IBM® Automated Problem Determination Tool

User's Guide

Version 1 Release 2.2.2

March 30, 2007

Table of Contents

Introduction

The IBM® Automated Problem Determination (AutoPD) Tool is provided as an aid for troubleshooting IBM WebSphere® Portal, IBM WebSphere Application Server, and other IBM software products. The tool focuses on automatic collection of problem data; it also provides symptom analysis support for the various categories of problems encountered by IBM software products. Information pertinent to a type of problem is collected and analyzed to help identify the origin of the problem under investigation. The AutoPD tool assists customers by reducing the amount of time it takes to reproduce a problem with the proper RAS tracing levels set (trace levels are set automatically by the tool), as well as by reducing the effort required to send the appropriate log information into IBM Support. In addition, the AutoPD tool performs symptom analysis to help streamline the problem determination process. In this user's guide we describe how the AutoPD tool can be used in the course of performing problem determination for IBM software products.

Tool Installation

Getting the Latest Version of the Tool

The latest version of the AutoPD Tool (version 1.2.2.2) can be obtained from the following URL:

http://www.ibm.com/support/docview.wss?rs=688&uid=swg24008662

In addition to three download files for the tool itself (a .zip file for Windows®, a .zip file for iSeries, and a .tar file for the other supported environments), this site contains links to additional download files that support internationalization of the text strings that the tool retrieves from the WebSphere Portal message catalog. See **Message Catalog Lookup** below for more information about these additional download files. This User's Guide is also available as a separate download.

Installation on Windows®

Extract RasGUI_Win.zip to the WebSphere Portal *install_root* directory, such as `c:\WebSphere\PortalServer`. It will create the install_root/RasGUI directory. You can also extract the file to the WebSphere Application Server *install_root* directory, such as `c:\WebSphere\AppServer`.

Installation on Linux®, AIX®, HP-UX®, and Solaris®

1. Download RasGUI_Unix.tar to the WebSphere Portal or WebSphere Application Server *install_root* directory, such as `/usr/WebSphere/PortalServer`.
2. Run `tar -xvf RasGUI_Unix.tar`. It will create the *`install_root`*`/RasGUI` directory.

Installation on iSeries™

Special procedures are required to install tool on iSeries:

1. On the iSeries server (e.g., iSeries_hostname), make the WebSphere Portal or WebSphere Application Server UserData root directory available as a network share.
2. On a Windows machine that contains the RasGUI_iSeries.zip file, perform the following steps:
    a. In the DOS console, issue **net use w: \\iSeries_hostname\root** to map the root network share from the iSeries machine onto an available drive (e.g., w:).
    b. Extract the RasGUI_iSeries.zip file into the WebSphere Portal or WebSphere Application Server UserData root directory on the chosen drive.

Note: If the iSeries machine isn't currently configured to allow network shares from a Windows machine, you will need to refer to iSeries documentation for instructions on how to enable this function.

Installation to a Location other than Your Product's Root Directory

For some Portal Install problems, a WebSphere Portal root directory may not have been created. Or you may simply prefer not to install the WebSphere Portal version of the tool into the root directory of a production WebSphere Portal. In either of these situations, you can install the tool in a directory of your choosing. A second step is required in this case, however, for the tool to work properly: you must ensure that the JAVA_HOME environment variable is set to a JDK™ version of 1.4 or greater. If the WebSphere Application Server root directory has been successfully created, you can set JAVA_HOME as follows:

- On a Windows machine, go to the **%WAS_HOME%\bin** directory and run the setupCmdLine.bat batch file, by typing "**setupCmdLine**".
- On a Linux/AIX/HP-UX/Solaris machine, go to the **$WAS_HOME/bin** directory and run the setupCmdLine.sh script, by typing "**. ./setupCmdLine.sh**".
- On the iSeries platform, you can use the procedure described below in **Starting the Tool When No WebSphere Application Server Root Directory Exists**.

Similarly, if you choose to install the WebSphere Application Server version of the tool into a location other than the WebSphere Application Server root directory, you must set the JAVA_HOME variable manually. If the WebSphere Application Server root directory exists, then you can use the setupCmdLine batch file or shell script to do this, as described above. If the WebSphere Application Server root directory does not exist, then you must use the procedure described below in **Starting the Tool When No WebSphere Application Server Root Directory Exists**.

Keeping a Backup Copy of the Tool

Since by default you are installing a copy of the AutoPD tool into the directory in which the WebSphere Portal or WebSphere Application Server has been installed, you should keep a backup copy of the tool's zip or tar file in another location, in case you need to uninstall and reinstall WebSphere Portal or WebSphere Application Server.

Tool Usage

Starting the Tool in GUI Mode on Windows

To start the AutoPD tool in a Windows environment, go to the RasGUI folder that was created when the tool was installed, and run the ***runAutoPD.bat*** batch file contained in this directory. For this batch file to complete properly, it must in turn execute the ***%WAS_HOME%\bin\setupCmdLine.bat*** batch file to set the %JAVA_HOME% environment variable. For a typical WebSphere Application Server or WebSphere Portal install, ***runAutoPD.bat*** automatically finds this second batch file and executes it. If ***runAutoPD.bat*** cannot find this second batch file, you will need to run it manually, by going to the ***%WAS_HOME%\bin*** directory and typing "**setupCmdLine**". Once you have done this, ***runAutoPD.bat*** will complete successfully, and the tool will start in GUI mode.

Starting the Tool in GUI Mode on Linux, AIX, HP-UX, and Solaris

To start the AutoPD tool in a Linux, AIX, HP-UX, or Solaris environment, go to the RasGUI folder that was created when the tool was installed, and run the ***runAutoPD.sh*** script contained in this directory. For this script to complete properly, it must in turn execute the ***$WAS_HOME/bin/setupCmdLine.sh*** script to set the $JAVA_HOME environment variable. For a typical WebSphere Application Server or WebSphere Portal install, ***runAutoPD.sh*** automatically finds this second script and executes it. If ***runAutoPD.sh*** cannot find this script, you will need to run it manually, by going to the ***$WAS_HOME/bin*** directory and typing "**. ./setupCmdLine.sh**". Once you have done this, ***runAutoPD.sh*** will complete successfully, and the tool will start in GUI mode.

Support for GUI Mode in the iSeries Environment

A GUI version of the AutoPD tool is not currently supported for the iSeries platform. You must use the tool in console mode for this environment, using the procedure described in the next section.

Starting the Tool in Console Mode

The AutoPD tool supports a non-graphical console mode in all environments, for situations where a graphical user interface is not appropriate. An example of such a situation is when the **telnet** command is used to login to a particular server remotely. To start the AutoPD tool in console mode on Windows, run the ***runAutoPDConsole.bat*** batch file found in the tool's installation directory***.*** To start the AutoPD tool in console mode on Linux, AIX, HP-UX, or Solaris, run the ***runAutoPDConsole.sh*** script located in the tool's installation directory. For the iSeries platform, you must first set up the JAVA_HOME variable (by, for example, issuing the command `export JAVA_HOME=/QIBM/ProdData/Java400/jdk14`), and then run the ***runAutoPDConsole_iSeries.sh*** script, which is also located in the tool's installation directory.

Starting the Tool When No WebSphere Application Server Root Directory Exists

In some situations, a WebSphere Application Server root directory may not have been created. This will prevent you from running the **setupCmdLine.bat** or **setupCmdLine.sh** contained under this root directory to set the JAVA_HOME environment variable used by the tool. If this situation occurs, you can still start the tool by manually insuring that the JAVA_HOME environment variable is set to a JRE version of 1.4 or greater. For example, if on a Windows platform you have jdk1.4 installed at c:\jdk1.4, you would set JAVA_HOME using the following command:

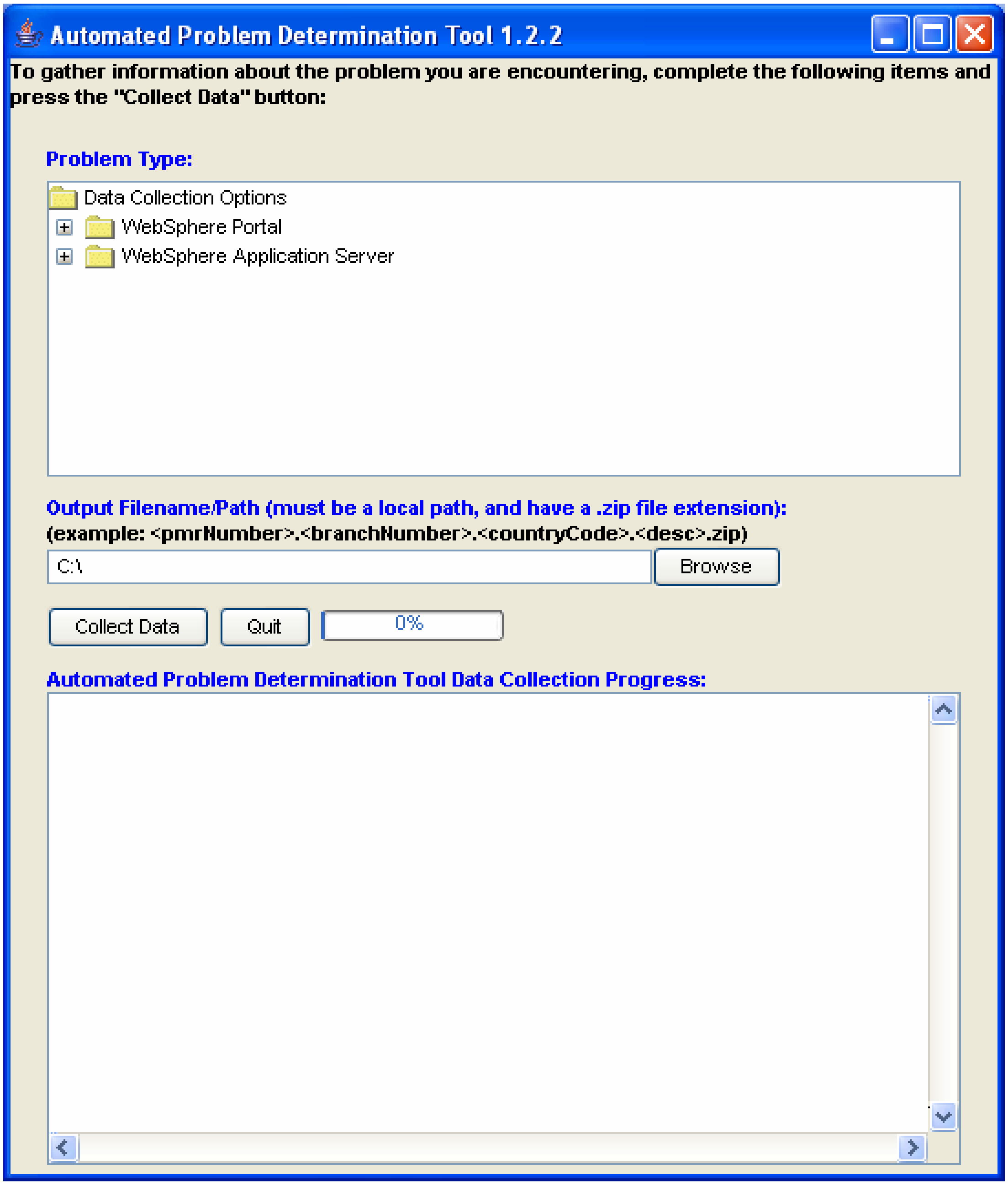
```
SET JAVA_HOME=c:\jdk1.4
```

On a Linux, AIX, HP-UX, Solaris, or iSeries platform, the exact command syntax to set JAVA_HOME varies depending on which shell you are using. For example, if you are using bash shell and you have the JDK installed in /opt/jdk14, you would set JAVA_HOME using the following command:

```
export JAVA_HOME=/opt/jdk14
```

After installing the tool into one of the default locations, you may find that it fails to start with the error message "ERROR: This tool requires JRE 1.4 or greater to run." This message indicates that the JRE included with WebSphere Application Server, which the tool automatically uses when it has been installed into one of the default locations, is at a level lower that 1.4. In this case you should follow the instructions in this section to install the tool into a different location and set the JAVA_HOME variable manually to point to a JRE of the proper level.

Interacting with the Tool in GUI Mode



Automated Problem Determination Tool 1.2.2

To gather information about the problem you are encountering, complete the following items and press the "Collect Data" button:

Problem Type:

Data Collection Options
- WebSphere Portal
- WebSphere Application Server

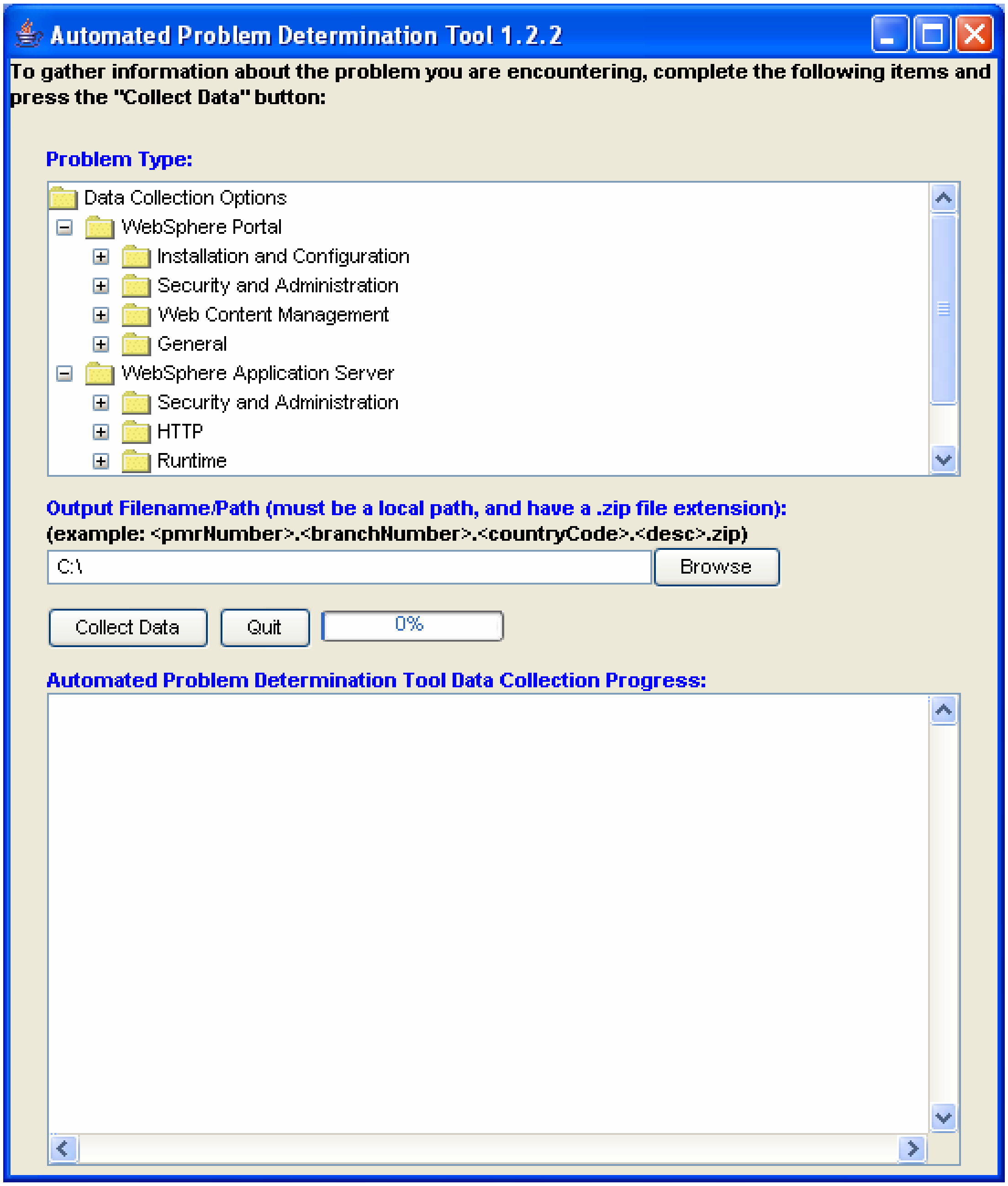
Output Filename/Path (must be a local path, and have a .zip file extension):
(example: <pmrNumber>.<branchNumber>.<countryCode>.<desc>.zip)

C:\

Automated Problem Determination Tool Data Collection Progress:

Figure 1

Once the AutoPD tool has been started, the graphical interface shown in Figure 1 will be displayed. Before the tool can be used to perform data collection and analysis, you must first select a problem type in the Problem Type window. In the initial view, the tree is collapsed to show only its top-level folders. You can expand the top-level folders to reveal the folders nested below them (Figure 2).



Automated Problem Determination Tool 1.2.2

To gather information about the problem you are encountering, complete the following items and press the "Collect Data" button:

Problem Type:

Output Filename/Path (must be a local path, and have a .zip file extension):
(example: <pmrNumber>.<branchNumber>.<countryCode>.<desc>.zip)

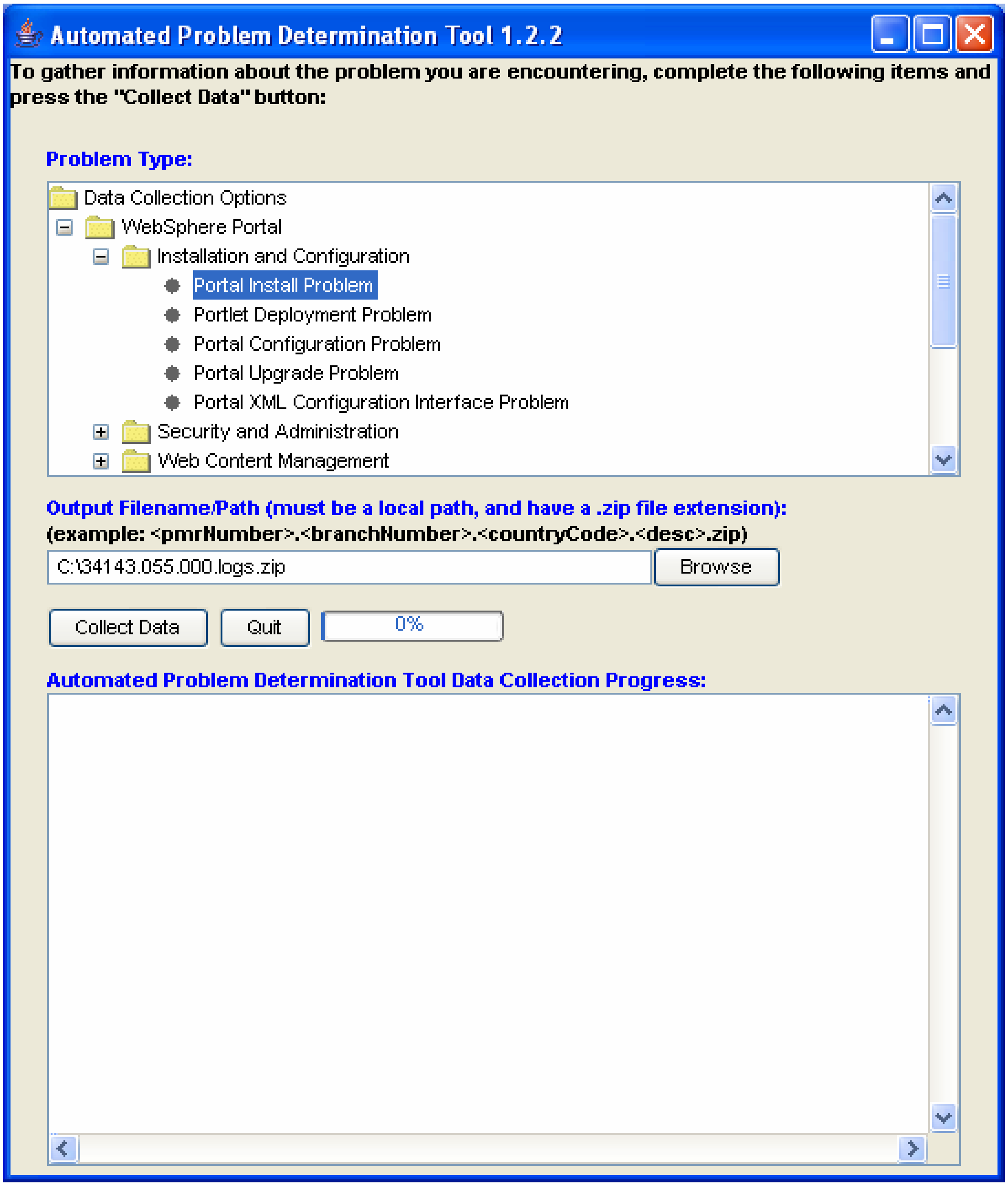
C:\

Automated Problem Determination Tool Data Collection Progress:

Data Collection Options
WebSphere Portal
Installation and Configuration
Security and Administration
Web Content Management
General
WebSphere Application Server
Security and Administration
HTTP
Runtime
Browse
Collect Data
Quit
0%

Figure 2

Continue expanding folders until the problem type you want is revealed, then highlight it (Figure 3).


Automated Problem Determination Tool 1.2.2

To gather information about the problem you are encountering, complete the following items and press the "Collect Data" button:

Problem Type:

- Data Collection Options
  - WebSphere Portal
    - Installation and Configuration
      - Portal Install Problem
      - Portlet Deployment Problem
      - Portal Configuration Problem
      - Portal Upgrade Problem
      - Portal XML Configuration Interface Problem
    - Security and Administration
    - Web Content Management

Output Filename/Path (must be a local path, and have a .zip file extension):
(example: <pmrNumber>.<branchNumber>.<countryCode>.<desc>.zip)

C:\34143.055.000.logs.zip

Automated Problem Determination Tool Data Collection Progress:

Figure 3

Prior to v1.1.4, the tool's main GUI also included fields asking you to supply the root directories for WebSphere Application Server, WebSphere Portal, and the Deployment Manager. To provide for greater flexibility, these interactions have been moved to pop-ups issued by the individual collection scripts. With this new approach, a script can solicit (and then validate) only the root directories relevant to its specific collection activities. A root directory that you supply through one of these pop-ups is saved by the tool, and presented to you the next time that root directory is requested. Thus you will never need to reenter a directory name, unless you want to change the instance against which collection is being performed.

Other than Problem Type, the only input field value that you must supply on the main GUI is a filename for the data collection zip file. The filename can be anything you want: the tool will automatically append the "zip" extension to it. You might, for example, use the filename

Install_failed_2005_01_03. If the problem you are dealing with results in a problem record being opened with the IBM Support Center, we recommend that after the PMR has been opened you change the filename to **<pmrnumber>.<branchnumber>.<countrycode>.<short description>.zip.** An example of a filename of this type would be *34143.055.000.logs.zip.* For compatibility with a previous format, commas are also accepted in place of the first three periods (but not in place of the fourth one prior to the "zip" extension). However, periods are the preferred delimiter. Figure 3 shows the tool with the Problem Type highlighted and the Output Filename/Path field filled in.

With these values filled in the next step is to press the Collect Data button. As it proceeds, the collection script will ask you for any additional information it needs to complete its collection activities. Most WebSphere Portal scripts ask for the WebSphere Application Server and WebSphere Portal root directories, as well as for whether the WebSphere Portal for which you're collecting data is running as part of a cluster. A script may also ask you for additional configuration information, for information about the sequence of events leading up to the problem you are dealing with, or for your preferences regarding how it should complete the collection. Once it has all the information it needs, the script will proceed with the remainder of the collection. Details regarding each of the collection scripts the tool supports for WebSphere Portal appear in **Appendix A**.

For some WebSphere Portal collection scripts, the AutoPD tool must reset the trace levels for a portal instance to the appropriate values for a particular problem type. For WebSphere Portal version 5, resetting these trace levels requires the tool to stop and restart the portal instance. Before it does this, it will ask you whether you are prepared to have your portal instance recycled in this way. If you answer "No," some scripts will terminate the entire collection at that point, since nothing of value can be obtained without recycling the portal instance. Other scripts, however, will proceed to collect the log files that already exist, since these files may contain sufficient data to diagnose the problem.

WebSphere Portal version 6 introduces a new capability to reset trace levels in a running portal instance without recycling it. You may, however, prefer to recycle the portal instance, even in a WebSphere Portal version 6 environment. The collection scripts let you choose how you would like to reset traces in this case: with or without recycling the portal instance.

If you answer "Yes" to the question about recycling the portal instance, the tool will do this, and then pause while you reproduce the problem. Alternatively, if you choose to reset trace levels in a WebSphere Portal version 6 environment without recycling the portal instance, the tool will do this, and then pause while you reproduce the problem. In both cases, the tool will then complete the data collection and analysis operations, and FTP the results to IBM Support if you indicate that you want it to do this. Finally, the tool will put WebSphere Portal back into the state it was in when the script began: with its previous trace levels and other configuration settings and (with one exception) the server either started or stopped depending on which of these states it was in before.

The one exception to this rule is the case where you initially requested that the trace levels be reset without recycling the portal instance, but then allowed the tool to recycle it when you were informed that it was not currently started. In this case, the collection leaves the portal instance in the started state, on the theory that this is the state you had believed it was in. If you need the portal instance to be returned to the stopped state, you can do this manually after the collection has completed.

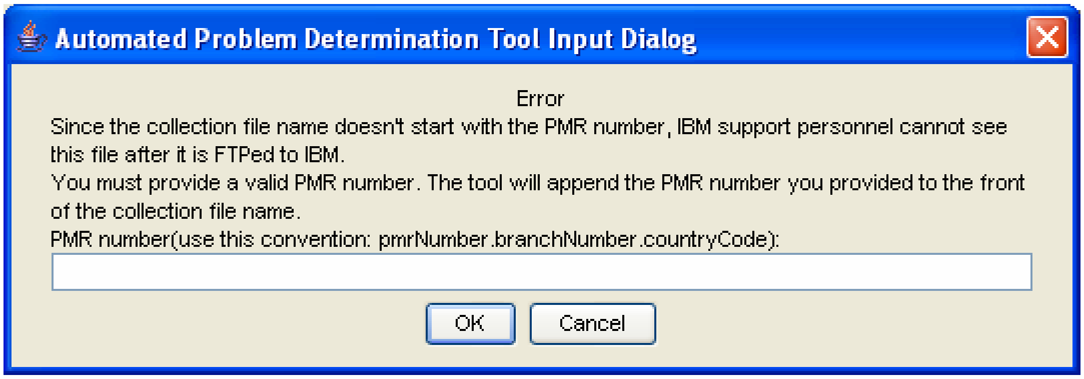
The name of the zip file the tool sends to IBM Support via FTP will be the filename entered in the Output Filename/Path box, with the server hostname and current timestamp appended to it.

Example: if the Output Filename/Path filename used was ***34143.055.000.logs.zip***, the name of the zip file sent to IBM would be ***34143.055.000.logs.zip-<hostname>-<currentTimestamp>.zip.***

Validation of the Collection Zip File Name

The automation at the IBM Support FTP site ftp.emea.ibm.com is entirely dependent on the file-naming convention described on the AutoPD tool's main GUI. If it receives a file with a name that doesn't follow this convention, then that file is never seen by IBM Support. Consequently, before the AutoPD tool invokes an FTP operation to send a collection zip file to ftp.emea.ibm.com, it validates the collection zip file's name against the convention. If the file name isn't in the correct form, the pop-up window shown below in Figure 4 is presented to the user, so that the file name can be corrected.

The current file naming convention for the FTP site is the one documented here, where the individual fields in the composite file name are separated by periods. Previously, however, the convention was to use commas as the separators. Since the FTP site still handles files named according to the previous convention, the AutoPD tool's validation logic accepts both periods and commas as delimiters.



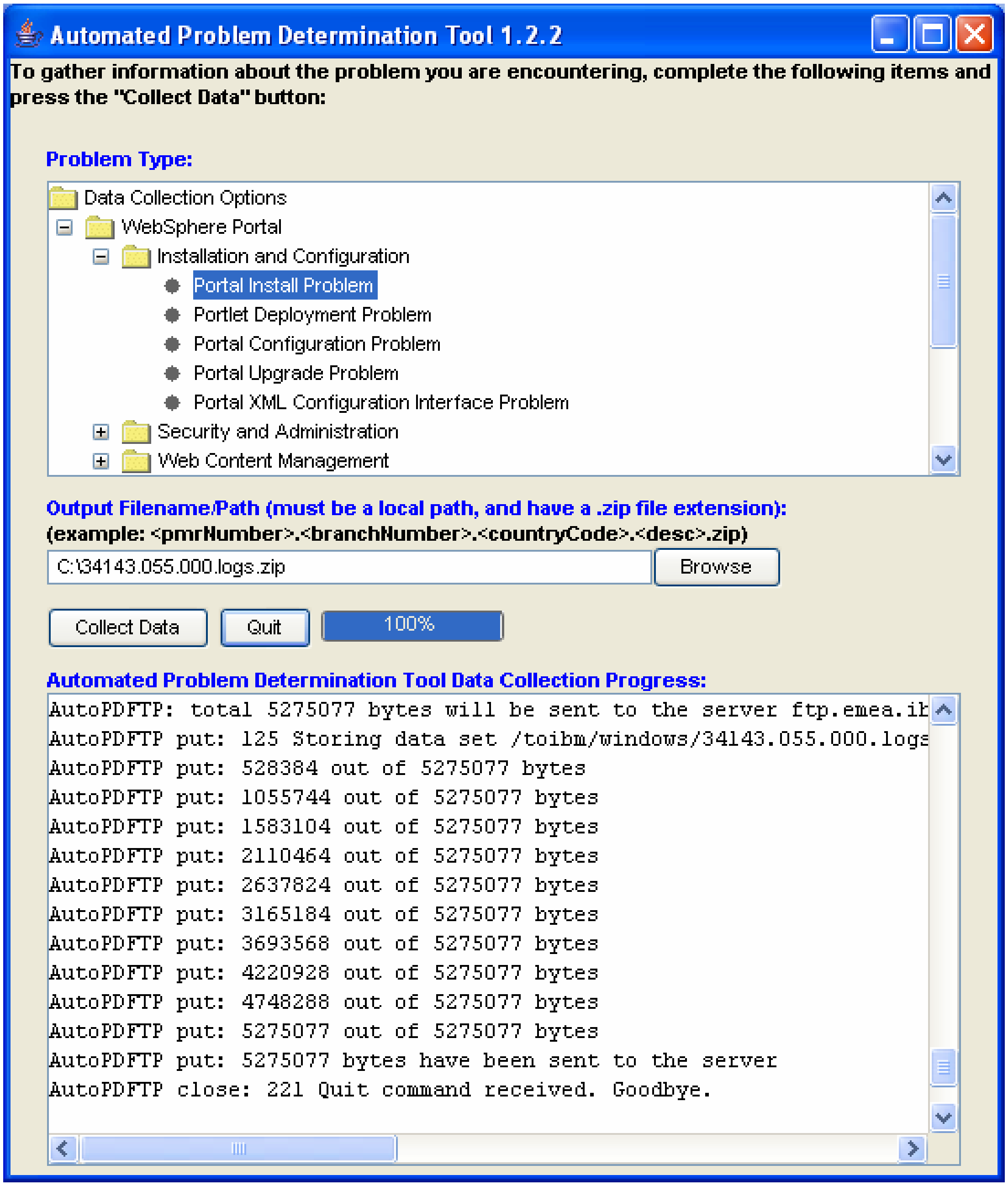
Error
Since the collection file name doesn't start with the PMR number, IBM support personnel cannot see this file after it is FTPed to IBM.
You must provide a valid PMR number. The tool will append the PMR number you provided to the front of the collection file name.
PMR number(use this convention: pmrNumber.branchNumber.countryCode):

Figure 4

If the FTP destination is anything other than ftp.emea.ibm.com, this validation step is skipped. This leaves open the possibility of using the AutoPD tool's FTP capabilities to send a collection zip file to a different destination (for example, an enterprise problem center or an IBM business partner) which doesn't use the file-naming convention that IBM Support uses.

Feedback regarding FTP Progress

Once an FTP operation has started, the AutoPD tool provides you with feedback in the progress window regarding its progress. As Figure 5 illustrates, the tool presents you with a message each time 10% of the collection zip file has been successfully transmitted to the FTP server.



To gather information about the problem you are encountering, complete the following items and press the "Collect Data" button:

Problem Type:

- Data Collection Options
  - ⊟ WebSphere Portal
    - ⊟ Installation and Configuration
      - ● Portal Install Problem
      - ● Portlet Deployment Problem
      - ● Portal Configuration Problem
      - ● Portal Upgrade Problem
      - ● Portal XML Configuration Interface Problem
    - ⊞ Security and Administration
    - ⊞ Web Content Management

Output Filename/Path (must be a local path, and have a .zip file extension):
(example: <pmrNumber>.<branchNumber>.<countryCode>.<desc>.zip)

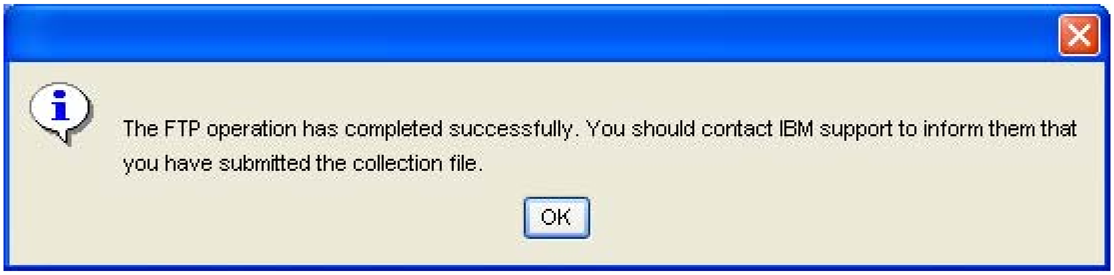
C:\34143.055.000.logs.zip

Automated Problem Determination Tool Data Collection Progress:

```
AutoPDFTP: total 5275077 bytes will be sent to the server ftp.emea.ib
AutoPDFTP put: 125 Storing data set /toibm/windows/34143.055.000.logs
AutoPDFTP put: 528384 out of 5275077 bytes
AutoPDFTP put: 1055744 out of 5275077 bytes
AutoPDFTP put: 1583104 out of 5275077 bytes
AutoPDFTP put: 2110464 out of 5275077 bytes
AutoPDFTP put: 2637824 out of 5275077 bytes
AutoPDFTP put: 3165184 out of 5275077 bytes
AutoPDFTP put: 3693568 out of 5275077 bytes
AutoPDFTP put: 4220928 out of 5275077 bytes
AutoPDFTP put: 4748288 out of 5275077 bytes
AutoPDFTP put: 5275077 out of 5275077 bytes
AutoPDFTP put: 5275077 bytes have been sent to the server
AutoPDFTP close: 221 Quit command received. Goodbye.
```

Figure 5

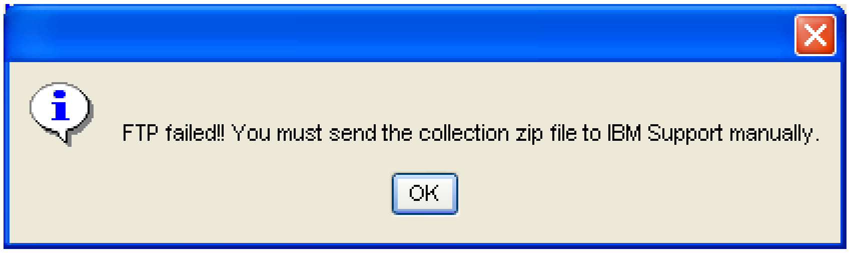
Figure 6 shows a pop-up window that appears when the entire FTP operation completes successfully.



The FTP operation has completed successfully. You should contact IBM support to inform them that you have submitted the collection file.

Figure 6

Figure 7 shows the pop-up that appears if the FTP operation fails. In this case, you should review the specific messages in the progress window to determine why the operation failed. The collection itself will continue to completion, at which time the collection zip file will have been saved in the location you chose when the collection began. Depending on why the script's automated FTP operation failed, you may be able to FTP the collection zip file to IBM Support manually.



FTP failed!! You must send the collection zip file to IBM Support manually.

Figure7

Manual FTP of the Data Collection Zip File

If for some reason the automatic FTP support provided by the AutoPD tool cannot be utilized, the data collection zip file can be sent to IBM Support using a manual approach. You can accomplish this by performing the following actions:

Note: use cmdline FTP, not browser FTP

1) **Enter: FTP ftp.emea.ibm.com.**
2) **Login as *anonymous* and enter your e-mail ID as password.**
3) **Change directories to /toibm by typing *cd /toibm* at the command prompt.**
4) **Change to one of the following directories:**
   aix, linux, os400, unix, or windows
5) **Enter: *bin* at the command prompt.**
6) **Use the following file naming convention to name your file and put it on the server. Your PMR will be updated to list where files are stored, using the format: *xxxxx.bbb.ccc.desc.yyy****
7) **Enter the *quit* command.**
8) **Notify IBM Support via e-mail that you have uploaded the requested data collection zip file.**

For step 4, you should use the operating system value "unix" if the WebSphere Application Server or WebSphere Portal for which data is being collected is running on HP-UX or Solaris.

* The PMR file name must use the format **xxxxx.bbb.ccc.desc.yyy** as described below. Use periods in the file name and a period between the file name and file type.

Field	**Explanation**	**Sample**
xxxxx	PMR Number	34143
bbb	Branch Office	055
ccc	IBM Country Code	000
desc	short description of file	Logs
yyy	A short description for the file type	zip or tar

Example: **34143.055.000.logs.zip**

For more information on submitting a WebSphere Portal collection zip file, see the link:
http://www-1.ibm.com/support/docview.wss?uid=swg21201571.

For more information on submitting a WebSphere Application Server collection zip file, see the link:
http://www.ibm.com/support/docview.wss?rs=180&context=SSEQTP&uid=swg21153852

Interacting with the Tool in Console Mode

The console mode provides you with command-line control of the AutoPD tool's collection scripts. In this section, we describe some specific features designed to improve your interactions with the tool when you use it in console mode.

Reduced Typing

The AutoPD tool provides features that assist you when you use the tool in console mode, by reducing the amount of repetitive typing you will need to do. There are two separate features that fall into this category:

1. When you are asked to supply a value such as a product's root directory, the AutoPD tool may already have a value available to it. (This value might be a default value that ships with the tool, a value that the tool has retrieved from a local configuration file, or a value that the tool saved after you entered it during a previous collection. For this discussion, the way in which the tool came to have this value is irrelevant.) If the tool has such a value, it presents it to you on the console interface between angle brackets; for example,

   ```
   Input Required
   WebSphere Application Server root directory<c:\ibm\AppServer>:
   ```

   If you want to accept the value provided by the tool, you may simply press the enter key: there is no need to re-type the value. If you want to supply a different value, you type it in as usual and then press the enter key.

   If the tool doesn't have a value at the point where the collection asks for it, it indicates that this is the case by omitting the angle brackets:

   ```
   Input Required
   WebSphere Application Server root directory:
   ```

   This indicates that you must supply a value before the collection can proceed.
2. When the tool presents you with a series of choices, such as whether to proceed with a collection or cancel it, or whether to proceed down the data collection or problem recreation path, each of the choices is associated with a number. For example,

   ```
   Do you want to proceed immediately to data collection or to
   recreate the problem prior to data collection?
   1:Proceed to Data Collection
   2:Recreate the Problem prior to Data Collection
   ```

   You can select any of the choices offered by typing its number and pressing the enter key. In addition, you can select the *first* choice (script writers typically make this choice the most common one) simply by pressing the enter key. Note that this second alternative is not supported for the initial menu where you choose which collection script to run, since there in nothing special about the first script in the list. In order to invoke the first script, you must enter its number (1), just as you must enter the number for any other script.

Support for Input Dialogues

When you are using the AutoPD tool in GUI mode, collection scripts often solicit input from you through Swing elements such as text fields and text boxes. In console mode, these input requests are handled via command-line display and input. In order to maintain the same interaction patterns that the GUI mode achieves with the three dialogue-closing buttons **OK**, **Skip**, and **Cancel**, the console mode provides additional delimiters that bracket an input sequence, and then gives you the same opportunity to accept or cancel the results of the sequence that the buttons give you in the GUI mode. For example,

```
**************************************************
* Input Required
* WebSphere Application Server root directory<c:\ibm\AppServer>:

* WebSphere Portal root directory<c:\ibm\PortalServer>:

**************************************************
OPTIONS FOR COMPLETING THE INPUT DIALOG
1: OK<Continue the collection using the values you set during the INPUT
DIALOG>
2: Cancel<Stop the collection>
```

Notice how the input dialogue is bracketed with lines of asterisks, to indicate how many inputs you are being asked about in the OPTIONS FOR COMPLETING THE INPUT DIALOGUE. Notice also that you have taken advantage of the “reduced typing” capability described in the previous section: by pressing the enter key twice (causing the two blank lines to appear), you have accepted the two cached values that the tool has proposed.

In this case, you are presented with only two options for terminating the input dialog: you may either continue the collection with the values you supplied (1 = OK), or you may stop the collection entirely (2 = Cancel). You are limited to the two options in this case because the collection can’t proceed without these two root directory values. In other cases, however, three options may be presented to you:

```
OPTIONS FOR COMPLETING THE INPUT DIALOG
1: OK<Continue the collection using the values you set during the INPUT
DIALOG>
2: Skip<Continue the collection without using the values you set during the
INPUT DIALOG>
3: Cancel<Stop the collection>
```

The new option (2 = Skip) allows the collection to continue, but without the values requested by the input dialog. This option applies only to requests for “extra” information that might prove valuable in diagnosing a problem, but that is not essential to the completion of the collection itself.

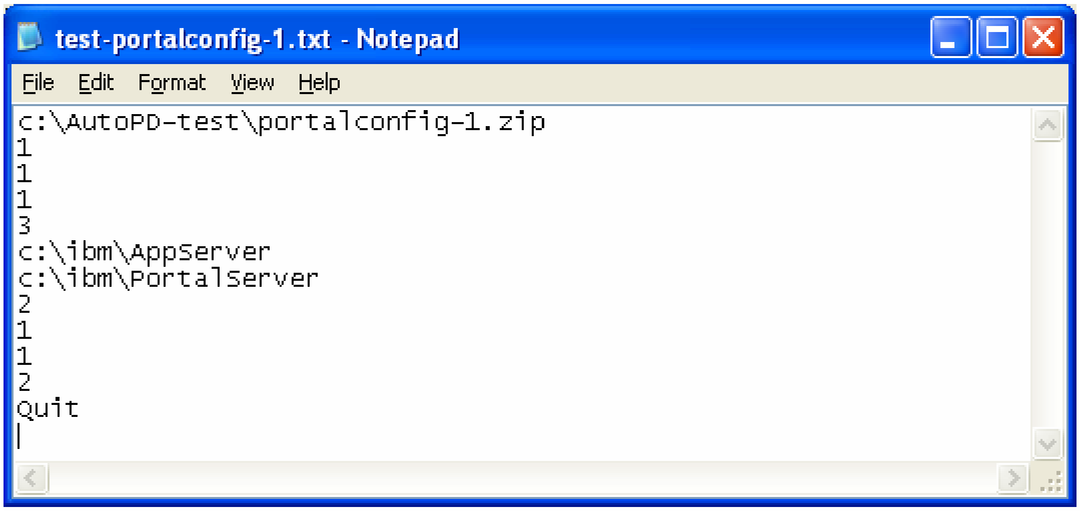
In some cases, collection script authors may elect to use “input” dialogues solely for the purpose of providing you with information at a particular point in a script, without actually soliciting any input from you. To continue the collection in these cases, you may simply press the enter key, since the **OK** option is always mapped to the default choice ‘1’.

Using the AutoPD Tool’s Script Input Mode

The AutoPD tool may be run in script input mode, in which the tool receives input to direct its operations from a text file containing the input commands that are normally entered by a human user. The script input mode can be invoked from the console mode version of the tool by passing the name of the file containing the tool commands as an argument to the console batch file or shell script being used to start the tool. For example, the following command:

runAutoPDConsole.bat test-portalconfig-1.txt

will start the tool in console mode and will read its command line input from the test-portalconfig-1.txt file. The ***runAutoPDConsole.sh*** and ***runAutoPDConsole_iSeries.sh*** shell scripts can be used in a similar fashion. Figure 8 provides a sample input script file. The first line specifies the collection zip file name for the collection. The next '1' indicates that this file should be used for this script execution. The sequence of numbers following it navigates down through the menu tree, to reach the point where it can invoke the Portal Configuration collection script. Next come the input fields where the user supplies the WebSphere Application Server and WebSphere Portal roots. Since there is no way of knowing when the script is written whether values like these will already be known to the AutoPD engine when the script is used, they are always specified explicitly. The final numbers represent the user's responses to various dialogues within the Portal Configuration collection script.



```
c:\AutoPD-test\portalconfig-1.zip
1
1
1
3
c:\ibm\AppServer
c:\ibm\PortalServer
2
1
1
2
Quit
```

Figure 8

AutoPD Logging

The AutoPD tool generates several log files that are included in the data collection zip file sent to IBM Support. These log files address two distinct requirements:

- Providing a record of how the tool was used to diagnose a WebSphere Application Server or WebSphere Portal problem. In this case, the log contains the same information that appears in the scrolling progress window when the tool is executed in the GUI mode.
- Providing detailed information for diagnosing problems with the AutoPD tool itself.

Three significant changes were introduced into the AutoPD tool's logging implementation in version 1.2.1

- **Redirection of the log files:** Prior to version 1.2.1, the AutoPD log files were always written to the same location: the /logs subdirectory underneath the tool's /RasGUI root directory. You can now specify this location explicitly, using one of two approaches:
  - By using the system variable autopd.instance.area. You can specify a value for this system variable by adding the qualifier -Dautopd.instance.area=c:\TheLocation to the statement in the batch file that launches the tool (on the Windows operating system),

    or by adding the qualifier -Dautopd.instance.area=/User/TheLocation to the statement in the shell script that launches the tool (on the other operating systems),
    - By setting the property autopdinstancearea in the autopd.properties file, for example autopdinstancearea=c:\TheLocation.

  If locations are specified in both of these ways, the one specified via the system variable takes precedence.

  This new redirection function introduced in version 1.2.1 actually applies to more than just the AutoPD log files. These variables specify a directory location for all the files that the AutoPD tool writes for its own purposes. This includes not only the AutoPD log files, but also all temporary files that the tool writes and then later includes in a collection zip file or reads for values to use later in a collection script. Thus if the indicated directory location is c:\TheLocation, then the AutoPD log files will be written to the subdirectory c:\TheLocation\log and the temporary files will be written to the subdirectory c:\TheLocation\tmp. Finally, files that the AutoPD uses to recover from a failed script execution will be written to the subdirectory c:\TheLocation\recovery.
- **Cleanup of old log files:** Prior to version 1.2.1, there was no mechanism for automatic cleanup of old AutoPD log files. You can now enable automatic cleanup of these files by specifying a number of log file “generations” to be retained. A generation of log files is a count applied to each of three types of log file instances: autopdsetup.log instances, autopd.log instances, and autopdecho.log instances. If, for example, you specify that three generations of log files should be retained, then when the tool detects that three instances of one of these types of log files exist, it will delete the oldest one prior to writing a new one.

  As with the location for writing log files, there are two ways to specify the number of log file generations the tool should retain: via the system variable autopd.logfile.generations and via the property maxLogInstanceFiles in the autopd.properties file. If both of these are specified, the system variable takes precedence. If neither is specified, no log file cleanup takes place. Note, however, that the tool ships with the default setting maxLogInstanceFiles=10 in the autopd.properties file. So to disable log file cleanup, it will be necessary to remove this property setting from the file.
- **Elimination of the merged, comprehensive log files:** Prior to version 1.2.1, the collection scripts shipped with the AutoPD tool always merged the instance logs created prior to and during a script execution into a pair of comprehensive logs: autopd.log for the autopdsetup.log and autopd.log instance logs, and autopdecho.log for the autopdecho.log instance logs. Both the instance logs and the comprehensive logs were then included in the collection zip file. Experience has shown that the comprehensive logs simply double the space required for AutoPD logs, both in the local file system and in the collection zip files, without providing any real benefits over those provided by the instance logs alone. Thus the collection scripts shipped with the tool no longer create the comprehensive logs.

Figure 9 shows a portion of the directory hierarchy present in a collection zip file. This hierarchy contains a subdirectory /log containing the instance log files, which fall into three categories. Those whose names start with “autopdinstance” provide complete information for one execution of a collection script, including diagnostic information for the AutoPD tool itself. Those whose names start with “autopdechoinstance” capture the text that was displayed to the user in the progress window during one execution of a collection script. This text may include not only messages originating from the AutoPD tool itself, but also those coming from any separate tools or other executable components that a collections script invokes. Finally, those whose names start with “autopdsetupinstance” provide messages related to the time when the tool is first brought up, until the first collection script is invoked. Note how the tool distinguishes among different instances of these log files by appending current time stamps to their filenames.

```
 ...
autopdzip/autopd/log/autopdinstance2005.10.05-12.03.37.304EDT.log
```

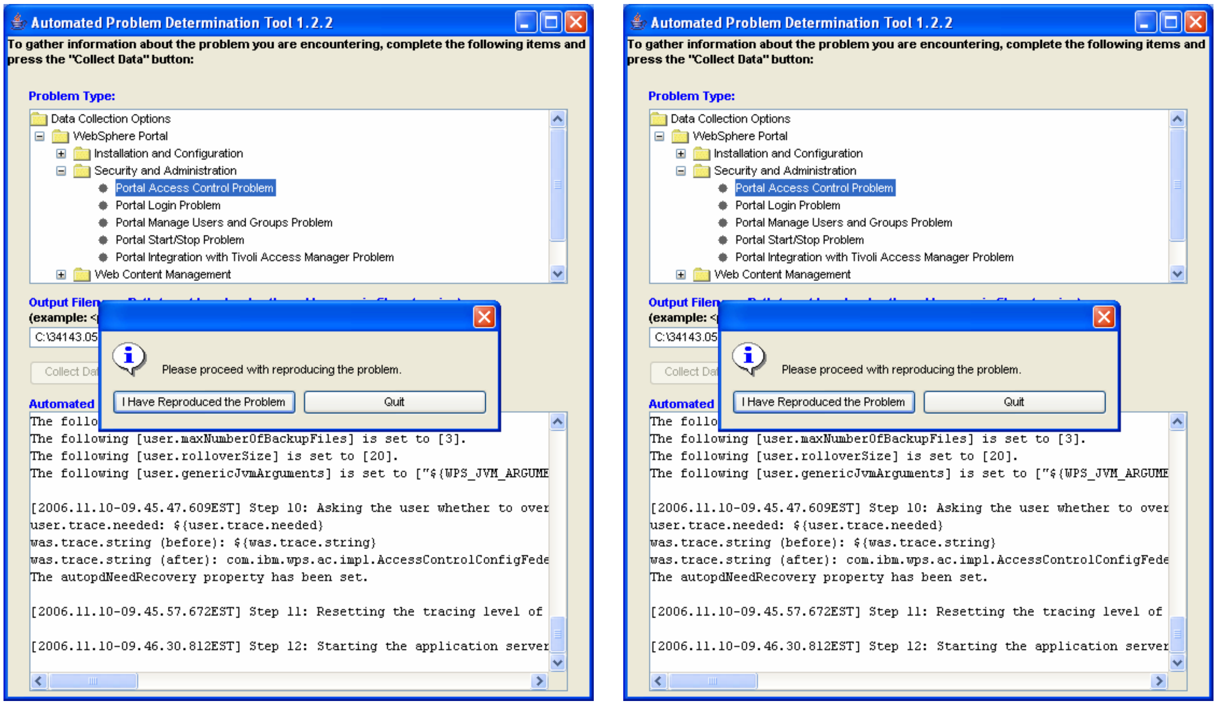
```
autopdzip/autopd/log/autopdinstance2005.10.05-12.04.49.808EDT.log
autopdzip/autopd/log/autopdinstance2005.10.05-12.06.14.820EDT.log
autopdzip/autopd/log/autopdinstance2005.10.05-12.07.31.240EDT.log
autopdzip/autopd/log/autopdechoinstance2005.10.05-12.03.37.304EDT.log
autopdzip/autopd/log/autopdechoinstance2005.10.05-12.04.49.808EDT.log
autopdzip/autopd/log/autopdechoinstance2005.10.05-12.06.14.820EDT.log
autopdzip/autopd/log/autopdechoinstance2005.10.05-12.07.31.240EDT.log
autopdzip/autopd/log/autopdsetupinstance2005.10.05-12.03.18.608EDT.log
```

Figure 9

Using the AutoPD Tool in a Vertical Cluster

The primary reason for introducing log instances into the logging design for the AutoPD tool was to enable the tool to be used effectively when WebSphere Portal is deployed in a vertical cluster. At least for some problem types, the recommended way to use the tool in a vertical cluster is to launch a separate copy of the tool for each cluster member (Figure 10). With collection scripts running on the different copies of the tool, it becomes possible to set up tracing on all of the WebSphere Portal instances in the cluster, then reproduce the problem, and then collect and analyze the logs from all of the instances. You, as the tool user, provide the synchronization among the different collection activities, by pausing all of the scripts until they all indicate that it is time for you to reproduce the problem: this point in time is illustrated in Figure 10. In order to prevent one collection zip file's overwriting the other one, you will need to give them different names. You should use the <desc> field in the recommended file name format for this purpose, so that both files can be associated with the PMR for the problem you are dealing with.

The AutoPD log instances come into the picture because, with multiple instances of the tool running concurrently, it is necessary for each instance to have its own logs – interleaving entries from all of the instances into a single log would make that log very difficult to interpret. By noting how the time stamps in the different instance logs relate to each other, it is possible to reconstruct how the different collection activities were themselves related to each other.



Automated Problem Determination Tool 1.2.2

To gather information about the problem you are encountering, complete the following items and press the "Collect Data" button:

Problem Type:
Data Collection Options
WebSphere Portal
Installation and Configuration
Security and Administration
Portal Access Control Problem
Portal Login Problem
Portal Manage Users and Groups Problem
Portal Start/Stop Problem
Portal Integration with Tivoli Access Manager Problem
Web Content Management
Output Filen
(example: <
Collect Da
Automated

C:\34143.05

Please proceed with reproducing the problem.

I Have Reproduced the Problem
Quit

```
The follo
The following [user.maxNumberOfBackupFiles] is set to [3].
The following [user.rolloverSize] is set to [20].
The following [user.genericJvmArguments] is set to ["${WPS_JVM_ARGUME

[2006.11.10-09.45.47.609EST] Step 10: Asking the user whether to over
user.trace.needed: ${user.trace.needed}
was.trace.string (before): ${was.trace.string}
was.trace.string (after): com.ibm.wps.ac.impl.AccessControlConfigFede
The autopdNeedRecovery property has been set.

[2006.11.10-09.45.57.672EST] Step 11: Resetting the tracing level of

[2006.11.10-09.46.30.812EST] Step 12: Starting the application server
```

Automated Problem Determination Tool 1.2.2

To gather information about the problem you are encountering, complete the following items and press the "Collect Data" button:

C:\34143.05

Please proceed with reproducing the problem.

The follo
The following [user.maxNumberOfBackupFiles] is set to [3].
The following [user.rolloverSize] is set to [20].
The following [user.genericJvmArguments] is set to ["${WPS_JVM_ARGUME

[2006.11.10-09.45.47.609EST] Step 10: Asking the user whether to over
user.trace.needed: ${user.trace.needed}
was.trace.string (before): ${was.trace.string}
was.trace.string (after): com.ibm.wps.ac.impl.AccessControlConfigFede
The autopdNeedRecovery property has been set.

[2006.11.10-09.45.57.672EST] Step 11: Resetting the tracing level of

[2006.11.10-09.46.30.812EST] Step 12: Starting the application server

Figure 10

Inclusion of Topology Data in WebSphere Portal Collections

When IBM Support diagnoses a problem, it is often helpful for them to have a view of the overall topology of the environment in which the problem occurred. For selected WebSphere Portal collection scripts, you will be given the option of including WebSphere Portal and WebSphere Application Server topology information in the collection. Inclusion of this additional data is completely optional – if you elect to omit it, the collection will complete normally, and the collection zip file will contain all of the files identified for the problem type except for the topology-related ones.

Including topology-related files in a collection has only minimal impact on the size of the collection zip file. Its effect on the time it takes the collection to complete varies, depending on the state of the system. The WebSphere Portal topology data is gathered by issuing an xmlaccess export command, which requires that the Portal Server be started. If the Portal Server is already started at the time the collection is performed, the time penalty for collecting the topology data is relatively small. If, however, the Portal Server must be started prior to issuing the xmlaccess export command, then the time penalty is significantly greater. Regardless of whether the Portal Server was started prior to the collection, the AutoPD tool returns it to its previous state before a collection completes.

Hiding Sensitive Information in the Collection Zip File

Some of the files that the AutoPD tool collects contain passwords and other sensitive information that an enterprise might not want to share with IBM Support. Typically, IBM Support needs these files in order to diagnose a problem, but they do not need the sensitive information contained in the files. So prior to sending a collection zip file to IBM Support, the collection scripts for WebSphere Portal and WebSphere Application Server invoke a function that does two things. It detects and hides passwords in all properties files and XML documents included in the collection zip file, and it removes from the collection zip file entirely a few files that contain other types of sensitive data.

Like many aspects of the AutoPD tool, this function's behavior is controlled by means of XML documents. In this case, the documents are located in the /properties/wps/security subdirectory. By editing these documents, it is possible to make the function hide more or less information, and to make it remove additional files. However, the documents as they are shipped with the tool invoke password-hiding for all XML documents and properties files in the collection zip file, so it is unlikely that they would need to be modified.

Usage Scenarios

There are three main usage scenarios for the AutoPD tool: the problem reproduction scenario, the basic log collection scenario, and the combined basic log collection / problem reproduction scenario, which combines the previous two in a single script. There are also three options for invoking the tool in a more limited way. One of these options makes modifications to the state of the system that persist after the script completes. The second performs analysis on a pre-assembled set of log files. The third triggers collection of product data only, with no log collection or analysis at all.

Each of the collection (problem) types the tool supports for WebSphere Portal falls under one of these usage scenarios. See **Appendix A** for a complete listing of the WebSphere Portal collection types, with indications of which usage scenario applies to each of them. **Appendix B** provides a similar mapping between the WebSphere Application Server collection types and their respective usage scenarios.

Basic Log Collection Scenario

In the basic log collection scenario, the problem is of a type that does not need to be reproduced -- it is sufficient for the AutoPD tool to collect and analyze a set of logs that already exist. An example of such a problem is when an administrator is unable to install WebSphere Portal, or has been instructed by IBM Support to invoke the RasUtil collector tool (which has been incorporated into the AutoPD tool for convenience) on an instance of WebSphere Portal. In this scenario, the AutoPD tool does not wait for the user to reproduce the problem. Instead, it proceeds with the data collection and analysis operations, and then FTPs the data it has collected to IBM Support.

Problem Reproduction Scenario

In the problem reproduction scenario, the problem is of a type for which the AutoPD tool must first reset the WebSphere Portal and/or WebSphere Application Server RAS tracing levels, and then give the user the opportunity to take whatever actions are necessary to reproduce the problem. Examples of problems that fall into this category include an end user's being unable to login to the portal, and problems related to configuring access control for WebSphere Portal. For these types of problems, the AutoPD tool will set the tracing values to capture the appropriate information specific to the problem being encountered. In some cases, this may require the tool to stop and restart the WebSphere Portal. If this is necessary, the user will be informed that this step is going to occur. Once the WebSphere Portal has been restarted, the tool will pause to permit the user to reproduce the problem.

For WebSphere Application Server and for WebSphere Portal version 6, most collections scripts give you two options for changing the trace settings:

- If your application server is currently running, you can choose to have the runtime trace settings changed dynamically. In this case it is not necessary to stop and restart the application server. After the problem has been reproduced, the previous runtime trace settings will be restored.
- Whether or not your application server is currently running, you can choose to have the startup trace settings changed. In this case, the application server will be stopped and restarted, at which point the new startup trace settings will take effect. After the problem has been reproduced, the results depend on whether the application server was running when the collection script was invoked:
  - If the application server was running when the script was invoked, then the application server will be left in the running state, and both the runtime and startup trace settings will be restored to their previous values.
  - If the application server was not running when the script was invoked, then the application server will be stopped, and the startup trace settings will be restored to their previous values. The one exception to this rule is the case where you initially requested that the trace levels be reset without recycling the application server, but then allowed the tool to recycle it when you were informed that it was not currently started. In this case, the trace settings are still restored to their previous values, but the collection leaves the application server in the started state, on the theory that this is the state you had believed it was in. If you need the application server to be returned to the stopped state, you can do this manually after the collection has completed.

After the problem has been reproduced, the user informs the tool that this has been done, and the tool then proceeds with collecting all the necessary trace and system information, performing its analysis, and FTPing the collected data to IBM Support.

Combined Basic Log Collection / Problem Reproduction Scenario

The combined basic log collection / problem reproduction scenario combines the basic log collection scenario with the problem reproduction scenario. It is used for problem types where reproducing the problem (with its attendant disruptions to normal operations) will provide useful diagnostic data, but where doing basic collection and analysis of existing log records may also prove valuable. The choice of which way to proceed is left up to the user.

System Environment Modification Scenario

The goal of the system environment modification scenario is to leave the system in a different state after the script has finished, from the one it was in when the script started. It is used, for example, to configure the system to capture a core dump the next time a crash occurs. Unlike the problem reproduction scenario, this scenario makes no provision for saving the prior state of the system, and restoring it before the collection script terminates.

Problem Analysis Scenario

In the problem analysis scenario, only the symptom-analysis capabilities of the AutoPD tool are used. When the tool is used in this way, the user must have already assembled in a local directory the set of log files to be analyzed. The user is prompted for the name of this directory, as well as for some parameters that identify the exact type of analysis required. When the tool finishes, the results of the analysis are available locally in an analysis report document.

Product Data Collection Scenario

The product data collection scenario does not involve a problem at all. The AutoPD tool is instead used to assemble system information, as well as information about the IBM HTTP Server, WebSphere Application Server, WebSphere Portal, and other IBM software products installed on the system, and to package this information into an easy-to-read analysis report document that is saved locally. This collected information can be used to record and determine the exact environment in which an instance of WebSphere Portal is running.

User-Provided Trace Strings in the Problem Reproduction Scenarios

Each collection script for a problem reproduction or combined basic log collection / problem reproduction scenario refers to a trace string to set up the appropriate tracing for the script’s particular problem type. When the problem is recreated with the collection’s trace string applied, the necessary diagnosis data will ordinarily be captured. Sometimes, however, additional tracing, or simply different tracing, must be enabled in order to diagnose the problem.

The flexibility to perform a collection using a trace string different from its customary one is provided through a user dialog that appears prior to the problem-reproduction step. This dialog invites you to supply a custom trace string. You must also indicate whether this trace string should be appended to the collection’s trace string or replace it completely. These custom trace strings will most often be supplied to you by IBM Support, based on their analysis of your particular problem.

As always, the collection script will restore all product trace settings to their previous values prior to ending the collection. Thus the custom trace settings will only be active for the time when the problem is being reproduced.

Troubleshooting and Recovery

In the problem reproduction scenario (and optionally in the combined basic log collection / problem reproduction scenario), the AutoPD tool automates the process of enabling log tracing for WebSphere Application Server and WebSphere Portal, permits the user to reproduce the problem, and then collects the required information from these products. This process requires the tool to modify some properties files to enable log tracing, and to create new log trace files. In some rare situations, the AutoPD tool itself might fail for some reason. If this occurs, the tool has an auto-recovery feature to restore all the files it modified to the state they were in before it began. If the auto-recovery feature does not function properly, it may be necessary to recover the modified files manually. Below we describe how to perform a manual recovery.

To describe the recovery process we need to introduce some environment variables used by the AutoPD tool. See **AutoPD Logging** for a discussion of how you can change the default locations of the two directories identified here as RasGUI_LOG and RasGUI_RECOVERY.

- WAS_HOME: WebSphere Application Server installation directory
- WPS_HOME: WebSphere Portal installation directory
- RasGUI_HOME: the AutoPD tool installation directory: WPS_HOME/RasGUI
- RasGUI_LOG: the AutoPD tool's log directory: by default, RasGUI_HOME/log
- RasGUI_RECOVERY: the AutoPD tool's recovery directory: by default, RasGUI_HOME/recovery
- PMRFILENAME: the data collection zip file name

Note: You will need to use '\' instead of '/' as the directory-path separator for Windows, and denote variables as %VARIABLE% instead of as $VARIABLE.

Recovery Procedures

1. Examine the latest instance logs in RasGUI_LOG, or the AutoPD collection progress window in the tool's GUI, to determine if there has been a failure.

2. For usage scenarios other than the problem reproduction and combined basic log collection / problem reproduction scenarios, recovery is never needed because no changes were made to the WebSphere Application Server or WebSphere Portal configuration.

3. For collections that use the problem reproduction or combined basic log collection / problem reproduction scenario, if the failure happens before the message "The autopdNeedRecovery property has been set" has appeared in the progress window, then recovery is not needed because no changes have yet been made to the WebSphere Application Server or WebSphere Portal configuration.

4. For collections that use the problem reproduction or combined basic log collection / problem reproduction scenario, if the failure happens after the message "The autopdRecoveryPerformed property has been set" has appeared in the progress window, then recovery is not needed because the AutoPD tool has already completed it.

5. For collections that use the problem reproduction or combined basic log collection / problem reproduction scenario, if the failure happens after the message "The autopdNeedRecovery property has been set" has appeared in the progress window, but before the message "The autopdRecoveryPerformed property has been set" has appeared there, then manual recovery is needed. You can recover the files from backup files located in RasGUI_RECOVERY for different problems. For each problem type, the tool creates a subdirectory in the RasGUI_RECOVERY directory. The subdirectory is named with the naming convention: wps_timestamp (e.g., wps_2004.12.10-01.29.48.508EST). For convenience, RasGUI_RECOVERY_WITH_TIMESTAMP is used to represent the directory below. Its name can be found from the latest autopdechoinstance log under the message "AutoPD recovery dir with timestamp" for a problem scenario. The original WebSphere Application Server or WebSphere Portal log files are backed up under the directory RasGUI_RECOVERY/RasGUI_RECOVERY_WITH_TIMESTAMP/log. These can be used to manually restore the WebSphere Application Server or WebSphere Portal log files back to the original log directory if necessary. If the collection process is completed successfully, the new WebSphere Portal log files are stored in WPS_HOME/log/RasGUI/log_timestamp and the original log files are returned to where they were originally located.

   - Portal Access Control, Portal Login, Portal Manage Users and Groups, and Portal Integration with Tivoli Access Manager problem types:

    - Copy log.properties under RasGUI_RECOVERY/RasGUI_RECOVERY_WITH_TIMESTAMP to WPS_HOME /shared/app/config
    - Copy server.xml under RasGUI_RECOVERY/RasGUI_RECOVERY_WITH_TIMESTAMP to WAS_HOME/config/cells/{cell}/nodes/{node}/servers/ {server}

  - Portlet Deployment, Portal Config, Portal General, and Portal XML Configuration Interface problem types:
    - Copy log.properties under RasGUI_RECOVERY/RasGUI_RECOVERY_WITH_TIMESTAMP to WPS_HOME /shared/app/config

  - WebSphere Application Server General, WebSphere Application Server JMS, WebSphere Application Server ORB, WebSphere Application Server Database Connection, WebSphere Application Server Security, WebSphere Application Server Start, WebSphere Application Server Stop, WebSphere Application Server Synchronization on Deployment Manager, WebSphere Application Server Synchronization on Node Agent, WebSphere Application Server Session Management, WebSphere Application Server Out of Memory Problem, WebSphere Application Server Memory Heap Leak Problem, WebSphere Application Server Memory Native Leak Problem, and Auxiliary Websphere Application Server Collection for IBM HTTP Server problem types:
    - Restore user default trace values through wsadmin (For Websphere Application Server problem types, trace setting levels and other trace related values, such as the maximum trace log file size, are modified through wsadmin commands. During recovery, these settings are set back to their original values through wsadmin, the same way they were set.)

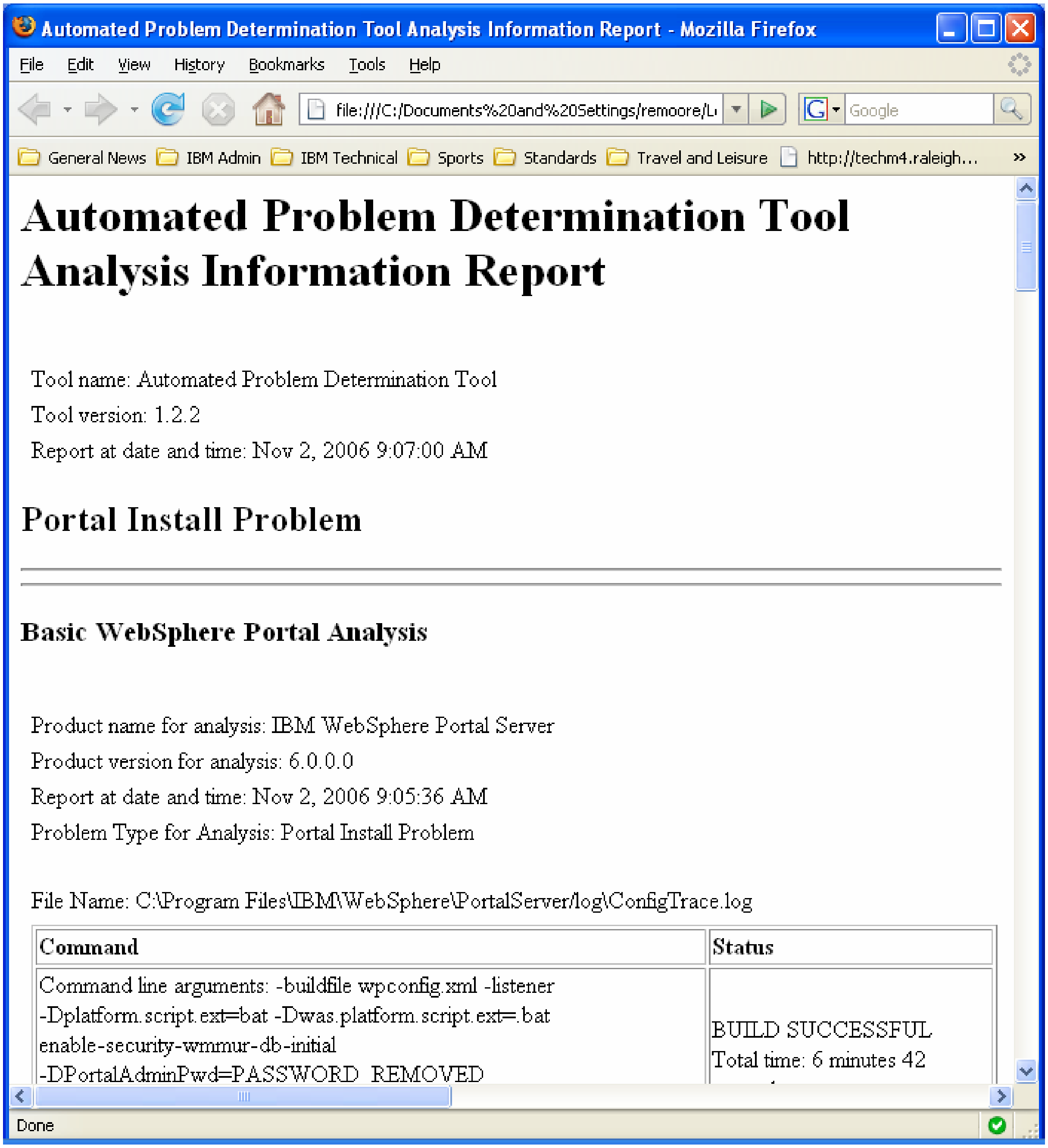
6. If the failure happened in one of the steps of the FTP process, manually FTP the zip file PMRFILENAME to the desired FTP site.

7. If you still have questions, you can send the entire RasGUI_LOG directory to IBM Support for additional help.

Symptom Analysis and System Level Reporting

Once the tool has completed a collection activity, the data collection zip file will contain all of the information collected for that activity. The zip file will also contain an ***analysis_report.html*** file, which has the symptom analysis results as well as system information such as the amount of memory on the machine, number of CPUs, OS type, and the various types and levels of IBM software that are installed on the machine. Figure 11 shows the initial part of an analysis_report.html file for a Portal Install problem.

Please note that some zip tools are not able to properly “view” the zip file created by the AutoPD tool: the symptom is that the zip file appears to be practically empty. If your zip tool provides such a view, you should still try to extract the contents to an empty directory. Then you can verify that all the files that should have been collected actually were collected.



Automated Problem Determination Tool Analysis Information Report

Tool name: Automated Problem Determination Tool

Tool version: 1.2.2

Report at date and time: Nov 2, 2006 9:07:00 AM

Portal Install Problem

Basic WebSphere Portal Analysis

Product name for analysis: IBM WebSphere Portal Server

Product version for analysis: 6.0.0.0

Report at date and time: Nov 2, 2006 9:05:36 AM

Problem Type for Analysis: Portal Install Problem

File Name: C:\Program Files\IBM\WebSphere\PortalServer/log\ConfigTrace.log

Command	Status
Command line arguments: -buildfile wpconfig.xml -listener -Dplatform.script.ext=bat -Dwas.platform.script.ext=.bat enable-security-wmmur-db-initial -DPortalAdminPwd=PASSWORD_REMOVED	BUILD SUCCESSFUL Total time: 6 minutes 42

Figure 11

Adding New Symptoms

The AutoPD tool uses a symptom database XML file to guide its symptom data analysis. This file employs two different approaches to identify the information the tool needs to extract from the configuration and log files it collects for different problems and different products:

- For files that are not XML documents, the files uses regular expression patterns to identify the information.

- For files that are XML documents, the file uses a different approach to identify the XML fields (element text and/or attribute values) of interest, and then uses regular expression patterns to match the values of these XML fields.

The symptom database file also provides guidance on how to display the analysis results.

You can use any XML editor tool to create your own symptom database XML file; detailed instructions for doing this appear below. A schema file pattern_template.xsd is included with the tool, to provide a formal description of a symptom database XML file. If your XML tools support validation of an XML document against its schema, you can use this schema file to check your symptom database XML file to make sure it's valid.

By default, the AutoPD tool uses the symptom database XML file pattern_template.xml, located in the RasGUI_HOME/properties/wps directory, when doing symptom analysis for the WebSphere Portal product.

Control of symptom analysis can be shifted to a different file by changing the value of the `patternFile` attribute of the `<analyze_files>` task in an Ant script. This change does not have to be global. Different `<analyze_files>` elements can point to different symptom database XML files, by assigning different values to this attribute.

```
<analyze_files problem="${collect_wps_information_common_ProblemType}"
      patternFile="${bundle.basedir}/properties/wps/pattern_template.xml"
      productname="${wps.product.name}"
      productversion="${wps.product.version}"
      includeType="inline"
      fragmentTitle="Basic WebSphere Portal Analysis"
      timeout="${infocollector.timeout}">
      <autopdfileset filesetName="wpslog"
            filesetDir="${portal.latest.file}" />
</analyze_files>
```

To maintain backward compatibility with previously written collection scripts, version 1.2.2.1 of the AutoPD tool continues to support invoking analysis via the original `<infocollect>` task. Just as with the **`<analyze_files>`** task, control of symptom analysis can be shifted to a different file by changing the value of the `patternFile` attribute of the `<infocollect>` `task` in an Ant script. Again, this change does not have to be global. Different `<infocollect>` elements can point to different symptom database XML files, by assigning different values to this attribute.

```
<infocollect problem="portalinfo"
      patternFile="${autopdproperties}/wps/pattern_template.xml"
      levelreport="${autopdtmp}/autopd/levelreport.html"
      autopdreport="${autopdtmp}/autopd/autopd_analysis_report.html"
      productname="${product.name}" productversion="${product.version}"
      autopdname="${autopd_name}" autopdversion="${autopd_version}" >
      <autopdfileset
            filesetName="wpslog"
            filesetDir="${portal.root}/log"
            filterByLatestTime="true" />
</infocollect>
```

Problem Description

A problem element contains instructions for obtaining symptom data information for single type of problem that might be encountered in WebSphere Portal or in another product. It contains a name and description as attributes.

For example, in the listing below, a `<problem>` element is created for the Portal Login problem with the name “portallogin” and the description “Portal Login Problem”. The description text is used in the analysis report.

```
<extractors>
      <problem name="portallogin" description="Portal Login Problem">
      </problem>
</extractors>
```

`<problem>` elements can have multiple `<analysisProfile>` elements as child nodes. Each `<analysisProfile>` element associates a set of symptoms with some specific set of products and/or product versions. For example, the XML fragment below contains an `<analysisProfile>` named `wps_portal_5.1`**.** The name itself is arbitrary, although this value suggests that the profile applies to one or more WebSphere Portal versions in the 5.1.x.x range. A separate element in the symptom database XML file, `<productInfo>`, identifies exactly which versions of WebSphere Portal will use this profile. A more detailed description of the `<productInfo>` element appears in the **Product Information** subsection below.

```
      <problem name="portallogin" description="Portal Login Problem">
         <analysisProfile name="wps_portal_5.1" >
         </analysisProfile>
      </problem>
```

An `<analysisProfile>` element contains one or more `<fileset>` elements as child elements. These `<fileset>` elements, which identify the logs that need to be analyzed for a particular problem type, are described in the next section.

Fileset Element

A `<fileset>` element, which is a child of the `<analysisProfile>` element, is used to describe the group of files on which the tool is to perform symptom analysis for a given type of problem. It contains a name attribute to identify the file set, and a value attribute. The value attribute is a regular expression pattern identifying the file names to match. A `<delimiterid>` element provides a reference either to a `<delimiter>` element or to an `<xmlDelimiter>` element. Each `<delimiter>` or `<xmlDelimiter>` element describes the matching information to use for any file set that refers to it. The delimiters are more thoroughly described in the **Delimiter and Symptom Matching Information** subsection of this document.

For example, the value of the `<fileset>` name attribute in the example listed below is *wpslog*. This value corresponds to the fileset name found in the `<autopdfileset>` child element of the `<analyze_files>` or `<infocollect>` element in the collection Ant script. Inside this `<autopdfileset>` element, the filesetDir attribute identifies the directory where the tool should search for the files on which to perform the analysis. The `<fileset>` element here contains a value attribute with the regular expression *(wps[_.].*)* as its value; this regular expression identifies filenames starting with either *`wps.`* or *`wps_`*.

```
      <problem name="portallogin" description="Portal Login Problem">
         <analysisProfile name="wps_portal_5.1" >
```

```
                <fileset name="wpslog" value="(wps[_.].*)">
                      <delimiterid id="delimiter1" />
            </fileset>
        </analysisProfile>
    </problem>
```

In some cases, a product such as WebSphere Portal may save multiple, timestamped versions of the same basic log file in a common directory. It is natural for a `<fileset>` element to be defined in such a way that it matches all of these timestamped files, even though the user would prefer for the tool to perform symptom analysis only on the latest of them. To deal with this situation, the `<autopdfileset>` task supports an attribute `filterByLatestTime` that tells the tool whether to analyze all of the files in a fileset, or only the latest one:

```
<autopdfileset
            filesetName="wpslog"
            filesetDir="${portal.root}/log"
            filterByLatestTime="true" />
```

Note that if this attribute is set to `true`, the "latest file" test is applied to all of the files in the fileset. This may not produce the desired result. Suppose, for example, that the `${portal.root}/log` directory contained the following log files:

wpsconfig.log
wps2005-05-05.log
wps2005-05-04.log
wps2005-05-03.log
wps2005-05-02.log

The `<fileset>` value `"(wps[_.].*)"` would place all five of these files into the fileset initially, so that selecting the latest one for analysis would choose either wpsconfig.log or wps2005-05-05.log. In order to have both wpsconfig.log and the latest of the timestamped log files analyzed, it is necessary to place wpsconfig.log into one fileset, and all of the timestamped log files into a second fileset.

Product Information

A `<productinfo>` element is used to describe the product information for a certain product. It is always a child element of the `<extractors>` element, which is the root element in a symptom database XML document. The AutoPD tool uses the information in this element, together with the name and version of the actual product it is dealing with, to determine which analysis profile to use.

The element contains name and version attributes, whose values are matched against those of the actual product for which analysis is being performed. The Ant script is responsible for setting two Ant properties: *product.name* and *product.version.* These two property values are then compared to the name and version attribute values in all the `<productinfo>` elements in the symptom database document, looking for the best match. For product name, a case-insensitive exact match is performed between the values of the *product.name* property and the `<productinfo>` element's name attribute. For version, the tool performs a longest prefix match between the values of the *product.version* property and the `<productinfo>` elements' version attributes, for all `<productinfo>` elements that matched on product name. Because version matching is based on a longest prefix match, a single `<productinfo>` element can match multiple minor product versions, so that analysis will not be disabled when a product is updated to a later version.

The AutoPD tool provides a custom Ant task `<wpsversiontask>` to set the *product.name* and *product.version* properties for WebSphere Portal. This custom task extracts the product name and version values from a properties file, assuming these values are stored using product and version key value pairs such as the following:

```
# Sample Product Information properties file that can be read by wpsversiontask
product=IBM WebSphere Portal Server
version=5.0.2.1
```

Please see the `wps_common_setup` target in the portal-sharedtargets.xml document that resides in the RasGUI_HOME/scripts/wps directory for an example of how to use the `<wpsversiontask>`. If you are doing symptom analysis for a product not currently supported by the AutoPD tool, and if this product does not have a properties file that `<wpsversiontask>` can understand, then you can just use standard Ant tasks in your script to set the *product.name* and *product.version* property values.

For example, the `<productinfo>` element shown below represents WebSphere Portal version 5.1.0.1, and instructs the tool to use the analysis profile wps_portal_5.1 when it performs symptom analysis for this version of WebSphere Portal.

```
<extractors>
 <productinfo analysisProfileRef="wps_portal_5.1"
                        name="IBM WebSphere Portal"
                        version="5.1.0.1">
      </productinfo>
 </extractors>
```

Output Format

An `<output>` XML element is used to describe the display format of the analysis report. It contains a format attribute to indicate the type of output – currently HTML is the only output type supported by the tool. An `<output>` element has one or more child `<format>` elements, each of which describes a display order for a set of selected information groups. A `<format>` element contains name and order attributes, and optionally one or more `<msgCatalogLookup>` child elements.

The `<format>` element's name attribute allows us to reference this defined output format when a symptom match occurs. Use of the output format element in this fashion is described in the **Delimiter and Symptom Matching Information** subsection below.

The order attribute's value contains a list of group names separated by |'s. These groups represent the only information to be used in this output format. The group information is defined in the **Delimiter and Symptom Matching Information** subsection below.

The optional `<msgCatalogLookup>` elements identify the information groups for which message catalog lookup should be performed. See the **Message Catalog Lookup** subsection below for a description of this function.

In the example shown below the output format used is HTML. Furthermore, the output element contains one child format element with the name attribute value “format1”, and the order attribute value “timestamp|errorID|description”. The format1 is used in the **Delimiter and Symptom Matching Information** section that follows. Timestamp, errorID and description are also defined in this following subsection as well. Finally, the element instructs the tool to provide message catalog lookup for the two information groups `errorID` and `description`.

```
<extractors>
   <output format="html">
      <format name="format1" order="timestamp|errorID|description" >
            <msgCatalogLookup group="errorID" />
            <msgCatalogLookup group="description" />
      </format>
   </output>
```

```
</extractors>
```

It is also possible to subset the information included in the analysis report for a particular group, for example, to exclude detailed stack trace information from the description group. The details of how to do this are described below in the subsection **Controlling the Information Reported for a Group**.

Delimiter and Symptom Matching Information

The goal of the AutoPD analysis function is easy to describe: extract “interesting” log records from a log file (typically, a very large log file), and present relevant information from each selected log record in an easy-to-read format. This goal, however, is achieved in very different ways, depending on whether the log file being analyzed is an XML document or simply a text file with non-XML formatting. The following two sections describe how to specify, in each of these cases, what makes a log record interesting, which information from an interesting log record is relevant, and how this relevant information should be presented in the analysis report.

Log Files that are not XML Documents

A `<delimiter>` XML element is used to describe the pattern that separates individual log records from each other. Since some log records can be multi-line, one cannot assume that end of a line ends a log record. Thus, it is necessary to explicitly define the initial substring that should be used to identify the start of a new log record. The `<delimiter>` element contains an `id` attribute to identify the delimiter pattern, so that it may be referenced from inside multiple `<fileset>` elements. Using the `id` attribute in this way is more thoroughly discussed in the **Fileset Element** subsection of this document. The `value` attribute contains a regular expression describing the pattern to be used to identify the start of a log record.

In the following example, the delimiter’s **`id`** attribute is **`delimiter1`**. The **`value`** attribute is a regular expression pattern that matches the timestamps found at the beginning of each log record in the portal wps*.log log file. An example of a pattern the regular expression below would match is 2004.09.17 16:07:05.386.

```
<extractors>
<delimiter id ="delimiter1" value="([0-9]{1,4}\.[0-9]{1,2}\.
      [0-9]{1,2}\ [0-9]{1,2}:[0-9]{1,2}:[0-9]{1,2}\.[0-9]{1,3})"
</delimiter>
</extractors>
```

Once a `<delimiter>` element identified an individual log record, its child `<pattern` **>** elements are applied to that log record. A `<pattern>` element does two things: it identifies which log records should be included in the analysis report, and it identifies the individual pieces of information the tool should extract from the log records that were selected for inclusion in the report. Both of these ideas are expressed in a single attribute: the `<pattern` **>** element’s `value` attribute.

The `value` attribute contains a regular expression pattern describing the text content to match to identify this symptom. This same regular expression also provides a mechanism whereby substrings matching the pattern can be easily broken into individual subcomponents by wrapping these subcomponents in parentheses. Consider, for example, the following pattern element:

```
<pattern value="([0-9]{1,4}\.[0-9]{1,2}\.[0-9]{1,2}\ [0-9]{1,2}:
 [0-9]{1,2}:[0-9]{1,2}\.[0-9]{1,3})(.*?)([A-Z]{4,5}[0-9]{1,4}[EW]):(.*)" >
```

This element has a very long regular expression that it uses to match a symptom. Since all the log records in the set of log files for which this pattern is used have timestamps of the indicated form, they all match most of the regular expression. The key for matching only selected log records in the log file lies in the subpattern `[A-Z]{4,5}[0-9]{1,4}[EW]`. This subpattern matches only those log records containing substrings of the form (4 or 5 characters) + (1 to 4 digits) + either an ‘E’ or a ‘W’ – in other words, all WebSphere Portal log records containing either an Error message or a Warning message.

Notice also that there are four subcomponents of this regular expression that are wrapped in parentheses. To make it easier to read, the four sections wrapped in parentheses have been alternately marked in either bold case or underlined case. The first set of parentheses matches the timestamp contained in a log record, such as “2004.10.01 11:34:21.095”. The second set of parentheses, marked in underline case, matches all content that follows the timestamp, up to but not including the Error or Warning code discussed above. The third set of parentheses, marked in bold case, matches this Error or Warning code. The fourth set of parentheses matches all information that follows the colon that comes right after the previously matched error code.

For each piece of information captured by a set of parentheses, the tool can be instructed to extract that piece of information using the `<group>` element. For example, the following `<pattern>` element has `<group>` elements instructing the tool to extract the timestamp and error code described above.

```
<delimiter id ="delimiter1" value="([0-9]{1,4}\.[0-9]{1,2}\.[0-9]{1,2}\
 [0-9]{1,2}:[0-9]{1,2}:[0-9]{1,2}\.[0-9]{1,3})" timeRangeApplies="false" >
       <pattern value="([0-9]{1,4}\.[0-9]{1,2}\.[0-9]{1,2}\
            [0-9]{1,2}:[0-9]{1,2}:[0-9]{1,2}\.[0-9]{1,3})(.*?)
            ([A-Z]{4,5}[0-9]{1,4}[EW]):(.*)" >
            <group name="timestamp" number="1" />
            <group name="errorID" number="3" />
            <group name="description" number="0" />
            <formatref outputName="html-1" formatName="format1" />
       </pattern>
</delimiter>
```

The `<group>` elements also associate names to the sets of parentheses that are extracting the strings we are interested in. For example, the first set of parentheses, identified by the number attribute that has a value of “1”, is identified by the name attribute of “timestamp” in the first `<group>` element. Similarly, the error code captured by the third set of parentheses is identified by the number attribute with a value of “3”, and this has been associated with the name attribute of “errorID” in the second group element. The third `<group>` element has a number attribute with a value of “0”, which is a default for extracting the complete string identified by the regular expression in the pattern element.

The `<formatref>` element here contains a two-part reference to a `<format>` element, identifying first the `<output>` element containing the `<format>` element, and then the `<format>` element itself. The `<format>` element, which is described in the **Output Format** section of this document, specifies how the tool should format the information extracted by a group element that refers to it.

Log Files that are XML Documents

The AutoPD tool’s initial steps in interpreting its analysis instructions are identical for XML and non-XML log records, down to the point where a particular delimiter is identified by a `<delimiterid>` element. In the case of an XML-formatted log file, however, the `<delimiterid>` points to an `<xmlDelimiter>` element, rather than to a `<delimiter>` element. It is this `<xmlDelimiter>` element that answers the four questions identified above: what constitutes a log record, what makes a log record interesting, which parts of an interesting log record should be included in the analysis report, and how should these parts of the log record be formatted in the report? Here is an example of an `<xmlDelimiter>` element:

```
<xmlDelimiter id="eventhistorydelim" documentXsd="event-history">
       <xmlLogRecord elementType="update-event" level="top"/>
       <xmlMatch matchAll="true"/>
       <xmlGroups>
           <xmlGroup xmlTarget="updateEventId" displayName="Fix" />
           <xmlGroup xmlTarget="updateEventResult" displayName="Status" />
           <xmlGroup xmlTarget="updateAction" displayName="Action" />
           <xmlGroup xmlTarget="updateEventStartTimestamp"
                      displayName="Date/Time" />
           <formatref outputName="html-1" formatName="format-eh1" />
        </xmlGroups>
</xmlDelimiter>
```

In the case of a log file that is an XML document, it is not necessary to use a regular expression to separate out the individual log records in the file: instead, you simply identify which XML element represents a log record, with the `elementType` attribute of the `<xmlLogRecord>` element. The `level` attribute supports two values that complete the specification of what constitutes a log record: “top” indicates that only top-level elements of the indicated type qualify, and “all” indicates that nested elements of this type qualify as well.

The `<xmlMatch>` element identifies which log records qualify as interesting. The example shown above is something of a special case, since it says that all log records identified by the `<xmlLogRecord>` element are interesting. If the `matchAll` attribute is set to false (or allowed to default to false), then the `<xmlMatch>` element can specify detailed matching criteria involving ANDs, ORs, and NOTs. Here, for example, is an `<xmlMatch>` element for CBEs, identifying CBEs having both a severity > 10 and a situation category other than ReportSituation:

```
<xmlMatch>
      <xmlNumericAtom xmlTarget="cbeSeverity"
                 comparisonOperator="greaterThan" >10</xmlNumericAtom>
      <xmlStringAtom xmlTarget="cbeSituationCategory"
                 negated="true">ReportSituation</xmlStringAtom>
</xmlMatch>
```

For more details on `<xmlMatch>` and its child elements, see the section **Symptom Schema** below.

The `<xmlGroup>` element in an `<xmlDelimiter>` plays basically the same role as that played by the `<group>` element in a regular delimiter: identifying which fields from the selected log records to include in the analysis report, and what to call them in the report. The `<formatref>` element works in exactly the same way that it does in a regular delimiter.

There is one additional point to understand about the `<xmlDelimiter>` element: the role of the `xmlTarget` attribute. This attribute identifies a “piece” of a log record, either for comparison in an `<xmlMatch>` context, or for extraction in an `<xmlGroup>` context. In both of these cases, the value of the `xmlTarget` attribute is chosen from a small number of specific values built into the AutoPD tool itself. These values have been chosen for ease of use in building `<xmlDelimiter>` elements, not for generality. They thus represent a very different design decision from that behind a mechanism such as XPath.

This point is easily illustrated using the CBE example above. When the AutoPD tool is told to find the `xmlTarget` named “cbeSeverity”, it invokes logic that returns the value of the CBE’s “severity” attribute. However, it does something very different when it is told to find the `xmlTarget` named “cbeSituationCategory”: it finds a child element <situation> of the CBE, and then returns the value of that <situation> element’s “categoryName” attribute. The difference between these two cases is hidden from the symptom author, who can simply think “severity greater than 10 and situation category other than ReportSituation.”

Message Catalog Lookup

Some IBM products, including WebSphere Portal, provide message catalogs containing additional information for some or all of the error codes the products write to their log files. The AutoPD tool includes message catalogs for the 5.1.0.0 and 5.1.0.1 releases of WebSphere Portal, so that this additional information can be linked to the occurrences of the error codes in the analysis report. When it is instructed to do so for an information group in the analysis report, the tool first scans the text for that group for occurrences of message IDs of the right format, and then uses any suitable IDs that it finds as keys for lookups in the message catalog for the version of WebSphere Portal it is dealing with. If any of these lookups succeed, it copies the relevant information from the message catalog into an HTML file that it builds, and links the occurrence of the message ID in the analysis report to the corresponding information in the HTML file. This HTML file is included in the collection zip file along with the main analysis report: when the user unzips the two files to the same directory, the links from the message IDs in the analysis report to the additional information in the HTML file will work.

As described above in the section **Output Format**, the `<format>` element's child element `<msgCatalogLookup>` identifies the information groups for which the tool is to perform message catalog lookup. Detailed instructions on how to perform it, however, may vary between different products and product releases, so these detailed instructions are associated with the `<productinfo>` element. Here is an example:

```
<productinfo analysisProfileRef="wps_portal_5.1"
             name="IBM WebSphere Portal"
             version="5.1.0.0">
             <msgPrefixMappingInfo
                  catalogDir="portal_message_catalog_510"
                  fileName="wps5100msgprefixmapping.properties"
                  mappingType="PortalCatalogMessage"
                  keyPattern="([A-Z]{4,5}[0-9]{1,4}[EW])"/>
</productinfo>
```

The `<productinfo>` element was discussed above in the section **Product Information**. Under this element, the `<msgPrefixMappingInfo>` element tells the tool exactly how to do message catalog mapping for any product that matches that `<productinfo>` element. These detailed instructions include the directory where the product's message files reside (under the tool's /messagecatalog directory), the name of the message catalog file for the product (located in the tool's /properties directory), the type of mapping the file supports, and the pattern the tool should use when scanning for potential message keys in the analysis report text for any selected information groups.

By default, the tool retrieves the English-language version of the message catalog information for the HTML file that it builds. The message catalogs themselves, however, support a number of additional languages. To have the tool build the HTML file using a different language, you must do two things:

- Because of the large number of messages involves and the large number of languages supported, the full message catalogs are not included in the initial download of the tool described earlier under **Tool Installation**. (The English-language files are included in the initial download, so it is not necessary to get the additional downloads described here unless the language for the message text is being set to something other than `en`.) There are separate downloads for each of the message catalogs that the tool supports: RasGUI_Portal_Msg_Catalog5100.zip and RasGUI_Portal_Msg_Catalog5100.tar for WebSphere Portal v5.1.0.0, and RasGUI_Portal_Msg_Catalog5101.zip and RasGUI_Portal_Msg_Catalog5101.tar for WebSphere Portal v5.1.0.1. After one or both of the .zip or .tar files has been retrieved for an environment where the tool has already been installed, they should be extracted to the tool's RasGUI directory, which was created earlier during the tool's initial installation.

- The second step is to edit the autopd.properties file (located in the tool's /properties directory), and change the value of the `catalogMappingLang` property from `en` to one of the other values supported by the message catalogs. These values are listed in the following table.

`ar`	Arabic
`cs`	Czech -- file strings are still English
`da`	Danish
`de`	German
`el`	Greek, modern -- file strings are still English
`en`	English (AutoPD tool default)
`es`	Spanish
`fi`	Finnish
`fr`	French
`hu`	Hungarian
`it`	Italian
`iw`	Hebrew -- file strings are still English
`ja`	Japanese
`ko`	Korean
`nl`	Dutch
`no`	Norwegian
`pl`	Polish
`pt`	Portuguese
`pt_BR`	Brazilian Portuguese
`ro`	Romanian
`ru`	Russian
`sv`	Swedish
`th`	Thai
`tr`	Turkish
`uk`	Ukrainian
`zh`	Simplified Chinese
`zh_TW`	Traditional Chinese

Controlling the Information Reported for a Group

Sometimes it is convenient to include in the analysis report only a subset of the information that the regular expression selects for a group. One common example of this is the often lengthy stack trace information included in the groups identified by a number attribute value of 0, which represents the entire strings selected by their regular expressions. The **`<group>`** and **`<xmlGroup>`** elements contain four attributes to specify such subsetting, based on the substring methods in the java.lang.String class:

- beginIndex
- beginPattern
- endIndex
- endPattern

It is not necessary to specify these attributes at all for a group, in which case all of the information for the group is included in the analysis report. If subsetting of the information for a group is desired, only one of the "begin" attributes and/or one of the "end" attributes should be specified. The following examples illustrate how the subsetting works:

initial string: `dogs_and_cats`		
attributes	**result**	**comments**

beginIndex = “3”
endIndex = “10”
	`s_and_c`	String index starts at 0

beginIndex=”3”	`s_and_cats`	
endIndex = “10”	`dogs_and_c`	different from java.lang.String, since XML attributes allow for end delimiter without a begin delimiter

beginIndex=”3”
endPattern=”ts”
	`dogs_and_ca`	OK to mix index and pattern attributes

beginIndex=”1”
endPattern=”s”
	`og`	match on first ‘s’

beginIndex=”3”
endPattern=”birds”
	`s_and_cats`	endPattern not matched, so no subsetting at the end of the string

beginIndex=”8”
endIndex=”3”
	`dogs_and_cats`	endIndex < beginIndex: for all errors, include the entire string

beginIndex=”20”
endIndex=”25”
	`dogs_and_cats`	indexes out of range: for all errors, include the entire string

beginIndex=”3”
endIndex=”25”
	`dogs_and_cats`	endIndex out of range: even though beginIndex is valid, the entire string is included because of the error in endIndex

The following line illustrates how to exclude stack traces from the information reported for the description group:

```
<group name="description" number="0" endPattern="at com.ibm" />
```

Symptom Database XML File Example

The following example illustrates how all the symptom elements can be utilized to create a sample symptom database XML file. The excerpt included here is only a small subset of the symptom database file pattern_template.xml included in the AutoPD tool.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- (C) COPYRIGHT International Business Machines Corp., 2004-2005. All Rights
Reserved * Licensed Materials - Property of IBM -->
<extractors xmlns="http://www.ibm.com/autopd/pattern"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.ibm.com/autopd/pattern pattern_template.xsd ">

<problem name="cbeanalysis" description="CBE Analysis">
   <analysisProfile name="wps_portal_5.1" >
       <fileset name="cbelog" value="(.*CBE.*)" >
             <delimiterid id="cd1" type="XML" />
       </fileset>
   </analysisProfile>
</problem>

<problem name="portallogin" description="Portal Login Problem">
   <analysisProfile name="wps_portal_5.1" >
       <fileset name="wpslog" value="(wps[_.].*)" >
             <delimiterid id="delimiter1" />
       </fileset>
       <fileset name="tracelog" value="(trace.*)"  >
             <delimiterid id="delimiter2" />
       </fileset>
       <fileset name="systemoutlog" value="SystemOut.*\.log"  >
             <delimiterid id="delimiter4" />
       </fileset>
       <fileset name="systemerrlog" value="SystemErr.*\.log"  >
             <delimiterid id="delimiter4" />
       </fileset>
       <fileset name="configtrace" value="ConfigTrace\.log"  >
             <delimiterid id="configtracedelim" />
       </fileset>
       <fileset name="eventhistory" value="(event\.history.*)"  >
             <delimiterid id="eventhistorydelim" type="XML"/>
       </fileset>
   </analysisProfile>
   <analysisProfile name="wps_portal_5.0" >
       <fileset name="wpslog" value="(wps[_.].*)" >
             <delimiterid id="delimiter1" />
       </fileset>
       <fileset name="tracelog" value="(trace.*)"  >
             <delimiterid id="delimiter2" />
       </fileset>
       <fileset name="systemoutlog" value="SystemOut.*\.log"  >
             <delimiterid id="delimiter4" />
       </fileset>
       <fileset name="systemerrlog" value="SystemErr.*\.log"  >
             <delimiterid id="delimiter4" />
       </fileset>
       <fileset name="configtrace" value="ConfigTrace\.log"  >
             <delimiterid id="configtracedelim" />
       </fileset>
       <fileset name="eventhistory" value="(event\.history.*)"  >
             <delimiterid id="eventhistorydelim" type="XML"/>
       </fileset>
   </analysisProfile>
</problem>

<delimiter id ="delimiter1" value="([0-9]{1,4}\.[0-9]{1,2}\.[0-9]{1,2}\ [0-
9]{1,2}:[0-9]{1,2}:[0-9]{1,2}\.[0-9]{1,3})" timeRangeApplies="false" >
        <timestamp group="1" pattern="yyyy.MM.dd hh:mm:ss.SSS" />
        <pattern value="([0-9]{1,4}\.[0-9]{1,2}\.[0-9]{1,2}\ [0-9]{1,2}:[0-
9]{1,2}:[0-9]{1,2}\.[0-9]{1,3})(.*?)([A-Z]{4,5}[0-9]{1,4}[EW]):(.*)" >
```

```
              <group name="timestamp" number="1" />
              <group name="errorID" number="3" />
              <formatref outputName="html-1" formatName="format1" />
        </pattern>
</delimiter>
<delimiter id="delimiter2" value="\[(([0-9][0-9]?/[0-9][0-9]?/[0-9]{1,2}\ [0-
9]{1,2}:[0-9]{1,2}:[0-9]{1,2}:[0-9]{1,3}\ [A-Z]{1,3})\]" timeRangeApplies="false">
        <timestamp group="1" pattern="yyyy.MM.dd hh:mm:ss.SSS z" />
        <pattern value="\[([0-9][0-9]?/[0-9][0-9]?/[0-9]{1,2}\ [0-9]{1,2}:[0-
9]{1,2}:[0-9]{1,2}:[0-9]{1,3}\ [A-Z]{1,3})\](.*)Exception:(.*)" >
              <group name="timestamp" number="1" />
              <group name="exception" number="0" endPattern="at com.ibm" />
              <formatref outputName="html-1" formatName="format2" />
       </pattern>
        <pattern value="\[([0-9][0-9]?/[0-9][0-9]?/[0-9]{1,2}\ [0-9]{1,2}:[0-
9]{1,2}:[0-9]{1,2}:[0-9]{1,3}\ [A-Z]{1,3})\](.*)[Ff]ailed(.*)" >
              <group name="timestamp" number="1" />
              <group name="timestamp" number="1" />
              <group name="errorID" number="3" />
              <group name="exception" number="0" endPattern="at com.ibm" />
              <formatref outputName="html-1" formatName="format2" />
       </pattern>
</delimiter>
<delimiter id="delimiter3" value="\[(([0-9]{1,2}/[0-9]{1,2}/[0-9]{1,2}\ [0-
9]{1,2}:[0-9]{1,2}:[0-9]{1,2}:[0-9]{1,3}\ [A-Z]{1,3})\]" timeRangeApplies="false">
        <timestamp group="1" pattern="yyyy.MM.dd hh:mm:ss.SSS z" />
        <pattern value="\[([0-9]{1,2}/[0-9]{1,2}/[0-9]{1,2}\ [0-9]{1,2}:[0-
9]{1,2}:[0-9]{1,2}:[0-9]{1,3}\ [A-Z]{1,3})\](.*)Exception:(.*)" >
              <group name="timestamp" number="1" />
              <group name="exception" number="0" endPattern="at com.ibm" />
              <formatref outputName="html-1" formatName="format2" />
       </pattern>
</delimiter>
<delimiter id="delimiter4" value="\[(([0-9]{1,2}/[0-9]{1,2}/[0-9]{1,2}\ [0-
9]{1,2}:[0-9]{1,2}:[0-9]{1,2}:[0-9]{1,3}\ [A-Z]{1,3})\]" timeRangeApplies="false">
        <timestamp group="1" pattern="yyyy.MM.dd hh:mm:ss.SSS z" />
        <pattern value="\[([0-9]{1,2}/[0-9]{1,2}/[0-9]{1,2}\ [0-9]{1,2}:[0-
9]{1,2}:[0-9]{1,2}:[0-9]{1,3}\ [A-Z]{1,3})\](.*?)([A-Z]{4,5}[0-9]{1,4}[EW]):(.*)" >
              <group name="timestamp" number="1" />
              <group name="errorID" number="3" />

              <group name="description" number="0" endPattern="at com.ibm" />
              <formatref outputName="html-1" formatName="format1" />
       </pattern>
</delimiter>

<xmlDelimiter id="eventhistorydelim" documentXsd="event-history">
        <xmlLogRecord elementType="update-event" level="top"/>
        <xmlMatch matchAll="true"/>
        <xmlGroups>
            <xmlGroup xmlTarget="updateEventId" displayName="Fix" />
            <xmlGroup xmlTarget="updateEventResult" displayName="Status" />
            <xmlGroup xmlTarget="updateAction" displayName="Action" />
            <xmlGroup xmlTarget="updateEventStartTimestamp" displayName="Date/Time" />
            <formatref outputName="html-1" formatName="format-eh1" />
        </xmlGroups>
</xmlDelimiter>

<output name="html-1" format="html">
       <format name="format1" order="timestamp|errorID|description" >
              <msgCatalogLookup group="errorID" />
              <msgCatalogLookup group="description" />
       </format>
       <format name="format2" order="timestamp|exception" />
       <format name="config_trace_format" order="Command|Status" />
       <format name="format-c1" order="timestamp|situation category|CBE" />
       <format name="format-eh1" order="Fix|Status|Action|Date/Time" />
</output>

<productinfo analysisProfileRef="wps_portal_5.1"
              name="IBM WebSphere Portal"
              version="5.1.0.1">
       <msgPrefixMappingInfo
              catalogDir="portal_message_catalog_5101"
              fileName="wps5101msgprefixmapping.properties"
              mappingType="PortalCatalogMessage"
              keyPattern="([A-Z]{4,5}[0-9]{1,4}[EW])"/>
</productinfo>
<productinfo analysisProfileRef="wps_portal_5.1"
              name="IBM WebSphere Portal"
              version="5.1.0.0">
       <msgPrefixMappingInfo
              catalogDir="portal_message_catalog_510"
              fileName="wps5100msgprefixmapping.properties"
              mappingType="PortalCatalogMessage"
              keyPattern="([A-Z]{4,5}[0-9]{1,4}[EW])"/>
</productinfo>
<productinfo analysisProfileRef="wps_portal_5.1"
              name="IBM WebSphere Portal"
              version="5.1"/>
<productinfo analysisProfileRef="wps_portal_5.0"
              name="IBM WebSphere Portal"
              version="5.0"/>
</extractors>
```

Regular Expressions Primer

There exists a lot of documentation regarding regular expressions. A quick Internet search should find lots of information. As a quick reference, we provide short descriptions of many common regular expressions and what they match:

a   the character a

[abc] a, b, or c

[^abc] Any character except a, b, or c

[a-zA-Z] a through z or A through Z, inclusive

[0-9] any digit

\ followed by a space ‘ ’ match a space

\n newline

\r carriage return

\t tab

\f formfeed

\d digit [0-9]

\D non-digit [^0-9]

\w word character [0-9a-z_A-Z]

\W non-word character [^0-9a-z_A-Z]

\s a whitespace character [\t\n\r\f]

\S a non-whitespace character [^ \t\n\r\f]

\x*nn* the hexadecimal representation of character *nn*

\c*D* the corresponding control character

nn or *nnn* the octal representation of character

*? 0 or more times

+? 1 or more times

{*n*}? exactly *n* times
{*n*,}? at least *n* times
{*n*,*m*}? at least *n* but not more than *m* times

Escape characters
Sometimes we need to match the special characters that are used in the regular expression syntax. For example, suppose we need to match a period. We can't just put a period in our regular expression, because the period in the regular expression syntax matches any single character. To say "match a period", we need to escape the period character by doing the following:

\. Matches a period.

Similarly, we can match other special characters that are part of the regular expression syntax by escaping them as well

\\ Matches a single backslash
\{ Matches a single left brace
\} Matches a single right brace
\(Matches a single left parenthesis
\) Matches a single right parenthesis

Other Useful Hints

| separates alternatives.
Expressions within parentheses are matched as subpattern groups and saved for later use.
By default, a quantified subpattern matches as many times as possible without causing the rest of the pattern not to match. To change the quantifiers to match the minimum number of times possible, without causing the rest of the pattern not to match, use a '?' immediately after the quantifier.

Symptom Schema

For completeness, the schema for our symptom document is included below. Since WebSphere Portal does not currently support Common Base Events (CBEs), the enumeration values related to CBE processing are not relevant for WebSphere Portal symptom analysis. However, the elements related to general XML document processing, as well as the enumeration values related to event-history documents, are relevant, because the AutoPD tool for WebSphere Portal includes symptom analysis for the XML-formatted event.history files.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- (C) COPYRIGHT International Business Machines Corp., 2004-2005. All Rights
Reserved * Licensed Materials - Property of IBM -->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
       targetNamespace="http://www.ibm.com/autopd/pattern"
       xmlns="http://www.ibm.com/autopd/pattern"
       elementFormDefault="qualified">

<xsd:element name="delimiter">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="0" ref="timestamp"/>
            <xsd:element maxOccurs="unbounded" minOccurs="1" ref="pattern"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="required"/>
        <xsd:attribute name="value" type="xsd:string" use="required"/>
        <xsd:attribute name="timeRangeApplies" type="xsd:boolean" use="optional"
             default="false"/>
    </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="delimiterid">
    <xsd:complexType>
        <xsd:attribute name="id" type="xsd:string" use="required"/>
        <xsd:attribute name="type" type="xsd:string" use="optional"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="timestamp">
    <xsd:complexType>
        <xsd:attribute name="group" type="xsd:string" use="required"/>
        <xsd:attribute name="pattern" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>

<!-- Begin schema elements for XML document processing -->
<xsd:element name="xmlDelimiter">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="1" ref="xmlLogRecord"/>
            <xsd:element maxOccurs="1" minOccurs="1" ref="xmlMatch"/>
            <xsd:element maxOccurs="1" minOccurs="1" ref="xmlGroups"/>
        </xsd:sequence>
        <xsd:attribute name="id" type="xsd:string" use="required"/>
        <xsd:attribute name="documentXsd" type="xsd:string" use="required"/>
     </xsd:complexType>
</xsd:element>

<!-- The <xmlLogRecord> element identifies the XML elements representing
     the log records that are candidates for extraction.
-->
<xsd:element name="xmlLogRecord">
    <xsd:complexType>
       <xsd:attribute name="elementType" type="xsd:string" use="required"/>
       <xsd:attribute name="level" type="XmlLevelType" use="required"/>
    </xsd:complexType>
</xsd:element>

<!-- The <xmlMatch> element specifies the criteria for selecting an
        XML element for inclusion in the analysis report.  It plays the
        role of the SQL "WHERE" clause.
-->
<xsd:element name="xmlMatch">
    <xsd:complexType>
        <xsd:sequence>
             <!-- Multiple <choice> elements are ANDed together -->
             <xsd:choice minOccurs="0" maxOccurs="unbounded">
                    <xsd:element ref="xmlStringAtom"/>
                    <xsd:element ref="xmlNumericAtom"/>
                    <xsd:element ref="xmlOr"/>
             </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="matchAll" type="xsd:boolean" default="false"/>
    </xsd:complexType>
</xsd:element>

<!-- The <xmlOr> element expresses a disjunction (OR) of a set of
 <xmlAtom> elements.
-->
<xsd:element name="xmlOr">
    <xsd:complexType>
        <xsd:sequence>
             <xsd:choice minOccurs="2" maxOccurs="unbounded">
```

```
                    <xsd:element ref="xmlStringAtom"/>
                    <xsd:element ref="xmlNumericAtom"/>
             </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<!-- The <xmlStringAtom> element identifies one field of a XML record to
        match against.  The value is a regular expression, against which the
        corresponding field of the XML record is tested.  The attribute
        "negated" determines whether the match should be negated, so that XML
        records that do not satisfy the indicated  match are selected.
-->
<xsd:element name="xmlStringAtom">
       <xsd:complexType>
             <xsd:simpleContent>
                    <xsd:extension base="xsd:string">
                           <xsd:attribute name="xmlTarget" type="XmlTargetType"
                                  use="required"/>
                           <xsd:attribute name="negated" type="xsd:boolean"
                                  use="optional" default="false"/>
                    </xsd:extension>
             </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>

<!-- The <xmlNumericAtom> element identifies one field of a XML record
        to match against.  The value is an integer, against which the
        corresponding field of the XML record is tested.  The attribute
        "comparisonOperator" indicates the test to be performed - for example,
        the comparisonOperator value "greaterThan" tests whether the value
        of the field in the XML record is greater than the value of this
        element.  The attribute "negated" determines whether the match should
        be negated, so that XML records that do not satisfy the indicated match
        are selected.
-->
<xsd:element name="xmlNumericAtom">
       <xsd:complexType>
             <xsd:simpleContent>
                    <xsd:extension base="xsd:integer">
                           <xsd:attribute name="xmlTarget" type="XmlTargetType"
                                  use="required"/>
                           <xsd:attribute name="comparisonOperator"
                                  type="ComparisonOperatorType" use="required"/>
                           <xsd:attribute name="negated" type="xsd:boolean"
                                  use="optional" default="false"/>
                    </xsd:extension>
             </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>

<!-- The XmlTargetType contains groups of enumeration values for
        each XML schema covered by the analysis function.  These values
        represent easy-to-understand designations for items to be
        extracted from the corresponding documents.  Each group
        requires its own Java class to perform the extractions.
-->
<xsd:simpleType name="XmlTargetType">
       <xsd:restriction base="xsd:string">

             <!-- Enumeration values for CBE documents. -->
             <xsd:enumeration value="entireCBE"/>
             <xsd:enumeration value="cbeSituationCategory"/>
             <xsd:enumeration value="cbeSeverity"/>
             <xsd:enumeration value="cbeTimestamp"/>
             <xsd:enumeration value="cbeMsgId"/>

             <!-- Enumeration values for event.history documents. -->
             <xsd:enumeration value="entireUpdateEvent"/>
             <xsd:enumeration value="updateEventId"/>
             <xsd:enumeration value="updateAction"/>
             <xsd:enumeration value="updateType"/>
             <xsd:enumeration value="updateEventStartTimestamp"/>
             <xsd:enumeration value="updateEventEndTimestamp"/>
             <xsd:enumeration value="updateEventResult"/>

       </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="XmlLevelType">
       <xsd:restriction base="xsd:string">
             <xsd:enumeration value="top"/>
             <xsd:enumeration value="all"/>
       </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="ComparisonOperatorType">
       <xsd:restriction base="xsd:string">
             <xsd:enumeration value="equalTo"/>
             <xsd:enumeration value="lessThan"/>
             <xsd:enumeration value="lessThanOrEqualTo"/>
             <xsd:enumeration value="greaterThan"/>
             <xsd:enumeration value="greaterThanOrEqualTo"/>
       </xsd:restriction>
</xsd:simpleType>

<!-- The <xmlGroups> element specifies which parts of a selected record
        are to be extracted for inclusion in the analysis report.  It plays
        the role of the SQL "SELECT" clause.
-->
<xsd:element name="xmlGroups">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" ref="xmlGroup"/>
            <xsd:element ref="formatref"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<!-- The <xmlGroup> element specifies one part of a selected record
     to be extracted for inclusion in the analysis report.
-->
 <xsd:element name="xmlGroup">
    <xsd:complexType>
        <xsd:attribute name="xmlTarget" type="XmlTargetType" use="required"/>
        <xsd:attribute name="displayName" type="xsd:string" use="required"/>
        <xsd:attribute name="beginIndex" type="xsd:integer" use="optional"/>
        <xsd:attribute name="beginPattern" type="xsd:string" use="optional"/>
        <xsd:attribute name="endIndex" type="xsd:integer" use="optional"/>
        <xsd:attribute name="endPattern" type="xsd:string" use="optional"/>
    </xsd:complexType>
</xsd:element>
<!-- End schema elements for xml processing -->

 <xsd:element name="extractors">
    <xsd:complexType>
```

```
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" ref="problem"/>
            <xsd:element maxOccurs="unbounded" minOccurs="0" ref="delimiter"/>
            <xsd:element maxOccurs="unbounded" minOccurs="0" ref="xmlDelimiter"/>
            <xsd:element ref="output"/>
            <xsd:element maxOccurs="unbounded" minOccurs="1" ref="productinfo"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name="fileset">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" ref="delimiterid"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="value" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="format">
    <xsd:complexType>
         <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="0"
                    ref="msgCatalogLookup"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="order" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="msgCatalogLookup">
    <xsd:complexType>
        <xsd:attribute name="group" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="formatref">
    <xsd:complexType>
        <xsd:attribute name="outputName" type="xsd:string" use="required"/>
        <xsd:attribute name="formatName" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="group">
    <xsd:complexType>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="number" type="xsd:string" use="required"/>
        <xsd:attribute name="beginIndex" type="xsd:integer" use="optional"/>
        <xsd:attribute name="beginPattern" type="xsd:string" use="optional"/>
        <xsd:attribute name="endIndex" type="xsd:integer" use="optional"/>
        <xsd:attribute name="endPattern" type="xsd:string" use="optional"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="output">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" ref="format"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="format" type="xsd:string" use="required"/>
    </xsd:complexType>
```

```
</xsd:element>

<xsd:element name="pattern">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" ref="group"/>
            <xsd:element ref="formatref"/>
        </xsd:sequence>
        <xsd:attribute name="value" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="problem">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" ref="analysisProfile"/>
        </xsd:sequence>
        <xsd:attribute name="description" type="xsd:string" use="optional"/>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="analysisProfile">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element maxOccurs="unbounded" minOccurs="1" ref="fileset"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="productinfo">
    <xsd:complexType>
       <xsd:sequence>
            <xsd:element maxOccurs="1" minOccurs="0" ref="msgPrefixMappingInfo"/>
       </xsd:sequence>
       <xsd:attribute name="analysisProfileRef" type="xsd:string" use="required"/>
       <xsd:attribute name="name" type="xsd:string" use="required"/>
       <xsd:attribute name="version" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>

<xsd:element name="msgPrefixMappingInfo">
    <xsd:complexType>
        <xsd:attribute name="catalogDir" type="xsd:string" use="required"/>
        <xsd:attribute name="fileName" type="xsd:string" use="required"/>
        <xsd:attribute name="mappingType" type="xsd:string" use="required"/>
        <xsd:attribute name="keyPattern" type="xsd:string" use="required"/>
    </xsd:complexType>
</xsd:element>

</xsd:schema>
```

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
iSeries
IBM
The IBM logo
WebSphere

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. 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.

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

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

HP-UX is a registered trademark of Hewlett-Packard Company.

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

Appendix A: Collection Details for Each WebSphere Portal Problem Type

The following sections provide specific details of what the tool does for each of the WebSphere Portal problem types it supports. Each section contains information that falls into four categories:

1. The usage scenario for that problem type, described above in **Usage Scenarios**.
2. The list of files collected by the tool and included in the collection zip file for that problem type.
3. The trace enablement settings, if traces are enabled for that problem type.
4. The problem type value used when symptom analysis is invoked.

Note that the lists of files shown for the various problem types represents the *maximum* sets of files that may be gathered for each problem type – in some case, only a subset of the indicated files actually exist. Also note that the first five files listed do not come from a fixed location. Depending on whether the WebSphere Portal is part of a cluster, and if so, depending on which cluster member is the subject of the collection, these files may come from a variety of locations. Note finally that the collections are set up in such a way that .jar, .ear, and .war files are *always excluded* from the collections, since they can greatly increase the size of the collection zip file without adding any real value to it.

For the files related to WebSphere Application Server, there are several locations from which files are typically collected:

1. WAS_HOME indicates the home directory where WebSphere Application Server was installed. Files in this location are collected for all releases of WebSphere Application Server. In the case of an iSeries system, this is typically a subdirectory under /ProdData.
2. WAS_PROFILE_HOME, which applies only to WebSphere Application Server v6 and above, indicates the home directory for the particular profile that WebSphere Portal is using. This directory is often in the location WAS_HOME/profiles/<profile-name>. This directory does not apply to an iSeries system.
3. ISERIES_USER_INSTALL_HOME, which applies only to iSeries systems, indicates the WebSphere Application Server user-installation home directory that WebSphere Portal is using for a run-time Portal. This is typically a subdirectory under /UserData.

For files related to WebSphere Portal, the location WPS_HOME indicates on most systems the home directory where WebSphere Portal was installed. The one exception to this rule is an iSeries system. In the case of an iSeries system, WPS_HOME indicates the location (typically a subdirectory under /UserData) where a run-time Portal instance is located. The product install directory itself, which is typically a subdirectory under /ProdData, is identified as WPS_ISERIES_PRODDATA_HOME.

Common to all problem types except for Portal Problem Analysis Report and Collect Portal Product Information is the option of automatically sending the collection zip file to IBM Support via FTP. You will always be given the option of bypassing this step, if this is not appropriate for your specific environment.

Portal Install Problem

This problem type is used for problems related to installation of WebSphere Portal. Depending on where the problem occurred in the overall install process, the usual file locations for the WebSphere Portal logs may not exist. Consequently, most of the collection activities for this problem type are directed to the OS-specific temporary directory in which the install logs are saved.

Usage Scenario

The basic log collection scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version: entire subdirectory *except* WPS_HOME/version/backup subdirectory
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files

- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

The following files are collected on all Windows systems, from the directory identified by the %TEMP% environment variable:

- baseInstallSummary.File
- baseResponse.File
- dbg*: entire subdirectory
- efixes: entire subdirectory
- external.mq: entire subdirectory
- icmrmcrdb_*.log
- installmessages.txt
- installtraces.txt
- installtraces*.txt
- isj*: entire subdirectory
- log.txt
- LocalizeConfigMessages.log
- LocalizeConfigTrace.log
- LocalizeConfigTrace*.log
- LocalizeProgress.txt
- LocalizeTrace.log
- mq_prereq.log
- platform: entire subdirectory
- pzninstall.log
- pznInstallLog.txt
- regfix*.reg
- resume.bat
- serverstatus.err
- wbisf_fixpacklogs: entire subdirectory
- wbisf_launcher.log
- wpinstalllog.txt
- wpsinstalllog.txt
- wpsinstallprogress.txt
- wps_CPPWIN32_dll*.tmp
- WSBAA*.uid

The vpd.properties file is also collected on Windows, from either the c:\WINNT or the c:\WINDOWS directory.

The following files are collected on Linux, from the /tmp directory:

- baseInstallSummary.File
- baseResponse.File
- dbg*: entire subdirectory
- efixes: entire subdirectory
- external.mq: entire subdirectory
- icmrmcrdb_*.log
- installmessages.txt
- installtraces.txt
- installtraces*.txt
- isj*: entire subdirectory
- log.txt

- LocalizeConfigMessages.log
- LocalizeConfigTrace.log
- LocalizeConfigTrace*.log
- LocalizeProgress.txt
- LocalizeTrace.log
- mq_prereq.log
- platform: entire subdirectory
- pzninstall.log
- pznInstallLog.txt
- regfix*.reg
- resume.bat
- serverstatus.err
- wbisf_fixpacklogs: entire subdirectory
- wbisf_launcher.log
- wpinstalllog.txt
- wpsinstalllog.txt
- wpsinstallprogress.txt
- WSBAA*.uid

The vpd.properties file is also collected on Linux, either from the system's root directory or from the /root directory.

The following files are collected on HP-UX, from the /tmp directory:

- baseInstallSummary.File
- baseResponse.File
- dbg*: entire subdirectory
- efixes: entire subdirectory
- external.mq: entire subdirectory
- icmrmcrdb_*.log
- installmessages.txt
- installtraces.txt
- installtraces*.txt
- isj*: entire subdirectory
- log.txt
- LocalizeConfigMessages.log
- LocalizeConfigTrace.log
- LocalizeConfigTrace*.log
- LocalizeProgress.txt
- LocalizeTrace.log
- mqcheck.log
- mq_prereq.log
- platform: entire subdirectory
- pzninstall.log
- pznInstallLog.txt
- regfix*.reg
- resume.bat
- serverstatus.err
- wbisf_fixpacklogs: entire subdirectory
- wbisf_launcher.log
- wpinstalllog.txt
- wpsinstalllog.txt
- wpsinstallprogress.txt

- WSBAA*.uid

The vpd.properties file is also collected on HP-UX, either from the system's root directory or from the /root directory.

The following files are collected on Aix, from the /tmp directory:

- baseInstallSummary.File
- baseResponse.File
- dbg*: entire subdirectory
- efixes: entire subdirectory
- external.mq: entire subdirectory
- icmrmcrdb_*.log
- installmessages.txt
- installtraces.txt
- installtraces*.txt
- isj*: entire subdirectory
- log.txt
- LocalizeConfigMessages.log
- LocalizeConfigTrace.log
- LocalizeConfigTrace*.log
- LocalizeProgress.txt
- LocalizeTrace.log
- mq_prereq.log
- mqcheck.log
- platform: entire subdirectory
- pzninstall.log
- pznInstallLog.txt
- regfix*.reg
- resume.bat
- serverstatus.err
- wbisf_fixpacklogs: entire subdirectory
- wbisf_launcher.log
- wpinstalllog.txt
- wpsinstalllog.txt
- wpsinstallprogress.txt
- WSBAA*.uid
- PaxHeaders.*: entire subdirectory
- xlogfile
- xlogfile*

The vpd.properties file is also collected on Aix, from the /usr/lib/objrepos directory.

The following files are collected on Solaris, from the /tmp directory:

- baseInstallSummary.File
- baseResponse.File
- installmessages.txt
- installtraces.txt
- installtraces*.txt
- log.txt
- mq_prereq.log

- pzninstall.log
- pznInstallLog.txt
- regfix*.reg
- resume.bat
- serverstatus.err
- wbisf_fixpacklogs: entire subdirectory
- wbisf_launcher.log
- wpinstalllog.txt
- wpsinstalllog.txt
- wpsinstallprogress.txt
- WSBAA*.uid
- pznInstallLog.txt
- pzninstall.log
- sdtvolcheck*

The following files are collected on Solaris, from the /var/tmp directory:

- dbg*: entire subdirectory
- efixes: entire subdirectory
- external.mq: entire subdirectory
- icmrmcrdb_*.log
- isj*: entire subdirectory
- LocalizeConfigMessages.log
- LocalizeConfigTrace.log
- LocalizeConfigTrace*.log
- LocalizeProgress.txt
- LocalizeTrace.log
- platform: entire subdirectory

The vpd.properties file is also collected on Solaris, either from the system's root directory or from the /root directory.

The following files are collected on iSeries, from the /tmp/InstallShield directory:

- baseInstallSummary.File
- baseResponse.File
- dbg*/**/*
- efixes/**/*
- external.mq/**/*
- icmrmcrdb_*.log
- installmessages.txt
- installtraces.txt
- installtraces*.txt
- isj*/**/*
- log.txt
- LocalizeConfigMessages.log
- LocalizeConfigTrace.log
- LocalizeConfigTrace*.log
- LocalizeProgress.txt
- LocalizeTrace.log
- mq_prereq.log
- platform/**/*
- pzninstall.log

- pznInstallLog.txt
- regfix*.reg
- resume.bat
- serverstatus.err
- wbisf_fixpacklogs/**/*
- wbisf_launcher.log
- wpinstalllog.txt
- wpsinstalllog.txt
- wpsinstallprogress.txt
- wps_CPPWIN32_dll*.tmp
- WSBAA*.uid

Trace Enablement Settings

No traces are enabled for this problem type.

Symptom Analysis Problem Type

portalinstall

Portlet Deployment Problem

This problem type is used for problems that arise when deploying portlets into a working WebSphere Portal.

Usage Scenario

The combined basic log collection / problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells/${CellName}/admin-authz.xml
- WAS_HOME/config/cells/${CellName}/nodes/${NodeName}/serverindex.xml
- WAS_HOME/config/cells/${CellName}/applications: entire subdirectory
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/config/wpconfig.properties
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml

- WPS_HOME/version/history/event.history
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- com.ibm.ssl.keyStore (directory location varies)
- com.ibm.ssl.trustStore (directory location varies)
- plugin-cfg.xml (directory location varies)

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/credentialslots_output.xml
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/tmp/XmlAccessForPortletInstall: entire subdirectory
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

When the basic log collection path is followed, no traces are enabled for this problem type.

When the problem reproduction path is followed, the following traces are enabled for this problem type:

Portal tracing level setting:

```
com.ibm.wps.pe.mgr.*=all=enabled:
com.ibm.wps.services.deployment.*=all=enabled:
com.ibm.wps.command.applications.*=all=enabled:
com.ibm.wps.command.portlets.*=all=enabled
```

Note: This trace setting string is contiguous; it has been reformatted here for readability.

Symptom Analysis Problem Type

portletdeployment

Portal Configuration Problem

This problem type is used for problems related to configuring WebSphere Portal.

Usage Scenario

The combined basic log collection / problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells: entire subdirectory *except* WAS_HOME/ config/cells/*/applications subdirectories other than WAS_HOME/config/cells/${CellName}/applications
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/config: entire subdirectory *except* WPS_HOME/config/DBTransfer, WPS_HOME/config/templates, WPS_HOME/config/wizard, WPS_HOME/config/deployed, WPS_HOME/config//deployable/ear, WPS_HOME/config/gather/deployed, and WPS_HOME/config/tmp subdirectories
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version: entire subdirectory *except* WPS_HOME/version/backup subdirectory
- WPS_HOME/wcm/config: entire subdirectory
- WPS_HOME/wcm/connect: entire subdirectory
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/config/wmm: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

When the basic log collection path is followed, no traces are enabled for this problem type.

When the problem reproduction path is followed, the following traces are enabled for this problem type:

Portal tracing level setting:

```
com.ibm.wps.ac.*=all=enabled:
com.ibm.wps.engine.commands.*=all=enabled:
com.ibm.wps.services.*=all=enabled:
com.ibm.wps.puma.*=all=enabled:
com.ibm.wps.engine.servlet=all=e autopdzip/autopd/portal-info/exportconfig_output.xmnabled:
com.ibm.wps.sso.*=all=enabled:
com.ibm.wps.composition.*=all=enabled:
com.ibm.wps.portlets.*=all=enabled:
org.apache.jetspeed.portlet.PortletLog=all=enabled:
com.ibm.wps.pe.*=all=enabled:
com.ibm.wps.command.applications.*=all=enabled:
com.ibm.wps.command.portlets.*=all=enabled:
com.ibm.wps.command.puma.*=all=enabled
```

Note: This trace setting string is contiguous; it has been reformatted here for readability.

Symptom Analysis Problem Type

portalconfig

Portal Configuration Problem with Topology Data

This collection script is identical to the one for the Portal Configuration Problem, with one exception: it gives you the option of including WebSphere Portal topology data in the collection. If you choose to do this, the script will issue an xmlaccess export command to gather the data. The data appears in the file

- autopdzip/autopd/portal-info/exportconfig_output.xml

Portal General Problem

This problem type is used for WebSphere Portal problems that cannot be assigned to one of the other problem types.

Usage Scenario

The combined basic log collection / problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells/${CellName}/applications: entire subdirectory
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/config: entire subdirectory *except* WPS_HOME/config/DBTransfer, WPS_HOME/config/templates, WPS_HOME/config/wizard, WPS_HOME/config/deployed, WPS_HOME/config//deployable/ear, WPS_HOME/config/gather/deployed, and WPS_HOME/config/tmp subdirectories
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version: entire subdirectory *except* WPS_HOME/version/backup subdirectory
- WPS_HOME/wcm/config: entire subdirectory
- WPS_HOME/wcm/connect: entire subdirectory
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

When the basic log collection path is followed, no traces are enabled for this problem type.

When the problem reproduction path is followed, the following traces are enabled for this problem type:

Portal tracing level setting:

```
com.ibm.wps.*=all=enabled:
org.apache.jetspeed.portlet.PortletLog=all=enabled
```

Note: This trace setting string is contiguous; it has been reformatted here for readability.

Symptom Analysis Problem Type

portalgeneral

Portal Upgrade Problem

This problem type is used for problems related to upgrading an existing WebSphere Portal to a later version.

Usage Scenario

The basic log collection scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells: entire subdirectory *except* WAS_HOME/ config/cells/*/applications subdirectories other than WAS_HOME/config/cells/${CellName}/applications
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/config: entire subdirectory *except* WPS_HOME/config/DBTransfer, WPS_HOME/config/templates, WPS_HOME/config/wizard, WPS_HOME/config/deployed, WPS_HOME/config//deployable/ear, WPS_HOME/config/gather/deployed, WPS_HOME/config/work, and WPS_HOME/config/tmp subdirectories
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version: entire subdirectory *except* WPS_HOME/version/backup subdirectory
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log/autopd.log
- RasGUI_HOME/log/autopdecho.log
- RasGUI_HOME/log: entire subdirectory

- <user-supplied directory containing updatePortal.bat/sh> : entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files

- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/config/wmm: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

No traces are enabled for this problem type.

Symptom Analysis Problem Type

portalupgrade

Portal Access Control Problem

This problem type is used for problems related to configuring access control for WebSphere Portal.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells/${CellName}/nodes/${NodeName}/servers/${ServerName}/server.xml
- WAS_HOME/config/cells/${CellName}/security.xml
- WAS_HOME/config/cells/${CellName}/applications: entire subdirectory
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files

- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version/history/event.history
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Collection of Topology Data

This collection script gives you the option of including WebSphere Portal topology data in the collection. If you choose to do this, the script will issue an xmlaccess export command to gather the data. The data appears in the file

- autopdzip/autopd/portal-info/exportconfig_output.xml

Trace Enablement Settings

The following traces are enabled for this problem type:

Portal tracing level setting for WebSphere Portal v5.0:

```
com.ibm.wps.portlets.resourceview.*=all=enabled:
com.ibm.wps.puma.*=all=enabled:
com.ibm.wps.services.puma.*=all=enabled:
com.ibm.wps.command.puma.*=all=enabled:
```

```
com.ibm.wps.ac.*=all=enabled:
com.ibm.wps.command.ac.*=all=enabled:
com.ibm.wps.ac.impl.*=all=enabled
```

Portal tracing level setting for WebSphere Portal v5.1:

```
com.ibm.wps.portlets.permissions.*=all=enabled:
com.ibm.wps.puma.*=all=enabled:
com.ibm.wps.services.puma.*=all=enabled:
com.ibm.wps.command.puma.*=all=enabled:
com.ibm.wps.ac.*=all=enabled:
com.ibm.wps.command.ac.*=all=enabled:
com.ibm.wps.ac.impl.*=all=enabled
```

Portal tracing level setting for WebSphere Portal v6.0 and later:

```
com.ibm.wps.ac.impl.AccessControlConfigFederator=all:
com.ibm.wps.ac.impl.AccessControlFederator=all:
com.ibm.wps.services.puma.*=all:
com.ibm.wps.puma.*=all:
com.ibm.wps.engine.commands.*=all:
com.ibm.wps.portlets.admin.*=all:
com.ibm.wps.portlets.permissions.*=all
```

WMM tracing level setting:

```
com.ibm.websphere.wmm.*=all=enabled:
com.ibm.ws.wmm.*=all=enabled:
WSMM=all=enabled
```

Note: These trace setting strings are contiguous; they have been reformatted here for readability.

Symptom Analysis Problem Type

portalaccesscontrol

Portal Login Problem

This problem type is used when a user has difficulty logging into WebSphere Portal.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells/${CellName}/nodes/${NodeName}/servers/${ServerName}/server.xml
- WAS_HOME/config/cells/${CellName}/security.xml
- WAS_HOME/config/cells/${CellName}/applications: entire subdirectory
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory

- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version/history/event.history
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

The following traces are enabled for this problem type:

Portal tracing level setting for WebSphere Portal v5.0 and v5.1:

```
com.ibm.wps.engine.*=all=enabled:
com.ibm.wps.command.puma.*=all=enabled:
com.ibm.wps.puma.*=all=enabled:
```

```
com.ibm.wps.services.puma.*=all=enabled:
com.ibm.wps.sso.*=all=enabled
```

Portal tracing level setting for WebSphere Portal v6.0

```
com.ibm.wps.engine.*=all:
com.ibm.wps.services.puma.*=all:
com.ibm.wps.puma.*=all:
com.ibm.wps.sso.*=all:
com.ibm.wps.services.authentication.*=all:
com.ibm.ws.security.*=all
```

WMM tracing level setting:

```
com.ibm.websphere.wmm.*=all=enabled:
com.ibm.ws.wmm.*=all=enabled:
WSMM=all=enabled
```

Note: These trace setting strings are contiguous; they have been reformatted here for readability.

Symptom Analysis Problem Type

portallogin

Portal Manage Users and Groups Problem

This problem type is used for problems related to user and group management for WebSphere Portal.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells/${CellName}/nodes/${NodeName}/servers/${ServerName}/server.xml
- WAS_HOME/config/cells/${CellName}/security.xml
- WAS_HOME/config/cells/${CellName}/applications: entire subdirectory
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml

- WPS_HOME/version/history/event.history
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

The following traces are enabled for this problem type:

Portal tracing level setting for WebSphere Portal:

```
com.ibm.wps.portlets.manageprincipals.*=all=enabled:
com.ibm.wps.portlets.admin.search.*=all=enabled:
com.ibm.wps.portlets.manageprincipals.util.*=all=enabled:
com.ibm.wps.portlets.manageprincipals.render.*=all=enabled:
com.ibm.wps.puma.*=all=enabled:
com.ibm.wps.services.puma.*=all=enabled:
com.ibm.wps.commands.puma.*=all=enabled:
com.ibm.wps.ac.*=all=enabled:
com.ibm.wps.command.ac.*=all=enabled:
com.ibm.wps.ac.impl.*=all=enabled
```

WMM tracing level setting:

```
com.ibm.websphere.wmm.*=all=enabled:
```

com.ibm.ws.wmm.*=all=enabled:
WSMM=all=enabled

Note: These trace setting strings are contiguous; they have been reformatted here for readability.

Symptom Analysis Problem Type

portalmanageusersandgroups

Portal Start/Stop Problem

This problem type is used for problems related to starting and stopping WebSphere Portal.

Usage Scenario

The basic log collection scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells: entire subdirectory *except* WAS_HOME/ config/cells/*/applications subdirectories other than WAS_HOME/config/cells/${CellName}/applications
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/config: entire subdirectory *except* WPS_HOME/config/DBTransfer, WPS_HOME/config/templates, WPS_HOME/config/wizard, WPS_HOME/config/deployed, WPS_HOME/config//deployable/ear, WPS_HOME/config/gather/deployed, and WPS_HOME/config/tmp subdirectories
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version: entire subdirectory *except* WPS_HOME/version/backup subdirectory
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/config/wmm: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

No traces are enabled for this problem type.

Symptom Analysis Problem Type

portalstartstop

Portal XML Configuration Interface Problem

This problem type is used for problems related to WebSphere Portal’s XML-based configuration interface (i.e., for problems related to use of the xmlaccess command).

Usage Scenario

The combined basic log collection / problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells/${CellName}/applications: entire subdirectory

- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version/history/event.history
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/tmp/XmlIntf: entire subdirectory
- RasGUI_HOME/log: entire subdirectory

- user's xmlaccess input file (if available)
- user's xmlaccess output file (if available)

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Collection of Topology Data

This collection script gives you the option of including WebSphere Portal topology data in the collection. If you choose to do this, the script will issue an xmlaccess export command to gather the data. The data appears in the file

- autopdzip/autopd/portal-info/exportconfig_output.xml

Trace Enablement Settings

The following traces are enabled for this problem type:

Portal tracing level setting:

com.ibm.wps.command.xml.*=all=enabled

Symptom Analysis Problem Type

portalxmlintf

Portal Integration with Tivoli Access Manager Problem

This problem type is used for integration problems involving WebSphere Portal and Tivoli Access Manager.

Usage Scenario

The combined basic log collection / problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells/${CellName}/nodes/${NodeName}/servers/${ServerName}/server.xml
- WAS_HOME/config/cells/${CellName}/security.xml
- WAS_HOME/config/cells/${CellName}/applications: entire subdirectory
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version/history/event.history
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/tmp/TamIntegration: entire subdirectory
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

The following traces are enabled for this problem type:

WebSphere Portal trace setting for an authentication problem:

```
com.ibm.ws.security.web.*=all=enabled
```

WebSphere Portal trace setting for an authorization problem:

```
com.ibm.wps.ac.esm.*=all=enabled:
com.ibm.wps.ac.authtable.*=all=enabled
```

WebSphere Portal trace setting for an authorization problem involving Credential Vault:

```
com.ibm.wps.sso.*=all=enabled
```

Note: These trace setting strings are contiguous; they have been reformatted here for readability.

Symptom Analysis Problem Type

portaltamintegration

Portal Document Manager Problem

This problem type is used for problems involving the WebSphere Portal Document Manager.

Usage Scenario

The combined basic log collection / problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells: entire subdirectory *except* WAS_HOME/ config/cells/*/applications subdirectories other than WAS_HOME/config/cells/${CellName}/applications
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/config: entire subdirectory *except* WPS_HOME/config/DBTransfer, WPS_HOME/config/templates, WPS_HOME/config/wizard, WPS_HOME/config/deployed, WPS_HOME/config//deployable/ear, WPS_HOME/config/gather/deployed, WPS_HOME/config/work, and WPS_HOME/config/tmp subdirectories
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version: entire subdirectory *except* WPS_HOME/version/backup subdirectory
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory

- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/config/wmm: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

Portal tracing level setting for WebSphere Portal

com.ibm.wps.pdm.*=all=enabled:
com.ibm.dm.*=all=enabled:
com.ibm.icm.*=all=enabled:
com.ibm.wps.services.puma.*=all=enabled:
com.ibm.wps.puma.*=all=enabled:
com.ibm.wps.pdm.servlets.*=all=enabled

Symptom Analysis Problem Type

portalcommon

Portal Personalization Problem

This problem type is used for problems involving WebSphere Portal Personalization.

Usage Scenario

The combined basic log collection / problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells: entire subdirectory *except* WAS_HOME/ config/cells/*/applications subdirectories other than WAS_HOME/config/cells/${CellName}/applications
- WAS_HOME/config/.repository: entire subdirectory

- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/config: entire subdirectory *except* WPS_HOME/config/DBTransfer, WPS_HOME/config/templates, WPS_HOME/config/wizard, WPS_HOME/config/deployed, WPS_HOME/config//deployable/ear, WPS_HOME/config/gather/deployed, WPS_HOME/config/work, and WPS_HOME/config/tmp subdirectories
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version: entire subdirectory *except* WPS_HOME/version/backup subdirectory
- WPS_HOME/wmm: entire subdirectory

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/config/wmm: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

Portal tracing level setting for WebSphere Portal

com.ibm.websphere.personalization.*=all=enabled:
com.ibm.dm.pzn.ui.*=all=enabled

Symptom Analysis Problem Type

portalcommon

IBM Workplace Web Content Management™ Problem

This problem type is used for general problems with IBM Workplace Web Content Management.

Usage Scenario

The combined basic log collection / problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells: entire subdirectory *except* WAS_HOME/ config/cells/*/applications subdirectories other than WAS_HOME/config/cells/${CellName}/applications
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/config: entire subdirectory *except* WPS_HOME/config/DBTransfer, WPS_HOME/config/templates, WPS_HOME/config/wizard, WPS_HOME/config/deployed, WPS_HOME/config//deployable/ear, WPS_HOME/config/gather/deployed, WPS_HOME/config/work, and WPS_HOME/config/tmp subdirectories
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version: entire subdirectory *except* WPS_HOME/version/backup subdirectory
- WPS_HOME/wmm: entire subdirectory
- WPS_HOME/wcm/config: all files (WebSphere Portal v5.x only)
- WPS_HOME/wcm/connect/log: all files (WebSphere Portal v5.x only)
- WPS_HOME/wcm/migration: all files (WebSphere Portal v6.x only)
- WPS_HOME/wcm/shared: all files (WebSphere Portal v6.x only)

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory
- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/config/wmm: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

The following traces are enabled.

Portal tracing level setting for WebSphere Portal v5.1:

com.ibm.dm.*=all=enabled:com.ibm.icm.*=all=enabled

Portal tracing level setting for WebSphere Portal v6.0:

com.ibm.icm.jcr.*=all=enabled

Symptom Analysis Problem Types

portalcommon, wcm

IBM Workplace Web Content Management Syndication Problem

This problem type is used for syndication problems involving IBM Workplace Web Content Management.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- all trace_[timestamp].log files

- WAS_HOME/config/cells: entire subdirectory *except* WAS_HOME/ config/cells/*/applications subdirectories other than WAS_HOME/config/cells/${CellName}/applications
- WAS_HOME/config/.repository: entire subdirectory
- WAS_HOME/config/wmm: entire subdirectory
- WAS_HOME/logs: entire subdirectory *except* WAS_HOME/logs/ffdc subdirectory and WAS_HOME /logs/lost+found subdirectory
- WAS_HOME/properties: all files
- WAS_HOME/properties/version: all files
- WAS_HOME/properties/version/history: all files
- WAS_HOME/properties/version/update/backup: all XML files

- WPS_HOME/log: all files
- WPS_HOME/config: entire subdirectory *except* WPS_HOME/config/DBTransfer, WPS_HOME/config/templates, WPS_HOME/config/wizard, WPS_HOME/config/deployed, WPS_HOME/config//deployable/ear, WPS_HOME/config/gather/deployed, WPS_HOME/config/work, and WPS_HOME/config/tmp subdirectories
- WPS_HOME/shared/app/config: entire subdirectory
- WPS_HOME/shared/app/wmm/wmm.xml
- WPS_HOME/version: entire subdirectory *except* WPS_HOME/version/backup subdirectory
- WPS_HOME/wmm: entire subdirectory
- WPS_HOME/wcm/config: all files (WebSphere Portal v5.x only)
- WPS_HOME/wcm/connect/log: all files (WebSphere Portal v5.x only)
- WPS_HOME/wcm/migration: all files (WebSphere Portal v6.x only)
- WPS_HOME/wcm/shared: all files (WebSphere Portal v6.x only)

- DMGR_HOME/logs: entire subdirectory

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME /config/cells: entire subdirectory *except* WAS_PROFILE_HOME/config/cells/*/applications subdirectories other than WAS_PROFILE_HOME/config/cells/${CellName}/applications
- WAS_PROFILE_HOME/config/.repository: entire subdirectory
- WAS_PROFILE_HOME /config/wmm: entire subdirectory
- WAS_PROFILE_HOME/logs: entire subdirectory *except* WAS_PROFILE_HOME/logs/ffdc subdirectory and WAS_PROFILE_HOME/logs/lost+found subdirectory

- WAS_PROFILE_HOME /properties: all files
- WAS_PROFILE_HOME /properties/version: all files
- WAS_PROFILE_HOME /properties/version/history: all files
- WAS_PROFILE_HOME /properties/version/update/backup: all XML files

The following files are collected on iSeries only:

- ISERIES_USER_INSTALL_HOME/config/cells: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/config/cells/*/applications subdirectories other than ISERIES_USER_INSTALL_HOME/config/cells/${CellName}/applications
- ISERIES_USER_INSTALL_HOME/config/.repository: entire subdirectory
- ISERIES_USER_INSTALL_HOME/config/wmm: entire subdirectory
- ISERIES_USER_INSTALL_HOME/logs: entire subdirectory *except* ISERIES_USER_INSTALL_HOME/logs/ffdc subdirectory and ISERIES_USER_INSTALL_HOME/logs/lost+found subdirectory
- ISERIES_USER_INSTALL_HOME/properties: all files
- ISERIES_USER_INSTALL_HOME/properties/version: all files
- ISERIES_USER_INSTALL_HOME/properties/version/history: all files
- ISERIES_USER_INSTALL_HOME/properties/version/update/backup: all XML files
- WPS_ISERIES_PRODDATA_HOME/logs: all files

Trace Enablement Settings

For WebSphere Portal v5.1:

In WPS_HOME/wcm/config/connect.cfg

TraceLevel=3 and Buffered=false for all values in:

```
<LogManager>
  <ErrorLog>
    <File LogFile="../connect/log/error.log" FlushLog=false Buffered=false TraceTime=true
TraceDate=true TraceLevel=3 Rollover=Size MaxFileSize=5M />^M
  </ErrorLog>
  <FullLog>
    <File LogFile="../connect/log/connect.log" FlushLog=false Buffered=true TraceTime=true
TraceLevel=1 TraceDate=true TraceThread=true Rollover=Size MaxFileSize=5M />
    <Screen Buffered=false TraceTime=true TraceLevel=0 TraceDate=true TraceThread=true />
  </FullLog>
  <DebugLog>
    <File LogFile="../connect/log/debug.log" FlushLog=false Buffered=false TraceTime=true
TraceLevel=0 Rollover=Size MaxFileSize=5M>^M
      <Packages>
        <!-- by default not debugging any package -->
      </Packages>
    </File>
  </DebugLog>
</LogManager>
```

AND

DebugMode value=true for:

```
<Syndication>
  <DebugMode value=false />
  <Subscriber value=com.aptrix.deployment.subscriber.PlutoSubscriberInterface />
  <Syndicator value=com.aptrix.deployment.syndicator.PlutoSyndicatorInterface />
</Syndication>
```

Portal tracing level setting for WebSphere Portal v5.1:

```
com.aptrix.deployment.*=all=enabled:
com.aptrix.syndication.*=all=enabled
```

For WebSphere Portal v6.0:

In WPS_HOME/wcm/shared/app/config/wcmservices/WCMConfigService.properties

replace

```
connect.moduleconfig.syndication.debugmode=false
```

with

```
connect.moduleconfig.syndication.debugmode=true
```

Portal tracing level setting for WebSphere Portal v6.0:

```
com.aptrix.deployment.*=all=enabled:
com.aptrix.syndication.*=all=enabled
```

Symptom Analysis Problem Type

portalcommon, wcm_syndication

Portal Problem Analysis Report

This is a special-case problem type that invokes the AutoPD symptom analysis function against an existing set of files. No collection zip file is created for this problem type – the tool simply produces an analysis report and makes it available to the user.

This problem type can prove useful in cases where WebSphere Portal log files have been submitted to IBM Support via means other than the AutoPD tool. In such cases, the IBM Support personnel can invoke AutoPD symptom analysis, and use it to extract the relevant log records from the files that were submitted.

Usage Scenario

This problem type has its own usage scenario, the problem analysis scenario.

List of Files Collected

As indicated earlier, there is no collection zip file for this problem type. The results of the symptom analysis are saved locally in the RasGUI_HOME/tmp/autopd directory, with a filename constructed as follows:

```
autopd_analysis_report_[problem-type]_[timestamp].html
```

Trace Enablement Settings

No traces are enabled for this problem type.

Symptom Analysis Problem Type

There is no pre-assigned value for this problem type. You instead choose the value at run-time, from the following list of alternatives:

Portal Install Problem	portalinstall
Portlet Deployment Problem	portletdeployment
Portal Config Problem	portalconfig
Portal General Problem	portalgeneral
Portal Upgrade Problem	portalupgrade
Portal Access Control Problem	portalaccesscontrol
Portal Login Problem	portallogin
Portal Manage Users and Groups Problem	portalmanageusersandgroups
Portal Start/Stop Problem	portalstartstop
Portal XML Configuration Interface Problem	portalxmlintf
Portal Integration with Tivoli Access Manager Problem	portaltamintegration
Common Base Event Analysis	cbeanalysis

Portal Collect Product Information

This is a special-case problem type that invokes the AutoPD product data collection function for the system where AutoPD is running.

This problem type can prove useful in cases where you need to know the exact maintenance levels of the various IBM software products that are installed on your system.

NOTE: This collection type is not supported on the iSeries platform.

Usage Scenario

This problem type has its own usage scenario, the product data collection scenario.

List of Files Collected

There is no collection zip file for this problem type. The tool creates an analysis report containing only the product information that usually appears at the bottom of these reports. This analysis report is saved locally in the RasGUI_HOME/tmp/autopd directory, under the filename autopd_analysis_report.html.

Trace Enablement Settings

No traces are enabled for this problem type.

Symptom Analysis Problem Type

There is no AutoPD symptom analysis associated with this problem type.

Appendix B: Collection Details for Each WebSphere Application Server Problem Type

The following sections provide specific details of what the tool does for each of the WebSphere Application Server problem types it supports. Each section contains information that falls into four categories:

1. The usage scenario for that problem type, described above in **Usage Scenarios**.
2. The list of files collected by the tool and included in the collection zip file for that problem type.
3. The trace enablement settings, if traces are enabled for that problem type.
4. The problem type value used when symptom analysis is invoked.

Note that the lists of files shown for the various problem types represents the *maximum* sets of files that may be gathered for each problem type – in some case, only a subset of the indicated files actually exist. Also note that the first five files listed do not come from a fixed location. Depending on whether the WebSphere Application Server is part of a cluster, and if so, depending on which cluster member is the subject of the collection, these files may come from a variety of locations. Note finally that the collections are set up in such a way that .jar files are *excluded* from the collections, since they can greatly increase the size of the collection zip file without adding any real value to it. The only .jar file that will be included in the collection zip file is the file that that collector script outputs.

For the files related to WebSphere Application Server, there are three locations from which files are typically collected:

1. WAS_HOME indicates the home directory where WebSphere Application Server was installed. Files in this location are collected for all releases of WebSphere Application Server.
2. WAS_PROFILE_HOME, which applies only to WebSphere Application Server v6 and above, indicates the home directory for the particular profile that WebSphere Portal is using. This directory is often in the location WAS_HOME/profiles/<profile-name>.
3. ISERIES_USER_INSTALL_HOME, which applies only to the OS/400 operating system on the iSeries hardware platform, indicates the user-installation home directory that WebSphere Portal is using.

Common to all problem types except for WebSphere Application Server Analysis Report and WebSphere Application Server Collect Product Information is the option of automatically sending the collection zip file to IBM Support via FTP. You will always be given the option of bypassing this step, if this is not appropriate for your specific environment.

WebSphere Application Server General Problem

This problem type is used to collect traces for general WebSphere Application Server problems.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Trace Enablement Settings

Traces are set by the user for this problem type.

Symptom Analysis Problem Type

was-general

WebSphere Application Server Java Message Service Problem

This problem type is used to collect traces for problems related to the Java Message Service components in WebSphere Application Server.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log

- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Trace Enablement Settings

The following traces are enabled for this problem type:

WebSphere Application Server tracing level setting:

```
JMSApi=all=enabled:
JMSServer=all=enabled:
JMSQueueManager=all=enabled:
Messaging=all=enabled:
com.ibm.ejs.jts.*=all=enabled
```

Note: These trace setting strings are contiguous; they have been reformatted here for readability.

Java Virtual Machine traceing level settings:

```
-DMQJMS_TRACE_LEVEL=base
```

Symptom Analysis Problem Type

was-jms

WebSphere Application Server Object Request Broker Problem

This problem type is used to collect traces for problems related to the Object Request Broker components in WebSphere Application Server.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt

- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

- RasGUI_HOME/tmp/autopd/orbtrc*
- RasGUI_HOME/tmp/autopd/orb.general.txt
- RasGUI_HOME/tmp/autopd/orb.interop.txt

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Trace Enablement Settings

The following traces are enabled for this problem type:

Websphere Application Server tracing level settings:

ORBRas=all=enabled

Java Virtual Machine tracing level settings:

-Dcom.ibm.CORBA.Debug=true -Dcom.ibm.CORBA.CommTrace=true

Symptom Analysis Problem Type

was-orb

WebSphere Application Server Database Connection Pooling Problem

This problem type is used to collect traces for problems related to database connection from WebSphere Application Server.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties
- RasGUI_HOME/tmp/DBconnection: entire subdirectory

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Trace Enablement Settings

The following traces are enabled for this problem type:

Websphere Application Server tracing level settings for database connection pooling for Application Server V5:

```
RRA=all=enabled:
WAS.j2c=all=enabled:
com.ibm.ejs.jts.*=all=enabled:
com.ibm.ws.LocalTransaction.*=all=enabled:
EJBContainer=all=enabled:
PMGR=all=enabled
```

Websphere Application Server tracing level settings for database connection pooling for Application Server V6:

```
RRA=all:
WAS.j2c=all:
com.ibm.ws.Transaction.*=all=enabled:
com.ibm.ws.LocalTransaction.*=all=enabled
EJBContainer=all:
PMGR=all
```

Websphere Application Server tracing level settings for JMS connection pooling for Application Server V5:

Messaging=all=enabled:
JMSApi=all=enabled:
com.ibm.ejs.jts.*=all=enabled:
com.ibm.ws.LocalTransaction.*=all=enabled

Websphere Application Server tracing level settings for JMS connection pooling for Application Server V5:

Messaging=all:
JMSAPI=all:
com.ibm.ws.Transaction.*=all=enabled:
com.ibm.ws.LocalTransaction.*=all=enabled

Websphere Application Server tracing level settings for EIS connection pooling for Application server V5:

com.ibm.ejs.jts.*=all=enabled:
com.ibm.ws.LocalTransaction.*=all=enabled

Websphere Application Server tracing level settings for EIS connection pooling for Application server V5:

com.ibm.ws.Transaction.*=all=enabled:
com.ibm.ws.LocalTransaction.*=all=enabled

Note: These trace setting strings are contiguous; they have been reformatted here for readability.

Symptom Analysis Problem Type

was-dbconnection

WebSphere Application Server Security Problem

This problem type is used to collect traces for problems related to security in WebSphere Application Server.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- collector.[sh/bat] output

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

- RasGUI_HOME/tmp/autopd/RequiredInfoFile.txt

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Trace Enablement Settings

The following traces are enabled:

WebSphere Application Server trace level settings for problems not related to SAS, ORB, or COMM (WebSphere Application Server v5):

```
com.ibm.ws.security.*=all=enabled
```

WebSphere Application Server trace level settings for problems related to SAS, ORB, or COMM (WebSphere Application Server v5):

```
SASRas=all=enabled
com.ibm.ws.security.*=all=enabled
ORBRas=all=enabled
```

Note: These trace setting strings are contiguous; they have been reformatted here for readability.

WebSphere Application Server trace level settings for WebSphere Application Server v6:

```
*=info:com.ibm.ws.security.*=all
```

Java Virtual Machine trace level settings problems related to SAS, ORB, or COMM (WebSphere Application Server v5:

```
-Dcom.ibm.CORBA.Debug=true -Dcom.ibm.CORBA.CommTrace=true
```

Symptom Analysis Problem Type

was-security

WebSphere Application Server Start Problem

This problem type is used to collect traces for problems related to starting the WebSphere Application Server.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- collector.[sh/bat] output

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Trace Enablement Settings

The following trace settings are enabled:

WebSphere Application Server trace level settings for Application Server v5:

*=all=enabled

WebSphere Application Server trace level settings for Application Server v6:

=info:=all

Symptom Analysis Problem Type

was-start

WebSphere Application Server Stop Problem

This problem type is used to collect traces for problems related to stopping WebSphere Application Server.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- collector.[sh/bat] output

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

In addition to these common files, various OS-specific files for Linux, AIX, and Solaris are also included in the collection.

Trace Enablement Settings

The following traces are enabled:

WebSphere Application Server trace level settings for Application Server v5:

```
*=all=enabled
```

WebSphere Application Server trace level settings for Application Server v6

*=info: *=all

Symptom Analysis Problem Type

was-stop

WebSphere Application Server Synchronization Problem on Deployment Manager

This problem type is used to collect traces for problems on the server deployment manager related to synchronization of the WebSphere Application Server deployment manager and nodes.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- collector.[sh/bat] output

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Trace Enablement Settings

The following traces are enabled:

WebSphere Application Server tracing level settings:

Sync=all

Symptom Analysis Problem Type

was-synchronization

WebSphere Application Server Synchronization Problem on Node Agent

This problem type is used to collect traces on the server node agent for problems related to synchronizing the WebSphere Application Server deployment manager and nodes.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- collector.[sh/bat] output

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory

- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Trace Enablement Settings

The following traces are enabled

WebSphere Application Server tracing level settings:

Sync=all:Admin=all

Symptom Analysis Problem Type

was-synchronization

WebSphere Application Server Session Management Problem

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Trace Enablement Settings

The following traces are enabled:

WebSphere Application Server tracing level settings if memory persistence is enabled:

```
com.ibm.ws.webcontainer.httpsession.*=all=enabled:
com.ibm.ws.webcontainer.srt.*=all=enabled:
com.ibm.ws.drs.*=all=enabled
```

WebSphere Application Server tracing level settings if memory persistence is disabled:

```
com.ibm.ws.webcontainer.httpsession.*=all=enabled:
com.ibm.ws.webcontainer.srt.*=all=enabled
```

Note: These trace setting strings are contiguous; they have been reformatted here for readability.

Symptom Analysis Problem Type

was-session

WebSphere Application Server Out of Memory Problem

This problem type is used to collect traces for problems related to WebSphere Application Server running out of memory.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties
- RasGUI_HOME/tmp/autopd/historyReport.html

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- All javacores and heapdumps (these files are contained within a separate collection zip file – PMRFILENAME_heapdump.zip)

The following files are collected on Windows:

- Perfmon logs
- User.dmp

The following files are collected on AIX:

- Ouput from memory leak script

The following files are collected on Linux:

- Mtrace.log
- Linux_memory_leak.sh log file

The following files are collected on Solaris:

- jtc*.log

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Environment Variable Settings

No traces are enabled for this problem type, but the following environment variables are set.

Environment Entries for WebSphere Application v5.x

IBM_HEAPDUMP	true
IBM_HEAP_DUMP	true
IBM_HEAPDUMPDIR	*your_directory*
IBM_HEAPDUMP_OUTOFMEMORY	true
IBM_JAVADUMP_OUTOFMEMORY	true

Environment Entries for WebSphere Application Server v6

IBM_HEAPDUMP	true
IBM_HEAP_DUMP	true
IBM_HEAPDUMPDIR	*your_directory*
IBM_HEAPDUMP_OUTOFMEMORY	true
IBM_JAVADUMP_OUTOFMEMORY	true
IBM_JAVA_HEAPDUMP_TEXT	true

Environment Entries for Linux

MALLOCTRACE	true
MALLOC_TRACE	true

JVM arguments for Solaris

```
-XX:+PrintGCDetails
-XX:+PrintGCTimeStamps
-XX:+PrintHeapAtGC
```

Symptom Analysis Problem Type

WAS_Memory_Problem

WebSphere Application Server Memory Heap Leak Problem

This problem type is used to collect traces for problems related to a memory heap leak.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties
- RasGUI_HOME/tmp/autopd/historyReport.html

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- All javacores and heapdumps (these files are contained within a separate collection zip file – PMRFILENAME_heapdump.zip)

The following files are collected on AIX:

- Ouput from memory leak script

The following files are collected on Linux:

- Linux_memory_leak.sh log file

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Environment Variable Settings

No traces are enabled for this problem type, but the following environment variables are set.
Environment Entries for WebSphere Application Server v5.x

IBM_HEAPDUMP	true
IBM_HEAP_DUMP	true
IBM_HEAPDUMPDIR	*your_directory*
IBM_HEAPDUMP_OUTOFMEMORY	true
IBM_JAVADUMP_OUTOFMEMORY	true

Environment Entries for WebSphere Application Server v6

IBM_HEAPDUMP	true
IBM_HEAP_DUMP	true
IBM_HEAPDUMPDIR	*your_directory*
IBM_HEAPDUMP_OUTOFMEMORY	true
IBM_JAVADUMP_OUTOFMEMORY	true
IBM_JAVA_HEAPDUMP_TEXT	true

Environment Entries for Linux

MALLOCTRACE	true
MALLOC_TRACE	true

Symptom Analysis Problem Type

WAS_Memory_Heap_Problem

WebSphere Application Server Memory Native Leak Problem

This problem type is used to collect traces for problems relating to native memory leak problem for WebSphere Application Server.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties
- RasGUI_HOME/tmp/autopd/historyReport.html

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log
- All javacores and heapdumps (these files are contained within a separate collection zip file – PMRFILENAME_heapdump.zip)

The following files are collected on Windows:

- Perfmon logs
- User.dmp

The following files are collected on AIX:

- Ouput from memory leak script

The following files are collected on Linux:

- Mtrace.log
- Linux_memory_leak.sh log file

The following files are collected on Solaris:

- jtc*.log

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Trace Enablement Settings

No trace strings or environment variables are enabled for this collection.

Symptom Analysis Problem Type

WAS_Memory_Native_Problem

Auxiliary WebSphere Application Server Collection for HTTP Server Problem

This problem type is used to collect traces for problems on the WebSphere Application Server.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

Trace Enablement Settings

The following traces are enabled:

WebSphere Application Server tracing level settings:

```
Servlet_Engine=all=enabled:
HTTP_Transport=all=enabled
```

Note: These trace setting strings are contiguous; they have been reformatted here for readability.

HTTP.conf file tracing level settings:

```
LogLevel=debug
```

HTTP Plugin Configuration tracing level settings:

```
LogLevel=trace
```

Symptom Analysis Problem Type

There is no AutoPD symptom analysis associated with this problem type.

WebSphere Application Server Crash Set Core Problem

This problem type is used to perform setup activities that will make it possible in the future to collect cores for problems related to WebSphere Application Server crashes.

NOTE: This collection type is not supported on the iSeries and HP-UX platforms.

Usage Scenario

The system environment modification scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc

In addition to these common files, various OS-specific files for Linux, AIX, Solaris, and Windows are also included in the collection.

Trace Enablement Settings

No traces are enabled for this problem type.

Symptom Analysis Problem Type

was-nocore

WebSphere Application Server Crash Collect Core Problem

This problem type is used to collect cores for problems related to WebSphere Application Server crashes.

NOTE: This collection type is not supported on the iSeries and HP-UX platforms.

Usage Scenario

The basic log collection scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/errorMsgInfo.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files

In addition to these common files, various OS-specific files for Linux, AIX, Solaris, and Windows are also included in the collection.

Trace Enablement Settings

No traces are enabled for this problem type.

Symptom Analysis Problem Type

was-collect-core

WebSphere Application Server Hang Problem

This problem type is used for hang problems related to WebSphere Application Server.

NOTE: This collection type is not supported on the iSeries and HP-UX platforms.

Usage Scenario

The problem reproduction scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- native_stderr.log
- native_stdout.log
- SystemErr.log
- SystemOut.log
- trace.log

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/tmp/autopd/historyReport.html
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties

- core*.* from the temporary install directory
- javacore*.* from the JVM working directory

WebSphere Application Server v5 only:

- WAS_HOME/logs/addNode.log
- WAS_HOME/logs/removeNode.log
- WAS_HOME/logs/nodeAgent.log
- WAS_HOME/logs/${ServerName}: entire subdirectory
- WAS_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_HOME/config: entire subdirectory, excluding .ear files
- WAS_HOME/*.trc
- WAS_HOME/bin/core*.*
- WAS_HOME/bin/javacore*.*
- WAS_HOME/javacore*.*

WebSphere Application Server v6 only:

- WAS_PROFILE_HOME/logs/addNode.log
- WAS_PROFILE_HOME/logs/removeNode.log
- WAS_PROFILE_HOME/logs/nodeAgent.log
- WAS_PROFILE_HOME/logs/${ServerName}: entire subdirectory
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_exception.log
- WAS_PROFILE_HOME/logs/ffdc/${ServerName}_[0-9].* (latest file only)
- WAS_PROFILE_HOME/config: entire subdirectory, excluding .ear files
- WAS_PROFILE_HOME/*.trc
- WAS_PROFILE_HOME/bin/core*.*
- WAS_PROFILE_HOME/bin/javacore*.*
- WAS_PROFILE_HOME/javacore*.*

If IBM HTTP Server resides on the same system as the WebSphere Application Server, the following files will also be collected:

- plugin-cfg.xml (directory location varies)
- http_plugin.log(directory location varies)

In addition to these common files, various OS-specific files for Linux, AIX, and Solaris are also included in the collection.

Trace Enablement Settings

No traces are enabled for this problem type.

Symptom Analysis Problem Type

was-collect-hangs

WebSphere Application Server RAS Collector

This is a special-case problem type that invokes the WebSphere Application Server RAS Collect utility, and then includes the zip file it produces in the usual AutoPD collection zip file. AutoPD symptom analysis is performed on the usual set of WebSphere Application Server log files.

Usage Scenario

The basic log collection scenario is used for this problem type.

List of Files Collected

The following files are collected on all operating systems:

- RasGUI_HOME/tmp/autopd/autopd_analysis_report.html
- RasGUI_HOME/tmp/autopd/autoPD-feedback.txt
- RasGUI_HOME/log: entire subdirectory
- RasGUI_HOME/topology.properties
- collector.[sh/bat] output

Trace Enablement Settings

No traces are enabled for this problem type.

Symptom Analysis Problem Type

invokerasutil

WebSphere Application Server Analysis Report

This is a special-case problem type that invokes the AutoPD symptom analysis function against an existing set of files. No collection zip file is created for this problem type – the tool simply produces an analysis report and makes it available to the user.

This problem type can prove useful in cases where WebSphere Application Server log files have been submitted to IBM Support via means other than the AutoPD tool. In such cases, the IBM Support personnel can invoke AutoPD symptom analysis, and use it to extract the relevant log records from the files that were submitted.

Usage Scenario

This problem type has its own usage scenario, the problem analysis scenario.

List of Files Collected

As indicated earlier, there is no collection zip file for this problem type. The results of the symptom analysis are saved locally in the RasGUI_HOME/tmp/autopd directory, with a filename constructed as follows:

autopd_analysis_report_[problem-type]_[timestamp].html

Trace Enablement Settings

No traces are enabled for this problem type.

Symptom Analysis Problem Type

There is no pre-assigned value for this problem type. You instead choose the value at run-time, from the following list of alternatives:

• WebSphere Application Server JMS Problem	was-jms
• Websphere Application Server ORB Problem	was-orb
• Websphere Application Server Database Connection Problem	was-dbconnection
• Websphere Application Server Start Problem	was-start
• Websphere Application Server Stop Problem	was-stop
• Websphere Application Server Security Problem	was-security
• Websphere Application Server Syncronization Problem	was-syncronization
• Websphere Application Server Crash Set Core Problem	was-nocore
• Websphere Application Server Crash Collect Core Problem	was-collect-core
• Websphere Application Server Hang Problem	was-collect-hangs
• Websphere Application Server Session Management Problem	was-session

WebSphere Application Server Collect Product Information

This is a special-case problem type that invokes the AutoPD product data collection function for the system where AutoPD is running.

This problem type can prove useful in cases where you need to know the exact maintenance levels of the various IBM software products that are installed on your system.

NOTE: This collection type is not supported on the iSeries platform.

Usage Scenario

This problem type has its own usage scenario, the product data collection scenario.

List of Files Collected

There is no collection zip file for this problem type. The tool creates an analysis report containing only the product information that usually appears at the bottom of these reports. This analysis report is saved locally in the RasGUI_HOME/tmp/autopd directory, under the filename autopd_analysis_report.html.

Trace Enablement Settings

No traces are enabled for this problem type.

Symptom Analysis Problem Type

There is no AutoPD symptom analysis associated with this problem type.

Appendix C: Known Limitations

The following list identifies the limitations we are aware of in the current version of the AutoPD tool:

1. Because of a Swing bug in the 1.4 level of the JDK, the directory browser dialogues must be used in a certain way in order to return the correct paths. Suppose, for example, you want to navigate to a WebSphere Portal root directory located at `C:\ibm\PortalServer`. In order to get Swing to return this directory location, you must choose the parent directory `C:\ibm` as the location to look in, and highlight the subdirectory `PortalServer` as the "file" to be selected. With the current bug, there is no way to select the directory location `C:\ibm\PortalServer`, and highlight nothing under it. If you attempt to do this, the value returned will be `C:\ibm\PortalServer\PortalServer`, with the final subdirectory appearing twice in the hierarchy.
2. Because of platform differences, copy and paste operations in the GUI mode must be done differently for Windows and for other platforms. For Windows, after selecting the text you want to copy, you must use ctrl-v to paste it into the text box presented to you on the user interface. For the Linux, AIX, and Solaris platforms, you must first highlight the text with the left mouse button, and then paste the text into the text box using the middle mouse button.
3. Because the collection scripts do not use the WebSphere Application Server and WebSphere Portal administrator credentials until some time after you provide them, they are not able to provide you with an immediate indication if you enter them incorrectly. The following scenarios may occur.

    - Incorrect administrator credentials for WebSphere Application Server: while a collection script can detect and react to a server's rejection of the credentials you have supplied, it cannot determine the validity of the credentials when the application server it is attempting to present them to is down. This becomes problematic in a collection scenario where an application server must first be started and then later be stopped. Since WebSphere Application Server requires credentials for stopping an application server but not for starting one, a collection script may start a server, and then subsequently discover that it cannot stop it because it has rejected the credentials you have supplied. In this case the collection script will display a message indicating that you must stop the application server manually if you need to return it to the stopped state.

    - Incorrect administrator credentials for WebSphere Portal: when you are invoking an xmlaccess command to collect topology data, you will see an element similar to this one in the exportconfig_output.xml document in your collection zip file:

      ```
      <failure>
      com.ibm.wps.command.xml.XmlCommandServlet$AuthenticationException:
      EJPXA0011E: Authentication for user wpsadmin failed.
      </failure>
      ```

4. The function that hides sensitive information in the collection zip file is unable to process XML documents that do not contained well-formed XML. Consequently, these files are left unchanged in the collection zip file. Since, however, it is unlikely that a password value contained in an ill-formed XML document could be used successfully by a product for any authentication activity, the likelihood of disclosing an actual password in the collection zip file is small.

Appendix D: Change/Revision History

Revision level, date, and a basic explanation are listed below. Content changes from the previous to current level are indicated by a list of changes made.

VERSION	DATE	EXPLANATION
V1.0	12/24/2004	Initial Version of tool
V1.1	1/18/2005	Updated GUI, updated Portal Install Script
V1.1.1	3/18/2005	Enhanced cluster support, Collect Product Information
V1.1.2	5/23/2005	Enhanced Portal Config and Portal General support, updated FTP support, added log analysis script, added filterByLatestTime, added message catalog support, added Common Base Event support
V1.1.3	7/22/2005	Added WebSphere Application Server v6 support, enhanced Portlet Install support, added scripts for Portal Upgrade, Portal Access Control, and Portal Manage Users and Groups, modified directory structure in collection zip files to match structure on source system
V1.1.4	10/18/2005	Added scripts for Portal Start/Stop, Portal XML Configuration Interface, Portal Integration with Tivoli Access Manager, WebSphere Application Server Crash Set Core, WebSphere Application Server Crash Collect Core, and WebSphere Application Server Hang problems; added full logging support, including registered message identifiers (CWPDC prefix), separation of autopd.log and autopdecho.log, and creation of separate log files for each script invocation; added support for vertical clusters; added analysis for XML-based event.history log files for both WebSphere Application Server and Portal./
V1.1.5	11/11/2005	Generalized installation information and added WebSphere Application Server script details for the WebSphere Application Server version of the tool.
V1.2	3/31/2006	Introduced tree-based menus to both the GUI and console modes, making it possible to deliver a single AutoPD tool containing scripts for multiple products. Added improved feedback and validation for FTP processing. Added automatic generation of Log and Trace Analyzer manifest files for AutoPD collection zip files. Introduced fourteen additional scripts for WebSphere Application Server, in addition to the three that were previously available in v1.1.4. These three scripts were also updated in this release.
V1.2.1	8/15/2006	Added scripts for Portal Document Manager, Portal Personalization, and Web Content Manager. Extended all WebSphere Portal scripts to support WebSphere Portal v6. Made miscellaneous fixes and other updates to WebSphere Portal and WebSphere Application Server scripts, as well as to the AutoPD engine. Added support for a second version of the Log and Trace Analyzer manigest document. Added support for retrieving iSeries CCSIDs. Introduced the Incremental Analysis function. Made several enhancements to AutoPD logging. Added support for the HP-UX platform.
V1.2.2	1/4/2007	Improved some of the user interactions related to product install roots and related properties. Added the directory containing the updatePortal scripts to the Portal Upgrade collection. Added a dialog to all problem-recreation collections for WebSphere Portal to allow the user to provide a custom trace string (often supplied by IBM Support) in addition to or in place of the collection's customary trace string. Added several functions to support graphical display of logical topology for WebSphere Application Server and WebSphere Portal. Added the ability to invoke topology collection to several WebSphere Portal scripts. Limited the size of Portal collection zip files by systematically excluding .ear and .war files in addition to .jar files. Renamed the Portlet Install collection to Portlet Deployment, to eliminate confusion with Portal Install. Changed the preferred delimiter in collection zip file names from comma to period. Added three additional collections for WebSphere Application Server. Added a new RasGUI download file specifically for iSeries. Made miscellaneous fixes and other updates to WebSphere Portal collections.

V1.2.2.1	2/1/2007	Fixed several small bugs in the Portal collections: failing to exclude .ear, .jar, and .war files from some subdirectories; failure to return the Portal Server to its initial state in some cases; handling of WebSphere Application Server profile root in some configurations. Improved the user feedback when incorrect WebSphere Application Server administrator credentials are entered. Added functions to blank out passwords in configuration files contained in the collection zip file. Updated the analysis function to deal with the timestamped trace logs introduced in WebSphere Portal v6. Retired the collection "Invoke WebSphere Portal RAS Collect Tool", directing the user to use the Portal General collection instead. Improved the handling of WebSphere Portal versions in the collection scripts. Enhanced the WebSphere Application Server memory collections to use previously set environment variables to locate heap dumps and javacores.
V1.2.2.2	3/30/2007	Fixed several bugs in the scripts and in the custom Ant tasks. Moved some of the WebSphere Portal analysis results into a separate document linked to the main analysis report. Attached a suffix to the collection zip file name to enable tracking of AutoPD-created zip files.