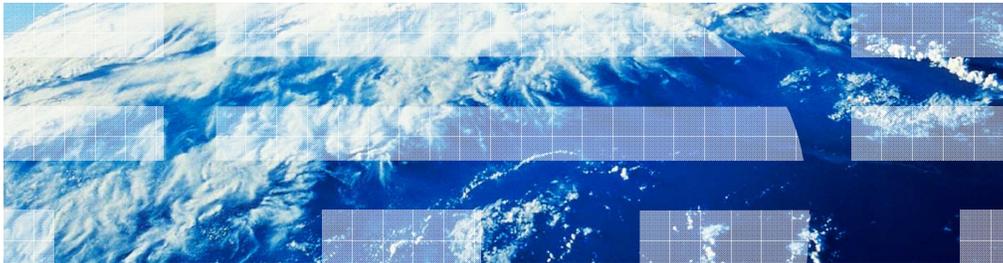

WebSphere Process Server for z/OS V7.0 WebSphere Enterprise Service Bus for z/OS V7.0

z/OS configuration troubleshooting



This presentation covers troubleshooting the configuration of WebSphere® Process Server for z/OS V7.0 and WebSphere Enterprise Service Bus for z/OS V7.0.

Goals

- Troubleshoot errors encountered when configuring WebSphere Process Server for z/OS V7.0 and WebSphere Enterprise Service Bus for z/OS V7.0

The goal of this presentation is to show you how to determine the cause of errors when trying to configure WebSphere Process Server for z/OS or WebSphere Enterprise Service Bus for z/OS.

Troubleshooting

ASCII files

- **zWPSInstall.sh -install** logs

- Found in `<app_server_root>/logs/wbi/install` directory

- `installconfig.log`

Configuration manager task log

- Found in `<app_server_root>/logs/wbi` directory

- `100SCleanOSGICache.ant.log`
 - `80SCopyInstallValidatorLog.ant.log`
 - `85SConfigNoProfileFirstStepsWBI.ant.log`
 - `90SCleanDeployTool.ant.log`
 - `90SConfigureWSProfileForAugmentation.ant.log`
 - `90SConfigureWSProfileForWBI.ant.log`
 - `91SConfigNoProfileFirstStepsCharset.ant.log`
 - `93SDeployBPCAdminConsolePlugins.ant.log`
 - `93SDeployServerAdminConsolePlugins.ant.log`
 - `93SDeployWBICommonAdminConsolePlugins.ant.log`
 - `94SDeployCoreAdminConsolePlugins.ant.log`
 - `zWPSInstall.log`
 - `zWPSInstall.trace`

Logs from
.ant tasks
called by the
configuration
manager task

When you run the `zWPSInstall` script, many log files are written to your configured logs directory as shown on the slide. The `zWPSInstall.log` and `zWPSInstall.trace` files are pretty basic, but if you turn traces on, the trace file might prove interesting. The `zWPSInstall` script calls the configuration manager, which is responsible for calling many ant tasks to configure the node with WebSphere Process Server or WebSphere Enterprise Service Bus functions. The `installconfig.log` file, found in the `wbi/install` directory, is a record of the configuration manager's progress overall. Each of the ant tasks that are called writes its own detailed log of its progress as well. These logs are shown here.

Return codes

- **installconfig.log** (ASCII file)
 - Go to end of file and should see one of these messages:
 - **Return Code=0**
 - `<message>Returning with return code: INSTCONFSUCCESS</message>`
 - **non-zero Return Code**
 - `<message>Returning with return code: INSTCONFFAILED</message>`
 - `<message>Returning with return code: INSTCONFPARTIALSUCCESS</message>`

At the end of the `installconfig.log` is a return code indicating whether the `zWPSInstall` script ran successfully. There are three possible outcomes: `SUCCESS`, `PARTIALSUCCESS` and `FAILED`. In the case of `SUCCESS`, you are done and will continue with the `zWPSConfig` script. In the case of `PARTIALSUCCESS` or `FAILED`, you need to determine the cause of the error. The next slide will look at the `FAILED` return code.

Return codes -- FAILED

- **installconfig.log** (ASCII file)
 - Look here if error occurred during execution of the configuration manager task
 - Find 'SEVERE' to determine error
 - If ant task was running, find 'Buildfile' **previous** to discover what ant task was running when the failure occurred
 - Look in corresponding .ant log file for more information

See example with zWPSConfig.sh

If the zWPSInstall script ended with a return code of FAILED during configuration manager processing, you can look in the installconfig.log for a record indicating 'SEVERE'. If an ant script was running at the time, from that point, you can look for the last 'Buildfile' occurrence before the SEVERE record. That will tell you what ant task was running when the failure occurred. It is possible that the SEVERE record will give you enough information to see what went wrong. If not, you can look in the corresponding ant log file to get more detailed information about the error that occurred. There is an example of this process on the slide that covers the zWPSConfig script logs. The same basic technique is used there. A PARTIALSUCCESS example will also be seen when looking at the zWPSConfig logs.

Keep in mind that all these log files are encoded in ASCII so you will either need to look at them from a Windows® machine or use an editor like viascii.

Troubleshooting

- **zWPSConfig.sh** or **zWESBConfig.sh -augment** logs
 - `<app_server_root>/logs/wbi` directory
 - zWPSConfig.log
 - zWPSConfig.trace
 - `<app_server_root>/logs/manageprofiles` directory
 - default_augment.log Main log
 - `<app_server_root>/logs/manageprofiles/default` directory
 - commonDBUtility.ant.log
 - configAdminConsoleEnv.ant.log
 - configAppScheduler.ant.log
 - configAppScheduler.ant.wsadmin.log
 - configAppSchedulerDBTables.ant.log
 - configAppSecurity.ant.log
 - configAppSecurity.ant.wsadmin.log
 - configBSpace.ant.log
 - . . .

Partial listing
of ant task
logs from
augment

Here you see where the logs are located when running the zWPSConfig and zWESBConfig scripts. They are found in a few different directories. To start, you will find the log and trace files in the logs/wbi directory. Like the equivalent zWPSInstall logs, the zWPSConfig and zWESBConfig trace and log files are pretty basic unless you have turned on tracing. The main log file is found in the logs/manageprofiles directory and is called default_augment.log. It has a very detailed record of all the ant tasks being called and the results from calling them. The next slides will take you through the process of finding a problem using this log file. Finally, there are detailed logs in the logs/manageprofiles/default directory for each of the ant tasks that are called. You will notice that if the ant task calls wsadmin for any reason, there is also a log for that processing as seen by the circled files on the slide.

Return codes

- **default_augment.log** (ASCII file)
 - Go to end of file and should see these messages:
 - **Return Code=0**
 - `<message>Returning with return code: INSTCONFSUCCESS</message>`
 - **non-zero Return Code**
 - `<message>Returning with return code: INSTCONFFAILED</message>`
 - `<message>Returning with return code: INSTCONFPARTIALSUCCESS</message>`

Same return codes as zWPSInstall

Again, if you go to the end of the `default_augment.log`, you will find a return code indicating whether the `zWESBConfig` or the `zWPSConfig` script ran successfully. Just like the `zWPSInstall` script, there are three possible outcomes: `SUCCESS`, `PARTIALSUCCESS` and `FAILED`. In the case of `SUCCESS`, you are done and will continue with your configuration. In the case of `PARTIALSUCCESS`, you might be instructed to re-run a command. This will depend on if the configuration action is deemed important in your particular configuration. For instance, you may see that it failed trying to deploy some sample applications. You may not care about those sample applications so you may decide to continue with your configuration. If you receive a `FAILED` return code, you need to determine the cause of the error and fix it before continuing the configuration. The next slides will take you through the process of determining what the error was.

Return codes - FAILED (1 of 5)

- **Standard output** directs you to

<app_server_root>/logs/manageprofiles/default_augment.log on failure:

```
- INSTCONFFAILED: Profile augmentation failed. For more information, consult
/etc/wasv7config/s7basea/s7nodea/AppServer/logs/manageprofiles/default_augm
ent.log
```

- **default_augment.log – find 'SEVERE'**

```
<record>
  <date>Mar 22, 2010 11:37:04 PM</date>
  <millis>1269315424323</millis>
  <sequence>7688</sequence>
  <logger>com.ibm.ws.install.configmanager.ConfigManager</logger>
  <level>SEVERE</level>
  <class>com.ibm.ws.install.configmanager.ConfigManager</class>
  <method>launch</method>
  <thread>0</thread>
  <message>ConfigManager action execution failed on a fatal action
</message>
</record>
```

Here is an example of looking for the problem. The standard output from running the zWPSConfig shell script directs you to the default_augment.log, as shown here. That is the main log for the augmentation process. If you look into that log file, you will see many records that detail the augmentation. In order to find where you hit a problem, you will do a search for 'SEVERE' as you saw earlier with the zWPSInstall error. This slide shows an actual example of what a record might look like when an error was recorded. In some cases, the error might give you enough information to know what went wrong. In this example though, you will need to look further to find the cause of the error. You will need to find the ant script that was running when the error occurred and go look at its error log.

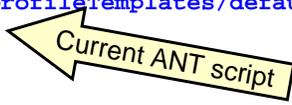
Return codes - FAILED (2 of 5)

▪ **default_augment.log** – find 'Buildfile' PREV

```
<record>
  <date>Mar 22, 2010 11:33:24 PM</date>
  <millis>1269315204378</millis>
  <sequence>6877</sequence>

  <logger>com.ibm.ws.install.configmanager.actionengine.ant.utils.InProcessAntRunner</logger>
  <level>INFO</level>

  <class>com.ibm.ws.install.configmanager.actionengine.ant.utils.InProcessAntRunner</class>
  <method>logOutputMessage</method>
  "
  <thread>0</thread>
  <message>Buildfile:
  /etc/wasv7config/s7basea/s7nodea/AppServer/profileTemplates/default.wbicore/actions/configDatabase.ant</message>
</record>
```

Current ANT script

In order to do that, you want to find the last 'Buildfile' occurrence before the SEVERE record. In this example, you see that the `configDatabase.ant` script was running when the SEVERE error occurred. You can now go look in the `configDatabase.ant` script's log to get further information.

Return codes - FAILED (3 of 5)

- `<app_server_root>/logs/manageprofiles/default/configDatabase.ant.log`

BUILD FAILED

```
/etc/wasv7config/s7basea/s7nodea/AppServer/profileTemplates/default.wbi
core/actions/configDatabase.ant:56: The following error occurred
while executing this line:
/etc/wasv7config/s7basea/s7nodea/AppServer/profileTemplates/default.wbi
core/actions/configDatabase.ant:72: The following error occurred
while executing this line:
/etc/wasv7config/s7basea/s7nodea/AppServer/util/dbUtils/profileHelpers/
commonDBUtility.ant:600: Failed to connect database 's7celldb', Error
Info: /etc/wasv7config/s7basea/s7nodea/AppServer/util/dbUtils/profileH
elpers/commonDBUtility.ant:588: com.ibm.db2.jcc.c.SqlException:
Failure in loading T2 native library db2jct2zos, reason:
java.lang.UnsatisfiedLinkError: db2jct2zos (Not found in
java.library.path)
    at
    org.apache.tools.ant.ProjectHelper.addLocationToBuildException(Projec
tHelper.java:539)
    .
    .
    .
```

Looking at the `configDatabase.ant.log`, you see some more specific information about the error. In this case, it seems the Java™ library path that was provided was incorrect.

Many of the ANT tasks do wsadmin processing to change the configuration. If the error happened to occur during wsadmin processing, there is also a wsadmin log file that records all wsadmin processing. You'll see that on the next slide.

The names of the logs for the ANT tasks are pretty self-explanatory. If it does not seem obvious which log to look into, however, a record in the **default_augment.log** will point you to the log that was in use while the ANT script was running.

Return codes - FAILED (4 of 5)

▪ `<app_server_root>/logs/manageprofiles/default/configDatabase.ant.wsadmin.log`

```
[3/22/10 23:34:20:292 EDT] 00000000 AbstractShell A BBO00222I:
WASX7090I: Executing command: "$AdminTask configCommonDB {
  cdbdbDesign
  {/etc/wasv7config/s7basea/s7nodea/AppServer/profiles/default/properti
es/db/CommonDB.metadata.properties} -outputDir
{/etc/wasv7config/s7basea/s7nodea/AppServer/profiles/default/dbscript
s} }"
[3/22/10 23:34:44:238 EDT] 00000000 CommonDBComma 1
com.ibm.wbiserver.commondb.command.CommonDBCommandProvider
createCommand Entering createCommand
.
.
.
```

Drilling down into the wsadmin log for the configDatabase ant task, you see the commands that have been run during the ant task processing. In this case, it is calling '\$AdminTask configCommonDB'. If an error occurred during the processing of that wsadmin command, you can find more information here.

Return codes - FAILED (5 of 5)

- **Standard output** can also specify that validation errors were found during augmentation

The following validation errors were present with the command line arguments:

```
dbCreateNew: dbCreateNew must be either true or false.
dbCreateNew: dbCreateNew must be either true or false.
ceiDbName: A duplicate database name exists for the s7cellldb Common database an
dbType: CWLDB0558E: The "DB2UDBOS390_V7_0" database type is not supported.
dbType: CWLDB0568E: The database parameters are not valid for validating the database.
dbType: CWLDB0558E: The "DB2UDBOS390_V7_0" database type is not supported.
dbType: CWLDB0568E: The database parameters are not valid for validating the database.
CWPIZ0267I: augmenting profile(s) complete
```

- **default_augment.log – find ‘Validation Error’**

```
<record>
  <date>Mar 23, 2010 11:15:23 PM</date>
  <millis>1269400523416</millis>
  <sequence>4718</sequence>
  <logger>com.ibm.ws.profile.utils.ValidationManagerUtils</logger>
  <level>WARNING</level>
  <class>com.ibm.ws.profile.utils.ValidationManagerUtils</class>
  <method>printOutValidationResults</method>
  <thread>0</thread>
  <message>Validation error for dbCreateNew: {0} must be either true or
  false.</message>
</record>
```

Another type of error that will cause you to receive the FAILED return code is argument validation. The standard output, in this case, will show you the problems that it found with the parameters you specified in your response file. Before running the augmentation again, you need to correct them. If you look in the default_augment.log, you will find records there that specify the problems as well. You can do a find on ‘Validation Error’ to see them all there.

Return codes - PARTIALSUCCESS (1 of 2)

- **default_augment.log – go to end**

```
<message>Returning with return code: INSTCONFPARTIALSUCCESS</message>
```

- **Find 'actions that failed':**

```
<record>
  <date>Mar 24, 2010 2:24:06 PM</date>
  <millis>1269455046932</millis>
  <sequence>20944</sequence>
  <logger>com.ibm.ws.install.configmanager.ConfigManager</logger>
  <level>INFO</level>
  <class>com.ibm.ws.install.configmanager.ConfigManager</class>
  <method>launch</method>
  <thread>0</thread>
  <message>Information about actions that failed, follows</message>
</record>
<record>
  <date>Mar 24, 2010 2:24:06 PM</date>
  <millis>1269455046933</millis>
  <sequence>20945</sequence>
  <logger>com.ibm.ws.install.configmanager.ConfigManager</logger>
  <level>INFO</level>
  <class>com.ibm.ws.install.configmanager.ConfigManager</class>
  <method>dumpNonFatalFailedActionsInfoToLogFile</method>
  <thread>0</thread>
  <message>This action failed to execute:
  /etc/wasv7config/s7basea/s7nodea/AppServer/profileTemplates/default.wbiserver/actions/configSamples.ant</message>
</record>
```

This slide shows an example of the PARTIALSUCCESS return code during augmentation. Remember, the errors documented here are considered recoverable and you can choose to ignore them if they are not essential to your configuration. The example here shows that the configuration of the samples was unsuccessful. To get more information, you can follow the steps shown previously to look at the output from the ant script.

Return codes - PARTIALSUCCESS (2 of 2)

- Gives directions on command that can be manually executed:

```
<record>
  <date>Mar 24, 2010 2:24:06 PM</date>
  <millis>1269455046946</millis>
  <sequence>20951</sequence>
  <logger>com.ibm.ws.install.configmanager.ConfigManager</logger>
  <level>INFO</level>
  <class>com.ibm.ws.install.configmanager.ConfigManager</class>
  <method>dumpNonFatalFailedActionsInfoToLogFile</method>
  <thread>0</thread>
  <message>To manually apply this action, please execute this command: sh -c
  "/etc/wasv7config/s7basea/s7nodea/AppServer/bin/ws_ant.sh -buildfile
  /etc/wasv7config/s7basea/s7nodea/AppServer/profileTemplates/default.wbiserver/
  actions/configSamples.ant"</message>
</record>
```

You will also see in the log directions on how you can manually run the command later. This is shown on the slide.

Troubleshooting

- Collect various log files to debug
 - `<app_server_root>/logs/manageprofiles`
`jar -cf logs.jar <app_server_root>/logs/manageprofiles/*`
 - `<app_server_root>/logs/manageprofiles/default`
`jar -cf profileLogs.jar <app_server_root>/logs/manageprofiles/default/*`
- Extract them on workstation to analyze

As noted earlier, most of the logs that are written by the configuration process have an ASCII encoding. If you do not have a good way to look at them on the z/OS system (such as the viascii editor), you can FTP them to a workstation in binary and look at them there. This slide shows a way to .jar up all the logs easily so that they can be extracted on a workstation and analyzed there instead. This is also a good set of information to provide the service team when opening a PMR.

Summary

- Troubleshooting involves looking at a variety of log files that are written during installation and augmentation.
 - This presentation gave you a methodology to find your problem quickly

In summary, there are many logs written to record the process of the configuration. This presentation showed you how to use them to troubleshoot a problem.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WBPMv70_zOSTroubleshooting.ppt

This module is also available in PDF format at: ..\\WBPMv70_zOSTroubleshooting.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the Web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. Java, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2010. All rights reserved.