
  - WebSphere Application Server - Express V5.1.1 Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp511.html)

Troubleshoot: Universal Description, Discovery, and Integration (UDDI), Web Service, and SOAP

These resources are available to help you determine what might be causing problems when you deploy or run applications which use Web Services, UDDI, or SOAP.

- Check the application server log files and HTTP Server log files for errors. For information on the application server log files and where they are located, see "WebSphere Application Server log files" on page 17.

- Read the release notes:
  - WebSphere Application Server - Express V5.1.0.X Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp51.html)
  - WebSphere Application Server - Express V5.1.1 Release Notes

    (http://www.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/express/docs/relnotesexp511.html)

Resources for monitoring the WebSphere Application Server - Express environment

WebSphere Application Server - Express has several resources for monitoring your application server environment. These topics provide information on the different methods you can use to monitor your application server environment:

"Monitor the application server jobs with iSeries commands"
This topic describes the jobs that make up the application server runtime and how you can use
iSeries commands to monitor them.

This topic describes how you can use a message queues to monitor the jobs that make up the
application server runtime. The same messages that are sent to the job log for the application server
jobs can also be sent to a user-specified message queue.

This topic describes how you can use the prerequisite validator to verify whether you have all the
required product prerequisites.

This topic describes how you can use the port validator to check for port conflicts.

The servicetools script provides serviceability tools that you can use to help troubleshoot WebSphere
Application Server - Express. This topic describes the servicetools script and the tools that it
provides.

Product history report scripts
This topic describes how to use the scripts to generate product history reports.

Product version report scripts
This topic describes how to use the scripts to generate product version reports.

Monitor the application server jobs with iSeries commands

This topic describes the jobs used by an application server and how you can monitor jobs by using
iSeries command language (CL) commands.

- Each application server instance consists of one or more jobs. These jobs run in the QASE51 subsystem.
  The job name is the first 10 characters of the application server name. If the first 10 characters do not
  provide a valid iSeries job name, the application server runtime creates a valid job name. If the runtime
  cannot create a valid job name for the application server, the application server will not start.

  The application server provides the runtime environment for server-side Java components. The
  application server plug-in interfaces with the Web server to handle client requests for server-side
  resources and route them to their application server for processing.

- If using the IBM Developer Kit for Java JDBC driver for database access, your instance uses one or
  more QSQSRVR jobs that run in the QSYSWRK subsystem. To determine what QSQSRVR job your
  application server job is using, view the server job log. For each JDBC connection obtained, the
  message SQL7908 with message text similar to the following:

  ```
  Job 163707/QUSER/QSQSRVR used for SQL server mode processing.
  ```

- If you are using the IBM Toolbox for Java JDBC driver for database access, your instance uses one or
  more QZDASOINIT jobs that run in subsystem QUSRWRK.

- Depending on which Web server you are using to serve Web resources, your instance uses jobs from
  the Web server instance:

  - **IBM HTTP Server (powered by Apache)**
    Each HTTP Server (powered by Apache) instance consists of two or more jobs that run in the
    QHTTPSVR subsystem. The name of each job is the same as the name of the HTTP Server instance.
    The application server plug-in code runs in the second job listed for HTTP Server (powered by
    Apache) instance.

  - **Domino Web Server**
    Each Domino Web Server instance has a corresponding subsystem in which the jobs for the instance
    run. The subsystem for the first Web server instance created is DOMINO001, the subsystem for the

second Web server instance created is DOMINO002, and so on. The application server plug-in code runs in the job named HTTP in the subsystem in which your Web server instance is running.

OS/400 command language (CL) commands for monitoring

Use iSeries CL commands to monitor jobs that run in your application server environment. You can view the job logs, display message details, and view spooled files associated with the jobs. The job logs and associated spooled files for the jobs can contain valuable information for determining the root of a problem.

This table lists some iSeries commands that you can use to monitor application server jobs:

Task	iSeries command and description
Work with active application server jobs	For application server jobs that run in the QASE51 subsystem: `WRKACTJOB SBS(QASE51)`
Work with all jobs with a specific name	This command lists all jobs, active or not, named *job_name*. The jobs are listed by date, most recent first: `WRKJOB JOB(job_name)` This command is very useful when an application server ends abnormally or fails to start successfully. For example, if an application server started then ended immediately, you could do the following with the job log: • Run the command `WRKJOB JOB(SERVER1)` . • Specify option **1** on the option line next to the first job listed that has a status of OUTQ. • Specify option **4** (Work with spooled files) on the command line of the Work with Job screen. • Specify option **5** next to the QPJOBLOG spooled file to view the contents.
Work with a specific job	This command displays the Work with Job screen for the specified job: `WRKJOB JOB(job_number/job_user/job_name)` This command is useful when you know the fully qualified job information for the job.
View the job log for an active application server job	1. Run the following command: • `WRKJOB SBS(QASE51)` 2. Specify option **5** (Work with) in the option line next to the active job whose job log you want to view. 3. In the Work with Job display, specify option **10** (Display job log, if active or on job queue). 4. Press the **F10** key to display all messages. 5. In the job log, position the cursor on a message for which you want to display extended message information. 6. Press the **F1** (Help) key.

Task	iSeries command and description
Work with application server jobs that run under the QEJBSVR user profile	`WRKUSRJOB USER(QEJBSVR)`
Delete spooled files for QEJBSVR user profile	`DLTSPLF FILE(*SELECT) SELECT(QEJBSVR)`
View the spooled files for a job	1. Run the following command: • `WRKJOB SBS(QASE51)` 2. Specify option **8** (Work with Spooled Files) in the option line next to the active job whose job log you want to view. 3. Specify option **5** (Display) in the option line next to the spooled file you want to view. You can view ended jobs by using the Work with User Jobs (WRKUSRJOB) or Work with Job (WRKJOB) commands.

Monitor the application server jobs with a message queue

The application server allows you to specify an iSeries message queue object where product messages are sent. The product messages are the same application server messages that are sent to the job log.

To enable message queue support for an application server, use the administrative console to specify the os400.WebSphere.message.queue system property for your server. The message queue is specified by using the fully qualified pathname of the object. For message queues in library QSYS, the format is /QSYS.LIB/*messageQueue*.MSGQ. For message queues in libraries other than QSYS, the format is /QSYS.LIB/*yourLib*.LIB/*messageQueue*.MSGQ. The message queue must exist and the QEJBSVR user profile must have *CHANGE authority to the message queue.

To use the os400.WebSphere.message.queue system property for your application server, follow these steps:

1. Start HTTP Server Administration interface.
2. Click the **Manage** tab.
3. Select your application server from the **Server** list.
4. Expand **Tools**.
5. Click **Launch Express Console**.
6. Expand **Servers**.
7. Click **Application Server**s.
8. Select your application server from the table.
9. Click **Process Definition**.
10. Click **Java virtual machine**.
11. Click **Custom Properties**.
12. Click **New**.
13. Specify **os400.WebSphere.message.queue** in the **Name** field.
14. Specify the fully qualified pathname for the message queue in the **Value** field.
15. Click **OK**.
16. Click the **Save** link at the top of the page to save your configuration changes. Click **Save** on the resulting page.
17. Stop and restart your application server.

Verify product prerequisites

WebSphere Application Server - Express may not run properly if all of the required software prerequisites are not installed. If the server does not start, run the prerequisite validator. It verifies your installation, ensures that you have the appropriate prerequisite software and PTFs installed, and verifies configuration settings required by WebSphere Application Server - Express.

To run the prerequisite validator, run the checkprereqs script from the Qshell command line.

1. On the CL command line, enter the STRQSH (Start Qshell) command.
2. On the Qshell command line, use the cd command to change to the directory that contains the script:

   `cd /QIBM/ProdData/WebASE51/ASE/bin`
3. Run the checkprereqs script:

   `checkprereqs`

   For more information and options for this script, see "The checkprereqs script."

Note: The prerequisite validator can also be run from the servicetools script. For more information see "The prerequisite validator tool" on page 13.

The checkprereqs script

The checkprereqs script runs the prerequisite validator tool. This tool verifies your WebSphere Application Server - Express installation, ensures that you have the appropriate prerequisite software and PTFs installed, and verifies certain configuration settings required by the product.

Authority

To run this script, your user profile must have *ALLOBJ authority.

Syntax

The syntax of the script is:

```
checkprereqs [ -xmlinput xml_input_file ] [ -xmloutput xml_output_file ]
  [ -output output_file ] [ -trace trace_file ] [ -help ]
```

Parameters

The parameters of the script are:

- **-xmlinput**
  This is an optional parameter. The value *xml_input_file* specifies the fully qualified path of an XML file that the script uses. If you do not specify this parameter, the script uses a default XML file that is included in the ServiceTools.jar file.
- **-xmloutput**
  This is an optional parameter. The value *xml_output_file* specifies the fully qualified path of an XML file to which the script writes its results. If you do not specify this parameter, the script writes its output to the display.
- **-output**
  This is an optional parameter. The value *output_file* specifies the fully qualified path of a file to which the script writes its standard output. If you do not specify this parameter, the script does not generate standard output.
- **-trace**
  This is an optional parameter. The value *trace_file* specifies the fully qualified path of a file to contain trace information. If you do not specify this parameter, the script does not generate trace information.
- **-verbose**
  This is an optional parameter. If you specify this parameter, the script displays all result information. If you do not specify this parameter, the script does not display successful result information.

- **-help**
  This optional parameter displays the usage statement for the script.

The prerequisite validator tool

The prerequisite validator tool verifies your WebSphere Application Server - Express installation, ensures that you have the appropriate prerequisite software and PTFs installed, and verifies certain configuration settings required by the product.

Note: The prerequisite validator tool can also be run from the checkprereqs script. See The checkprereqs script for more information.

For descriptions of the syntax and parameters that apply to the servicetools script, see "The servicetools script" on page 15.

Syntax

The syntax to run the prerequisite validator tool is:

```
servicetools [ parameters ] -checkprereqs [ -xmlinput xml_input_file ]
  [ -xmloutput xml_output_file ] [ -output output_file ]  [ -verbose ] [ -help ]
```

Parameters

The parameters for the prerequisite validator tool are:

- *parameters*
  This optional parameter list specifies parameters for the servicetools script. See "The servicetools script" on page 15 for more information. The servicetools parameters must be specified before the -checkprereqs parameter.
- **-checkprereqs**
  This parameter invokes the prerequsite validator.
- **-xmlinput**
  This is an optional parameter. The value *xml_input_file* specifies the fully qualified path of the XML file that lists the prerequisites that the script checks. If this parameter is not specified, the script uses the default XML input file that is included in the ServiceTools.jar file.
- **-xmloutput**
  This is an optional parameter. The value *xml_output_file* specifies the fully qualified path of the XML file to which the script writes the results. If neither this parameter nor the -output parameter is specified, output information is written to the display.
- **-verbose**
  This is an optional parameter. If you specify this parameter, the script displays all result information. If you do not specify this parameter, the script does not display successful result information.
- **-output**
  This is an optional parameter. If you specify this parameter, the script displays all result information. If you do not specify this parameter, the script does not display successful result information.
- **-help**
  This optional parameter displays the usage statement for the script.

Check for port conflicts

WebSphere Application Server - Express may not run properly if you have port conflicts. For example, your server may not start. The port validator verifies your configuration to ensure that you do not have port conflicts across WebSphere Application Server instances and products. It detects port conflicts between instances and servers in different Version 4 and Version 5 products (Version 4 Advanced Edition, Version 4 Advanced Edition Single Server, Version 5 Base, Version 5 Network Deployment, Version 5 Express, Version 5.1 Base, Version 5.1 Network Deployment, and Version 5.1 Express).

To run the port validator, run the servicetools script from the Qshell command line and specify the
-portconflict parameter.

1. On the CL command line, enter the STRQSH (Start Qshell) command.

2. On the Qshell command line, use the cd command to change to the directory that contains the script:

   `cd /QIBM/ProdData/WebASE51/ASE/bin`

3. Run the servicetools script to check for port conflicts:

   `servicetools -portconflict -products all`

   For more information, see "The port validator tool."

The port validator tool

The port validator tool verifies your WebSphere Application Server - Express configuration to ensure that
you do not have port conflicts between WebSphere Application Server instances and products. The port
validator tool is one of the tools available with the servicetools script.

For descriptions of the syntax and parameters that apply to the servicetools script, see "The servicetools
script" on page 15.

Syntax

The syntax to run the port validator tool is:

```
servicetools [ parameters ] -portconflict [ -products product_list ]
  [ -instances instance_list ] [ -ports port_list ]
  [ -comparetoproduct compare_prod ] [ -comparetoinstance compare_inst ]
  [ -xmloutput output_file ]
```

Parameters

The parameters for the port validator tool are:

- *parameters*
  This optional parameter list specifies parameters for the servicetools script. See "The servicetools
  script" on page 15 for more information. The servicetools parameters must be specified before the
  -portconflict parameter.

- **-portconflict**
  Specify this parameter to invoke the port validator tool.

- **-products**
  This is an optional parameter. The value *product_list* specifies the product or products for which you
  want to check for port conflicts. If you specify multiple products, separate them with a colon (:). Valid
  products are WAS40Adv, WAS40AEs, WAS50Base, WAS50Express, WAS50ND, WAS50PME,
  WAS50PMEND, WAS51Base, WAS51Express, WAS51ND, WAS51PME, WAS51PMEND, and ALL.

  The port validator tool uses the -products and -instances parameters to determine the set of ports that
  it checks for conflicts. The tool generates a default set of ports that includes the ports used by the
  specified products and instances.

  If you do not specify this parameter, the port validator tool checks for port conflicts in the products
  specified by the servicetools -products parameter.

  If you specify this parameter and the -products parameter for servicetools, the port validator tool
  checks only the products that you specify for the -portconflict -products parameter.

  You must specify the -product parameter for either the servicetools script or the port validator tool, or
  both.

- **-instances**
  This is an optional parameter. The value *instance_list* specifies the instance or instances for which you
  want to check for port conflicts. If you specify multiple instances, separate them with a colon (:).

The port validator tool uses the -products and -instances parameters to determine the default set of ports that it checks for conflicts. The tool generates a default set of ports that includes the ports used by the specified products and instances.

If you do not specify this parameter, the port validator tool checks for port conflicts in the instances specified by the servicetools -instances parameter, instances within the products specified by the -products parameter.

If you specify this parameter and the -instances parameter for servicetools, the port validator tool checks only the instances that you specify for the -portconflict -instances parameter.

If you do not specify the -instances parameter for either the service tools script or the port validator tool, the tool checks for port conflicts in all instances in the specified products.

- **-ports**
  This is an optional parameter. The value *port_list* is a list of ports that you want to check for conflicts. If you specify this parameter, the tool generates a set of ports based on the value *port_list*. The tool checks for conflicts between this set of ports and the default set that is generated based on the -products and -instances parameters.

  You can specify individual ports, or port blocks. Separate ports or port blocks with a colon (:). For example, to check for conflicts on ports 6680, 7600-7610, and 13320, include specify this parameter:

  ```
  -ports 6680:7600-7610:13320
  ```

  If you do not specify this parameter, the port validator tool uses the values specified by the -products, -instances, -comparetoproduct, and -comparetoinstance parameters to check for conflicts.

  Note: You cannot specify the -ports parameter with the -comparetoproduct parameter or with the -comparetoinstance parameter.

- **-comparetoproduct**
  This is an optional parameter. The port validator tool compares port usage between the product specified by *compare_prod* and the products and instances specified by -products and -instances. If you specify this parameter, the port validator tool generates a set of ports based on the specified product. The tool checks for conflicts between this set of ports and the default set that is generated based on the -products and -instances parameters.

  Note: You cannot specify the -comparetoproduct parameter with the -ports parameter.

- **-comparetoinstance**
  This is an optional parameter. The port validator tool compares port usage between the instance specified by *compare_inst* and the products and instances specified by -products and -instances. If you specify this parameter, the port validator tool generates a set of ports based on the specified instance. The tool checks for conflicts between this set of ports and the default set that is generated based on the -products and -instances parameters.

  Note: You cannot specify the -comparetoinstance parameter with the -ports parameter.

- **-xmloutput**
  This is an optional parameter. The value *output_file* is the fully qualified path of an XML file. If you specify this parameter, the port validator tool writes the output in XML format to the specified file, in addition to writing the output to the display.

The servicetools script

The servicetools script includes serviceability tools that you can use to help troubleshoot WebSphere Application Server - Express.

Authority

To run this script, your user profile must have *ALLOBJ authority.

Syntax

The syntax of the script is:

```
servicetools [ -products product_list [ -instances instance_list ] ]
  [ -input input_file ] [ -output output_file ] [ -trace trace_file ]
  [ -tools tool_options ] [ -help]
```

Parameters

The parameters of the script are:

- **-products**
  This is an optional parameter. The value *product_list* specifies the list of products for which you want to run the tools. If you specify multiple products, separate them with a colon (:). Valid products are WAS40Adv, WAS40AEs, WAS50Base, WAS50Express, WAS50ND, WAS50PME, WAS50PMEND, WAS51Base, WAS51Express, WAS51ND, WAS51PME, WAS51PMEND, and ALL. If you do not specify this parameter, you must specify the -products parameter for each tool as appropriate.

  Note: Not all tools require that you specify a product. This parameter applies only to those tools that require that you specify a product.

- **-instances**
  This is an optional parameter. The value *instance_list* specifies the instance or instances for which you want to run the tools. If you specify multiple instances, separate them with a colon (:). If you do not specify this parameter, the tools run for all of the instances in the products specified by the -products parameter.

- **-input**
  This is an optional parameter. The value *input_file* specifies an alternate XML Configuration file for the service tools. By default, the servicetools script uses the validator.xml file that is included in the ServiceTools.jar file.

- **-output**
  This is an optional parameter. The value *output_file* specifies a file to which the script writes standard output. If this value is not specified, the script writes the standard output to the display.

- **-trace**
  This is an optional parameter. The value *trace_file* specifies a file to which the script writes trace information. If you do not specify this parameter, the script does not generate trace information.

- ***-tools***
  This parameter specifies the tool or tools that you want to run. The value *tools* specifies the tool that you want to run. The value *tool_options* specifies the parameters for the tool. These are the valid values for *-tools*:

  - **-checkprereqs**
    If you specify this parameter, the script starts the prerequisite validator tool. For information on this tool and its parameters, see "The prerequisite validator tool" on page 13.

  - **-portconflict**
    If you specify this parameter, the script starts the port validator tool. This parameter requires the -product parameter. For information on this tool and its parameters, see "The port validator tool" on page 14.

  You must specify all other parameters before you specify the *tools* parameter. You must specify the options for each tool after its parameter. You can specify more than one tool when you run the servicetools script. To run multiple tools, use this format:

  ```
  servicetools parameters -tool1 tool1_options -tool2 tool2_options
  ```

- **-help**
  This optional parameter displays the usage statement for the script.

WebSphere Application Server log files

The application server has a variety of logs to which messages are written. For example, system messages, which can be written by any application server component or application are written to general-purpose logs such as the Java virtual machine (JVM) logs and the IBM Service logs. Other logs are very specific in nature and are scoped to a particular component or activity. For example, the HTTP Server plug-in maintains a component-specific plug-in log.

The general purpose logs, such as JVM and IBM Service, are used to monitor the health of the application server and assist in troubleshooting. Troubleshooting for specific components may require additional log analysis, such as component or product specific log files.

The topics below describe the log files available for the application server and how you can configure and view the log files.

"Java virtual machine log files"
This topic describes the JVM log files. All application server message events go to these files as well as System.out and System.err print statements for your applications. This topic also provides information on how to configure, view, and interpret the JVM log files.

"Process log files" on page 20
This topic describes the process log files. Any output for the native code supporting the JVM or native code for an application is written to these files. This topic also provides information on how to configure and view the process log files.

"IBM Service log files" on page 22
This topic describes the IBM Service, or activity, log files. The IBM Service log file is a binary file to which the application server writes the message events for any servers running under and instance (node). This topic provides information on how to configure and view the IBM Service log files.

Set up logs on HTTP Server (powered by Apache)
This topic describes how to configure different log files for the HTTP Server (powered by Apache).

Java virtual machine log files

The Java virtual machine (JVM) log files contain the output for the System.out and System.err output streams for the application server process. One log file is created for the System.out output stream and another for the System.err output stream. The JVM logs contain print data written by applications. The data may be written directly by the application in the form of System.out.println(), System.err.print(), or other method calls. Data may also be written indirectly by the application calling a JVM function, such as Exception.printStackTrace(). In addition, the System.out JVM log contains system messages, also known as message events, written by the application server.

The JVM log files are self-managing. The maximum file size and the number of log files to archive and when to archive log files are user defined.

The log files must have the following granted authorities:
- *Public *Exclude
- QEJBSVR *RW

If the application server is running under a user profile other than the default (QEJBSVR) and the user profile does not have QEJBSVR specified as a group profile, the user profile must have explicit *RW authority to the JVM log file.

Application print data may be formatted with the JVM to display as a WebSphere system message or as plain text. WebSphere Application Server system messages are always preformatted.

See these topics for information on how to use the JVM log files:

- "Configure the Java virtual machine log files"
  Learn how to configure the JVM log files for your application server through the administrative console .
- **"View the Java virtual machine log files"**
  Learn how to view the JVM log files through the administrative console and other methods.
- "**Interpret the contents for the Java virtual machine log files**" on page 19
  Learn about the formatting of the information contained in the log files.

Configure the Java virtual machine log files

Use the WebSphere administrative console to configure the Java virtual machine (JVM) logs for your application server.

Note: Changes to a JVM log file while an application is running are not applied until the application server is restarted.

To configure a JVM log file, follow these steps:

1. Start the administrative console.
2. Expand **Troubleshooting**.
3. Click **Logs and Trace**.
4. Click your application server from the table.
5. Click **JVM Logs** from the **Logging and Tracing** table.
6. Click the **Configuration** tab. The panel displays the attribute values currently assigned to your JVM log file. Make any configuration changes and continue.
7. Click **OK**.
8. Click **Save** at the top of the page to save the log changes.
9. Click **Save** your changes for your application server configuration.

View the Java virtual machine log files

The Java virtual machine (JVM) logs are written as plain ASCII text files. By default, the logs are contains in the subdirectory logs/*appservername* of the /QIBM/UserData/WebASE51/ASE/*appservername* directory, where *appservername* is the name of your application server. You can view the JVM log files using any of the methods listed below. For information on interpreting the contents of the logs, see "**Interpret the contents for the Java virtual machine log files**" on page 19.

View the JVM logs with the HTTP Server Administration interface

1. Start HTTP Server Administration interface.
2. Click the **Manage** tab.
3. Select your application server from the **Server** list.
4. Expand **Problem Determination**.
5. Click **View Logs**.
6. Select the log file you want to view from the table.
7. Click **View**.

View the JVM logs from the WebSphere administrative console

1. Start the administrative console.
2. Expand **Troubleshooting**.
3. Click **Logs and Trace**.

4. Click your application server from the table.
5. Click **JVM Logs** from the **Logging and Tracing** table.
6. Click the **Runtime** tab.
7. Select a file name from the **General Properties** table.
8. Click **View**.

View the JVM logs with iSeries CL commands
1. Start a 5250 emulation session on your iSeries.
2. Type one of the following:
   - For SystemOut log file type EDTF
     FILE('/QIBM/UserData/WebASE51/ASE/*appservername*/logs/*appservername*/SystemOut.log')
   - For SystemErr log file type EDTF
     FILE('/QIBM/UserData/WebASE51/ASE/*appservername*/logs/*appservername*/SystemErr.log')

View the JVM logs from a non-iSeries workstation Perform these steps to view the JVM logs from a
mapped or mounted driver:
1. Map (Windows 32-bit workstation) or mount (Unix workstation) a drive to the iSeries.
2. Open the file in a text editor or drag and drop the file into a file editing and viewing application.

Interpret the contents for the Java virtual machine log files
The Java virtual machine (JVM) writes data to the logs using two different formats, Basic and Advanced
format. The Basic and Advanced formats use the same fields and formatting techniques with the
exception that Advanced format adds information about events when possible.

Basic and Advanced format fields

TimeStamp
> The timestamp is formatted using the locale of the process where it is formatted. It includes a
> fully qualified date (for example, YYMMDD), 24 hour time with millisecond precision and a time
> zone.

ThreadID
> An 8 character hexadecimal value generated from the hash code of the thread that issued the
> message.

ShortName
> The abbreviated name of the logging component that issued the message or trace event. This is
> typically the class name for application server internal components, but may be some other
> identifier for user applications.

LongName
> The full name of the logging component that issued the message or trace event. This is typically
> the fully qualified class name for application server internal components, but may be some other
> identifier for user applications.

EventType
> A one character field that indicates the type of the message or trace event. Message types are in
> upper case. Possible values include:
> - **A** - An audit message.
> - **I** - An informational message.
> - **W** - A Warning message.
> - **E** - An Error message.
> - **F** - A Fatal message.
> - **O** - A message that was written directly to System.out by the user application or application
>   server internal components.

- **R** - A message that was written directly to System.err by the user application or application server internal components.
- **u** - A specialized message type used by the message logging component of the application server runtime.
- **Z** - A placeholder to indicate the type was not recognized.

ClassName
> The class that issued the message or trace event.

MethodName
> The method that issued the message or trace event.

Organization
> The organization that own the application that issued the message or trace event.

Product
> The product that issued the message or trace event.

Component
> The component within the product that issued the message or trace event.

UOW   The unit of work identifier for the event. This field is not currently used.

Basic format

Message events displayed in basic format use this format. The notation <name> indicated mandatory fields that are always displayed in the Basic format message. The notation [name] indicates optional or conditional fields that are included if they can be determined.

TimeStamp ThreadID ShortName EventType [ClassName] [MethodName] *messages*

Advanced format

Message events displayed in advanced format use this format. The notation <name> is used to indicate mandatory fields that are always displayed in the advanced format for message entries. The notation [name] is used to indicate optional or conditional fields that are included if they can be determined.

TimeStamp ThreadID EventType UOWsource=LongName [ClassName] [methodName *OrganizationProductComponentmessage*

Process log files

Application server processes contain two output streams that are accessible to native code running in the process. These streams are the **stdout** and **stderr** streams. Platform-specific code, including the Java virtual machine (JVM), may write data to these process streams.

By default, the **stdout** and **stderr** streams are redirected to log files at application server startup, which contain text written to the **stdout** and **stderr** streams by native modules (for example, *SRVPGM object). The application server does not have a special process or format of the output that is written to the process log.

The log files must have the following granted authorities:
- *Public *Exclude
- QEJBSVR *RW

If the application server is running under a user profile other than the default (QEJBSVR) and the user profile does not have QEJBSVR specified as a group profile, the user profile must have explicit *RW authority to the process log file.

See these topics for information on how to use the process log files:

* **"Configure the process log files"**
  Learn how to configure the process log files for your application server through the WebSphere administrative console.
* **"View the process log files"**
  Learn how to view the JVM log files through the WebSphere administrative console and other methods.

Configure the process log files

Use the WebSphere administrative console to configure the process logs for your application server. The file name for a process log is the only attribute that can be changed. Unlike the "Java virtual machine log files" on page 17, the process logs are not self-managing, meaning the file size and number of archives are not user defined. Generally, these files are empty or contain a very small amount of information and do not require self-management.

Note: Changes to a process log file while an application is running are not applied until the application server is restarted.

To configure a process log file, follow these steps:
1. Start the administrative console.
2. Expand **Troubleshooting**.
3. Click **Logs and Trace**.
4. Click your application server from the table.
5. Select **Process Logs** from the **Logging and Tracing** table.
6. Click the **Configuration** tab. The panel displays the attribute values currently assigned to your process log file. Make any configuration changes and continue.
7. Click **OK**.
8. Click **Save** at the top of the page to save the log changes.
9. Click **Save** your changes for your application server configuration.

View the process log files

The process logs are written as plain ASCII text files. By default, the logs are contains in the subdirectory logs/*appservername* of the /QIBM/UserData/WebASE51/ASE/*appservername* directory, where *appservername* is the name of your application server. You can view the process log files using any of the methods listed below.

View the process logs with the HTTP Server Administration interface
1. Start HTTP Server Administration interface.
2. Click the **Manage** tab.
3. Select your application server from the **Server** list.
4. Expand **Problem Determination**.
5. Click **View Logs**.
6. Select the log file you want to view from the table.
7. Click **View**.

View the process logs from the WebSphere administrative console
1. Start the administrative console.
2. Expand **Troubleshooting**.
3. Click **Logs and Trace**.
4. Click your application server from the table.

5. Click **Process Logs** from the **Logging and Tracing** table.
6. Click the **Runtime** tab.
7. Select a file name from the **General Properties** table.
8. Click **View**.

View the process logs with iSeries CL commands
1. Start a 5250 emulation session on your iSeries.
2. Type one of the following:
   - For stdout log file type EDTF
     FILE('/QIBM/UserData/WebASE51/ASE/*appservername*/logs/*appservername*/native_stdout.log')
   - For stderr log file type EDTF
     FILE('/QIBM/UserData/WebASE51/ASE/*appservername*/logs/*appservername*/native_stderr.log')

View the process logs from a non-iSeries workstation Perform these steps to view the JVM logs from a mapped or mounted driver:
1. Map (Windows 32-bit workstation) or mount (Unix workstation) a drive to the iSeries.
2. Open the file in a text editor or drag and drop the file into a file editing and viewing application.

IBM Service log files

The IBM Service, or activity, log file is a binary file to which the application server writes the message events for any servers running under an instance (node). The application server runtime creates the file, named activity.log by default, in the logs subdirectory of your application server instance. By default, the logs are contained in the subdirectory logs/*appservername* of the /QIBM/UserData/WebASE51/ASE/*appservername* directory, where *appservername* is the name of your application server.

The log files must have the following granted authorities:
- *Public *Exclude
- QEJBSVR *RW

If the application server is running under a user profile other than the default (QEJBSVR) and the user profile does not have QEJBSVR specified as a group profile, the user profile must have explicit *RW authority to the activity.log file.

See these topics for information on how to use the JVM log files:
- **"Configure the IBM Service log files"**
  Learn how to configure the IBM Service log files for your application server through the WebSphere administrative console.
- **"Collect data for IBM Service" on page 23**
  The WebSphere Application Server - Express includes a Qshell script that collects information about your WebSphere Application Server - Express installation and packages it in a JAR file that you can send to IBM Customer Support to assist in problem determination and analysis.
- "The showlog script" on page 24
  The WebSphere Application Server - Express includes a Qshell script that displays the contents of the IBM Service log.

Configure the IBM Service log files
Use the HTTP Server Administrative interface to access the WebSphere administrative console to configure the IBM Service logs for your application server.

Note: Changes to a IBM Service log file while an application server is running are not applied until the application server is restarted.

To configure a IBM Service log file, follow these steps:

1. Start the administrative console.
2. Expand **Troubleshooting**.
3. Click **Logs and Trace**.
4. Click your application server from the table.
5. Select **IBM Service Logs** from the **Logging and Tracing** table.
6. Click the **Configuration** tab. The panel displays the attribute values currently assigned to your IBM Service log file. Make any configuration changes and continue.
7. Click **OK**.
8. Click **Save** at the top of the page to save the log changes.
9. Click **Save** your changes for your application server configuration.

Collect data for IBM Service

WebSphere Application Server - Express includes a Qshell script that collects information about your WebSphere Application Server - Express installation and packages it in a JAR file that you can send to IBM Customer Support to assist in problem determination and analysis. The information includes logs, property files, configuration files, operating system and Java data, and prerequisite software presence and levels. For information on running Qshell scripts, see Configure Qshell to run WebSphere Application Server - Express scripts.

To collect as much data as possible, the collector script is designed to run to completion regardless of errors such as files or commands not found.

Note: You should only run the collector tool when requested to by IBM Service personnel.

Authority

To run this script, your user profile must have *ALLOBJ authority.

Usage

To run the collector script, follow these steps:

1. On the CL command line, run the STRQSH (Start Qshell) command.
2. Run the cd command to change to the directory that contains the script:
   ```
   cd /home/myprofile
   ```
3. Run the collector script:
   ```
   /QIBM/ProdData/WebASE51/ASE/bin/collector -instance myInstance -JarOutName jarfile
   ```
   where *myInstance* is the name of the instance for which you want to collect information.

Syntax

The syntax of the collector script is:
```
[fully_qualified_directory]/collector -instance instance
  [ -JarOutName jarfile ] [ -Summary ] [ -IncludeInstalledApps ]
```

where *fully_qualified_directory* is the directory that contains the script.

Parameters

The parameters of the collector script are:

- **-instance**
  This is a required parameter. The value *instance* specifies the name of the WebSphere Application Server - Express instance for which you want to collect information.

- **-JarOutName**
  This is an optional parameter. The value *jarfile* specifies the fully qualified path name of the JAR file to which the collector tool writes the output. The location of the JAR file must be outside of the /QIBM/ProdData/WebASE51/ASE directory structure for the product. By default, the script creates a JAR file in the current working directory. The default file name is WASenv.jar.

  It is recommended that you specify the -JarOutName parameter.

  Notes:
  – If you specify a jar file name that exceeds sixty characters, it is truncated to the first sixty characters of the name.
  – The QEJBSVR user profile must have *RWX authority to the directory to contain the output jar file.

- **-Summary**
  This optional parameter is supported in versions 5.0.2 and later. If you specify this option, the script generates a summary report of the WebSphere Application Server - Express version and fix level, in addition to information about the i5/OS release level and Java Development Kit level. The summary report is written to the display as well as to file Collector_Summary.txt in the current working directory. When you report a problem to IBM Support, include the Collector_Summary.txt file or output. When you run the collector script with the -Summary parameter, the script does not generate a JAR file.

- **-IncludeInstalledApps**
  This optional parameter is supported in versions 5.0.2 and later. By default, the installedApps directory for your instance is not included in the JAR file that the collector tool generates. Specify the -IncludeInstalledApps parameter if you want to include the installedApps directory the generated JAR file.

Example

```
collector -instance testinst -JarOutName /home/myprofile/testinst_svc.jar
```

This example collects information on the instance testinst and writes the output to testinst_svc.jar in the /home/myprofile directory.

The showlog script
The showlog script displays the contents the "IBM Service log files" on page 22. For information on running Qshell scripts, see Configure Qshell to run WebSphere Application Server - Express scripts.

Authority

To run this script, your user profile must have *ALLOBJ authority.

Usage

To display the IBM Service log, follow these steps:
1. On the CL command line, run the STRQSH (Start Qshell) command.
2. Run the cd command to change to the directory that contains the script:
   ```
   cd /QIBM/ProdData/WebASE51/ASE/bin
   ```
3. Run the showlog script:
   ```
   showlog -instance instance binaryfile
   ```
   where *instance* is the name of your application server instance and *binaryfile* is the name of the IBM Service log file that you want to view.

Syntax

The syntax of the showlog script is:

```
showlog -instance instance binaryfile [ outputfile ]
```

Parameters

The parameters of the showlog script are:

- *-instance*
  This is a required parameter. The value *instance* specifies the instance for which you want to display the IBM service log.
- *binaryfile*
  This is a required paramter. The value *binaryfile* specifies the of the IBM Service log file. If you do not specify a fully qualified file name, the script searches for the specified file in the /QIBM/UserData/WebASE51/ASE/*instance*/logs directory, where *instance* is the name of your instance.
  Note: The QEJBSVR user profile must have *R authority to the specified file.
- *outputfile*
  This is an optional parameter. The value *outputfile* specifies the name of the file to which the script writes the formatted contents of the IBM service log. If you do not specify a fully qualified file name, the script creates the output file in the current working directory. By default, showlog writes results to standard out.
  Note: The QEJBSVR user profile must have *RW authority to the directory to contain the output file.

Example

```
showlog -instance myinst /QIBM/UserData/WebASE51/ASE/myAppSvr/logs/activity.log
 /home/myprofile/formatted_activity.log
```

This example writes the formatted contents of the activity.log file for the myAppSvr instance to the file formatted_activity.log in the /home/myprofile directory.

Use the WebSphere Application Server - Express trace service

Tracing is useful for determining errors from the application server log files. Trace allows you to obtain detailed information about the execution of application server components, including clients, and other processes in the environments. Trace files show the time and sequence of methods called by the application server runtime classes, and can be used to pinpoint the failure.

Refer to these topic for more information on enabling and using the application server trace service.

"Enable the trace service"
This topic describes how you can use the administrative console to enable and disable the trace service for a server.

"Interpret the trace service output" on page 26
This topic describes the format of the information that is written to the trace file.

"Use wsadmin to configure a trace for your application server" on page 28
This topic describes how to use the wsadmin to configure a trace for your application server

"Use wsadmin to turn traces on and off for your application server" on page 28
This topic describes how to use wsadmin to turn traces on and off in your application server

Enable the trace service

Use the WebSphere administrative console to enable the trace service for an application server. The trace service can be enabled dynamically or statically for an application server.

When you enable trace dynamically, the trace settings are in effect only for the lifetime of the server. The trace settings are not saved to the application server configuration. Use dynamic trace when the problem you are diagnosing occurs after the application server has started successfully.

When you enable trace statically, the trace settings are not enabled until the application server is started (or restarted). The trace settings are used every time you start the application server. Use static trace when the problem you are diagnosing occurs during application server startup.

Note: Depending on the amount of trace event data being collected, the trace service can negatively affect performance of the application server. Once you have gathered the appropriate trace, be sure to disable the trace service.

Enable the trace service

To enable a trace service, follow these steps:
1. Start the administrative console.
2. Expand **Troubleshooting**.
3. Click **Logs and Trace**.
4. Click your application server from the table.
5. Click **Diagnostic trace**.
6. Select a tracing option:
   - To enable trace statically, select the **Configuration** tab.
   - To enable trace dynamically, select the **Runtime** tab.

   The panel displays the attribute values currently assigned. Make any configuration changes and continue.
7. Click **Apply**.
8. If enabling static trace, click the **Save** link at the top of the page to save your configuration changes. Click **Save** on the resulting page.

Disable the trace service

To disable a trace service, follow these steps:
1. Start the administrative console.
2. Expand **Troubleshooting**.
3. Click **Logs and Trace**.
4. Click your application server from the table.
5. Click **Diagnostic trace**.
6. Select a tracing option:
   - To disable a static trace, select the **Configuration** tab. Uncheck the **Enable Trace** checkbox.
   - To disable a dynamic trace, select the **Runtime** tab. Change the **Trace Specification** to *=all=disabled.

   The panel displays the attribute values currently assigned. Make any configuration changes and continue.
7. Click **Apply**.
8. If enabling static trace, click the **Save** link at the top of the page to save your configuration changes. Click **Save** on the resulting page.

Interpret the trace service output

On an application server, trace output can be directed either to a file or to an in memory circular buffer. If trace output is directed to the in memory circular buffer, it must be dumped to a file before it can be viewed.

On a standalone process, trace output can be directed either to a file or to the process console window.

In all cases, trace output is generated as plain text in either basic or advanced format as specified by the user when enabling the trace service.

Trace output formats

Formatted trace events may be written to the trace file in one of three formats:

Basic format
    The format used in WebSphere Application Server - Express.

Advanced format
    Extends the basic format by adding information about an event.

Basic and advanced format fields

Basic and advanced formats use many of the same fields and formatting techniques. The various fields that may be found in these formats include:

TimeStamp
    The timestamp is formatted using the locale of the process where it is formatted. It includes a fully qualified date (YYMMDD), 24 hour time with millisecond precision and a Time zone.

ThreadID
    An 8 character hexadecimal value generated from the hash code of the thread that issued the trace event.

ShortName
    The abbreviation name of the logging component that issued the trace event. This is typically the class name for application server internal components, but may be some other identifier for user applications.

LongName
    The full name of the logging component that issued the trace event. This is typically the fully qualified class name for application server internal components, but may be some other identifier for user applications.

EventType
    A one character field that indicates the type of the trace event. Trace types are in lower case. Possible values include:
    - **>** - A trace entry of type method entry.
    - **<** - A trace entry of type method exit.
    - **e** - A trace entry of type event.
    - **d** - A trace entry of type debug.
    - **m** - A trace entry of type dump.
    - **u** - A trace entry of type unconditional.
    - **Z** - A placeholder to indicate that the trace type was not recognized.

ClassName
    The class that issued the message or trace event.

MethodName
    The method that issued the message or trace event.

Organization
    The organization that owns the application that issued the message or trace event.

Product
    The product that issued the message or trace event.

Component
> The component within the product that issued the message or trace event.

UOW The unit of work identifier for the event. This field is not currently used.

Basic format

Trace events displayed in basic format use this format, where *name* indicates mandatory fields that are always displayed in the formatted message and [name] indicates optional fields that are displayed if they can be determined.

```
TimeStamp ThreadId ShortName EventType [ClassName] [MethodName]
textmessage[parameter 1] [parameter 2]
```

Advanced format

Trace events displayed in advanced format use this format, where *name* indicates mandatory fields that are always displayed in the formatted message and [name] indicates optional fields that are displayed if they can be determined.

```
TimeStamp ThreadId EventType UOWsource=LongName [ClassName] [MethodName]
Organization Product Component textMessage [parameter 1=paremterValue]
[parameter 2=paremterValue]
```

Use wsadmin to configure a trace for your application server

To use wsadmin to configure a trace for your application server, follow these steps:

1. On the CL command line, run the STRQSH (Start Qshell) command.
2. Run the cd command to change to the directory that contains the wsadmin tool:

   `cd /QIBM/ProdData/WebASE51/ASE/bin`
3. Start wsadmin.
4. At the wsadmin prompt, run this command to identify the server and assign it to the server variable:

   `set server [$AdminConfig getid /Cell:`*myCell*`/Node:`*myNode*`/Server:`*myAppSvr*`/]`

   where *myCell* is the name of the cell that contains your application server, *myNode* is the name of the node that contains your application server, and *myAppSvr* is the name of your application server.
5. Run this command to identify the trace service for the application server and assign it to the ts variable:

   `set ts [$AdminConfig list TraceService $server]`
6. Set the trace string:

   This example sets the trace string for a single component:

   ```
   $AdminConfig modify $ts {{startupTraceSpecification
    com.ibm.websphere.management.*=all=enabled}}
   ```

   The example sets the trace string for multiple components:

   ```
   $AdminConfig modify $ts {{startupTraceSpecification
    com.ibm.websphere.management.*=all=enabled:
    com.ibm.ws.management.*=all=enabled:com.ibm.ws.runtime.*=all=enabled}}
   ```

   Note: These commands have been wrapped for display purposes.
7. Run this command to save your changes:

   `$AdminConfig save`

Use wsadmin to turn traces on and off for your application server

To turn traces on and off in your application server, follow these steps:

1. On the CL command line, run the STRQSH (Start Qshell) command.
2. Run the cd command to change to the directory that contains the wsadmin tool:

   `cd /QIBM/ProdData/WebASE51/ASE/bin`

3. Start wsadmin.
4. At the wsadmin prompt, run this command to identify the object name for the TraceService MBean running in the process:

   `$AdminControl completeObjectName type=Server,name=server,*`
5. Run this command to obtain the name of the object and set it to a variable:

   `set ts [$AdminControl completeObjectName type=TraceService,process=myAppSvr,*]`

   where *myAppSvr* is the name of the application server for which you want to enable or disable traces.
6. Turn traces on or off for the application server.
   - Run this command to turn on traces for the server:

     `$AdminControl setAttribute $ts traceSpecification com.ibm.*=all=enabled`
   - Run this command to turn off traces for the server:

     `$AdminControl setAttribute $ts traceSpecification com.ibm.*=all=disabled`
7. Run this command to save your changes:

   `$AdminConfig save`

Appendix. 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 grant 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.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.
```

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 illustrate 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This WebSphere Application Server - Express publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

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

```
AIX
AIX 5L
e(logo)server
eServer
i5/OS
IBM
IBM (logo)
iSeries
pSeries
WebSphere
xSeries
zSeries
```

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

Linux is a trademark of Linus Torvalds 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, and service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the p <?Pub Caret?>ublications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM ®

Printed in USA