WebSphere® Process Server for Multiplatforms
Version 6.2.0.3

# *Troubleshooting and Support*

WebSphere® Process Server for Multiplatforms
Version 6.2.0.3

# *Troubleshooting and Support*

**Note**

Before using this information, be sure to read the general information in the Notices section at the end of this document.

**25 June 2010**

This edition applies to version 6, release 2, modification 0 of WebSphere Process Server for Multiplatforms (product number 5724-L01) and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, send an e-mail message to doc-comments@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

# PDF books and the information center

PDF books are provided as a convenience for printing and offline reading. For the latest information, see the online information center.

As a set, the PDF books contain the same content as the information center.

The PDF documentation is available within a quarter after a major release of the information center, such as Version 6.0 or Version 6.1.

The PDF documentation is updated less frequently than the information center, but more frequently than the Redbooks®. In general, PDF books are updated when enough changes are accumulated for the book.

Links to topics outside a PDF book go to the information center on the Web. Links to targets outside a PDF book are marked by icons that indicate whether the target is a PDF book or a Web page.

*Table 1. Icons that prefix links to topics outside this book*

| Icon | Description |
|---|---|
|  | A link to a Web page, including a page in the information center.<br><br>Links to the information center go through an indirection routing service, so that they continue to work even if target topic is moved to a new location.<br><br>If you want to find a linked page in a local information center, you can search for the link title. Alternatively, you can search for the topic id. If the search results in several topics for different product variants, you can use the search result **Group by** controls to identify the topic instance that you want to view. For example:<br>1. Copy the link URL; for example, right-click the link then select Copy link location. For example: `http://www14.software.ibm.com/webapp/wsbroker/redirect?version=wbpm620&product=wesb-dist &topic=tins_apply_service`<br>2. Copy the topic id after `&topic=`. For example: `tins_apply_service`<br>3. In the search field of your local information center, paste the topic id. If you have the documentation feature installed locally, the search result will list the topic. For example:<br>`1 result(s) found for`<br><br>`Group by: None \| Platform \| Version \| Product`<br>`Show Summary`<br><br>`Installing fix packs and refresh packs with the Update Installer`<br><br>4. Click the link in the search result to display the topic. |
|  | A link to a PDF book. |

# Contents

# Chapter 1. Overview of troubleshooting

Troubleshooting is a systematic approach to solving a problem. The goal is to determine why something does not work as expected and how to resolve the problem.

The first step in the troubleshooting process is to describe the problem completely. Without a problem description, neither you nor IBM® can know where to start to find the cause of the problem. This step includes asking yourself basic questions, such as:

- What are the symptoms of the problem?
- Where does the problem occur?
- When does the problem occur?
- Under which conditions does the problem occur?
- Can the problem be reproduced?

The answers to these questions typically lead to a good description of the problem, and that is the best way to start down the path of problem resolution.

## What are the symptoms of the problem?

When starting to describe a problem, the most obvious question is "What is the problem?" This might seem like a straightforward question; however, you can break it down into several more-focused questions that create a more descriptive picture of the problem. These questions can include:

- Who, or what, is reporting the problem?
- What are the error codes and messages?
- How does the system fail? For example, is it a loop, hang, crash, performance degradation, or incorrect result?
- What is the business impact of the problem?

## Where does the problem occur?

Determining where the problem originates is not always easy, but it is one of the most important steps in resolving a problem. Many layers of technology can exist between the reporting and failing components. Networks, disks, and drivers are only a few components to be considered when you are investigating problems.

The following questions can help you to focus on where the problem occurs in order to isolate the problem layer.

- Is the problem specific to one platform or operating system, or is it common for multiple platforms or operating systems?
- Is the current environment and configuration supported?

Remember that if one layer reports the problem, the problem does not necessarily originate in that layer. Part of identifying where a problem originates is understanding the environment in which it exists. Take some time to completely describe the problem environment, including the operating system and version, all corresponding software and versions, and hardware information. Confirm that you are running within an environment that is a supported configuration; many

problems can be traced back to incompatible levels of software that are not intended to run together or have not been fully tested together.

## When does the problem occur?

Develop a detailed timeline of events leading up to a failure, especially for those cases that are one-time occurrences. You can most easily do this by working backward: Start at the time an error was reported (as precisely as possible, even down to the millisecond), and work backward through the available logs and information. Typically, you need to look only as far as the first suspicious event that you find in a diagnostic log; however, this is not always easy to do and takes practice. Knowing when to stop looking is especially difficult when multiple layers of technology are involved, and when each has its own diagnostic information.

To develop a detailed timeline of events, answer the following questions:

- Does the problem happen only at a certain time of day or night?
- How often does the problem happen?
- What sequence of events leads up to the time that the problem is reported?
- Does the problem happen after an environment change, such as upgrading or installing software or hardware?

Responding to these types of questions can provide you with a frame of reference in which to investigate the problem.

## Under which conditions does the problem occur?

Knowing what other systems and applications are running at the time that a problem occurs is an important part of troubleshooting. These and other questions about your environment can help you to identify the root cause of the problem:

- Does the problem always occur when the same task is being performed?
- Does a certain sequence of events need to occur for the problem to surface?
- Do any other applications fail at the same time?

Answering these types of questions can help you explain the environment in which the problem occurs and correlate any dependencies. Remember that just because multiple problems might have occurred around the same time, the problems are not necessarily related.

## Can the problem be reproduced?

From a troubleshooting standpoint, the "ideal" problem is one that can be reproduced. Typically with problems that can be reproduced, you have a larger set of tools or procedures at your disposal to help you investigate. Consequently, problems that you can reproduce are often easier to debug and solve. However, problems that you can reproduce can have a disadvantage: If the problem is of significant business impact, you do not want it to recur! If possible, re-create the problem in a test or development environment, which typically offers you more flexibility and control during your investigation.

**Tip:** Simplify the scenario to isolate the problem to a suspected component.

The following questions can help you with reproducing the problem:

- Can the problem be re-created on a test machine?
- Are multiple users or applications encountering the same type of problem?

- Can the problem be re-created by running a single command, a set of commands, a particular application, or a stand-alone application?

# Chapter 2. Troubleshooting checklist for WebSphere Process Server

Asking questions about hardware and software requirements, product fixes, specific problems, error messages, and diagnostic data can help you troubleshoot WebSphere® Process Server.

The following questions can help you to identify the source of a problem that is occurring with WebSphere Process Server:

1. Is the configuration supported?

   Refer to the requirements for WebSphere Process Server to ensure that your system meets all hardware, operating system, and software requirements: WebSphere Process Server system requirements Web site.
2. Have you applied the latest fixes?
3. What is the problem?
   - Installing and configuring WebSphere Process Server
   - Migrating existing applications and configuration information to WebSphere Process Server
   - Deploying applications on WebSphere Process Server
   - Administering applications and components on WebSphere Process Server
   - Using WebSphere Application Server capabilities in WebSphere Process Server
4. Have any error messages been issued?
5. For additional help in finding error and warning messages, interpreting messages, and configuring log files, see Diagnosing problems with message logs in the Websphere Application Server information center.
6. Difficult problems can require the use of tracing, which exposes the low-level flow of control and interactions between components. For help in understanding and using traces, see Working with trace in the Websphere Application Server information center.
7. If the checklist does not guide you to a resolution, you can collect additional diagnostic data. This data is necessary for IBM Support to effectively troubleshoot and assist you in resolving the problem. For more information, see Chapter 16, “Contacting IBM Software Support,” on page 113.

# Chapter 3. Messages overview

When you receive a message from WebSphere Process Server, you can often resolve the problem by reading the entire message text and the recovery actions that are associated with the message.

You can find the full text of runtime messages, their explanations, and the recommended recovery actions by searching for the message identifier in the Messages section of the WebSphere Process Server Reference documentation.

Messages displayed during WebSphere Process Server product installation and profile creation are documented in topics under Related Topics at the bottom of this page..

Runtime message identifiers consist of a four- or five-character message prefix, followed by a four- or five-character message number, followed by a single-letter message type code. For example, zzzzL1042C. The message type code describes the severity of the error message, as follows:

**C** Indicates a severe message.

**E** Indicates an urgent message.

**I** Indicates an informational message.

**N** Indicates an error message.

**W** Indicates a warning message.

**Related reference**

"Messages: installation and profile creation" on page 16
Some of the most commonly found error messages encountered when installing and configuring can be addressed with actions that resolve the underlying problems.

# Chapter 4. WebSphere Process Server log files

There are two distinct groups of log files in the installed product. Logs detailing the product installation, product updates, and profile management are one group. Logs detailing the characteristics and runtime activities of individual profiles comprise the second group.

Various log files are created during the installation and uninstallation of WebSphere Process Server and during profile creation, augmentation, and deletion. Examine these logs when problems occur during the product installation and configuration process. The log files and their locations within the product installation are detailed in the "Installation and profile creation log files" topic.

There are also a number of log files that are created for each profile. Some of these logs describe the parameters used for the creation of the profile. These types of log files generally remain unchanged when the profile is fully configured. Other profile-specific logs are continually updated to capture error, warning, and information messages emitted during runtime. Some of these log files are also used to capture a Common Base Event (that may include business object data) that is selected for monitoring. This set of logs is described in "Profile-specific log files" topic.

**Related tasks**

Installation and profile creation log files

**Related reference**

“Profile-specific log files” on page 47
There are log files detailing the characteristics and runtime activities of individual profiles. These log files are located within the profile directory for each profile.

# Chapter 5. Transaction log file

The transaction (tranlog) log file stores critical transactional data that is written to databases. It is an internal file that WebSphere Application Server uses to manage in-flight transactions and attempt to recover them should the server crash.

**DO NOT delete the transaction log file from a production environment**. Deleting this file removes information on in-flight transactions from WebSphere Process Server memory. Without the transaction log file, there is no functionality to recover transactional information. In addition, long-running processes remain in an inconsistent state and you cannot complete the process flow except by deleting running instances. Deleting running instances might cause you to lose operational or business-critical data, which makes the database inconsistent with the message destination. Other inconsistencies that may be caused by deleting the transaction log file includes the following:

- Started transactions will neither be rolled back nor committed
- Artifacts will remain in the Java™ Virtual Machine (JVM) since they are referenced or allocated by a transaction but never garbage collected
- Database content (amongst others navigation state of long running BPEL processes) remains in the Business Process Choreographer related tables and are never deleted
- Navigation messages of the Business Process Engine (BPE) of long running processes are never processed further
- Service Component Architecture (SCA) messages that belong to a process navigation and transaction remain on SCA-related queues

**Note:** Deleting the transaction log from a development environment causes the same problems. Because you can recreate business processes, deleting the files from a test environment is not as damaging as deleting them from a production environment.

# Chapter 6. Troubleshooting installation and configuration

You can diagnose problems when the installation and configuration of WebSphere Process Server is unsuccessful.

## About this task

The installer program records the following indicators at the end of the primary log file:

- `INSTCONFSUCCESS`: installation was successful
- `INSTCONFPARTIALSUCCESS`: installation was partly successful. Some installation actions failed but can be retried.
- `INSTCONFFAILED`: installation was not successful. Recovery is not possible.

The primary log file, `log.txt`, is located in *`install_root`*`/logs/wbi/install/log.txt` on i5/OS®, Linux®, and UNIX® platforms or *`install_root`*`\logs\wbi\install\log.txt` on Windows® platforms, where *install_root* represents the product installation directory.

If the result is `INSTCONFPARTIALSUCCESS` or `INSTCONFFAILED`, continue analyzing the problem by following these steps. (For details on uninstalling any installed portions before reinstalling, see Preparing for reinstallation after a failed uninstallation.)

**Procedure**

## Procedure

1. Read any error messages from the installation process.

   See the following topic for an explanation: Error messages: installation and profile creation and augmentation. If the message corresponds to any of those described, correct the problem, clean up any installed portions, and try to reinstall.
2. Determine which product caused the partial success or failure. Review the `install_error.log` file located in the *`install_root`*`/logs/wbi/install` directory on Linux, UNIX, and i5/OS platforms or the *`install_root`*`\logs\wbi\install\` directory on Windows platforms. This file logs errors, warnings, and installation results extracted from the log files for installation of WebSphere Application Server Network Deployment, WebSphere Feature Pack for Web Services, and WebSphere Process Server. If profile creation failed or was partially successful during an installation, the results are extracted from the profile log files and included in this file as well.

   **Tip:** If no files exist in the *`install_root`*`/logs/wbi/install` directory, installation failed early in the process. See the log files in *`user_home`*`/wbilogs` instead.

   Perform one of the following tasks depending on which product did not install properly:

   - If WebSphere Application Server Network Deployment did not install properly, go to step 3 on page 14.

- If WebSphere Feature Pack for Web Services did not install properly (and WebSphere Application Server Network Deployment did), go to step 4.
- If WebSphere Process Server did not install properly (and WebSphere Application Server Network Deployment and WebSphere Feature Pack for Web Services did), go to step 5.

3. If the installation of WebSphere Application Server Network Deployment was not successful, review the `install_error.log` file for errors. If this file does not provide enough information to correct the problem, check Troubleshooting installation in the WebSphere Application Server Network Deployment information center and use the information found there to correct the problem before attempting to reinstall WebSphere Process Server.

   **Tip:** If a problem occurs during an installation of WebSphere Application Server Network Deployment as part of a WebSphere Process Server installation, the installation process will not continue and an error message will be displayed.

4. If the installation of WebSphere Feature Pack for Web Services was not successful (and installation of WebSphere Application Server Network Deployment was), review the `install_error.log` file for errors. If this file does not provide enough information to correct the problem, check Troubleshooting Web server plug-ins installation and removal in the WebSphere Application Server Network Deployment information center and use the information found there to correct the problem before attempting to reinstall WebSphere Process Server.

   **Tip:** If a problem occurs during an installation of WebSphere Feature Pack for Web Services as part of a WebSphere Process Server installation, the installation process will not continue and an error message will be displayed.

5. If the installation of WebSphere Process Server was not successful (and installation of WebSphere Application Server Network Deployment and WebSphere Feature Pack for Web Services were), review the `install_error.log` file for errors. If this file does not provide enough information to correct the problem, check other WebSphere Process Server installation log files. For information about the names, locations, and descriptions of these log files, see Installation and profile creation log files. Check the log files in this sequence:

   i5/OS **On i5/OS platforms:**

   a. log files in the *install_root*`/logs/wbi/install` directory
   b. log files in the *user_home*`/wbilogs` directory if no files are found in *install_root*`/logs/wbi/install`
   c. *user_data_root*`/profileRegistry/logs/manageprofiles/`*profile_name*`_create_error.log`
   d. *user_data_root*`/profileRegistry/logs/manageprofiles/`*profile_name*`_create.log` and *user_data_root*`/profileRegistry/logs/manageprofiles/pmt.log`
   e. *install_root*`/logs/wbi/installconfig.log` (indicates configuration problems that can prevent the product from working correctly). For more information about diagnosing failed configuration scripts, see "Diagnosing a failing Ant configuration script" on page 25.
   f. Any additional log or trace files generated by installation actions. Look in *install_root*`/logs/wbi/install` for trace files generated during the installation process. Look in *user_data_root*`/profileRegistry/logs/`

manageprofiles/*profile_name* for those generated by profile creation or augmentation. (For more information about *install_root* and *user_data_root* locations, see Default installation directories for the product, profiles, and tools.) These files are primarily intended for use by IBM technical support.

Linux UNIX **On Linux and UNIX platforms:**

a. log files in the *install_root*/logs/wbi/install directory
b. log files in the *user_home*/wbilogs directory if no files are found in *install_root*/logs/wbi/install
c. *install_root*/logs/manageprofiles/*profile_name*_create_error.log
d. *install_root*/logs/manageprofiles/*profile_name*_create.log and *install_root*/logs/manageprofiles/pmt.log
e. *install_root*/logs/wbi/installconfig.log (indicates configuration problems that can prevent the product from working correctly). For more information about diagnosing failed configuration scripts, see "Diagnosing a failing Ant configuration script" on page 25.
f. Any additional log or trace files generated by installation actions. Look in *install_root*/logs/wbi/install for trace files generated during the installation process. Look in *install_root*/logs/manageprofiles/*profile_name* for those generated by profile creation or augmentation. (For more information about *install_root* and *profile_root* locations, see Default installation directories for the product, profiles, and tools.) These files are primarily intended for use by IBM technical support.

Windows **On Windows platforms:**

a. log files in the *install_root*\logs\wbi\install directory
b. log files in the *user_home*\wbilogs directory if no files are found in *install_root*\logs\wbi\install
c. *install_root*\logs\manageprofiles*profile_name*_create_error.log
d. *install_root*\logs\manageprofiles*profile_name*_create.log and *install_root*\logs\manageprofiles\pmt.log
e. *install_root*\logs\wbi\installconfig.log (indicates configuration problems that can prevent the product from working correctly). For more information about diagnosing failed configuration scripts, see "Diagnosing a failing Ant configuration script" on page 25.
f. Any additional log or trace files generated by installation actions. Look in *install_root*\logs\wbi\install for trace files generated during the installation process. Look in *install_root*\logs\manageprofiles*profile_name* for those generated by profile creation or augmentation. (For more information about *install_root* and *profile_root* locations, see Default installation directories for the product, profiles, and tools.) These files are primarily intended for use by IBM technical support.

6. If the error logs do not contain enough information to determine the cause of the problem, uninstall the product, clean up any log files or other artifacts that are left behind, turn on tracing, and reinstall.
   - Report the stdout and stderr logs to the console window by adding the **-is:javaconsole** parameter to the install command:
     – i5/OS **On i5/OS platforms:**

       ```
       install -is:javaconsole
       ```

       Capture the stream to a file with the following commands:

       ```
       install -is:javaconsole > captureFileName.txt 2>&1
       ```

     – Linux UNIX **On Linux and UNIX platforms:**

```
install -is:javaconsole
```

Capture the stream to a file with the following commands:

```
install -is:javaconsole > captureFileName.txt 2>&1
```

– Windows **On Windows platforms:**

```
install.exe -is:javaconsole
```

Capture the stream to a file with the following commands:

```
install.exe -is:javaconsole > drive:\captureFileName.txt
```

- Capture additional information to a log of your choice with the `-is:log file_name` option.

7. If you have successfully created a server profile, use the First steps console or the command-line method to start the server.
8. Verify that the server starts and loads properly by looking for a running Java process and the *Open for e-business* message in the `SystemOut.log` and `SystemErr.log` files.

   If no Java process exists or if the message is not displayed, examine the same logs for any miscellaneous errors. Correct any errors and try again.

   You can find the `SystemOut.log` and `SystemErr.log` files in the following platform-specific directories:

   - i5/OS **On i5/OS platforms:** *profile_root*`/logs/`*servername*
   - Linux UNIX **On Linux and UNIX platforms:** *profile_root*`/logs/`*servername*
   - Windows **On Windows platforms:** *profile_root*`\logs\`*servername*
9. Use the First steps console or the command-line method to stop the server, if it is running.
10. To troubleshoot a WebSphere Process Server deployment environment, see Verifying your deployment environment.
11. If you want to use a Snoop Servlet to verify the ability of the Web server to retrieve an application from WebSphere Process Server, see the step "Start the Snoop servlet to verify the ability of the Web server to retrieve an application from the Application Server" in Troubleshooting installation in the WebSphere Application Server Network Deployment documentation.
12. Start the administrative console. For more information, see Starting and stopping the administrative console.
13. To resolve any IP address caching problems, see the step "Resolve any IP address caching problems" in Troubleshooting installation in the WebSphere Application Server Network Deployment documentation.

### What to do next

On the product support Web site, you can review current information about resolutions to known problems, and you can read documents that can save you time gathering the information that you need to resolve a problem. Before opening a PMR, see the IBM WebSphere Process Server support page.

## Messages: installation and profile creation

Some of the most commonly found error messages encountered when installing and configuring can be addressed with actions that resolve the underlying problems.

**Note:** Linux UNIX Windows The following WebSphere Process Server installation and configuration errors appear on Linux, UNIX, and Windows platforms.

**Tip:** For information about messages that might be generated by the installation of WebSphere Application Server Network Deployment, refer to the Business Process Management messages topic.

What kind of problem are you having while installing WebSphere Process Server?

- "Supported IBM JDK was not found. The IBM JDK shipped with this product must be located at *install_root*/JDK. Please correct this problem and try again."
- "Warning: Cannot convert string "<type_name>"to type FontStruct"

If you do not see an error message that resembles yours, or if the information provided does not solve your problem, contact WebSphere Process Server support at IBM for further assistance.

**Related tasks**

Chapter 3, "Messages overview," on page 7
When you receive a message from WebSphere Process Server, you can often resolve the problem by reading the entire message text and the recovery actions that are associated with the message.

## Supported IBM JDK was not found. The IBM JDK shipped with this product must be located at *install_root*/JDK. Please correct this problem and try again.

If you use symbolic links to point to the IBM Java Development Kit (JDK) shipped with the product, or to a JDK found in the PATH environment variable on your system, IBM SDK for Java validation might fail, resulting in a failed installation. This problem is caused by the way IBM SDK for Java validation code detects whether the JDK shipped with the product is the current JDK used for installation.

To resolve this problem, do not use symbolic links in JVMs supplied with the installation image of WebSphere Process Server and remove symbolic links from all JVMs that appear in your system's PATH environment variable.

## Warning: Cannot convert string "<type_name>"to type FontStruct

If you install the Web server plug-ins for WebSphere Application Server, you also install the ikeyman utility. The ikeyman utility is part of the Global Services Kit 7 (GSKit7).

Linux If you issue the `ikeyman.sh` script on a Linux system, you might see the following message:

```
Warning: Cannot convert string
   "-monotype-arial-regular-r-normal--*-140-*-*-p-*-iso8859-1"
   to type FontStruct
```

You can safely ignore the warning and use the ikeyman utility.

## Installation and profile creation log files

Various log files are created during installation and uninstallation of WebSphere Process Server and during profile creation, augmentation, and deletion. Consult the applicable logs if problems occur during these procedures.

Table 2 shows the logs, content, and indicators of success and failure for WebSphere Process Server.

If the `logs` directory does not exist on your system, the installation failed very early in the process. In this case, review the following:

- Linux UNIX **On Linux and UNIX platforms:** *user_home*/wbilogs
- Windows **On Windows platforms:** *user_home*\wbilogs
- i5/OS **On i5/OS platforms:** *user_home*/wbilogs

Some directory paths, file names, and indicator values in Table 2 contain spaces to allow the entries to fit in the table cells. The actual directory paths, file names, and indicator values do not contain spaces.

The variable *install_root* represents the installation directory of WebSphere Process Server. The variable *profile_root* represents the root location of a profile.

i5/OS **On i5/OS platforms:** The variable *user_data_root* represents the default user data directory.

For more information see Default installation directories for the product, profiles, and tools.

*Table 2. Installation and profile logs for WebSphere Process Server components*

| Log | Content | Indicators |
|---|---|---|
| • Linux UNIX *install_root*/logs/wbi/install/log.txt<br>• Windows *install_root*\logs\wbi\install\log.txt<br>• i5/OS *install_root*/logs/wbi/install/log.txt | Logs all installation events relating to WebSphere Process Server. | **INSTCONFFAILED**<br>Total installation failure.<br>**INSTCONFSUCCESS**<br>Successful installation.<br>**INSTCONFPARTIALSUCCESS**<br>Installation errors occurred but the installation is still usable. Additional information in other log files identifies the errors. |
| • Linux UNIX *install_root*/logs/wbi/install/install_error.log<br>• Windows *install_root*\logs\wbi\install\install_error.log<br>• i5/OS *install_root*/logs/wbi/install/install_error.log | Logs errors, warnings, and installation results extracted from the log files for installation of WebSphere Application Server Network Deployment, WebSphere Feature Pack for Web Services, and WebSphere Process Server. If profile creation failed or was partially successful during an installation, the results are extracted from the profile log files and included in this file as well. | N/A |

*Table 2. Installation and profile logs for WebSphere Process Server components (continued)*

| Log | Content | Indicators |
|---|---|---|
| • Linux UNIX *install_root*/logs/wbi/installconfig.log<br>• Windows *install_root*\logs\wbi\installconfig.log<br>• i5/OS *install_root*/logs/wbi/installconfig.log | Logs configuration actions that run at the end of the installation process to configure components, install system applications, and create Windows shortcuts and registry entries. | Contains a series of <record> elements that document the configuration actions. If a post-installation configuration action fails, text like the following appears in the log:<br><record><br><date>2005-05-26T11:41:17</date><br><millis>1117132877344</millis><br><sequence>742</sequence><br><logger>com.ibm.ws.install.configmanager.ConfigManager</logger><br><level>WARNING</level><br><class>com.ibm.ws.install.configmanager.ConfigManager</class><br><method>executeAllActionsFound</method><br><thread>12</thread><br><message>Configuration action failed: com.ibm.ws.install.configmanager.actionengine.ANTAction-D:\WBI\AS\properties\version\install.wbi\6.1.0.0\config\full\install\90SInstallCEI.ant</message><br></record><br><br>If no actions fail, the following message is included in the record in the log:<br><record><br>. . .<br><message>Returning with return code: INSTCONFSUCCESS</message><br></record> |
| • Linux UNIX *install_root*/logs/manageprofiles/pmt.log<br>• Windows *install_root*\logs\manageprofiles\pmt.log<br>• i5/OS *user_data_root*/profileRegistry/logs/manageprofiles/pmt.log | Logs all events from the Profile Management Tool. | **INSTCONFFAILED**<br>Total profile creation failure.<br>**INSTCONFSUCCESS**<br>Successful profile creation.<br>**INSTCONFPARTIALSUCCESS**<br>Profile creation errors occurred but the profile is still functional. Additional information in other log files identifies the errors. |
| • Linux UNIX *install_root*/logs/manageprofiles/*profile_name*_create.log<br>• Windows *install_root*\logs\manageprofiles*profile_name*_create.log<br>• i5/OS *user_data_root*/profileRegistry/logs/manageprofiles/*profile_name*_create.log | • Traces all events that occur during the creation of the named profile.<br>• Created when a profile is created during a Complete installation, when using the Profile Management Tool, or when using the manageprofiles command. | **INSTCONFFAILED**<br>Total profile creation failure.<br>**INSTCONFSUCCESS**<br>Successful profile creation.<br>**INSTCONFPARTIALSUCCESS**<br>Profile creation errors occurred but the profile is still functional. Additional information in other log files identifies the errors. |

*Table 2. Installation and profile logs for WebSphere Process Server components (continued)*

| Log | Content | Indicators |
| --- | --- | --- |
| • Linux UNIX *install_root*/logs/manageprofiles/*profile_name*_create_error.log<br>• Windows *install_root*\logs\wbi\update*profile_name*_create_error.log<br>• i5/OS *install_root*/logs/wbi/update/*profile_name*_create_error.log | Logs information extracted from the*profile_name*_create.log file. This information pertains to any failing configuration actions, validations, wsadmin calls and or any corresponding log files. | N/A |
| • Linux UNIX *install_root*/logs/manageprofiles/*profile_name*_augment.log<br>• Windows *install_root*\logs\manageprofiles*profile_name*_augment.log<br>• i5/OS *user_data_root*/profileRegistry/logs/manageprofiles/*profile_name*_augment.log | • Traces all events that occur during the augmentation of the named profile.<br>• Created when a profile is augmented, when using the Profile Management Tool, or when using the manageprofiles command. | **INSTCONFFAILED**<br>Total profile augmentation failure.<br>**INSTCONFSUCCESS**<br>Successful profile augmentation.<br>**INSTCONFPARTIALSUCCESS**<br>Profile augmentation errors occurred but the profile is still functional. Additional information in other log files identifies the errors. |
| • Linux UNIX *install_root*/logs/manageprofiles/*profile_name*_augment_error.log<br>• Windows *install_root*\logs\wbi\update*profile_name*_augment_error.log<br>• i5/OS *install_root*/logs/wbi/update/*profile_name*_augment_error.log | Logs information extracted from the *profile_name*_augment.log file. This information pertains to any failing configuration actions, validations, wsadmin calls and any corresponding log files. | N/A |
| • Linux UNIX *install_root*/logs/manageprofiles/*profile_name*_delete.log<br>• Windows *install_root*/logs/manageprofiles/*profile_name*_delete.log<br>• i5/OS *user_data_root*/profileRegistry/logs/manageprofiles/*profile_name*_delete.log | • Traces all events that occur during the deletion of the named profile.<br>• Created when profile deletion is performed with the manageprofiles command. | **INSTCONFFAILED**<br>Total profile deletion failure.<br>**INSTCONFSUCCESS**<br>Successful profile deletion.<br>**INSTCONFPARTIALSUCCESS**<br>Profile deletion errors occurred but the profile is still deleted. Additional information in other log files identifies the errors. |

*Table 2. Installation and profile logs for WebSphere Process Server components (continued)*

| Log | Content | Indicators |
| --- | --- | --- |
| • Linux UNIX *install_root*/logs/install/log.txt<br>• Windows *install_root*\logs\install\log.txt<br>• i5/OS *install_root*/logs/wbi/install/log.txt | • Logs all installation events relating to WebSphere Application Server Network Deployment.<br>• Created as part the underlying installation of WebSphere Application Server Network Deployment that is installed with WebSphere Process Server. | **INSTCONFFAILED**<br>Total installation failure.<br>**INSTCONFSUCCESS**<br>Successful installation.<br>**INSTCONFPARTIALSUCCESS**<br>Installation errors occurred but the installation is still usable. Additional information in other log files identifies the errors. |
| • Linux UNIX *install_root*/logs/installconfig.log<br>• Windows *install_root*\logs\installconfig.log<br>• i5/OS *install_root*/logs/wbi/installconfig.log | • Logs configuration actions that run at the end of the installation process to configure components, install system applications, and create Windows shortcuts and registry entries.<br>• Created as part the underlying installation of WebSphere Application Server Network Deployment that is installed with WebSphere Process Server. | Contains a series of <record> elements that document the configuration actions. |
| • Linux UNIX *install_root*/logs/wbi/uninstall/log.txt<br>• Windows *install_root*\logs\wbi\uninstall\log.txt<br>• i5/OS *install_root*/logs/wbi/uninstall/log.txt | Logs all uninstallation events relating to WebSphere Process Server. | **INSTCONFFAILED**<br>Total uninstallation failure.<br>**INSTCONFSUCCESS**<br>Successful uninstallation.<br>**INSTCONFPARTIALSUCCESS**<br>The uninstallation wizard successfully removed the core product files, but errors occurred during configuration. Additional information in other log files identifies the errors. |
| • Linux UNIX *install_root*/logs/wbi/update/updateconfig.log<br>• Windows *install_root*\logs\wbi\update\updateconfig.log<br>• i5/OS *install_root*/logs/wbi/update/updateconfig.log | Logs configuration actions that run at the end of the uninstallation process. | Contains a series of <record> elements that document the configuration actions. |
| • i5/OS *%TEMP%*\firststeps_i5.log | Logs errors that can occur when running the First steps console and provides suggestions on how to fix them. | If you experience any unexpected or erroneous behavior from the First steps console, check this log file. It is particularly useful if you run the First steps console from the command line, because of the chance of typographical errors. |

## Troubleshooting the launchpad application

If the launchpad application does not start, try the following troubleshooting tips.

Restart the launchpad after you make any changes.

- If you are using images from Passport Advantage®, make sure that you extract the contents of the images for *WebSphere Process Server V6.2 DVD*, *WebSphere Application Server Network Deployment Supplements V6.1* CD, and *WebSphere Application Server Toolkit V6.1.1 Disk 1* (if included for your platform) into three separate directories. Extracting the files from the images into the same directory will cause errors to occur. It is recommended that you use three sibling directories. For example, use a set of directories such as the following:

  **Note:** i5/OS The installation images obtained from Passport Advantage must be downloaded to a Windows workstation.

  – i5/OS

  ```
  %/downloads/WPS/image1
  %/downloads/WPS/image2
  %/downloads/WPS/image3
  ```

  – Linux UNIX

  ```
  %/downloads/WPS/image1
  %/downloads/WPS/image2
  %/downloads/WPS/image3
  ```

  – Windows

  ```
  C:\downloads\WPS\image1
  C:\downloads\WPS\image2
  C:\downloads\WPS\image3
  ```

- If you can start the launchpad, but selecting a link does not resolve to a page in the launchpad, you might have the media for the wrong operating system in the disk drive. Check the validity of the media.

- Windows If you are attempting to use the Mozilla browser on a Windows system, Internet Explorer might open instead. The launchpad does not recognize Mozilla as the default browser if Internet Explorer is also installed on the same system. The launchpad is fully functional with Internet Explorer, so no action is required.

  To create an environment variable that forces the use of Mozilla, issue the following case-specific command at a command prompt:

  ```
  set BROWSER=Mozilla
  ```

- Ensure that the JavaScript™ function is enabled in your browser.

  Linux UNIX Mozilla: Click **Edit > Preferences > Advanced > Scripts & Plugins**:

  – Enable JavaScript for: Navigator.

  – Allow scripts to ... (Select all boxes.)

  Linux UNIX Mozilla Firefox: Click **Tools > Options > Content**:

  – Select **Enable Java**.

  – Select **Enable JavaScript**.

  – Click **Advanced** and Allow scripts to ... (Select all boxes.)

  Windows Internet Explorer: Click **Tools > Internet Options > Security > Custom Level for Internet > Scripting > Active scripting > Enable**.

If the launchpad links still do not work after trying these tips, start the component installation programs directly. The locations of these programs are listed in Options on the launchpad.

# Troubleshooting a silent installation

If a silent installation using a response file fails, you can examine log files and error messages to determine what went wrong, and make changes to your response file.

## Before you begin

For information about using the response file for a silent installation of WebSphere Process Server, see Installing silently.

To troubleshoot a silent product installation, perform the following steps.

**Procedure**

## Procedure

1. Check your response file to make sure you are precise when supplying option values in the file so that the installation program can read the values. Incorrect specifications affect the silent interface of the installation wizard. For example, always use the correct case within property names, which are case-sensitive. In addition, always enclose values in double quotation marks. If the error is an incorrect option value, the InstallShield MultiPlatform program displays a warning message that you must confirm and stops the installation.
2. Compare your response file to the `responsefile.wbis.txt` file that is shipped with the product to make the necessary corrections. This file is in the *install_image*/WBI directory. After correcting the file, reinstall.
3. Review commonly found error messages in Messages: installation and profile creation and augmentation.
4. Examine log files. See the descriptions of relevant log files listed in Installation and profile creation log files.
5. Certain events can prevent InstallShield MultiPlatform from starting the installation wizard silently (for example, not having enough disk space to launch the installation wizard). If your installation fails and there is no information in the installation logs, record entries for events that cause the ISMP program to fail to start the installation wizard.

   The syntax of the install command for logging such events is as follows:

   AIX **On AIX® platforms:**

   ```
   install -options "/usr/IBM/WebSphere/silentFiles/myresponsefile.txt"
           -silent -log
   ```

   HP-UX Solaris **On HP-UX and Solaris platforms:**

   ```
   install -options "/opt/IBM/WebSphere/silentFiles/myresponsefile.txt"
           -silent -log
   ```

   i5/OS **On i5/OS platforms:**

   ```
   install -options responsefile.wbis.txt -silent -log log.txt @ALL
   ```

   **Note:** i5/OS **On i5/OS platforms:** You must change to the directory that contains the copied DVD image. Example: */MYDIR*/WBI

   Linux **On Linux platforms:**

```
install -options "/opt/ibm/WebSphere/silentFiles/myresponsefile.txt"
        -silent -log
```

Windows **On Windows platforms:**

```
install.exe -options "C:\IBM\WebSphere\silentFiles\myresponsefile.txt"
            -silent -log # !C:\IBM\WebSphere\silentFiles\log.txt  @ALL
```

6. For other tips on troubleshooting your installation, see Troubleshooting installation.
7. If your profile did not create successfully, see Recovering from profile creation or augmentation failure.

## i5/OS installation troubleshooting tips

You can refer to sources that might be helpful in troubleshooting an installation problem for a WebSphere Process Server product on the i5/OS operating system.

WebSphere Process Server offers several methods you can use to troubleshoot problems. Which method you use depends on the nature of the problem. Generally, you use a combination of these methods to determine the cause of a problem and then decide on an appropriate method for its resolution.

### Tip 1: Refer to troubleshooting documentation for WebSphere Application Server for i5/OS

These resources provide general troubleshooting assistance:

- WebSphere Process Server Release Notes.
- WebSphere Application Server FAQ database.
- WebSphere Application Server for OS/400® newsgroup. This System i® Technical Support Web-based forum is dedicated to WebSphere Application Server for i5/OS and OS/400.

### Tip 2: Install WebSphere Process Server Version 6.2 for i5/OS

- **Wrong version of i5/OS installed on your server.**

  WebSphere Process Server runs on i5/OS V5R4 and V6R1. The product cannot be installed on prior releases of i5/OS.
- **IBM Development Kit for Java V1.5 is not installed.**

  Local and remote command-line installations require JDK 1.5. Install product 5722-JV1, option 7 to obtain JDK 1.5. After installing option 7, you should reinstall the cumulative PTF package and Java group PTF to pick up any JDK 1.5 specific fixes.
- **Host servers are not started, or failed to start correctly.**

  The installation process requires that the i5/OS host servers be running. To start the host servers, run this command from the CL command line.

  ```
  STRHOSTSVR SERVER(*ALL)
  ```

  If errors other than "Host server daemon jobs unable to communicate using IPX." occur when starting the host servers, follow the instructions in the error message to fix the problem. After the problem is fixed, start the host servers and attempt to install WebSphere Process Server Server again.
- **Installation fails due to "Object not found" or "Not authorized" errors.**

  The user profile of the user installing the product must have *ALLOBJ and *SECADM special authorities.

### Tip 3: Start WebSphere Process Server for i5/OS

- **Port conflicts**

  Port conflicts may exist if you have multiple stand-alone installations of WebSphere Application Server or multiple installations of stack products that bundle WebSphere Application Server like WebSphere Enterprise Service Bus or WebSphere Process Server on the same physical i5/OS machine.

## Diagnosing a failing Ant configuration script

Determine whether a product installation problem on an operating system such as AIX, Linux, Windows, or i5/OS is caused by a failing Apache Ant configuration script.

### Before you begin

Start diagnosing installation problems by looking at the troubleshooting procedure. See Troubleshooting installation. After the installation completes successfully, several Ant scripts configure the product. The following procedure describes what to do when an Ant script fails. When the installation log does not indicate a failure, determine how to correct any problems with failing Ant configuration scripts.

### About this task

The *install_root*/logs/wbi/installconfig.log file, when present, describes any failure of an Ant script. Determine if any of the following configuration scripts failed. If so, use the configuration script recovery procedures. Use the investigative action to manually verify that the following configuration scripts ran successfully during the configuration of the WebSphere Process Server product. If any script failed, use the recovery action steps to complete the function of the script.

To diagnose failed Ant configuration scripts, perform the following steps.

### Procedure

- Diagnose the failed 90SConfigWBIMigrationScript.ant configuration script. This script changes the permissions of the following script to 755: *install_root*/bin/wbi_migration. This script also replaces the following tokens in the *install_root*/bin/wbi_migration script:

| From: | To the value that you selected during installation: |
|---|---|
| ${JAVAROOT} | *install_root*/java/jre/bin/java |
| ${MIGRATIONJAR} | *install_root*/bin/migration/migrationGUI/migrationGUI.jar |
| ${WASROOT} | *install_root* |
| ${PRODUCTID} | ${WS_CMT_PRODUCT_TYPE} |

1. Investigative action: Verify that the permissions are 755 for the *install_root*/bin/wbi_migration.sh script on Linux and UNIX platforms, *install_root*\bin\wbi_migration.bat on Windows platforms, or for the *install_root*/bin/wbi_migration script on i5/OS platforms.
2. Recovery action: Issue the following command: chmod 755 *install_root*/bin/wbi_migration.sh on Linux and UNIX platforms, chmod

755 *install_root*\bin\wbi_migration.bat on Windows platforms or chmod 755 *install_root*/bin/wbi_migration on i5/OS platforms.

3. Investigative action: Open the *install_root*/bin/wbi_migration.sh on Linux and UNIX platforms, *install_root*\bin\wbi_migration.bat on Windows platforms, or *install_root*/bin/wbi_migration script on i5/OS platforms in an editor and verify that real values exist instead of the following values: ${JAVAROOT}, ${MIGRATIONJAR}, ${WASROOT}, and ${PRODUCTID}.
4. Recovery action: Change the following tokens to actual values in the wbi_migration script: ${JAVAROOT}, ${MIGRATIONJAR}, ${WASROOT}, and ${PRODUCTID}.

- Diagnose the failed 85SConfigNoProfileFirstStepsWBI.ant. This script copies all files from the *install_root*/properties/version/install.wbi/firststeps.wbi directory to the *install_root*/firststeps/wbi/html/noprofile directory. This script also replaces the following tokens in the *install_root*/firststeps/wbi/firststeps.sh script (Linux, and UNIX), the *install_root*\firststeps\wbi\firststeps.bat script (Windows platforms), or the *install_root*/firststeps/wbi/firststeps script (i5/OS platforms):

| From: | To the value that you selected during installation: |
|---|---|
| ${JAVAROOT} | *install_root*/java/jre/bin/java |
| ${PROFILEROOT} | *install_root* |
| ${HTMLSHELLJAR} | *install_root*/lib/htmlshellwbi.jar |
| ${CELLNAME} | ${WS_CMT_CELL_NAME} |

1. Investigative action: Verify that all files are copied from the *install_root*/properties/version/install.wbi/firststeps.wbi directory to the *install_root*/firststeps/wbi/html/noprofile directory.
2. Recovery action: Copy all of the files from the *install_root*/properties/version/install.wbi/firststeps.wbi directory to the *install_root*/firststeps/wbi/html/noprofile directory.
3. Investigative action: Open the *install_root*/firststeps/wbi/firststeps script in an editor. Verify that real values exist instead of the following values: ${JAVAROOT}, ${PROFILEROOT}, ${HTMLSHELLJAR}, and ${CELLNAME}.
4. Recovery action: Change the following tokens to actual values in the *install_root*/firststeps/wbi/firststeps script. ${JAVAROOT}, ${PROFILEROOT}, ${HTMLSHELLJAR}, and ${CELLNAME}.

### Results

After you correct any installation errors and any Ant script configuration errors by performing the corrective actions in this procedure, the installation is complete.

### What to do next

Start the First steps console.

## Recovering from profile creation or augmentation failure

The Profile Management Tool can experience failures when creating new or augmenting existing profiles. The same can occur using the manageprofiles command. If such a failure occurs, first check the log files as described in this topic, then follow the recovery instructions described, depending on the situation.

## Log files

All manageprofiles log files are in *install_root*/logs/manageprofiles. Look at the following log files in the order given. Each log file must contain the entry "INSTCONFSUCCESS." If a file does not include this entry, a failure was detected. Look at the log files to determine why a failure was encountered and to determine a remedy.

1. The log file *profile_name*_create_error.log (where *profile_name* is the name of the profile).

   **Note:** Look at this file only if you were creating a new profile, not augmenting an existing one.

   - Linux UNIX *install_root*/logs/manageprofiles/*profile_name*_create_error.log
   - Windows *install_root*\logs\wbi\update*profile_name*_create_error.log
   - i5/OS *install_root*/logs/wbi/update/*profile_name*_create_error.log

   Search for the text Configuration action succeeded or Configuration action failed.

   **Note:** There can be multiple occurrences of Configuration action failed. Investigate and remedy each one. Also review the log files described in the following options, if the profile was created.

   **Note:** Additional information is available in the manageprofiles directory in the pmt.log, which logs all events that occur when a default profile is created during a complete installation using the Profile Management Tool.
2. Log file *profile_name*_augment_error.log (where *profile_name* is the name of the profile).

   This log file is located in the following directories:

   - Linux UNIX *install_root*/logs/manageprofiles/*profile_name*_augment_error.log
   - Windows *install_root*\logs\wbi\update*profile_name*_augment_error.log
   - i5/OS *install_root*/logs/wbi/update/*profile_name*_augment_error.log

   Search for the text Configuration action succeeded or Configuration action failed.

   **Note:** There can be multiple occurrences of Configuration action failed. Investigate and remedy each one. Also review the log files described in the following options, if the profile was created.

   **Note:** If you want to know the status of a profile you created during installation, run the following commands:

   - Linux UNIX *install_root*/bin/logProfileErrors.sh
   - Windows *install_root*\bin\logProfileErrors.bat
   - i5/OS *install_root*/logProfileErrors
3. Individual profile template action log files.

   If you discovered false values in the log files described in the preceding options, review the log files in the following directories:

   - i5/OS *user_data_root*/profileregistry/logs on i5/OS systems

- Linux UNIX *install_root*/logs/manageprofiles/*profile_name* on Linux and UNIX systems
- Windows *install_root*\logs\manageprofiles*profile_name* on Windows systems

where *profile_root* or *user_data_root* is the installation location of the profile.

These log files do not follow a consistent naming convention, but typically, each is the name of the Apache Ant script that failed followed by .log. For example, suppose the following entry is in the *profile_name*_augment.log file:

```
<messages>Result of executing
E:\o0536.15\profileTemplates\default.wbicore\actions\saveParamsWbiCore.ant
was:false</messages>
```

First look at the surrounding entries in the *profile_name*_augment.log file in the *install_root*/logs/manageprofiles directory. If you cannot determine the cause of the failure from the surrounding entries, look for the corresponding log file for any failing Ant script entries. In this case, the log file created by the saveParamsWbiCore.ant script is saveParamsWbiCore.ant.log. Look at that file to investigate why the failure occurred.

## Recovery for creation failure

After you determine why profile creation failed and address the cause of the failure, you can try to create the profile again.

**Note:** When you create a profile, it first creates a WebSphere Application Server profile and then augments it with WebSphere Process Server profile templates to create a WebSphere Process Server profile. Even if you encountered a profile creation failure, a profile can exist that does not have all the needed augmentations.

To determine if the profile exists, run the *install_root*/bin/manageprofiles -listProfiles command. If the profile name you used for creation does not exist, you can recreate the profile. If the profile name you used for creation exists, then the profile was created and you have encountered an augmentation failure. For tips on recovering from an augmentation failure, see "Recovery for augmentation failure."

## Recovery for augmentation failure

After you determine why profile augmentation failed and address the cause of the failure, you can try to augment the existing profile again to successfully create a complete WebSphere Process Server profile by following these steps:

1. Start the Profile Management Tool and, instead of creating a new profile, choose to augment an existing profile.
2. Choose the profile you were working with, and enter the correct information for it.

**Note:** Some of the augmentations might have completed successfully the first time you ran the Profile Management Tool. As a result, you might not see all of the panels that you saw the first time you tried to create the profile. This is because the Profile Management Tool detects which remaining augmentations must be completed and displays only the necessary panels.

## Troubleshooting the Business Process Choreographer configuration

For information on how to solve problems relating to the configuration of Business Process Choreographer and its Business Flow Manager or Human Task Manager components, go to the WebSphere Process Server for Multiplatforms, version 6.2, information center and review the topics under **Installing and configuring WebSphere Process Server > Troubleshooting installation and configuration > Troubleshooting the Business Process Choreographer configuration**. You can also find this information in the *Business Process Choreographer PDF*.

# Chapter 7. Troubleshooting migration

If you encounter problems during migration, the information described here could help.

## Troubleshooting version-to-version migration

Review this page for troubleshooting tips if you encounter problems while migrating from an older version of WebSphere Process Server

The following sections describe specific errors and exceptions that may occur in a version-to-version migration and provide steps you can follow to understand and resolve these problems.



### Application installation error

If you select the option for the migration process to install the enterprise applications that exist in the version 6.1.x or 6.0.2.x configuration into the new version 6.2 configuration, you might encounter some error messages during the application-installation phase of migration.

The applications that exist in the version 6.1.x or 6.0.2.x configuration might have incorrect deployment information—usually, incorrect XML documents that were not validated sufficiently in previous WebSphere Process Server runtimes. The runtime now has an improved application-installation validation process and will fail to install these malformed EAR files. This results in a failure during the application-installation phase of WBIPostUpgrade and produces an "E:" error message.

If the application installation fails in this way during migration, you can do one of the following:

- Fix the problems in the version 6.1.x or 6.0.2.x applications, and then remigrate.
- Proceed with the migration and ignore these errors.

  In this case, the migration process does not install the failing applications but does complete all of the other migration steps.

  Later, you can fix the problems in the applications and then manually install them in the new version 6.2 configuration using the administrative console or an install script.

## Application server error

After you migrate a managed node to version 6.2, the application server might not start.

When you try to start the application server, you might see errors similar to those in the following example:

```
[5/11/06 15:41:23:190 CDT] 0000000a SystemErr  R
    com.ibm.ws.exception.RuntimeError:
com.ibm.ws.exception.RuntimeError:  org.omg.CORBA.INTERNAL:
   CREATE_LISTENER_FAILED_4
vmcid: 0x49421000  minor code: 56  completed: No
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr  R  at
com.ibm.ws.runtime.WsServerImpl.bootServerContainer(WsServerImpl.java:198)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr  R  at
com.ibm.ws.runtime.WsServerImpl.start(WsServerImpl.java:139)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr  R  at
com.ibm.ws.runtime.WsServerImpl.main(WsServerImpl.java:460)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr  R  at
com.ibm.ws.runtime.WsServer.main(WsServer.java:59)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr  R  at
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr  R  at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:64)
[5/11/06 15:41:23:197 CDT] 0000000a SystemErr  R  at
sun.reflect.DelegatingMethodAccessorImpl.invoke
    (DelegatingMethodAccessorImpl.java:43)
```

Change the port number at which the managed node's server is listening. If the deployment manager is listening at port 9101 for ORB_LISTENER_ADDRESS, for example, the server of the managed node should not be listening at port 9101 for its ORB_LISTENER_ADDRESS. To resolve the problem in this example, perform the following steps:

1. On the administrative console, click **Application servers** → *server_name* → **Ports** → **ORB_LISTENER_ADDRESS**.
2. Change the ORB_LISTENER_ADDRESS port number to one that is not used.

## Exceptions: database connectivity, loading, or missing class

Never change any WebSphere Application Server variables that are configured as a part of profile creation.

If you modify these values incorrectly in old profile, you might get database connectivity, loading, or other missing class exceptions, such as:

```
10/25/08 13:22:39:650 GMT+08:00] 0000002e J2CUtilityCla E   J2CA0036E: An
exception occurred while invoking method setDataSourceProperties on
com.ibm.ws.rsadapter.spi.WSManagedConnectionFactoryImpl used by resource
jdbc/com.ibm.ws.sib/ewps6101.Messaging-BPC.cwfpcCell01.Bus :
com.ibm.ws.exception.WsException: DSRA0023E: The DataSource implementation
class "com.ibm.db2.jcc.DB2XADataSource" could not be found.DB2,
```

Derby, and SQL Embedded JDBC drivers are bundled with the WebSphere Process Server product installation. If you need to change these drivers to any higher version, you must copy drivers on same location where they exists in the product installation, as follows:

- **Derby**: *%was.install.root%*\derby\lib

- **DB2**: *%was.install.root%*/universalDriver_wbi/lib
- **SQL**: *%was.install.root%*lib

If you need a new JDBC provider and datasource for your application, you can create these resources by selecting a valid jdbcclasspath and setting the WebSphere Application Server variable accordingly. For example, if you need DB2 at cell level which doesn't exist earlier in your installation, you could use the following procedure.

1. In the administrative console, navigate to: **Resources** → **JDBC** → **JDBC Providers** → **DB2 Universal JDBC Driver Provider (XA)**.
2. In the **Class path** box, set the following paths:
   - DB2UNIVERSAL_JDBC_DRIVER_PATH =%was.install.root%/ universalDriver_wbi/lib
   - DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH=""

   If you need your own drivers, set the following path: DB2UNIVERSAL_JDBC_DRIVER_PATH=%myDriverLocation%

## Out of memory error

If either the WBIPreUpgrade or WBIPostUpgrade command-line utility fail due to Out of Memory problems, you can increase the heap size to a number that takes into consideration the size and scope of the environment being migrated, as well as what the machine will allow.

For instructions on how to increase the heap size, use the procedure described in Solution 4 of the following technote: Handling certain Out of Memory conditions when migrating an earlier version of WebSphere Application Server to V6.0.2, V6.1, or 7.0.

## Profile creation error

While you are using the version 6.2 migration wizard to create a profile when migrating a configuration, you might see the following profile-creation error messages.

```
profileName: profileName cannot be empty
profilePath: Insufficient disk space
```

These error messages might be displayed if you enter a profile name that contains an incorrect character such as a space. Rerun the migration wizard, and verify that there are no incorrect characters in the profile name such as a space, quotes, or any other special characters.

## Profile migration error

When you use the migration wizard to migrate a profile from WebSphere Process Server version 6.1.x or 6.0.2.x to version 6.2 on a Solaris x64 processor-based system, the migration might fail during the WBIPostUpgrade step.

You might see messages similar to the following in *profile_root*/logs/WASPostUpgrade.*time_stamp*.log:

```
MIGR0327E: A failure occurred with stopNode.
MIGR0272E: The migration function cannot complete the command.
```

WebSphere Process Server version 6.1.x or 6.0.2.x uses a Java virtual machine (JVM) in 32-bit mode. The migration wizard for WebSphere Process Server version 6.2 calls the `WBIPostUpgrade.sh` script, which attempts to run the JVM for version 6.1.x or 6.0.2.x in the 64-bit mode when the server stops the version 6.1.x or 6.0.2.x node.

Complete the following actions to remove the incomplete profile and enable WebSphere Process Server to correctly migrate the version 6.1.x or 6.0.2.x profile:

1. On a command line, change to the *install_root*`/bin` directory.

   For example, type the following command:

   ```
   cd /opt/IBM/WebSphere/Procserver/bin
   ```

2. Locate the `WBIPostUpgrade.sh` script in the *install_root*`/bin` directory, and make a backup copy.
3. Open the `WBIPostUpgrade.sh` or `WBIPostUpgrade.bat` file in an editor, and perform the following actions:
   a. Locate the following line of code:

      UNIX Linux

      ```
      "$binDir" /setupCmdLine.sh
      ```

      Windows

      ```
      call "%~dp0setupCmdLine.bat" %*
      ```

   b. Insert the following line of code after the code that was identified in the previous step:

      ```
      JVM_EXTRA_CMD_ARGS=""
      ```

   c. Save the changes.
4. Repeat steps 2 through 4 with the`WASPostUpgrade.sh` or the `WASPostUpgrade.bat` file.
5. Delete the incomplete version 6.2 profile that was created during the migration process. Use the following procedure.
   a. Open a command prompt and run one of the following commands, based on your operating system:

      - i5/OS **On i5/OS platforms:** `manageprofiles -delete -profileName` *profile_name*
      - Linux UNIX **On Linux and UNIX platforms:** `manageprofiles.sh -delete -profileName` *profile_name*
      - Windows **On Windows platforms:** `manageprofiles.bat -delete -profileName` *profile_name*

      The variable *profile_name* represents the name of the profile that you want to delete.

   b. Confirm that the profile deletion has completed by checking the following log file:

      - i5/OS **On i5/OS platforms:** *user_data_root*`/profileRegistry/logs/manageprofiles/`*profile_name*`_delete.log`
      - Linux UNIX **On Linux and UNIX platforms:** *install_root*`/logs/manageprofiles/`*profile_name*`_delete.log`
      - Windows **On Windows platforms:** *install_root*`\logs\manageprofiles\`*profile_name*`_delete.log`
6. Delete the *profile_root* directory of the version 6.2 profile that was removed in the previous step.

7. Rerun the migration wizard.

### Servlet error

In a network deployment environment, if the error `SRVE0026E: [Servlet Error]-[com/ibm/wbiservers/brules/BusinessRuleManager]: java.lang.NoClassDefFoundError` occurs when you access the Business Rules Manager after migrating, you must manually install the Business Rules Manager application on the deployment target before continuing with normal migration of that node. See Migrating Business Rules Manager in a network deployment environment for more information.

### Synchronization error

If synchronization fails when you migrate a managed node to version 6.2, the server might not start.

You might receive messages similar to the following when you migrate a managed node to version 6.2:

```
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0111E: Program exiting with error:
           com.ibm.websphere.management.exception.AdminException: ADMU0005E:
           Error synchronizing repositories
ADMU0211I: Error details may be seen in the file:
           /opt/WebSphere/62AppServer/profiles/AppSrv02/logs/syncNode.log
MIGR0350W: Synchronization with the deployment manager using the SOAP protocol
 failed.
MIGR0307I: The restoration of the previous WebSphere Application Server
 environment is complete.
MIGR0271W: Migration completed successfully, with one or more warnings.
```

These messages indicate the following:

- Your deployment manager is at a version 6.2 configuration level.
- The managed node that you are trying to migrate is at a version 6.2 configuration level on the deployment manager's repository (including applications).
- The managed node itself is not quite complete given that you did not complete the syncNode operation.

Perform the following actions to resolve this issue:

1. Rerun the syncNode command on the node to synchronize it with the deployment manager.

   See the syncNode command .
2. Run the GenPluginCfg command.

   See the GenPluginCfg command .

## Troubleshooting migration from WebSphere InterChange Server or WebSphere Business Integration Server Express

Find solutions to problems you encounter with migration as well as instructions for turning on logging and tracing.

# Enabling logging and tracing for supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs

Enable logging and tracing for the supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs through the administrative console.

## About this task

If your migrated application includes any supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs, you can enable logging and tracing for them for troubleshooting purposes.

**Procedure**

## Procedure

1. Launch the administrative console.
2. From the left (navigation) panel, select **Troubleshooting > Logs and Trace**.
3. In the right panel, select the name of the server on which you want to enable logging and tracing.
4. In the right panel, under "General properties," select **Change Log Level Details**.
5. Select the Runtime tab. (Selecting the Runtime tab allows you to make this change in real time without requiring you to restart the server.)
6. Add the name of the package followed by `=all` to the list of logged packages in the box on the screen. Separate this new entry from any existing entries with a colon. For example, `CxCommon=all`. In this case, `CxCommon` is the name of the package for a set of supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs. Specifying `all` enables all logging and tracing. See Supported WebSphere InterChange Server or WebSphere Business Integration Server Express APIs for a list of the APIs, including their package names.
7. Select **Apply**.
8. To keep this configuration after the server is restarted, select the **Save runtime changes to configuration as well** check box.
9. Select **OK**.
10. When the next screen appears, select **Save** to save your changes.

# Failure trying to serialize an object that is not serializable in a migrated BPEL file

If a serialization failure occurs in a BPEL file generated by the migration, you might be able to modify it to prevent the failure from occurring.

**Problem:** A serialization failure occurs in a custom snippet node of a business process execution language (BPEL) file generated by migration because an attempt is made to serialize an object that is not serializable.

**Cause:** In WebSphere InterChange Server or WebSphere Business Integration Server Express, a Collaboration Template is compiled into a single Java class. In WebSphere Process Server, each node in a BPEL file might compile into a separate Java class. In WebSphere InterChange Server or WebSphere Business Integration

Server Express, a variable can be declared once and shared throughout the various steps of a Collaboration Template. To simulate that behavior in the migrated BPEL file, each variable used in a code snippet must be retrieved at the start of the snippet and saved at the end of the snippet. Variables defined in WebSphere InterChange Server or WebSphere Business Integration Server Express Port definitions become BPEL variables. These are retrieved into BusObj variables at the beginning of each snippet (if referenced in the snippet) and saved back to the BPEL variables at the end of each snippet. For example, a retrieval at the beginning of snippets looks like this:

```
BusObj tempBusObj = null;if (tempBusObj_var != null) { tempBusObj =
      new BusObj(tempBusObj_var); };
```

and a save at the end of snippets looks like this:

```
if (tempBusObj == null) { tempBusObj_var = null; } else { tempBusObj_var =
      tempBusObj.getBusinessGraph(); }
```

Other variables used in the WebSphere InterChange Server or WebSphere Business Integration Server Express snippet code are serialized and stored as a String in a BPEL variable named *CollabTemplateName_var*. These variables are deserialized at the beginning of each BPEL snippet, and then serialized and saved at the end of each BPEL Snippet that they are referenced in. For example, objects are retrieved like this:

```
BusObj tempBusObj = (BusObj)BaseCollaboration.deserialize
  (FrontEndCollab_var.getString("tempBusObj"));
```

and objects are saved like this:

```
FrontEndCollab_var.setString("tempBusObj", BaseCollaboration.serialize(tempBusObj));
```

If the type of the object being serialized is not serializable, then using serialize and deserialize will fail when the BPEL is run.

**Solution:** After migration, modify the BPEL file as follows:

- For any variable that is not Java-serializable, update the BPEL snippets to remove the serialization and deserialization statements. If the variable needs to be shared across snippets (instead of being re-created in each snippet) another method must be used to preserve the value of the variable across snippets.
- Manually define BPEL variables for variables of type BusObj that are not declared in the WebSphere InterChange Server or WebSphere Business Integration Server Express Port definitions but are used on Partner Invokes. This is a manual step because variables used during invokes in WebSphere Process Server must be strongly typed, and the migration tools cannot accurately determine that type from the WebSphere InterChange Server or WebSphere Business Integration Server Express snippets.

  **Note:** The naming convention used by the migration tools is to add `_var` to the name of the variable in the snippet code when naming the BPEL variables. For example, for a variable called tempBusObj in the snippet code, the migration tools will create a BPEL variable named `tempBusObj_var`.
- For variables that must be declared manually as BPEL variables, change the BPEL snippet code so that it uses the "deserialize/serialize" method of preserving these variables rather than the "retrieve from/store into BPEL variable" method of preserving these variables.

# New behavior for heritage APIs in WebSphere Process Server, version 6.2

In version 6.2 of WebSphere Process Server, heritage APIs use WebSphere Process Server Service Data Objects to store attribute states and data that were formerly stored by the BusinessObjectInterface interface. As a result, the behavior of some method calls in the BusinessObjectInterface and CxObjectContainerInterface interfaces have changed.

The major change to the heritage APIs (HAPIs) in WebSphere Process Server version 6.2 is that the WebSphere InterChange Server BusinessObjectInterface interface is no longer the root storage object for HAPI. Instead, a WebSphere Process Server Service Data Object (SDO) is now used to store attribute states and data.

If you use the Java equivalence operator and weakly typed attribute principles, the behavior of the method calls in the BusinessObjectInterface and CxObjectContainerInterface interfaces is different, as described in the following sections:

- "Using the Java equivalence operator when performing a Set operation followed by a Get operation"
- "Using the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute" on page 39
- "Using the Java equivalence operator when you set and retrieve a BusinessObjectInterface object from the CxObjectContainerInterface interface" on page 40
- "Using weakly typed attribute data types for the BusObj class validData methods" on page 41

## Using the Java equivalence operator when performing a Set operation followed by a Get operation

A different BusinessObjectInterface object is returned when performing a Set operation followed by a Get operation of a BusinessObjectInterface object for one target attribute. The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when performing a Set operations followed by a Get operation.

*Table 3. Behavior changes: Using the Java equivalence operator with Set and Get operations*

| Behavior type | Description |
|---|---|
| **Behavior prior to WebSphere Process Server version 6.2** | The same BusinessObjectInterface container that was set was also retrieved, and you could use the Java equivalence operator "==" to determine if they were the same.<br><br>Example:<br><br>`boolean b = (JavaObjectA == JavaObjectB)` |
| **Behavior after WebSphere Process Server version 6.2** | The original BusinessObjectInterface container is discarded, and when you perform a Get operation to retrieve the BusinessObjectInterface object, a new container is created. The returned container is not the same object, but the root object that it wraps is the same object. A new method, isEquivalent, has been added to the BusinessObjectInterface class: BusinessObjectInterface.isEquivalent(BOI). When you want to determine if the two BusinessObjectInterface objects are equivalent, use the isEquivalent method to perform the comparison. |

*Table 3. Behavior changes: Using the Java equivalence operator with Set and Get operations (continued)*

| Behavior type | Description |
|---|---|
| **Example of new behavior** | The following example shows the use of isEquivalent. You have a BusinessObjectInterface object of type MasterBusinessObject with the attribute Attr_Nine, which is a BusinessObjectInterface object of type HelloWorld:<br>`BusinessObjectInterface mboBOI, hw1BOI, hw2BOI;`<br>`hw1BOI.setAttrValue("Message", "hw1BOI_message");`<br>`hw1BOI.setVerb("Create");`<br>`mboBOI.setAttrValue("Attr_Nine", hw1BOI);`<br>`hw2BOI = mboBOI.getAttrValue("Attr_Nine");`<br><br>Instead of:<br><br>`boolean result = (hw1BOI == hw2BOI); assertTrue(result);`<br><br>Use this:<br><br>`boolean result = hw1BOI.isEquivalent(hw2BOI); assertTrue(result);` |

## Using the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute

Setting a BusinessObjectInterface object to more than one target attribute sets a cloned object. This applies both to elements of a BusObjArray class and to multiple target attributes. The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when setting a BusinessObjectInterface object to more than one target attribute.

*Table 4. Behavior changes: Using the Java equivalence operator with more than one target attribute*

| Behavior type | Description |
|---|---|
| **Behavior prior to WebSphere Process Server version 6.2** | You could set a BusinessObjectInterface object to multiple locations, and all of the locations contained a reference to the original BusinessObjectInterface object. If you changed the attribute in one BusinessObjectInterface object, that change was reflected in all of the other references for that object. |
| **Behavior after WebSphere Process Server version 6.2** | Service Data Object (SDO) rules prevent you from setting the same SDO to more than one target property. If you try to set the SDO to more than one target property, the SDO moves from one attribute to the next, leaving a "null" value at the previous attribute location. Now, instead of leaving a "null" value when the BusinessObjectInterface object is set to a second, third, and so on, location, the object is cloned into the multiple locations.<br><br>For example, you have a BusinessObjectInterface object of the type MasterBusinessObject, with the attributes Attr_Nine and Attr_Eleven that are of the type HelloWorld. If you set the same HelloWorld object to both attributes, then Attr_Nine is assigned the original object, and Attr_Eleven is assigned a clone. The clone is a snapshot of the object at the time that it is cloned.<br><br>If you want to determine if two BusinessObjectInterface objects are equivalent, do not use the Java equivalence operator; instead, use the isEquivalent method to perform the comparison. |

*Table 4. Behavior changes: Using the Java equivalence operator with more than one target attribute (continued)*

| Behavior type | Description |
|---|---|
| **Example of new behavior** | The following example shows the use of isEquivalent and clones. You have a BusinessObjectInterface object of type MasterBusinessObject, with the attributes Attr_Nine and Attr_Eleven that are of type HelloWorld:<br>`BusinessObjectInterface mboBOI;`<br>`BusinessObjectInterface hw1BOI, hw2BOI, hw3BOI;`<br>`hw1BOI.setAttrValue("Message", "hw1BOI_message");`<br>`hw1BOI.setVerb("Create");`<br>`mboBOI.setAttrValue("Attr_Nine", hw1BOI);`<br>`mboBOI.setAttrValue("Attr_Eleven", hw1BOI);`<br>`hw2BOI = mboBOI.getAttrValue("Attr_Nine");`<br>`hw3BOI = mboBOI.getAttrValue("Attr_ Eleven ");`<br><br>Instead of:<br><br>`boolean result = hw2BOI == hw3BOI); assertTrue(result);`<br><br>Use isEquivalent instead:<br><br>`boolean result = hw2BOI.isEquivalent(hw3BOI); assertTrue(result);`<br><br>The cloned objects do not share a reference, and changes to the original BusinessObjectInterface object are not reflected in the cloned BusinessObjectInterface object:<br>`hw1BOI.setAttrValue("Message", "hw1BOI_message changed");`<br>`boolean result = hw1BOI.isEquivalent(hw2BOI);`<br>`assertTrue(result);`<br>`boolean result = hw1BOI.isEquivalent(hw3BOI);`<br>`assertFalse(result);`<br>`boolean result = hw2BOI.isEquivalent(hw3BOI);`<br>`assertFalse(result);` |

## Using the Java equivalence operator when you set and retrieve a BusinessObjectInterface object from the CxObjectContainerInterface interface

The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used the Java equivalence operator when setting and retrieving a BusinessObjectInterface object from the CxObjectContainerInterface interface.

*Table 5. Behavior changes: Using the Java equivalence operator with the CxObjectContainerInterface interface*

| Behavior type | Description |
|---|---|
| **Behavior prior to WebSphere Process Server version 6.2** | When you set and then retrieved a BusinessObjectInterface object from the CxObjectContainerInterface interface, you could use the Java equivalence operator "==" because the BusinessObjectInterface container that was retrieved was the same BusinessObjectInterface container that was set. |
| **Behavior after WebSphere Process Server version 6.2** | You must use the BusinessObjectInterface.isEquivalent(BOI) method. |

*Table 5. Behavior changes: Using the Java equivalence operator with the CxObjectContainerInterface interface (continued)*

| Behavior type | Description |
|---|---|
| **Example of new behavior** | The following JUnit test code demonstrates the old and new behavior.<br>`CxObjectContainerInterface testCxObjectContainerInt;`<br>`BusinessObjectInterface mB01, mB02, mB03;`<br>`testCxObjectContainerInt.insertBusinessObject(mB01);`<br>`testCxObjectContainerInt.setBusinessObject(1, mB01);`<br>`BusinessObjectInterface mB02 = testCxObjectContainerInt.getBusinessObject(0));`<br>`BusinessObjectInterface mB03 = testCxObjectContainerInt.getBusinessObject(1));`<br>`assertTrue(mB01 == mB02);`<br>`assertTrue(mB01 == mB03);`<br>`assertTrue(mB02 == mB03);`<br><br>This Java equivalence operator no longer works because the BusinessObjectInterface object that is returned by CxObjectContainerInterface.getBusinessObject(int index) is not the same Java object that was set to CxObjectContainerInterface.<br><br>In the following code, the equivalence operator is replaced by the method BusinessObjectInterface.isEquivalent(BOI):<br>`boolean result1 = mB01.isEquivalent(mB02)`<br>`assertTrue(result1);`<br>`boolean result2 = mB01.isEquivalent(mB03)`<br>`assertFalse(result2);`<br>`boolean result3 = mB02.isEquivalent(mB03)`<br>`assertFalse(result3);`<br><br>The cloned objects do not share a reference, and changes to the original BusinessObjectInterface object are not reflected in the cloned BusinessObjectInterface object:<br>`hw1BOI.setAttrValue("Message", "hw1BOI_message changed");`<br>`boolean result = mB01.isEquivalent(mB02;`<br>`assertTrue(result);`<br>`boolean result = mB01.isEquivalent(mB02;`<br>`assertFalse(result);`<br>`boolean result = mB02.isEquivalent(hw3BOI);`<br>`assertFalse(result);` |

## Using weakly typed attribute data types for the BusObj class validData methods

The following table describes the previous behavior, current behavior, and an example of what to change if you previously had used weakly typed attribute data types in WebSphere InterChange Server or WebSphere Business Integration Server Express for the BusObj class validData methods.

*Table 6. Behavior changes: Using weakly typed attribute data types for BusObj class validData methods*

| Behavior type | Description |
|---|---|
| **Behavior prior to WebSphere Process Server version 6.2** | For the BusObj class validData methods, attribute data types were weakly typed in WebSphere InterChange Server or WebSphere Business Integration Server Express. This allowed some odd data-type combinations. For example, if a business object had an attribute that was of type boolean but you used a set method that had a string parameter, you were able to set the string "not a boolean" into an attribute that was type boolean. As long as you used the getString method, they could get the string "not a boolean" back. |

*Table 6. Behavior changes: Using weakly typed attribute data types for BusObj class validData methods (continued)*

| Behavior type | Description |
|---|---|
| **Behavior after WebSphere Process Server version 6.2** | These attribute data types are now strongly typed. If a data type was valid but now is not valid, a CollaborationException exception is thrown, with message number 1802. Because WebSphere Process Server is strongly typed, you cannot put a String value into an Attribute of type boolean. Even if you used the Java conversions for strings into the boolean values of *true* and *false*, there is no way to return the original value of "not a boolean". The only possible returned value is *true* or *false*.<br><br>Therefore, attributes are now strongly typed for doubles-floats or int-long; these can be used interchangeably where Java provides automatic casting. However, as with any casting of types, you can expect some loss of precision when fields are demoted. If a type is not valid for the attribute to which it is set, but it used to be valid in WebSphere InterChange Server or WebSphere Business Integration Server Express, then a CollaborationException exception is thrown, with message number 1802. This is a new message number; the message definition is located in the InterchangeSystem.txt message file. |
| **Example of new behavior** | If a type is not valid for the Attribute to which it is set, but it used to be valid in WebSphere InterChange Server, a CollaborationException with message number 1802 will be thrown. This is a new message number, the Message definition is located in the InterchangeSystem.txt message file:<br>`try`<br>`{`<br>`BusObj mBO = new BusObj("MasterBusinessObject");`<br>`mBO.set("Attr_Two", "xxx");`<br>`fail("Expected CollaborationException not thrown");`<br>`}`<br>`catch (CollaborationException e)`<br>`{`<br>`int a = e.getMsgNumber();`<br>`String b = e.getSubType();`<br>`String c = e.getMessage();`<br>`String d = e.toString();`<br>`assertEquals("exception_msgNumber", 1802, a);`<br>`assertEquals("exception_type", "AttributeException", b);`<br>`assertEquals("exception_message",`<br>`"Error 1802 The attribute \"Attr_Two\" in SDO MasterBusinessObject is`<br>`of type boolean and is not allowed to be set with a value \"xxx\" of`<br>`type String. Error1802", c);`<br><br>`assertEquals("exception_toString", "AttributeException: Error 1802 The`<br>`attribute \"Attr_Two\" in SDO MasterBusinessObject is of type boolean`<br>`and is not allowed to be set with a value \"xxx\" of type String.`<br>`Error1802", d);`<br>`}` |

# Chapter 8. Troubleshooting a failed deployment

This topic describes the steps to take to determine the cause of a problem when deploying an application. It also presents some possible solutions.

## Before you begin

This topic assumes the following things:
- You have a basic understanding of debugging a module.
- Logging and tracing is active while the module is being deployed.

## About this task

The task of troubleshooting a deployment begins after you receive notification of an error. There are various symptoms of a failed deployment that you have to inspect before taking action.

**Procedure**

## Procedure

1. Determine if the application installation failed.

   Examine the `SystemOut.log` file for messages that specify the cause of failure. Some of the reasons an application might not install include the following:

   - You are attempting to install an application on multiple servers in the same Network Deployment cell.
   - An application has the same name as an existing module on the Network Deployment cell to which you are installing the application.
   - You are attempting to deploy J2EE modules within an EAR file to different target servers.

   **Important:** If the installation has failed and the application contains services, you must remove any SIBus destinations or J2C activation specifications created prior to the failure before attempting to reinstall the application. The simplest way to remove these artifacts is to click **Save > Discard all** after the failure. If you inadvertently save the changes, you must manually remove the SIBus destinations and J2C activation specifications (see Deleting SIBus destinations and Deleting J2C activation specifications in the Administering section).
2. If the application is installed correctly, examine it to determine if it started successfully.

   If the application did not start successfully, the failure occurred when the server attempted to initiate the resources for the application.

   a. Examine the `SystemOut.log` file for messages that will direct you on how to proceed.
   b. Determine if resources required by the application are available and/or have started successfully.

      Resources that are not started prevent an application from running. This protects against lost information. The reasons for a resource not starting include:
      - Bindings are specified incorrectly
      - Resources are not configured correctly

- Resources are not included in the resource archive (RAR) file
- Web resources not included in the Web services archive (WAR) file

c. Determine if any components are missing.

The reason for missing a component is an incorrectly built enterprise archive (EAR) file. Make sure that the all of the components required by the module are in the correct folders on the test system on which you built the Java archive (JAR) file. “Preparing to deploy to a server” contains additional information.

3. Examine the application to see if there is information flowing through it.

Even a running application can fail to process information. Reasons for this are similar to those mentioned in step 2b on page 43.

a. Determine if the application uses any services contained in another application. Make sure that the other application is installed and has started successfully.

b. Determine if the import and export bindings for devices contained in other applications used by the failing application are configured correctly. Use the administrative console to examine and correct the bindings.

4. Correct the problem and restart the application.

# Deleting J2C activation specifications

The system builds J2C application specifications when installing an application that contains services. There are occasions when you must delete these specifications before reinstalling the application.

## Before you begin

If you are deleting the specification because of a failed application installation, make sure the module in the Java Naming and Directory Interface (JNDI) name matches the name of the module that failed to install. The second part of the JNDI name is the name of the module that implemented the destination. For example in `sca/SimpleBOCrsmA/ActivationSpec`, **SimpleBOCrsmA** is the module name.

**Required security role for this task:** When security and role-based authorization are enabled, you must be logged in as administrator or configurator to perform this task.

## About this task

Delete J2C activation specifications when you inadvertently saved a configuration after installing an application that contains services and do not require the specifications.

**Procedure**

## Procedure

1. Locate the activation specification to delete.

The specifications are contained in the resource adapter panel. Navigate to this panel by clicking **Resources > Resource adapters**.

a. Locate the **Platform Messaging Component SPI Resource Adapter.**

To locate this adapter, you must be at the **node** scope for a standalone server or at the **server** scope in a deployment environment.

2. Display the J2C activation specifications associated with the Platform Messaging Component SPI Resource Adapter.

   Click on the resource adapter name and the next panel displays the associated specifications.
3. Delete all of the specifications with a **JNDI Name** that matches the module name that you are deleting.
   a. Click the check box next to the appropriate specifications.
   b. Click **Delete.**

### Results

The system removes selected specifications from the display.

### What to do next

Save the changes.

## Deleting SIBus destinations

Service integration bus (SIBus) destinations are used to hold messages being processed by SCA modules. If a problem occurs, you might have to remove bus destinations to resolve the problem.

### Before you begin

If you are deleting the destination because of a failed application installation, make sure the module in the destination name matches the name of the module that failed to install. The second part of the destination is the name of the module that implemented the destination. For example in `sca/SimpleBOCrsmA/component/test/sca/cros/simple/cust/Customer`, **SimpleBOCrsmA** is the module name.

**Required security role for this task:** When security and role-based authorization are enabled, you must be logged in as administrator or configurator to perform this task.

### About this task

Delete SIBus destinations when you inadvertently saved a configuration after installing an application that contains services or you no longer need the destinations.

**Note:** This task deletes the destination from the SCA system bus only. You must remove the entries from the application bus also before reinstalling an application that contains services (see Deleting J2C activation specifications in the Administering section of this information center.

**Procedure**

### Procedure

1. Log into the administrative console.
2. Display the destinations on the SCA system bus.
   a. In the navigation pane, click **Service integration → buses**
   b. In the content pane, click **SCA.SYSTEM.*cell_name*.Bus**
   c. Under Destination resources, click **Destinations**

3. Select the check box next to each destination with a module name that matches the module that you are removing.
4. Click **Delete.**

## Results

The panel displays only the remaining destinations.

## What to do next

Delete the J2C activation specifications related to the module that created these destinations.

# Chapter 9. Troubleshooting administration tasks and tools

Use the information in this group of topics to identify and resolve problems that can occur while you are administering the runtime environment.

## Profile-specific log files

There are log files detailing the characteristics and runtime activities of individual profiles. These log files are located within the profile directory for each profile.

There are a number of log files that are created for each profile. Some of these logs describe the parameters used for the creation of the profile. These types of log files generally remain unchanged once the profile is fully configured. Other profile-specific logs are continually updated to capture error, warning, and information messages emitted during runtime. Some of these log files are also used to capture a Common Base Event (that may include business object data) that is selected for monitoring.

The table below specifies the different types of profile-specific log files and the locations where you can find them within the product. Within the table, the variable *install_root* represents the installation directory of WebSphere Process Server. The variable *profile_root* represents the root location of a profile.

i5/OS The variable *user_data_root* represents the default user data directory.

For more information see Default installation directories for the product, profiles, and tools.

*Table 7. Profile-specific log files updated during runtime*

| Log | Contents |
|---|---|
| First failure data capture (ffdc) log and exception files (common to all profile types) are found in these directories:<br>• Linux UNIX *profile_root*/logs/ffdc<br>• Windows *profile_root*\logs\ffdc<br>• i5/OS *profile_root*/logs/ffdc | Contains the ffdc log and exception files for individual profiles. There are two types of ffdc logs: a single log file with a compilation of all the errors encountered during the profile runtime, and numerous text files with details such as stack traces and other information. The naming conventions for the different types of profiles are given for both files, as follows:<br>• **Deployment manager profile:**<br>– Log file: *deployment_manager_name*_exception.log.<br>– Text files: *deployment_manager_name_hex_id_date_time*.txt.<br>• **Custom profile:**<br>– Log files:*node_agent_name*_exception.log and *server_name*_exception.log .<br>– Text files: *node_agent_name_hex_id_date_time*.txt or *server_name_hex_id_date_time*.txt.<br>• **Stand-alone profile:**<br>– Log file: *server_name*_exception.log.<br>– Text files: *server_name_hex_id_date_time*.txt. |

*Table 7. Profile-specific log files updated during runtime (continued)*

| Log | Contents |
| --- | --- |
| Deployment manager logs (deployment manager profiles only) are found in these directories:<br>• Linux UNIX *profile_root*/logs/*deployment_manager_name*<br>• Windows *profile_root*\logs*deployment_manager_name*<br>• i5/OS *profile_root*/logs/*deployment_manager_name* | You will work primarily with four log files in this directory:<br>• `startServer.log` —<br>Contains the system parameters detected on the system and the messages emitted by the deployment manager during the start process<br>• `stopServer.log` —<br>Contains the system parameters detected on the system and the messages emitted when the deployment manager is shut down.<br>• `SystemErr.log` —<br>Contains error and exception messages generated by the deployment manager during runtime. Continually updated while server is running.<br>• `SystemOut.log` —<br>Contains all messages, including error, warning, and information messages generated by the deployment manager during runtime. Continually updated while server is running. |
| Node agent logs (custom profiles only) are found in these directories:<br>• Linux UNIX *profile_root*/logs/*node_agent_name*<br>• Windows *profile_root*\logs*node_agent_name*<br>• i5/OS *profile_root*/logs/*node_agent_name* | You will work primarily with four log files in this directory:<br>• `startServer.log` —<br>Contains the system parameters detected on the system and the messages emitted by the node agent during the start process<br>• `stopServer.log` —<br>Contains the system parameters detected on the system and the messages emitted when the node agent is shut down.<br>• `SystemErr.log` —<br>Contains error and exception messages generated by the node agent during runtime. Continually updated while node agent is running.<br>• `SystemOut.log` —<br>Contains all messages, including error, warning, and information messages generated by the node agent during runtime. Continually updated while the node agent is running. |

*Table 7. Profile-specific log files updated during runtime (continued)*

| Log | Contents |
| --- | --- |
| Server logs (custom and stand-alone profiles only) are found in these directories:<br>• Linux UNIX *profile_root*/logs/*server_name*<br>• Windows *profile_root*\logs*server_name*<br>• i5/OS *profile_root*/logs/*server_name* | You will work primarily with four log files in this directory:<br>• `startServer.log` —<br>Contains the system parameters detected on the system and the messages emitted by the server during the start process<br>• `stopServer.log` —<br>Contains the system parameters detected on the system and the messages emitted when the server is shut down.<br>• `SystemErr.log` —<br>Contains error and exception messages generated by the server during runtime. Continually updated while server is running.<br>• `SystemOut.log` —<br>Contains all messages, including error, warning, and information messages generated by the server during runtime. Also contains any events being monitoring that are emitted from the Common Event Infrastructure (CEI), in Common Base Event format. These events may also include the level of business object data (FINE, FINER, or FINEST) that is specified for the monitor. Continually updated while the server is running. |
| Node federation log files are found in these directories (only applies to non-deployment manager profiles):<br>• Linux UNIX **On Linux and UNIX platforms:** *profile_root*/logs<br>• Windows **On Windows platforms:** *profile_root*\logs<br>• i5/OS **On i5/OS platforms:** *profile_root*/logs | Two log files are generated when you attempt to federate a custom, augmented, or stand-alone profile to a deployment manager:<br>• `addNode.log` —<br>contains the pertinent server environment information and messages generated when you attempt to federate the profile.<br>• `isFederated.log` —<br>lists the commands used by the deployment manager to federate the profile. |
| The location of the Integrated Solutions Console application deployment log file is listed here (only for deployment manager and stand-alone profiles):<br>• Linux UNIX *profile_root*/logs/iscinstall.log<br>• Windows *profile_root*\logs\iscinstall.log<br>• i5/OS *profile_root*/logs/iscinstall.log | The iscinstall.log file contains information regarding the deployment of the administrative console application in a deployment manager or stand-alone profile. |
| The location of the Installation Verification Tool log file is listed here (only for deployment manager and stand-alone profiles):<br>• Linux UNIX *profile_root*/logs/ivtClient.log<br>• Windows *profile_root*\logs\ivtClient.log<br>• i5/OS *profile_root*/logs/ivtClient.log | This log file contains the output generated by the Installation Verification Tool. You can start this program from the First Steps console after you create a deployment manager or stand-alone profile. The log contains basic configuration information and the messages that are displayed when you run the tool. |

*Table 7. Profile-specific log files updated during runtime (continued)*

| Log | Contents |
| --- | --- |
| The location of the log file detailing the commands generated for a profile creation is listed here:<br>• Linux UNIX *profile_root*/logs/updateserverpolicy.log<br>• Windows *profile_root*\logs\updateserverpolicy.log<br>• i5/OS *profile_root*/logs/updateserverpolicy.log | This file contains the sequence of commands used by the product to set server environment variables and create a profile. All profile types will contain this file. |

**Related tasks**

Chapter 4, "WebSphere Process Server log files," on page 9
There are two distinct groups of log files in the installed product. Logs detailing the product installation, product updates, and profile management are one group. Logs detailing the characteristics and runtime activities of individual profiles comprise the second group.

Installation and profile creation log files

# Troubleshooting the failed event manager

This topic discusses problems that you can encounter while using the failed event manager.

**Note:** This topic does not discuss how to use the failed event manager to find, modify, resubmit, or delete failed events on the system. For information about managing failed events, see *Managing WebSphere Process Server failed events* in the information center.

Select the problem you are experiencing from the table below:

| Problem | Refer to the following |
| --- | --- |
| I am having trouble entering values in the Search page's **By Date** tab | "Values in the By Date and From Date field automatically change to default if entered incorrectly" |
| I am having trouble deleting expired events | "Using the Delete Expired Events function appears to suspend the failed event manager" on page 51 |
| I am having trouble with failed events not being created | "Failed events are not being created" on page 51 |
| I am having trouble retrieving or deleting large numbers of failed events | "The server fails when retrieving or deleting large numbers of failed events" on page 51 |

## Values in the By Date and From Date field automatically change to default if entered incorrectly

The Search page's **From Date** and **To Date** fields require correctly formatted locale-dependent values. Any inconsistency in the value's format (for example, including four digits in the year instead of 2, or omitting the time) will cause the failed event manager to issue the following warning and substitute a default value in the field:

CWMAN0017E: The date entered could not be parsed correctly: *your_incorrectly_formatted_date*. Date: *default_date* is being used.

The default value of the **From Date** field is defined as January 1, 1970, 00:00:00 GMT.

**Important:** The actual default value shown in your failed event manager implementation will vary depending on your locale and time zone. For example, the From Date field defaults to 12/31/69 7:00 PM for a workstation with an en_US locale in the Eastern Standard Time (EST) time zone.
The default value for the **To Date** field is always the current date and time, formatted for your locale and time zone.

To avoid this problem, always enter your dates and times carefully, following the example provided above each field.

## Using the Delete Expired Events function appears to suspend the failed event manager

If you use the Delete Expired Events button in situations where there are many failed events in the current search results, or where those events contain a large amount of business data, the failed event manager can appear to be suspended indefinitely.

In this situation, the failed event manager is not suspended: it is working through the large data set, and will refresh the results set as soon as the command completes.

## Failed events are not being created

If the Recovery subsystem is not creating failed events, go through the following checklist of potential causes:

- Ensure that the wpsFEMgr application is running. If necessary, restart it.
- Ensure that the failed event manager's database has been created, and that the connection has been tested.
- Ensure that the necessary failed event destination has been created on the SCA system bus. There should be one failed event destination for each deployment target.
- Ensure that the Quality of Service (QoS) **Reliability** qualifier has been set to Assured for any Service Component Architecture (SCA) implementation, interface, or partner reference that participates in events you want the Recovery service to handle.

## The server fails when retrieving or deleting large numbers of failed events

The server can fail if you use the failed event manager to retrieve or delete a large number of failed events at once. To prevent this problem, be sure to check the total failed event count on the main page of the failed event manager before performing search or delete actions. If you have a large number of events, do not try to retrieve or delete all of them at once. Instead, use the **Search failed events** option to return a subset of failed events that match specific criteria. You can then delete all of the events in the filtered result set without causing a server failure.

**Note:** There are several factors in your environment that can affect the number of failed events the server can return or delete in a single request, including other processes running on the same machine as the server and the amount of available memory.

# Troubleshooting the business rules manager

Some of the problems you might encounter using the business rules manager are login errors, login conflicts, and access conflicts.

You can take various steps to troubleshoot these problems.

## Resolving login errors

A log in error occurs upon logging in.

### Before you begin

### About this task

The login error message is as follows:

```
Unable to process login. Please check User ID and password and try again.
```

**Note:** Login errors occur only when administrative security is enabled and either the user ID, password, or both, are incorrect.

To resolve login errors, perform the following steps.

**Procedure**

### Procedure

1. Click **OK** on the error message to return to the Login page.
2. Enter the valid **User ID** and **Password**.
   - If passwords are case sensitive, make sure that Caps Lock key is not on.
   - Make sure the user ID and password are spelled correctly.
   - Check with the system administrator to be sure that the user ID and password are correct.
3. Click **Login**.

### What to do next

If you resolve the login error, you will now be able to log in to the business rules manager. If the error is not resolved, contact your system administrator.

## Resolving login conflict errors

A login conflict error occurs when another user with the same user ID is already logged in to the application.

### Before you begin

### About this task

The login conflict message is as follows:

```
Another user is currently logged in with the same User ID. Select from the
following options:
```

Usually this error occurs when a user closed the browser without logging out. When this condition occurs, the next attempted login before the session timeout expires results in a login conflict.

**Note:** Login conflict errors occur only when administrative security is enabled.

To resolve login conflict errors, select from the following three options:

- Return to the Login page.

  Use this option if you want to open the application with a different user ID.
- Log out the other user with the same user ID.

  Use this option to log out the other user and start a new session.

  **Note:** Any unpublished local changes made in the other session will be lost.
- Inherit the context of the other user with the same user ID and log out that user.

  Use this option to continue work already in progress. All unpublished local changes in the previous session that have been saved will not be lost. The business rules manager will open to the last page displayed in the previous session.

## Resolving access conflict errors

An access conflict error occurs when a business rule is updated in the data source by one user at the same time another user is updating the same rule.

### Before you begin

This error is reported when you publish your local changes to the repository.

### About this task

To correct access conflict errors, perform the following actions:

- Find the source of the business rule that is causing the error and check if your changes on the local machine are still valid. Your change may no longer be required after the changes done by another user.
- If you choose to continue working in the business rules manager, you must reload those business rule groups and rule schedules in error from the data source as your local changes of business rule groups and rule schedules in error are no longer usable. Reload a business rule group or rule schedule page, by clicking **Reload** in the Publish and Revert page of the rule for which the error was reported. You can still use local changes in other business rule groups and rule schedules that are not in error.

# Chapter 10. Troubleshooting WebSphere Application Server

Because IBM WebSphere Process Server is built on IBM WebSphere Application Server, the function that you are having problems with may be provided by the underlying WebSphere Application Server. You might want to consult troubleshooting information in the WebSphere Application Server documentation.

WebSphere Process Server is built on WebSphere Application Server Network Deployment, version 6.1.

For more information about troubleshooting in WebSphere Application Server, see the topic "Troubleshooting and support" in the WebSphere Application Server Information Center.

**Related reference**


Troubleshooting and support

# Chapter 11. Tools for troubleshooting your applications

WebSphere Process Server and WebSphere Integration Developer include several tools you can use to troubleshoot your applications that you develop and deploy on the server.

During development of your applications, you can use debugging tools in WebSphere Integration Developer. You can implement runtime troubleshooting capabilities into your applications using logging, tracing, and service component event monitoring. Administrators of running applications can use the failed event manager to view, modify, resubmit, and delete failed operations between Service Component Architecture (SCA) components.

## Debugging applications in WebSphere Integration Developer

In order to debug applications that are running on WebSphere Process Server, you must use your application development tool, such as IBM WebSphere Integration Developer.

### About this task

For more information about debugging applications, see **Debugging components** in the IBM WebSphere Business Process Management information center or in the online documentation installed with WebSphere Integration Developer.

**Related reference**

IBM WebSphere Business Process Management Version 6.2 information center

## Using logging, tracing, and monitoring in your applications

Designers and developers of applications that run on WebSphere Process Server can use capabilities such as monitoring and logging that add troubleshooting features to applications.

### About this task

WebSphere Process Server is built on IBM WebSphere Application Server, Network Deployment, version 6.1. For more information, see the topic "Adding logging and tracing to your application" in the WebSphere Application Server Information Center.

To use logging, tracing, and monitoring with your applications, perform the following steps.

### Procedure

- You can set up service component event monitoring for applications running on WebSphere Process Server. For more information, see the "Monitoring service component events" topic link in the Related Topics section at the bottom of this page.
- You can add logging and tracing to your applications using WebSphere Application Server.

**Related reference**

Adding logging and tracing to your application

Monitoring service component events

# Troubleshooting Service Component Architecture (SCA) processing and call chains

Cross-component tracing allows you to identify systemout.log or trace.log data that is associated with WebSphere Process Server and WebSphere Enterprise Service Bus modules and components. The trace.log data can include error and event information, such as corrupted data or runtime exceptions, captured during SCA processing. The input and output data passing between WebSphere Process Server and WebSphere Enterprise Service Bus components can also be captured and used for problem determination using WebSphere Integration Developer.

Events that can be captured include:

- Errors that occur during processing because of corrupted data.
- Errors when resources are not available, or are failing.
- Interpretation of code paths.

You can access the Cross-Component Trace page from the administrative console and then clicking **Troubleshooting** → **Cross-Component Trace**. On this page, you can select the servers from which you want to collect trace data. Use the **Configuration** column to specify trace settings for servers that will be used when the server starts or restarts. Use the **Runtime** column to specify trace settings for servers that are running.

For each server, you can use any of the following settings:

**enable**
: This setting enables tracing for SCA processing. Data collected from this setting is added to the systemout.log and trace.log files and is purged as those files are purged.

**enable with data snapshot**
: This setting enables tracing for SCA processing and input and output data that passes between WebSphere Process Server and WebSphere Enterprise Service Bus components. Data from SCA processing is added to the systemout.log and trace.log files. Input and output data from WebSphere Process Server and WebSphere Enterprise Service Bus components is placed in files that are created in the logs\XCT directory.

**disable**
: This setting disables tracing on the selected server.

## Handling and deleting collected data

- Data collected from SCA processing is added to added to the systemout.log and trace.log files and is purged as those files are purged.
- Input and output data added as files in the logs\XCT directory can be moved to a location where it can be viewed by WebSphere Integration Developer for problem determination. The files can then be manually deleted. These input and output files in logs\XCT are related to the systemout.log and trace.log files that were created at the same time. When WebSphere Application Server deletes old systemout.log and trace.log files, the associated input and output files in

logs\XCT can also be deleted. Generally, there will be many input and output files for a given systemout.log file. The timestamps in the systemout.log and trace.log files can be used to identify what input and output files to delete. It is safe to delete all input and output files older than the oldest date in the systemout.log and trace.log files.

# Managing failed events

The WebSphere Process Server Recovery service captures data about failed events. You can then use the failed event manager to view, modify, resubmit, or delete the failed event.

The WebSphere Process Server Recovery service manages failed operations between Service Component Architecture (SCA) components, failed JMS events, and failed operations within long-running business processes.

## Failed SCA events

In the context of SCA, an event is a request or response that is received by a service application. It can come from an external source (such as an inbound application adapter) or an external invocation to a Web service. The event is comprised of a reference to the business logic it wants to operate and its data, stored in a Service Data Object (a business object). When an event is received, it is processed by the appropriate application business logic.

A single thread of execution can branch off into multiple branches (or threads); the individual branches are linked to the main invoking event by the same session context.

If this business logic in one of these branches cannot execute completely due to system failure, component failure, or component unavailability, the event moves into the failed state. If multiple branches fail, a failed event is created for each. The Recovery service handles the following types of failed SCA events:

- Event failures that occur during an asynchronous invocation of an SCA operation
- Event failures that are caused by a runtime exception (in other words, any exception that is not declared in the methods used by the business logic)

The Recovery service does not handle failures from synchronous invocations.

Failed SCA events typically have source and destination information associated with them. The source and destination are based on the failure point (the location where the invocation fails), regardless of the type of interaction. Consider the following example, where Component A is asynchronously invoking Component B. The request message is sent from A to B, and the response (callback) message is sent from B to A.

- If the exception occurs during the initial request, Component A is the source and Component B is the destination for the purposes of the failed event manager.
- If the exception occurs during the response, Component B is the source and Component A is the destination for the purposes of the failed event manager.

This is true for all asynchronous invocations.

The Recovery service sends failed SCA asynchronous interactions to failed event destinations that have been created on the SCA system bus (SCA.SYSTEM.*cell_name*.Bus). The data for failed events is stored in the failed event

database (by default, WPCRSDB) and is made available for administrative purposes through the failed event manager interface.

## Failed JMS events

The Java Message Service (JMS) binding type and configuration determine whether a failed event is generated and sent to the failed event manager.

### JMS bindings

WebSphere Integration Developer provides a recovery binding property that allows you to enable or disabled recovery for each JMS binding, at authoring time. The `recoveryMode` property can be set to one of the following:

| | |
|---|---|
| `bindingManaged` | Allow binding to manage recovery for failed messages |
| `unmanaged` | Rely on transport-specific recovery for failed messages |

Recovery for JMS bindings is enabled by default. When it is enabled, JMS failed events are created in the following situations:

- The function selector fails
- The fault selector fails
- The fault selector returns the `RuntimeException` fault type
- The fault handler fails
- The data binding or data handler fails after a single retry in JMS

In addition, a Service Component Architecture (SCA) failed event is created when the `ServiceRuntimeException` exception is thrown in a JMS binding target component after a single retry in JMS.

These failures can occur during inbound or outbound communication. During outbound communication, JMSImport sends a request message and receives the response message; a failed event is generated if the JMS import binding detects a problem while processing the service response. During inbound communication, the sequence of events is as follows:

1. JMSExport receives the request message.
2. JMSExport invokes the SCA component.
3. The SCA component returns a response to JMSExport.
4. JMSExport sends a response message.

A failed event is generated if the JMS export binding detects a problem while processing the service request.

The Recovery service captures the JMS message and stores it in a Recovery table in the Common database. It also captures and stores the module name, component name, operation name, failure time, exception detail, and JMS properties of the failed event. You can use the failed event manager to manage failed JMS events, or you can use a custom program.

To disable recovery, you must explicitly disable it in WebSphere Integration Developer by setting the `recoveryMode` property to `unmanaged`.

**Note:** If the `recoveryMode` property is missing (for earlier versions of applications), the recovery capability is regarded as enabled.
When recovery is disabled, a failed message is rolled back to its original destination and retried. The system does not create a failed event.

**WebSphere MQ JMS bindings and generic JMS bindings**

WebSphere MQ JMS bindings and generic JMS bindings handle failures in a different way than JMS bindings. Problems during request and response handling do not generate a failed JMS event. Instead, they generate a failed SCA event if the following two conditions are met:

- The underlying messaging system is configured to automatically redeliver a failed message.
- The failure occurs in the export binding's target SCA component, not in the binding itself.

When both conditions are true, the Recovery system generates a failed SCA event that you can manage with the failed event manager.

In all other situations, the failed message rolls back to its original destination, where it is handled according to the messaging system configuration. No failed event is created.

## Failed Business Process Choreographer events

In the context of Business Process Choreographer, exceptions can occur that, if not handled by the process logic, cause an activity to stop or the process instance to fail. A failed event is generated when a long-running Business Process Execution Language (BPEL) process fails and one of the following happens:

- The process instance enters the failed or terminated state
- An activity enters the stopped state

The Recovery service captures the module name and component name for failed Business Process Choreographer events. Failed event data is stored in the Business Process Choreographer database (BPEDB) database.

Note that the Recovery service does not handle failures from business process and human task asynchronous request/reply invocations.

## Business Flow Manager hold queue messages

You can use the failed event manager to manage navigation messages that are stored in the Business Flow Manager hold queue. A navigation message might be stored in the hold queue if:

- An infrastructure, such as a database, is unavailable.
- The message is damaged.

In a long-running process, the Business Flow Manager can send itself request messages that trigger follow-on navigation. These messages trigger either a process-related action (for example, invoking a fault handler) or an activity-related action (for example, continuing process navigation at the activity). A navigation message always contains its associated process instance ID (piid). If the message triggers an activity-related action, it also contains the activity template ID (atid) and the activity instance ID (aiid).

You can use the failed event manager to manage Business Flow Manager hold queue messages, or you can use a custom program.

Business Flow Manager hold queue messages cannot be deleted directly in the failed event manager. If the related process instance does not exist, replaying the hold queue message will result in deletion of the message.

### How are failed events managed?

An administrator uses the failed event manager to browse and manage failed events. Common tasks for managing failed events include:

- Browsing all failed events
- Searching for failed events by specific criteria
- Editing data for a failed event
- Resubmitting failed events
- Deleting failed events

To access the failed event manager, click **Integration Applications → Failed Event Manager**.

## Security considerations for recovery

If you have enabled security for your WebSphere Process Server applications and environment, it is important to understand how role-based access and user identity affect the Recovery subsystem.

**Role-based access for the failed event manager**
: The failed event manager uses role-based access control for the failed event data and tasks. Only the administrator and operator roles are authorized to perform tasks within the failed event manager. Users logged in as either administrator or operator can view all data associated with failed events and can perform all tasks.

**Event identity and user permissions**
: A failed event encapsulates information about the user who originated the request. If a failed event is resubmitted, its identity information is updated to reflect the user who resubmitted the event. Because different users logged in as administrator or operator can resubmit events, these users must be given permissions to the downstream components required to process the event.

For more information about implementing security, see Securing applications and their environment.

## Finding failed events

Failed events are stored in a database and are retrieved through the search functionality of the failed event manager. You can search for all failed events on all the servers within the cell, or for a specific subset of events.

### Before you begin

If administrative security is enabled, you must be logged in as administrator or operator to perform this task.

## About this task

This topic describes how to find all failed events in the cell. This default query returns all SCA and JMS failed events.

If Business Process Choreographer is installed, the query also returns failed, terminated, and stopped Business Process Choreographer events.

To retrieve a complete list of failed events, use the following procedure.

**Procedure**

## Procedure

1. Ensure the administrative console is running.
2. Click **Integration Applications** → **Failed Event Manager** to enter the failed event manager.
3. From the **Failed events on this server** box, click **Get all failed events**.

## Results

The Search Results page opens, displaying a list of all the WebSphere Process Server failed events in the cell.

## What to do next

You can now view (and in some cases, modify) data in a failed event, resubmit it, or delete it.

## Searching for events by criteria

Use the failed event manager Search page to find only those events that meet specified criteria. You can search by failed event type and by criteria such as failure time, event destination or source, exception or business object type, session ID or, for WebSphere Process Server only, event sequencing qualifier.

### Before you begin

If administrative security is enabled, you must be logged in as administrator or operator to perform this task.

### About this task

To search for a specific subset of failed events on the server, perform the following steps.

**Procedure**

### Procedure

1. Ensure the administrative console is running.
2. Click **Integration Applications** → **Failed Event Manager** to enter the failed event manager.
3. From the **Failed events on this server** box, click **Search failed events**.
4. From the **Event type** box on the Search failed events page, select one or more types of events to search for:
   - SCA

- JMS
- Business Process Choreographer
- Business Flow Manager hold queue messages

5. If you are searching for Business Process Choreographer events, verify the event status selected in the Event status box. By default, the failed event manager returns all failed, stopped, and terminated Business Process Choreographer events, but you can modify the search to return only events with a particular status.
6. Optional: Specify any additional search criteria. The following table describes the available options. If you specify multiple criteria, the AND operator is used during the query; the failed event manager returns only events that meet all of the criteria.

*Table 8. Search criteria*

| Search criteria | Field or fields to use | Supported event types | Usage notes |
|---|---|---|---|
| The module, component, or method the event was en route to when it failed. | **Module**<br><br>**Component**<br><br>**Operation** | SCA<br><br>JMS<br><br>Business Process Choreographer<br><br>Business Flow Manager hold queue | Use one or more of these fields to search for failed events associated with a specific module, component, or method. |
| The time period during which the event failed | **From date**<br><br>**To date** | SCA<br><br>JMS<br><br>Business Process Choreographer<br><br>Business Flow Manager hold queue | Formats for date and time are locale-specific. An example is provided with each field. If the value you provide is not formatted correctly, the failed event manager displays a warning and substitutes the default value for that field.<br><br>The time is always local to the server. It is not updated to reflect the local time of the individual workstations running the administrative console. |
| The session in which the event failed | **Session ID** | SCA | None |
| The module or component from which the event originated | **Source module**<br><br>**Source component** | SCA | Use one or both of these fields to find only those failed events that originated from a specific source module or component. The failed event manager determines the source based on the point of failure, regardless of interaction type. |
| The type of business object in the failed event | **Business object type** | SCA | None |

*Table 8. Search criteria (continued)*

| Search criteria | Field or fields to use | Supported event types | Usage notes |
|---|---|---|---|
| Whether the event had the event sequencing qualifier specified | **Event sequencing qualified** | SCA | This search criteria is applicable only for WebSphere Process Server. |
| The exception thrown when the event failed | **Exception text** | SCA | Specify all or part of the exception text in the field to find all events associated with that exception. |

For detailed information about each field and the values it accepts, see the online help for the failed event manager Search page.

7. Click **OK** to begin the search.

### What to do next

You can now view (and in some cases, modify) data in a failed event, resubmit it, or delete it.

## Working with data in failed events

Each failed event has data associated with it; often, that data can be edited before an event is resubmitted. There are two basic types of data for a failed event: data about the event, and business data.

### Data about the failed event

Each type of failed event has the following data associated with it:

- SCA events
  - The event ID, type, status, and session ID
  - The service invocation type between SCA components
  - The names of the module and component from which the event originated (the source).
  - The names of the destination module, component and method for the event
  - The time the event failed
  - The deployment target associated with the event
  - Whether an event sequencing qualifier has been declared for this event
  - The destination module where the event has been or will be resubmitted
  - The correlation ID, if one exists
  - The exception thrown when the event failed
  - The expiration date for resubmitted events (this data can be edited)
  - The trace control set for the event (this data can be edited)
- JMS events:
  - The event ID, type, and status
  - The interaction type
  - The names of the destination module, component and method for the event
  - The time the event failed
  - The deployment target associated with the event
  - The exception thrown when the event failed
  - The destination module where the event has been or will be resubmitted

  - The correlation ID, if one exists
  - The expiration date for resubmitted events (this data can be edited)
  - The JMS-specific properties associated with the failed event: redelivered count, delivery mode, message priority, type, `replyTo` destination, and redelivered indicator (`true` or `false`).
- Business Process Choreographer events:
  - The event ID, type, and status
  - The names of the destination module and component for the event
  - The time the event failed
  - The deployment target associated with the event
  - The process instance name associated with the event
  - The top-level process ID associated with the event
- Business Flow Manager hold queue events:
  - The event ID, type, and status
  - The time the event failed
  - The deployment target associated with the event
  - The process instance ID (if the process instance does not exist, `0` is returned)
  - The name and state of the process instance
  - The name of the associated process template
  - The activity instance name and ID
  - The activity template ID

## Business data

SCA and Business Process Choreographer failed events typically include business data. Business data can be encapsulated in a business object, or it can be simple data that is not part of a business object. Business data for SCA failed events can be edited with the business data editor available in the failed event manager.

## Browsing data in failed events

Use the failed event manager to view failed event data and any business data associated with the event.

### Before you begin

If administrative security is enabled, you must be logged as administrator or operator to perform this task.

### About this task

To browse failed event data, use the following procedure.

**Procedure**

### Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the Search Results page of the failed event manager, click the ID (found in the Event ID column) of the failed event whose data you want to browse.

   The Failed Event Details page opens and displays all of the information about the event.

3. If your failed event has business data, you can browse it by clicking **Edit business data**.

   The Business Data Editor collection page opens, displaying the business data associated with the failed event. Each parameter name in the hierarchy is a link. If the parameter is a simple data type, clicking its name will open up a form so you can edit the parameter's value. If the parameter is a complex data type, clicking its name will expand the hierarchy further.

## Editing trace or expiration data in a failed SCA event

The Failed Event Details page enables you to set or modify values for the trace control and expiration date associated with a failed event.

### Before you begin

You must be logged in as administrator or operator to perform this task.

### About this task

**Important:** Any edits you make to the trace or expiration data are only saved locally until you resubmit the event. If you perform any other action before resubmitting the event, all edits are lost.

Failed SCA events can be resubmitted with trace to help you monitor the event processing. Tracing can be set for a service or a component, and it can be sent to a log or to the Common Event Infrastructure (CEI) server. When you view the failed event data on the Failed Event Details page, the default trace value `SCA.LOG.INFO;COMP.LOG.INFO` is shown for the event. If you resubmit the event with this default setting, no trace occurs when the session calls an SCA service or executes a component.

Some failed SCA events also have an expiration. If a user has specified an expiration with the asynchronous call that sends the event, that data persists even if the event fails, and the expiration time appears in the **Resubmit Expiration Time** field on the Failed Event Details page. Expired failed events cannot be resubmitted successfully. To prevent a second failure, you can edit the expiration date for the event to ensure that it is not expired when it is resubmitted.

To edit trace or expiration data in a failed event, use the following procedure.

**Procedure**

### Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the failed event manager's Search Results page, click the ID (found in the Event ID column) of the failed event whose data you want to edit.

   The Failed Event Details page opens.
3. If the event has an expiration date that causes it to expire before it is resubmitted, edit the expiration in the **Resubmit expiration time** field.

   The expiration time shown is local to the server. The value for this field must be formatted according to your specified locale. An example of the correct format for your locale is provided above the field.

4. If you want to enable tracing for the failed event, specify a new value in the **Trace Control** field. For detailed information about trace values, see the Monitoring topics in the WebSphere Business Process Management Information Center.
5. Do one of the following:
   - If the edited data is correct and you want to resubmit the event, click **Resubmit** to make the changes at a server level.
   - If you want to remove the changes you made, click **Undo local changes**.

   The edited failed event is resubmitted for processing and is removed from the failed event manager.

**Related tasks**

"Finding failed events" on page 62
Failed events are stored in a database and are retrieved through the search functionality of the failed event manager. You can search for all failed events on all the servers within the cell, or for a specific subset of events.

## Editing business data in a failed SCA event

Business data can be encapsulated into a business object, or it can be simple data that is not part of a business object. A failed event can have both simple data and a business object associated with it. Use the business data editor to edit the business data associated with a failed event before you resubmit it.

### Before you begin

If administrative security is enabled, you must be logged in as administrator or operator to perform this task.

### About this task

For each failed event, the editor displays the associated business data in a hierarchical format; the navigation tree at the top of the table is updated as you navigate through the parameters to give you a clear picture of where you are in the hierarchy.

You can edit only simple data types (for example, String, Long, Integer, Date, Boolean). If a data type is complex (for example, an array or a business object), you must navigate through the business data hierarchy until you reach the simple data types that make up the array or business object. Complex data is denoted by an ellipsis (`...`) in the Parameter Value column.

Note that you cannot use the failed event manager to edit business data for a Business Process Choreographer event. Instead, click the **Open calling process in Business Process Explorer** link from the failed event detail page and use Business Process Choreographer Explorer to make any permitted modifications.

**Important:** Any edits you make to business data are saved locally. Changes are not made to the corresponding business data in the server until you resubmit the failed event.

To edit business data associated with a failed SCA event, use the following procedure.

**Procedure**

### Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the failed event manager's Search Results page, click the ID (found in the Event ID column) of the failed event whose data you want to edit.
3. From the failed event detail page, click **Edit business data** to access the Business Data Editor collection page.

   This page displays a hierarchical view of all of the data associated with the failed event.
4. Navigate through the business data hierarchy by clicking on the name of each parameter (these appear as links in the Parameter Name column). When you have located the parameter whose value you want to edit, click its name.

   If the parameter has an editable value, the Business Data Editor page opens.
5. In the **Parameter value** field, specify the new value for the parameter.
6. Click **OK**.

   The change is saved locally and you are returned to the Business Data Editor collection page.
7. If you want to remove the changes you made, click **Undo local business data changes**.

   All of the edits are removed and the business data is returned to its original state.
8. If the edited business data is correct, click **Resubmit** to make the changes at a server level.

   The edited failed event is resubmitted for processing and is removed from the failed event manager.

## Resubmitting failed events

If you want to send an event again, you must resubmit it from the failed event manager. You can resubmit an event without changes, or, in some cases, you can edit the business data parameters before resubmitting it.

When a failed event is resubmitted, the processing resumes only for the failed branch, not for the entire event.

Tracing is available for resubmitted SCA events to help monitor the event's processing. Tracing can be set for a service or a component, and its output can be sent to a log or to the Common Event Infrastructure (CEI) server.

You can also use the event's unique event ID to track its success or failure. If a resubmitted event fails again, it is returned to the failed event manager with its original event ID and an updated failure time.

### Resubmitting an unchanged failed event

You can resubmit one or more unchanged failed events to be processed again. Processing resumes only for the failed branch, not for the entire event.

#### About this task

If administrative security is enabled, you must be logged in as administrator or operator to perform this task.

**Procedure**

### Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the Search Results page, select the check box next to each failed event you want to resubmit.
3. Click **Resubmit**.

### Results

Each selected event is resubmitted for processing and is removed from the failed event manager.

## Resubmitting a failed SCA event with trace

You can monitor the resubmission of a failed SCA event to determine whether it now succeeds. The failed event manager provides optional tracing for all failed events.

### About this task

Tracing can be set for a service or a component, and it can be output to a log or to the Common Event Infrastructure (CEI) server. For detailed information about setting and viewing trace, see the Monitoring topics in the information center.

If administrative security is enabled, you must be logged in as administrator or operator to perform this task.

**Procedure**

### Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the Search Results page, select the check box next to each failed event you want to resubmit.
3. Click **Resubmit with trace**.
4. From the Resubmit with Trace page, specify the level of trace you want to use in the **Trace control** field.

   By default, the value is SCA.LOG.INFO;COMP.LOG.INFO. With this setting, no trace occurs when the session calls an SCA service or executes a component.
5. Click **OK** to resubmit the failed event and return to the Search Results page.

### What to do next

To view the trace log for a resubmitted event, open the corresponding component logger or use the CEI log viewer.

# Managing failed JMS events

When problems processing a JMS request or response message create a failed JMS event in the Recovery subsystem, you must decide how to manage that event. Use the information in this topic to help you identify and fix the error and clear the event from the Recovery subsystem.

## About this task

To manage a failed JMS event, perform the following steps.

**Procedure**

## Procedure

1. Use the failed event manager to locate information about the failed JMS event, taking note of the exception type.
2. Locate the exception type in Table 9 to determine the location and possible causes of the error, as well as suggested actions for managing the failed event.

*Table 9. Failed JMS events*

| Exception type | Location of error | Possible cause of error | Suggested action |
|---|---|---|---|
| FaultServiceException | Fault handler or fault selector | There is malformed data in the JMS message. | 1. Inspect the JMS message and locate the malformed data.<br>2. Repair the client that originated the message so it creates correctly formed data.<br>3. Resend the message.<br>4. Delete the failed event. |
| | | There was an unexpected error in the fault handler or fault selector. | 1. Debug the custom fault selector or fault handler, fixing any errors identified.<br>2. Resubmit the failed event. |
| ServiceRuntimeException | Fault handler | The fault selector and runtime exception handler are configured to interpret the JMS message as a runtime exception. This is an expected exception. | Look at the exception text to determine the exact cause, and then take appropriate action. |
| DataBindingException or DataHandlerException | Data binding or data handler | There is malformed data in the JMS message. | 1. Inspect the JMS message and locate the malformed data.<br>2. Repair the client that originated the message so it creates correctly formed data.<br>3. Resend the message.<br>4. Delete the failed event. |
| | | There was an unexpected error in the data binding or data handler. | 1. Debug the custom data binding or data handler, fixing any errors identified.<br>2. Resend the message.<br>3. Delete the failed event. |

*Table 9. Failed JMS events (continued)*

| Exception type | Location of error | Possible cause of error | Suggested action |
|---|---|---|---|
| SelectorException | Function selector | There is malformed data in the JMS message. | 1. Inspect the JMS message and locate the malformed data.<br>2. Repair the client that originated the message so it creates correctly formed data.<br>3. Resend the message.<br>4. Delete the failed event. |
| | | There was an unexpected error in the function selector. | 1. Debug the custom function selector, fixing any errors identified.<br>2. Resend the message.<br>3. Delete the failed event. |

# Managing stopped Business Process Choreographer events

Use the failed event manager and Business Process Choreographer Explorer to manage stopped Business Process Choreographer events in any process state. Stopped events occur if a Business Process Execution Language (BPEL) instance encounters an exception and one or more activities enter the Stopped state.

## About this task

You can view, compensate, or terminate the process instance associated with a stopped Business Process Choreographer event. In addition, you can work with the activities associated with the event. viewing, modifying, retrying, or completing them as appropriate.

To manage stopped events originating from a long-running BPEL process, perform the following steps.

**Procedure**

## Procedure

1. Ensure the administrative console is running.
2. Open the failed event manager by clicking **Integration Applications → Failed Event Manager**.
3. Perform a search to find the stopped Business Process Choreographer event or events you want to manage.
4. For each stopped event you want to manage, do the following:
   a. Click the stopped event ID in the Event ID column of the Search Results page.
   b. From the event detail page, click **Open calling process in Business Process Choreographer Explorer**.
   c. Use Business Process Choreographer Explorer to manage the event and its associated activities.

# Finding business process instances related to a failed event

If a failed event is generated from a business process, the failed event manager provides a link to view that business process instance in Business Process Choreographer Explorer.

## Before you begin

You must be logged in as administrator or operator to perform this task.

## About this task

Examining the business process instance that generated the failed event can give you additional information about how or why the event failed. The business process instance and the failed event are linked by a common session ID.

**Note:** Not all failed events are generated from a business process instance.

To find and examine a business process instance related to a failed event, use the following procedure.

**Procedure**

## Procedure

1. From within the administrative console, use the failed event manager to locate the failed event you want to investigate. See "Finding failed events" on page 62 for instructions on how to search for failed events.
2. From the Failed Event Details page for that event, click **Open calling process in Business Process Choreographer Explorer**.

## Results

The Business Process Choreographer Explorer opens in a new browser window and displays information about the related process instance.

# Finding Common Base Events related to a failed event

A failed event can be related to one or more Common Base Events. The failed event manager provides a link to view related Common Base Events in the Common Base Event Browser.

## Before you begin

You must be logged in as administrator or operator to perform this task.

## About this task

Examining related Common Base Events can give you additional information about how or why the original event failed. The failed event and any related Common Base Events are linked by the same session ID.

To find and view related Common Base Events, use the following procedure.

**Procedure**

### Procedure

1. From within the administrative console, use the failed event manager to locate the failed event you want to investigate. See "Finding failed events" on page 62 for instructions on how to search for failed events.
2. From the Failed Event Details page for that event, click **Browse Related Common Base Events**.

### Results

The Common Base Event Browser opens in a new browser window and lists any Common Base Events related to the original failed event.

## Deleting failed events

If you do not want to resubmit a failed event, or if you have failed events that have expired, use the failed event manager to delete them from the server. The failed event manager provides three options for deleting failed events.

### Before you begin

You must be logged in as administrator or operator to perform this task.

### About this task

To delete one or more failed events, use the following procedure.

**Procedure**

### Procedure

1. Ensure that the failed event manager is open and that you have retrieved a list of the failed events on your system.
2. From the failed event manager's Search Results page, do one of the following:
   - If you want to delete one or more specific failed events, select the check box next to each event and then click **Delete**.
   - If you want to delete only those failed events that have expired, click **Delete expired events**. Note that this deletes only the expired events in the current set of search results.
   - If you want to delete all failed events on the server, click **Clear all**.

## Troubleshooting the failed event manager

This topic discusses problems that you can encounter while using the failed event manager.

**Note:** This topic does not discuss how to use the failed event manager to find, modify, resubmit, or delete failed events on the system. For information about managing failed events, see *Managing WebSphere Process Server failed events* in the information center.

Select the problem you are experiencing from the table below:

| Problem | Refer to the following |
| --- | --- |
| I am having trouble entering values in the Search page's **By Date** tab | "Values in the By Date and From Date field automatically change to default if entered incorrectly" on page 50 |

| Problem | Refer to the following |
| --- | --- |
| I am having trouble deleting expired events | "Using the Delete Expired Events function appears to suspend the failed event manager" on page 51 |
| I am having trouble with failed events not being created | "Failed events are not being created" on page 51 |
| I am having trouble retrieving or deleting large numbers of failed events | "The server fails when retrieving or deleting large numbers of failed events" on page 51 |

## Values in the By Date and From Date field automatically change to default if entered incorrectly

The Search page's **From Date** and **To Date** fields require correctly formatted locale-dependent values. Any inconsistency in the value's format (for example, including four digits in the year instead of 2, or omitting the time) will cause the failed event manager to issue the following warning and substitute a default value in the field:

```
CWMAN0017E: The date entered could not be parsed correctly:
your_incorrectly_formatted_date. Date: default_date is being used.
```

The default value of the **From Date** field is defined as `January 1, 1970, 00:00:00 GMT`.

**Important:** The actual default value shown in your failed event manager implementation will vary depending on your locale and time zone. For example, the From Date field defaults to `12/31/69 7:00 PM` for a workstation with an en_US locale in the Eastern Standard Time (EST) time zone.
The default value for the **To Date** field is always the current date and time, formatted for your locale and time zone.

To avoid this problem, always enter your dates and times carefully, following the example provided above each field.

## Using the Delete Expired Events function appears to suspend the failed event manager

If you use the Delete Expired Events button in situations where there are many failed events in the current search results, or where those events contain a large amount of business data, the failed event manager can appear to be suspended indefinitely.

In this situation, the failed event manager is not suspended: it is working through the large data set, and will refresh the results set as soon as the command completes.

## Failed events are not being created

If the Recovery subsystem is not creating failed events, go through the following checklist of potential causes:

- Ensure that the wpsFEMgr application is running. If necessary, restart it.
- Ensure that the failed event manager's database has been created, and that the connection has been tested.

- Ensure that the necessary failed event destination has been created on the SCA system bus. There should be one failed event destination for each deployment target.
- Ensure that the Quality of Service (QoS) **Reliability** qualifier has been set to `Assured` for any Service Component Architecture (SCA) implementation, interface, or partner reference that participates in events you want the Recovery service to handle.

## The server fails when retrieving or deleting large numbers of failed events

The server can fail if you use the failed event manager to retrieve or delete a large number of failed events at once. To prevent this problem, be sure to check the total failed event count on the main page of the failed event manager before performing search or delete actions. If you have a large number of events, do not try to retrieve or delete all of them at once. Instead, use the **Search failed events** option to return a subset of failed events that match specific criteria. You can then delete all of the events in the filtered result set without causing a server failure.

**Note:** There are several factors in your environment that can affect the number of failed events the server can return or delete in a single request, including other processes running on the same machine as the server and the amount of available memory.

# Chapter 12. Recovering from a failure

Recovering from a failure requires an understanding of standard system processing in the event of a failure, as well as an understanding of how to analyze problems that may be the cause of a failure.

## Overview of the recovery process

The recovery process encompasses a set of tasks that include both analysis and procedures.

When you must recover from a failure, these are the high-level steps to follow:

- Familiarize yourself with the possible kinds of failures. See *Triggers for recovery* for more information.
- Assess the state of the system. See *Assessing the state of the system* for more information.
- Form a hypothesis about what the problem is.
- Collect and analyze the data.
- Refer to other topics in this information center for instructions on fixing the problem.

## Triggers for recovery

The need for solution recovery can result from a variety of triggers.

### Situations from which solution recovery is necessary

Solution recovery is the process of returning the system to a state from which operation can be resumed. It encompasses a set of activities that address system failure or system instability that can be triggered by unforeseen circumstances.

You may need to perform solution recovery activities for the following circumstances:

- **Hardware failure**

  Abnormal termination or system down can be caused by a power outage or catastrophic hardware failure. This can cause the system (all if not most JVMs) to stop.

  In the case of a catastrophic hardware failure, the deployed solution may enter an inconsistent state on restart.

  Hardware failures and environmental problems also account for unplanned downtime, although by far not as much as the other factors.

  You can reduce the likelihood of hardware failures and environmental problems by using functions such as state-of-the-art LPAR capabilities with self-optimizing resource adjustments, Capacity on Demand (to avoid overloading of systems), and redundant hardware in the systems (to avoid single points of failure).
- **System is not responsive**

  New requests continue to flow into the system but on the surface it appears that all processing has stopped.
- **System is unable to initiate new process instance**

The system is responsive and the database seems to work correctly. Unfortunately, new process instance creation is failing.

- **Database, Network or Infrastructure Failure**

  In the case of fundamental infrastructure failure, the solution may require administration to restart/resubmit business transactions after the infrastructure failure is resolved.

- **Poor tuning or a lack of capacity planning**

  System is functional but is severely overloaded. Transaction time-outs are reported and there is evidence of an overflow of the planned capacity.

  Incomplete capacity planning or performance tuning can cause this type of solution instability.

- **Defects in application module development**

  The modules that are part of a custom developed solution can have bugs. These bugs can result in solution instability and failed services.

  Bugs in a custom developed solution can result from a variety of situations, including (but not limited to) the following:

  – Business data that was not planned for or unforeseen by the application design.

  – An incomplete error handling strategy for the application design.

    A detailed error handling design can reduce solution instability.

- **WebSphere software defect**

  A defect in the WebSphere product causes a backlog of events to process or clear.

## Assessing the state of the system

The first thing to do when an abnormal condition occurs is to take the *pulse* of the overall system and get a feel for how much or how little of the system is operational and how much of it is rendered 'out of service' by whatever the external stimuli was that caused this condition.

Address a predefined set of questions to assess the extent of the outage. The following list provides examples of predefined questions designed to help you gather the appropriate information:

1. Is this system still performing work?

   Determine if system is still operational. Often times, a system can be operational, but because of overload or inappropriate tuning, or both, the system is not completing tasks quickly and/or is attempting to do work that is indeed failing.

   The litmus test for each of these questions will be specific to the nature of the deployed solution.

2. What special error handling support is built-in to the application?

   If there is a lot of automated retry and various support logic, then the application itself might shield some errors from manifesting the IT operator.

   These conditions must be known and documented for reference by the recovery team.

Things you can do to help assess the state of the system include the following:

1. Check to see if the server is at least running.

   Do you see the PID or get a positive feedback from the deployment manager via the administrative console?

2. Check to see if there are locks in the database(s) or any unusual database traffic.

   Most databases will have facilities to look at locks. Depending on the deployment topology, there maybe multiple databases.

   - Messaging Engine Database
   - Business Process Container Database
   - WebSphere Process Server Common Database (Failed Events and Relationship data)

3. Check to see what the status of the messaging system is.

   Check for events or messages in the following locations:

   - Business Process Choreographer Hold and Retention Destinations
   - Number of failed events
   - Number of messages on the solutions module destinations

4. Check to see if the database is functioning.

   Can you perform some simple SELECT operation, on unlocked data in a reasonable amount of time?

5. Check to see if there are errors in the database log.

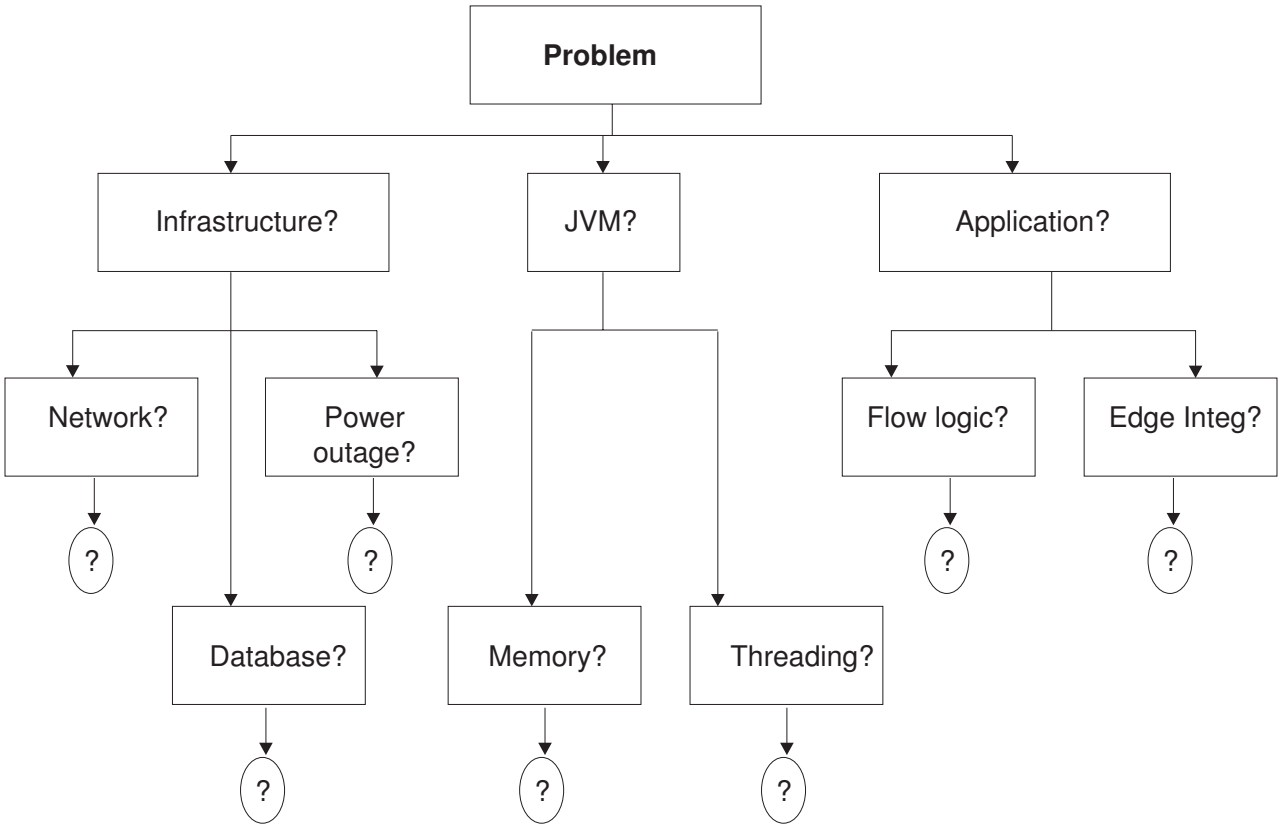
If the database is not working properly, then recovering the database (so that it can at least release locks and perform simple selects) is vital to system recovery.

If the messaging system is not working properly, then recovering the messaging subsystem, so that it can at least be viewed and managed, is also vital to system recovery.

**Note:** A 'bottoms up' approach is not always conclusive. However, chances of successful recovery vary based on these basic activities.

From these basic procedures and health check kinds of activities, we need to now start to look at some specific situations. Patterns will be described, specifics will be given and insights as to what is going on under the covers will be provided.

Realize that this situational analysis is a read-only activity. While it provides vital information from which to determine the appropriate recovery actions, it should not change the state of the system under review. It is impossible to predict and provide prescriptive actions for all possible causes of a system outage. For example, consider the following decision tree:




There are many broad categories to investigate in the event of an unplanned outage. These broad categories will have sub categories and so on. Defining prescriptive actions for each node and the subsequent node will depend on the results of each investigation. Because this type of relationship is difficult to convey in a document form, utilizing a support tool such as *IBM Guided Activity Assist* to interactively walk you through the investigative and decision making process is recommended. As we progress from the top to each child node, it is important to conduct the appropriate level of situational analysis.

## Recovery: Analyzing the problem

For all unplanned system events, a set of basic recovery procedures can be leveraged at the point of identification.

There are several well defined steps to situational analysis. The steps are listed below.

1. Define the question
2. Gather information and resources (observe)
3. Form hypothesis
4. Perform experiment and collect data
5. Analyze data
6. Interpret data and draw conclusions that serve as a starting point for new hypothesis

For each production scenario the symptoms that initiate a recovery action may vary.

It is important to follow the guidelines for situational analysis and take the corrective action relative to the symptoms that are presented.

## Situational analysis

Situational analysis is the cyclical execution of the scientific method and can take into account various situations that will initiate a recovery procedure.

The following list is of the different types of situations that will initiate a recovery procedure:

- Abnormal termination or system down

  A power outage or catastrophic hardware failure has caused the system (all if not most JVMs) to stop.
- System is not responsive

  New requests continue to flow into the system but on the surface it appears that all processing has stopped.
- System is functional but is severely overloaded

  Transaction time-outs are reported and there is evidence of an overflow of the planned capacity.
- System is unable to initiate new process instance

  The system is responsive and the database seems to work correctly. Unfortunately, new process instance creation is failing.

# Recovery: First steps

Administrators can facilitate solution recovery processes by following a first steps checklist of general practices.

The following list describes actions that you **SHOULD NOT TAKE** under normal circumstances when trying to recover a solution.

**Note:** There could be special situations for which you might need to perform some of the actions listed below. However, you should never initiate any of these actions without first consulting with the WebSphere Process Server support organization.

- Do not delete the transaction log file

  The transaction (tranlog) log file stores critical transactional data that is written to databases. It is an internal file that WebSphere Application Server uses to manage in-flight transactions and attempt to recover them should the server crash.
- Do not have the transaction logs local on the cluster members

  Put the transaction logs on a shared drive. This is the only way to allow peer recovery, which helps minimize the downtime during recovery.
- Do not attempt database operations where the result set is large enough to create additional resource contention (OutOfMemory)
- Avoid performing Business Process Choreographer Explorer operations that return large result sets.
- Avoid executing administrative scripts on process instances without considering the result set size.
- Do not drop and/or recreate databases in production
- Do not uninstall applications as part of your standard recovery procedures

  You should only uninstall applications with the direction from the IBM support organization.

- Do not enable too much trace if the system is overloaded.

  Too much trace will cause a slowdown in system throughput and might cause transaction time-outs. Too much trace can often add to the problems that need to be addressed, rather than providing insight as to how to solve the original problems.

  Get immediate assistance from IBM support to define the correct trace specification.
- Do not experiment or try out new scripts or new commands on production systems.
- Do not run your production servers in *development mode*

  Enabling the **Run in development mode** option may reduce the startup time of an application server. This may include JVM settings such as disabling bytecode verification and reducing JIT compilation costs.


The following list describes the recommended actions when it comes to recovery.

- Always take a *snapshot* of the configuration tree, the PI file of the application in question and the logs which are available.

  Logs may be overwriting themselves depending on the configuration. Capturing a set early and often is an important step for postmortem analysis. See the topic on *IBM Support Assistant (ISA)* for details on the IBM Support Assistant, which helps with this type of activity.
- Always understand your database settings, especially related to database transaction log file size, connection pools, and lock timeouts.

## Failed-event locations: Where does the data go?

For all (production and test) recovery activities there are a finite number of locations in the solution where events accumulate.

By adhering to guidelines and preventive measures described in *Planning error prevention and recovery*, all business events and associated data will reliably accumulate in one of these locations.

If you do not adhere to sound architectural and application development practices, then a percentage of inflight events may end up in an inconsistent state, from which recovery cannot be attained. Under such circumstances, (presumably

identified during testing cycles) post recovery investigation and clean up is necessary to correct the issue so that future recovery activities are completely successful.

In order to accurately describe the following scenarios, it is important to put the information in the context of a use case.

## Use case: recovering data from failed events

A use case provides a context for a recovery scenario. In the use case, a business has an application that receives a request to create a new Account.

The solution is comprised of multiple modules as recommended through module best practices.

The first module mediates the request and delegates work to an Account Creation process. In the example below we have implemented the solution as separate modules where the request is passed between the mediation module (AccountRouting) and the processing module (AccountCreation) via an SCA import/export. See the following screen capture for an illustration of the two modules.

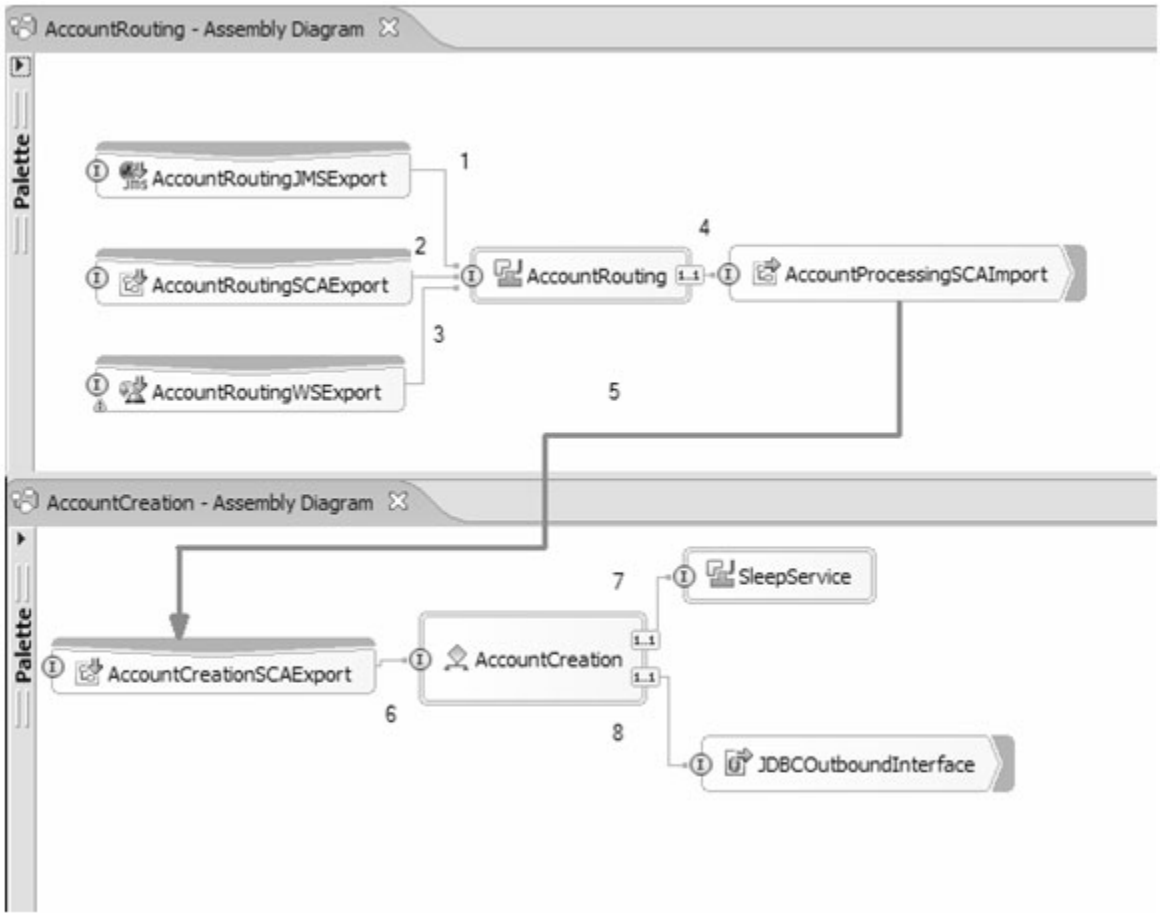



*Figure 1. Assembly diagram of account routing process*

From the assembly diagram shown in Figure 1, you can begin to see at what locations in the flow that failures might occur. Any of the invocation points in the

assembly diagram can propagate or involve a transaction. There are a few areas in the flow where data will collect as a result of application or system failures.

In general, transaction boundaries are created and managed by the interaction (synchronous and asynchronous) between components and import/export bindings and their associated qualifiers. Business data accumulates in specific recovery locations most often due to transaction failure, deadlock or rollback.

Transaction capabilities within WebSphere Application Server help WebSphere Process Server enlist transactions with service providers. These enlisted interactions are particularly important to understand with respect to import and export bindings. Understanding how imports and exports are used within your specific business cases is important in determining where events in need of recovery accumulate.

An error handling strategy should define interaction patterns, transactions used, import and export usage prior to developing the application. The solution architect should identify the preferences to use, guidelines, that are then used as the application is created. For example, the architect needs to understand when to use synchronous vs. asynchronous calls, when to use BPEL fault handling and so forth. The architect needs to know whether or not all services can participate in transactions, and for those services that can not participate, how to handle compensation if problems are encountered.

Additionally, the application shown in the assembly diagram in Figure 1 on page 83 leverages connectivity groups and module development best practices. By leveraging this pattern we now have the ability to stop the inbound flow of new events by stopping the `AccountRouting` module.

The following sections address the location of business data in the case of failure and recovery.

## Business Flow Manager or Human Task Manager

In our business case, we leverage a BPEL process for `AccountCreation` process.

With regard to recovery, there are a some questions you need to ask yourself with respect to BPEL and Human Task management as follows:

1. What type of process is being run (short running or long running, business state machine, human task) ?

   Short running processes are known as microflows.
2. Is the process developed properly and using fault handling to promote data integrity?
3. How are the invocation patterns and unit of work properties configured to predict and control transaction boundaries?

Knowing the answers to these questions will impact your recovery strategy for invocations 7 and 8 shown in the assembly diagram, as highlighted in the screen capture below:

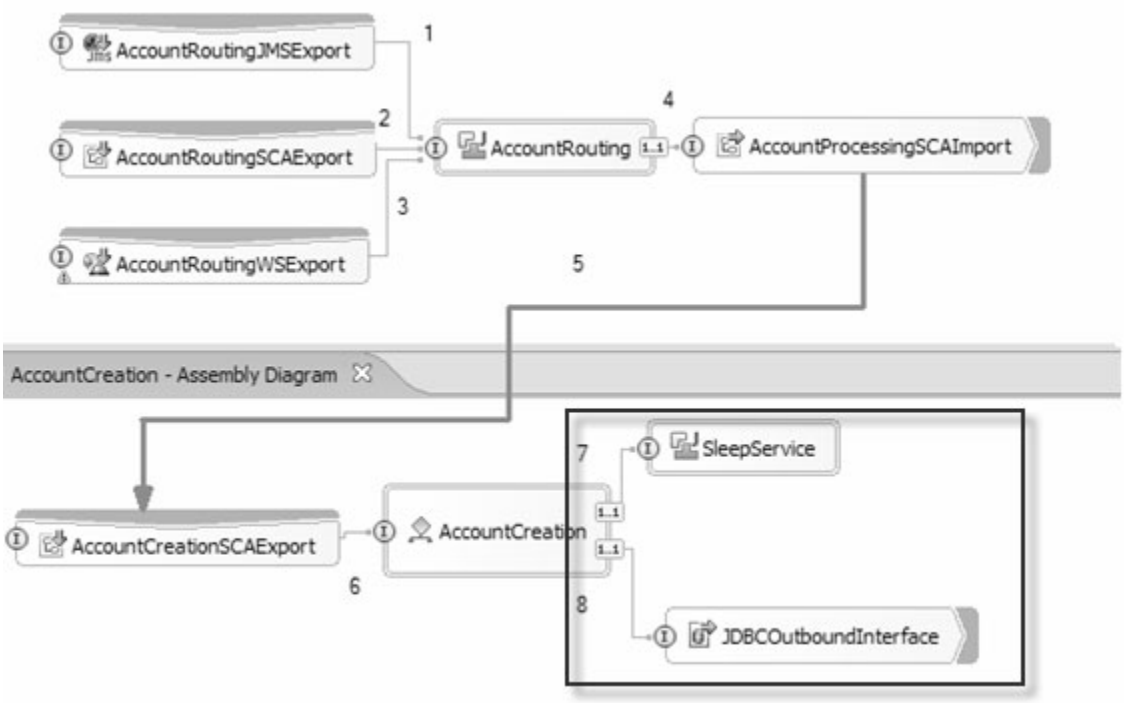



*Figure 2. Assembly diagram of account routing - invocations 7 and 8*

Stateful components, such as Long Running BPEL processes and Business State Machines, involve many database transactions where process activity changes and state changes are committed to the database. The work progresses by updating the database and placing a message on an internal queue that describes what is to be done next. More information on Macro flow transactions is available in the information center topic titled *Transactional behavior of long-running processes*.

If there are problems processing messages that are internal to the Business Flow Manager, these messages are moved to a *Retention Queue*. The system attempts to continue to process messages. If a subsequent message is successfully processed, the messages on the Retention Queue are resubmitted for processing. If the same message is placed on the Retention Queue five times, it is then placed on the Hold Queue. Information, such as what internal queues are used or retry algorithms for these queues is described in detail in the information center topic titled *Recovery from infrastructure failures*.

Additional information about viewing the number of messages and replaying messages can be found in Replaying Messages from the Retention Queue / Hold Queue.

## Failed event manager

The Failed event manager (FEM) is used to replay events or service invocation requests that are made asynchronously between *most* component types.

Failed events are created if the `AccountRouting` component makes an asynchronous call to the SCA Import binding `AccountCreationSCAImport` and a `ServiceRuntimeException` is returned.

It is important to note that failed events are not generated in most cases where BPEL is the client in the service interaction. This means that the invocation for 7 and 8 (as shown in Figure 2 on page 85) will not typically result in a failed event. BPEL provides fault handlers and other ways to model for failure. For this reason, if there is a `ServiceRuntimeException` (SRE) failure calling "JDBCOutboundInterface", the SRE is returned to the BPEL for processing. The error handling strategy for the project should define how runtime exceptions are consistently handled in BPEL.

It is important to note however, failed events are created for asynchronous response message for the BPEL client if these messages can not be delivered to the process instance due to an infrastructure failure.

The following diagram illustrates how the Failed event manager component works. Descriptions of the processing associated with each numbered step are provided following the diagram.

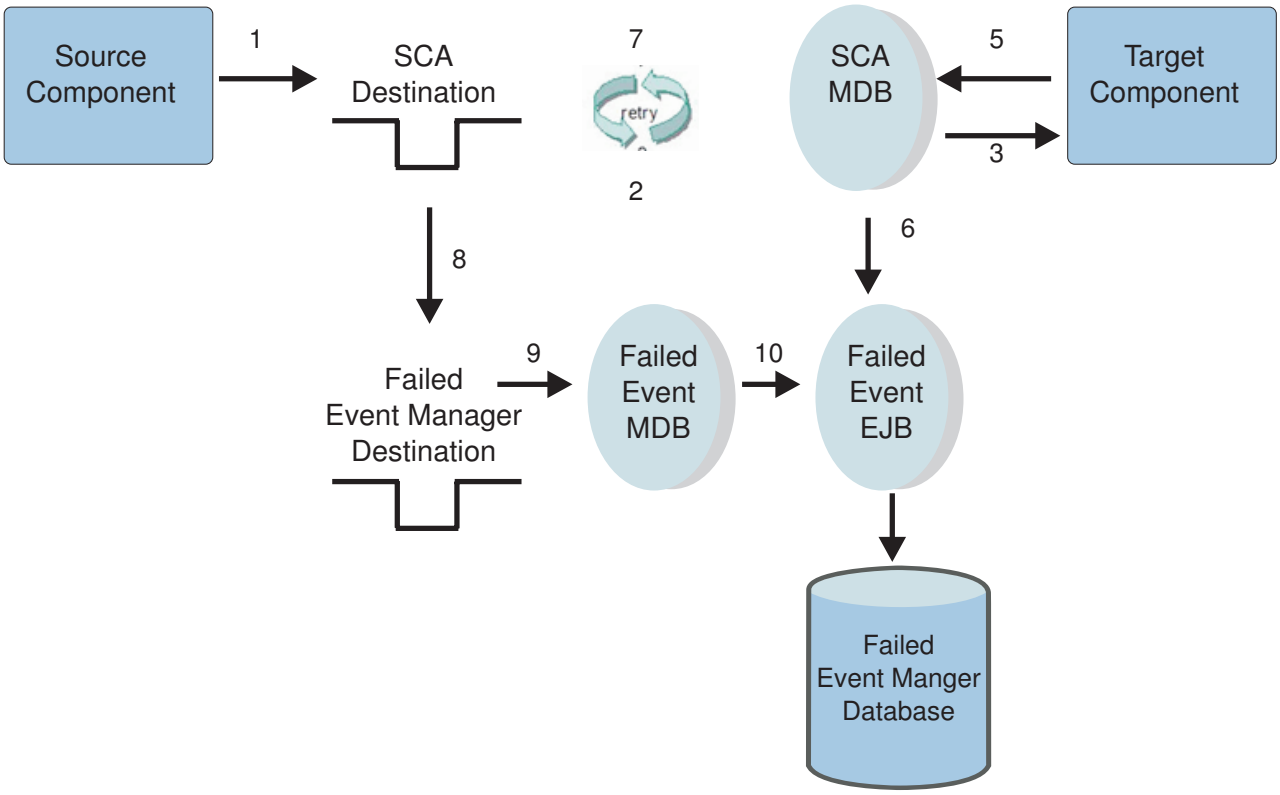



*Figure 3. Failed event manager processing*

**Failed event manager processing**

1. The source component makes a call using an asynchronous invocation pattern
2. The SCA MDB picks the message up off the SCA destination
3. The SCA MDB makes the call to the correct target component
4. The target component throws a `ServiceRuntimeException`
5. The SCA MDB transaction rolls back to the SCA destination
6. The exception information is stored to the Failed event manager database with a status of *not confirmed*
7. The invocation is retried by the SIBus n number of times

The retry limit default value is 5 - one original and 4 retries. You can change the default value in the administrative console. For example, given an SCA module M, you could navigate to **Buses → SCA.SYSTEM.<CELL>.BUS → Destinations → sca/M** and change the value in the *Maximum failed deliveries* field.

8. After the number of retries reaches the specified limit, the message is moved to the FEM destination.
9. The Failed Event Manager database picks up the message
10. The Failed Event Manager database updates the failed event in the database and the status is set to *failed*.

**When are "failed events" created?**

As stated, failed events are neither created for synchronous invocations nor typically for two-way business process interactions.

Failed events are generally created when clients use an asynchronous invocation pattern and a ServiceRuntimeException is thrown by the service provider.

If everything is done synchronously and in the same transaction, data is not collected anywhere. Instead it is all rolled back to the client that made the call. Where ever a commit is occurs, data collects. If the calls are all synchronous, but there are multiple commits, then these commits become an issue.

In general, you should use asynchronous processing calls or long running BPEL if multiple transactions are needed. So each ASYNC call is a chance for data to collect. Long running BPEL process are a collection point.

*Table 10. Invocation patterns and relationship to the creation of failed events:* **Service Business Exceptions**

| Invocation Pattern | Failed Event Created Y/N? | Notes |
|---|---|---|
| Synchronous | No | Failed events are not created for service business exceptions or when using a synchronous pattern |
| Asynchronous - One Way | No | By definition, one-way invocations cannot declare faults, meaning, it is impossible to throw a ServiceBusinessException. |
| Asynchronous - Deferred Response | No | Failed Events are not created for service business exceptions |
| Asynchronous - Callback | No | Failed Events are not created for service business exceptions |

*Table 11. Invocation patterns and relationship to the creation of failed events:* **Service Runtime Exceptions**

| Invocation Pattern | Failed Event Created Y/N? | Notes |
|---|---|---|
| Synchronous | No | Failed events are not created for service runtime exceptions or when using a synchronous pattern. |
| Asynchronous - One Way | Yes | |
| Asynchronous - Deferred Response | Yes | |
| Asynchronous - Callback | Yes | |
| BPEL - Two Way | No | Failed Events are not created when the source component is a business process.<br>**Note:** For an asynchronous call, if the response can not be returned to BPEL, then a failed event is created. |

*Table 11. Invocation patterns and relationship to the creation of failed events:* **Service Runtime Exceptions** *(continued)*

| Invocation Pattern | Failed Event Created Y/N? | Notes |
|---|---|---|
| BPEL - One Way | Yes | |

For additional information, please review the information center topic titled *Managing failed events*.

Additional information about viewing and resubmitting failed events can be found in section *Resubmitting Failed Events*.

## Service Integration Bus Destinations

Messages that are waiting to be processed may accumulate in a few Service Integration Bus (SIBus) destinations. For the most part these destinations are "system" destinations. Messages within these destinations typically are a mixture of three types as follows:

- Asynchronous requests for processing
- Asynchronous replies to requests
- Asynchronous messages that failed deserialization or function selector resolution

**Note:** Asynchronous replies can be valid Business Objects or faults returned as a result of a request.

**SCA Module Destination**

Again, referring back to our business case.

There would be two "SCA Module" destinations in the solution:

- sca/AccountRouting
- sca/AccountCreation

These destinations are created when the module is deployed to an application server or a cluster.

There are rare opportunities for messages to accumulate in these destinations. The accumulation of messages in these locations is a strong indication that there maybe a performance problem or an application defect. Investigate immediately. It is important to monitor the depth of the module destinations (with your chosen IT monitoring solution) as a back up of messages could lead to a system outage or a prolonged recycle time.

We call these "SCA Module" destinations because the generated name is the same as the module name with the additional "sca/". These destinations are pivotal in the functioning of SCA asynchronous invocations (brokering requests and responses). There are a varying number of additional destinations that are generated during application install on the SCA.SYSTEM bus but for the purpose of the discussion we'll be addressing the importance of the "SCA Module" destination.

**System Integration Bus Retry**

As we learned above, the FEM has a built-in retry mechanism with the SCA message driven bean (MDB). This retry behavior can be controlled by modifying the "Maximum Failed Deliveries" attribute on the module destination.

**Note:** Typically, there is no reason to adjust this retry capability. The information provided here is for completeness.

Referring to our business case, there are a number of SI Bus destinations created by SCA to support asynchronous communication.

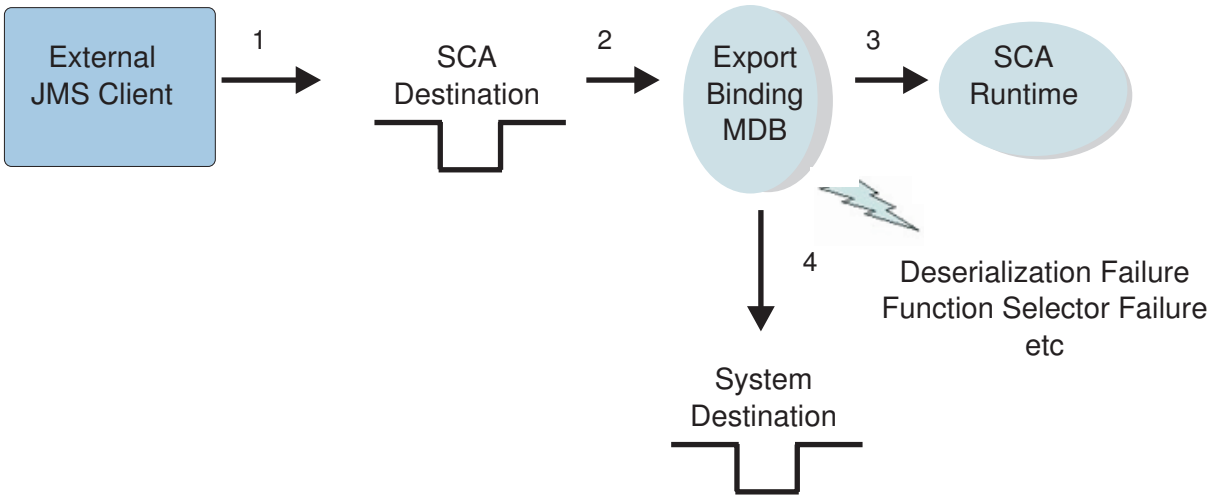
As we have learned, one of these destinations is called "sca/AccountRouting" You can adjust the number of retries that happen during a ServiceRuntimeException of an asynchronous service invocation by changing the value of the "Maximum Failed Deliveries" property via the admin console. However, you may not set the value less than 2 in modules with a BPEL process. The second delivery is required to return ServiceRuntimeExceptions back to the BPEL for processing.

**System Exception Destinations**

The failed event manager is one place where we can look to administer failures. When dealing with Imports and Exports that are JMS or EIS based, we have to consider another important location.

Destinations on the SCA.Application bus are configured to route failed messages to the SIB system exception destination for that bus. Thus, if a JMS export picks up a message from the SCA Application bus and runs into a rollback situation, the failed message will be routed to the SIB system exception destination instead of the WBI recovery exception destination. This scenario differs from the failed event discussion above in that a failure to deserialize a message on the SCA.Application bus will not result in a failed event. There is a system exception destination on every bus within the solution. These destinations must be monitored and administered much like the "dead letter queue" common to MQ infrastructures.

Consider the following scenario.




An external JMS client places a message on an inbound queue exposed via a JMS Export The JMS Export Binding MDB picks up the message for processing. From here, one of two things will happen:

1. The JMS export successfully parses the message and determines which operation on the interface to invoke at which point the message is sent to the SCA runtime for processing.
2. The JMS export fails to recognize the message body as a valid business object or the JMS export binding *deserializes* the message body but is unable to determine the appropriate operation on the interface to invoke. At this point the message is placed on the System Exception Destination for the bus.

We can have this type of failure when trying to receive requests from the `AccountRoutingJMSExport` (1). This export is a JMS export and there is a possibility that events can accumulate on the System Exception Destination on the SCA.Application.Bus. Use the chosen IT monitoring solution to observe the depth of this destination.

**Failed Event Manager and SIB Destinations**

For WebSphere Process Server, the exception destination is set to the WebSphere Process Server exception destination queue. This queue follows a naming convention as follows:

```
Node name: WPSNode
Server name: server1
Recovery exception destination: WBI.FailedEvent.WPSNode.server1
```

In general, all the destinations created on the SCA.System bus will be configured to route failed messages to the recovery exception destination.

When a system failure occurs, in addition to capturing the failed message in this exception destination, the WebSphere Process Server recovery feature also generates a failed event that represents the system error and stores it into the Recovery database as described in the Failed Event Manager section of this document.

## Summary

In summary, WebSphere Process Server provides administrative capabilities above and beyond the underlying WebSphere Application Server platform. Proper measures should be made to understand and use these capabilities along with following the guidance provided in the Planning error prevention section of *Planning error prevention and recovery*.

*Table 12. Administrative capabilities to help manage failures*

| Administrative Capability | Bundled With WebSphere Process Server Y/N? | Summary |
|---|---|---|
| Business Process Choreographer Explorer | Yes | Read/Write/Edit/Delete Access. This is the central place to administer business processes and human tasks. |
| Failed Event Manager | Yes | Read/Edit/Delete Access. This is the central place to administer Service Runtime Exceptions and other forms of infrastructure failures. |
| Service Integration Bus Browser | Yes | Read/Delete. Use the Service Integration Bus Browser on the administrative console for browsing and performing day-to-day operational tasks on service integration buses. |

**Note:** The number of events or records that can be simultaneously administered by these tools are specific to external factors such as memory allocation, result sets and DB tuning, connection timeout. Please run tests and set the appropriate thresholds to avoid exceptions (OOM, TransactionTimeOut).

**Related concepts**

"Retention queues and hold queues" on page 97
When a problem occurs while processing a message, it is moved to the retention queue or hold queue.

# Recovery troubleshooting tips

This section provides a list of tips for troubleshooting the recovery process.

## Restarting deployment environments

As one step in a recovery process, you may need to restart of you deployment environment.

### About restarting deployment environments

The procedure to restart a deployment environment varies depending on the topology. Topologies are based on system configuration patterns, each pattern designed to meet particular business requirements.

WebSphere Process Server supports a set of predetermined deployment environment configuration patterns. If none of the patterns meet your requirements, you can plan and create your own customized deployment environment.

In any given deployment environment configuration pattern there are a number of servers running as JVM processes. In general there are three types of servers as follows:

- Messaging Servers

  Messaging servers are responsible for providing the Service Integration Bus (SIB) messaging infrastructure.
- WebSphere ESB Servers

  Servers with profiles capable of only hosting and running mediation modules.
- WebSphere Process Servers

  Servers with profiles capable of hosting and running all module types. This profile hosts the Business Process Choreographer component.
- Support Servers

  This server is responsible for providing support and monitoring services such as the Common Event Infrastructure CEI.

The deployment patterns differ in how you group and organize all the functional components, so that the pattern can address your business requirements in the most cost effective fashion. For more advanced and highly available environments, the servers would reside in clusters that are distributed across physical resources.

### General practice for restarting servers as part of a recovery operation

A general model for starting servers is to start the messaging servers first, then the support servers and lastly the WebSphere Process Server servers. Each application architecture may have specific dependencies between application components that need to be taken into consideration.

Shutdown basically happens inverse to the startup procedure, starting with the application server clusters and ending with shutting down the messaging infrastructure after it has had time to quiesce and process any inflight transactions.

**Related tasks**

Choosing your deployment environment pattern
You can configure your deployment environment by choosing one of the IBM-supplied patterns or by creating your own custom deployment environment. This topic lists the features supported by each IBM-supplied pattern.

Planning your deployment environment
Setting up your deployment environment involves many decisions that affect everything from the number of physical servers to the type of pattern you choose. Each decision will affect how you set up your deployment environment.

**Related information**

WebSphere Process Server and WebSphere Enterprise Service Bus deployment patterns: Selecting your deployment pattern

## Viewing the service integration bus

Use the Service Integration Bus browser on the administrative console to view the service integration bus.

### Before you begin

Make sure you understand how the SCA system bus is used.

### About this task

The Service Integration Bus Browser provides a single location for browsing and performing day-to-day operational tasks on service integration buses.

Viewing the service integration bus is a useful way to determine if messages are accumulating on the SCA Module destinations.

The accumulation of messages on the SCA Module destinations is a strong indication that there maybe a performance problem or an application defect.

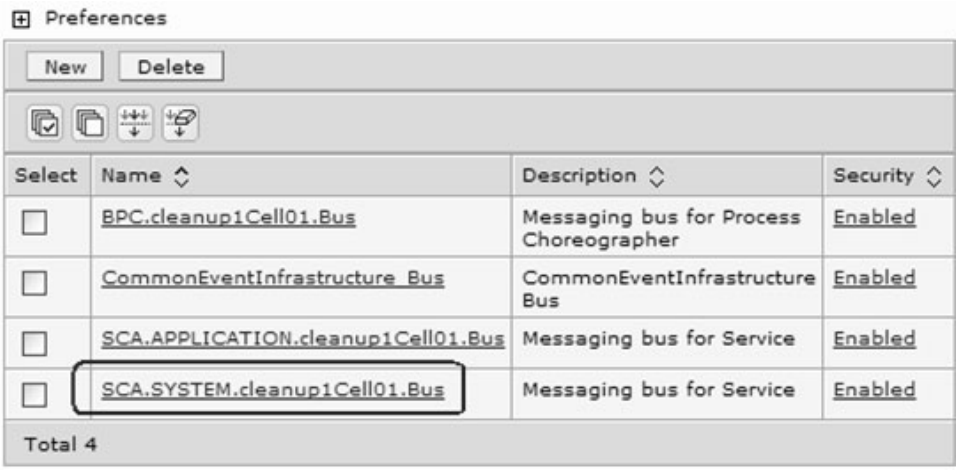
It is a good idea to periodically view the messages and determine if there are any messages have become locked for an extended duration of time as this may indicate that there are "indoubt transactions".

**Procedure**

### Procedure

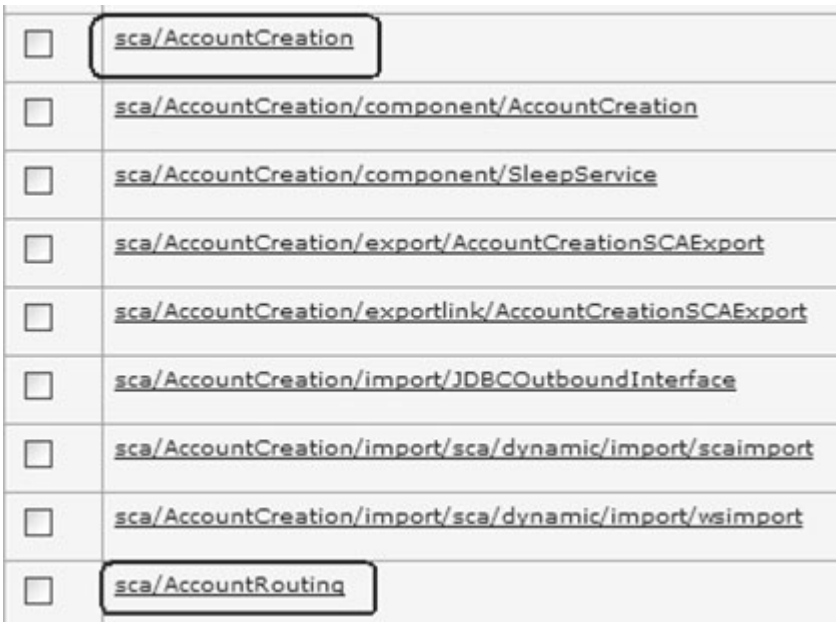
1. From the administrative console, expand **Service integration**.
2. Select **Buses**.


3. Select the appropriate messaging bus for the service. The following example shows the messaging bus named `SCA.System.cleanup1cell01.bus` highlighted, where `cleanup1cell01` is the name of the cell.



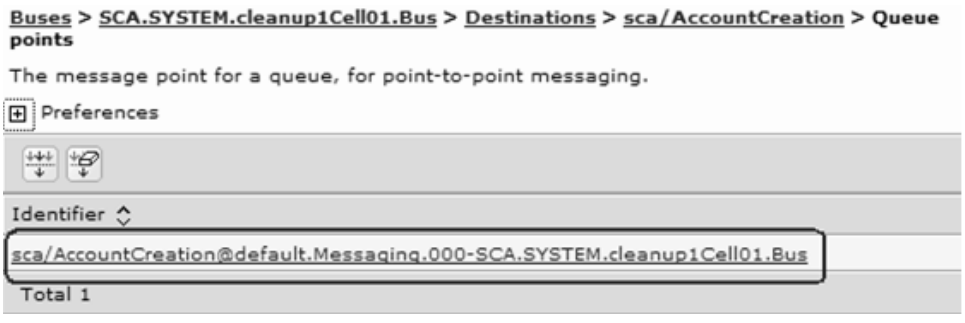

4. Select **Destinations**


5. Review the relevant information. You should look at the destinations named `sca/XYZ`, where `XYZ` is the name of the module. For example, for modules named `AccountRouting` and `AccountCreation`, you would look for the following destinations:

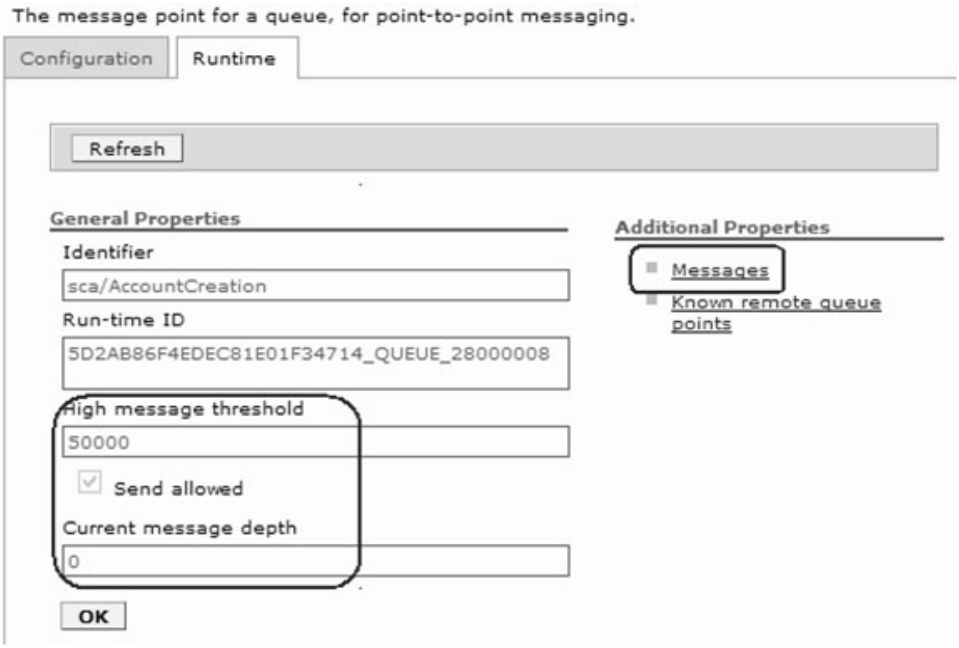



6. Select the link text for the destination that you are interested in viewing.
   This will link you to a general properties page for the destination that you want to view.
7. From the general properties page of the destination, select the **Queue points**


8. From the Queue points page, select the link for the message point.




9. Select the **Runtime** tab.
   From this screen you can see the current message "depth" and the threshold. Selecting the **Messages** link will allow you to view the message contents.




Ideally, use an appropriate IT monitoring tool and set alert thresholds for these destinations. The threshold value would be established during the performance test phase for the application.

Messages on a production system should never be deleted unless explicitly directed to do so by the SCA L3 team.

**Related tasks**

"Resolving indoubt transactions" on page 100
Transactions can become stuck in the indoubt state indefinitely due to exceptional circumstances, such as the removal of a node causing messaging engines to be destroyed.

**Related information**

SCA resources

SCA system bus

Considerations for Service Component Architecture support in servers and clusters

Service integration buses

Service Integration Bus Browser

# Capturing javacore

There are a number of methods that you can use to capture a javacore from an IBM JDK and thread dumps for non-IBM JDKs.

## Capturing javacore

A javacore dump, or a thread dump as it is also called, is one of the primary problem determination documents that an application server creates.

1. Use wsadmin to produce a javacore in the Profile directory:
   a. For Windows:

```
<PROFILE_DIR>\bin\wsadmin.bat [-host host_name] [-port port_number]
[-user userid -password password] -c
"$AdminControl invoke [$AdminControl queryNames WebSphere:name=JVM,process=server1,*]
dumpThreads"
```

b. For Unix (IBM JDKs):

```
<PROFILE_DIR>>/bin/wsadmin.sh[-host host_name]
[-port port_number] [-user userid -password password] -c
"\$AdminControl invoke [\$AdminControl queryNames WebSphere:name=JVM,process=server1,*]
dumpThreads"
```

**Note:** The braces, [] around the `AdminControl queryNames` command are part of the command, and not used to signify optional parameters as is the case for braces around host, port and user. The process name: server1 may need to be change to fit your configuration.

2. A signal can be sent to the server process:
   a. Windows:

      A launch script must be used to start the server process to allow the signal to be passed to the process. This does require special setup prior to starting the server.

      1) <PROFILE_DIR>\bin\startServer.bat server1 –script SERVER1.bat
      2) b. SERVER1.bat

         The server process will start in a command window. You will need to check the logs to verify that the server has successfully started since the intermediate JVM process which usually starts the server process is not used.
      3) <CTRL><BREAK>

         Issue a <CTRL><BREAK> into the command window where the server process is running. A javacore will be produced.

   b. **Unix (all JDKs)**: kill -3 <pid>

      Where <pid> is the process id of the WebSphere Process Server. For IBM JDKs a javacore will be produced in the <PROFILE_DIR>directory.

      For non-IBM JDKs, a thread dump will be written to the native_stdout.log.
3. An alternative method to dumping a windows core file is to use jvmdump.

   This does not require special setup prior to starting the server. However, it does require a special executable from the JVM team. The jvmdump.exe program can be requested by sending a note to jvmcookbook@uk.ibm.com. The advantage of this method is additional information can be obtained about native code being executed within JVM. The format of the dump differs from the IBM javacores.

   - jvmdump.exe <PID>
   - `<WAS_HOME>>\java\jre\bin\jextract.exe <core.name.dmp>`
   - `<WAS_HOME>\java\jre\bin\jdumpview.exe`
     - set dump <core.name.dmp>.zip
     - display thread

       Displays the current executing thread at the time of the dump
     - c. display thread *

       Display all of the threads from the dump.

   For more details about the jdumpview utility please consult the Diagnostics Guide for the IBM Developer Kit and Runtime Environment, Java Technology Edition, Version 5.0.

**Related information**

Configuring the hang detection policy

## Servers and recovery mode processing

When you restart an application server instance with active transactions after a failure, the transaction service uses recovery logs to complete the recovery process.

These recovery logs, which each transactional resource maintains, are used to rerun any Indoubt transactions and return the overall system to a self-consistent state. An *indoubt transaction* is one that has encountered environmental or other errors during commit processing. Logging occurs for normal inflight transactions, but those log entries are removed upon successful commit processing.

This recovery process begins as soon as all of the necessary subsystems within the application server are available during a server startup. If the application server is not restarted in recovery mode, the application server can start accepting new work as soon as the server is ready, which might occur before the recovery work has completed. This might be okay in many cases, but we have provided the more conservative option here. To be clear, recovery will run on a server restart even if the server is started in 'normal' start model.

For information on how to start a server, see the Starting a server topic in the WebSphere Process Server information center.

**Related information**

Starting a server

Profile-specific log files
There are log files detailing the characteristics and runtime activities of individual profiles. These log files are located within the profile directory for each profile.

## Retention queues and hold queues

When a problem occurs while processing a message, it is moved to the retention queue or hold queue.

You can perform administrative actions on the messages in the retention queue and hold queue using either the administrative console or through scripting.

In some cases, viewing and replaying messages on the retention queue or the hold queue can be part of a recovery procedure.

**Related concepts**

"Use case: recovering data from failed events" on page 83
A use case provides a context for a recovery scenario. In the use case, a business has an application that receives a request to create a new Account.

**Related tasks**

Querying and replaying failed messages, using the administrative console
This describes how to check for and replay any messages for business processes or human tasks that could not be processed.

Querying and replaying failed messages, using administrative scripts
Use the administrative scripts to determine whether there are any failed messages for business processes or human tasks, and, if there are, to retry processing them.

**Related information**

Business processes: Recovery from infrastructure failures

Failed event manager console help field descriptions

Managing failed events
The WebSphere Process Server Recovery service captures data about failed events. You can then use the failed event manager to view, modify, resubmit, or delete the failed event.

## Business Process Choreographer maintenance and recovery scripts

There are several maintenance-related scripts for Business Process Choreographer. Run these maintenance scripts as part of a general maintenance policy to help maintain database performance, or as part of a recovery process as deemed necessary.

You should run these scripts to remove from the database the templates and their associated objects, as well as completed process instances, that are not contained in any corresponding valid application in the WebSphere configuration repository.

There is also the possibility of having invalid process templates. This situation can occur if an application installation was canceled or not stored in the configuration repository by the user.

WebSphere Process Server also provides a service that automates Business Process Choreographer cleanup. You can run that service from the administrative console.

Use the following scripts for Business Process Choreographer recovery maintenance:

- deleteInvalidProcessTemplate.py

  Run this script to delete, from the Business Process Choreographer database, business process templates that are no longer valid.

  **Note:** These templates usually have no impact. They are not shown in Business Process Choreographer Explorer.

  You cannot use this script to remove templates of valid applications from the database. This condition is checked and a ConfigurationError exception is thrown if the corresponding application is valid.
- deleteInvalidTaskTemplate.py

Run this script to delete, from the Business Process Choreographer database, human task templates that are no longer valid.

You cannot use this script to remove templates of valid applications from the database. This condition is checked and a ConfigurationError exception is thrown if the corresponding application is valid.

- deleteCompletedProcessInstances.py

  Run this script when all completed process instances have to be deleted.

  A top-level process instance is considered completed when it is in one of the following end states:

  – Finished
  – Terminated
  – End
  – Failed

  You can specify the criteria to selectively delete top-level process instances and all their associated data (such as activity instances, child process instances, and inline task instances) from the database.

**Note:** When running these scripts from the command line, please make sure the SOAP client timeout is set high enough to complete the requested operation for the WAS Admin client.

## Deleting an allotment of completed process instances

You can delete an allotment of process instances from the development environment.

**Using a script that wrappers the provided deleteCompletedProcessInstances.py**

By editing and placing correct user names, passwords, and paths in this wrapper script, you can delete an allotment of process instances from the development environment.

Carefully selecting an adequate time slice prevents SOAP timeout exceptions when communicating with the deployment manager.

The “adequate time slice” of administrable instances depends on many factors including, but not limited to, the following:

- JVM tuning and memory allocations
- Transaction log configuration for the database server
- SOAP connection Time-Out configuration

**Example**

For example, after altering the script and running the command as:

```
wsadmin.<bat|sh> -user<USERNAME> -password<PASSWORD> -f loopDeleteProcessInstances.py
2008-04-02T21:00:00 3600
```

This command will run `deleteCompletedProcessInstances.py` while increasing the completed before time stamp by one hour (60 minutes * 60 seconds) after every execution.

The `deleteCompletedProcessInstances.py` script has a time stamp parameter which can be used to control the number of instances being deleted. The smaller

the interval, the fewer instances will be deleted per invocation of the `deleteCompletedProcessInstances.py`. This can be useful in situations where deletion of multiple process instances encounter transaction timeouts. The most common cause for transaction timeouts during process deletion involve the following:

- an untuned database
- an overloaded system
- attempting to delete “too many” process instances at once

**Related concepts**

Process instances
A process instance is the instantiation of a process template.

**Related tasks**

Using scripts to administer Business Process Choreographer
Describes the administrative actions that can be performed using scripts.

Deleting process templates that are no longer valid
Use the administrative scripts to delete, from the Business Process Choreographer database, business process templates that are no longer valid.

Deleting completed process instances
Use an administrative script to selectively delete from the Business Process Choreographer database or the Business Process Archive database any top-level process instances that have reached an end state of finished, terminated, or failed.

Deleting human task templates that are no longer valid
Use the administrative scripts to delete, from the Business Process Choreographer database, human task templates that are no longer valid.

Configuring the cleanup service and cleanup jobs
Use the administrative console to configure and schedule cleanup jobs that periodically delete instances of business processes and human tasks that are in particular states.

# Resolving indoubt transactions

Transactions can become stuck in the indoubt state indefinitely due to exceptional circumstances, such as the removal of a node causing messaging engines to be destroyed.

## Before you begin

Use the procedure to resolve indoubt transactions only if you have tried other procedures (such as restarting the server in recovery mode), unsuccessfully.

## About this task

When a transaction is stuck in the indoubt state, it must either be committed or rolled back so that normal processing by the affected messaging engine can continue.

You can use the administrative console to display the messages causing the problem by Listing messages on a message point.

If there are messages related to an indoubt transaction, the identity of the transaction displays in a panel associated with the message. You can then resolve the transaction in one of the following ways:

- Using the server's transaction management panels
- Using methods on the messaging engine's MBean

You should first attempt to resolve the indoubt transaction using the application server transaction management panels. If this does not work, then use methods on the messaging engine's MBean. Both procedures are described below.

**Procedure**

## Procedure

1. **Using the application server transaction management panels to resolve indoubt transactions**
   a. Navigate to the transaction management panels in the administrative console

      Click **Servers** → **Application servers** → **[Content Pane]** → *server-name* → **[Container Settings] Container Services** → **Transaction Service** → **Runtime** → **Imported prepared transactions - Review**
   b. If the transaction identity appears in the resulting panel, you can commit or roll back the transaction

      Please choose the option to roll back the transaction

      If the transaction identity does not appear in the panel, the transaction identity was not enlisted with the Transaction Service on the server. In this case only, you should use methods on the MBean (as described in the next step) to display a list of the identities of the indoubt transactions managed directly by the messaging engine.
2. **Using methods on the messaging engine's MBean to resolve indoubt transactions**

   **CAUTION:**
   **Only perform this step if you were unable to display the transaction identity by using the server's transaction management panels**
   a. The following methods on the messaging engine's MBean can be used to get a list of transaction identities (xid) and to commit and roll back transactions:
      - getPreparedTransactions()
      - commitPreparedTransaction(String xid)
      - rollbackPreparedTransaction(String xid)
   b. To invoke the methods, you can use a wsadmin command, for example, you can use a command of the following form to obtain a list of the indoubt transaction identities from a messaging engine's MBean:

      ```
      wsadmin> $AdminControl invoke [$AdminControl queryNames type=SIBMessagingEngine,*] getPreparedTransactions
      ```

      Alternatively, you can use a script such as the following to invoke the methods on the MBean:

      ```
      foreach mbean [$AdminControl queryNames type=SIBMessagingEngine,*] {
        set input 0

        while {$input >=0} {
          set xidList [$AdminControl invoke $mbean getPreparedTransactions]

          set meCfgId [$AdminControl getConfigId $mbean]
          set endIdx [expr {[string first "(" $meCfgId]  - 1}]
      ```

```
    set me [string range ${meCfgId} 0 $endIdx]

    puts "----Prepared Transactions for ME $me ----"
    set index 0
    foreach xid $xidList {
      puts "  Index=$index XID=$xid"
      incr index
    }
    puts "------- End of list ---------"
    puts "Select index of XID to commit/rollback (-1 to continue) :"
    set input [gets stdin]

    if {$input < 0 } {
puts "No index selected, going to continue."
    } else {
      set xid [lindex $xidList $input]
      puts "Enter c to commit or r to rollback XID $xid"
      set input [gets stdin]
      if {$input == "c"} {
        puts "Committing xid=$xid"
        $AdminControl invoke $mbean commitPreparedTransaction $xid
      }
      if {$input == "r"} {
        puts "Rolling back xid=$xid"
        $AdminControl invoke $mbean rollbackPreparedTransaction $xid
      }
    }
    puts ""
  }
}
```

This script lists the transaction identities of the transactions together with an index. You can then select an index and commit or roll back the transaction corresponding to that index.

## Results

In summary, to identify and resolve indoubt transactions:

1. Use the administrative console to find the transaction identity of indoubt transactions.
2. If a transaction identity appears in the transaction management panel, commit or roll back the transactions as required.
3. If a transaction identity does not appear in the transaction management panel, use the methods on the messaging engine's MBean. For example, use a script to display a list of transaction identities for indoubt transactions. For each transaction:
   a. Enter the index of the transaction identity of the transaction.
   b. Enter c to commit the transaction
   c. Enter r to roll back the transaction.
4. To check that transactions are no longer indoubt, restart the server and use the transaction management panel, or methods on the messaging engine's MBean.

**Related tasks**

"Viewing the service integration bus" on page 92
Use the Service Integration Bus browser on the administrative console to view the service integration bus.

# Reviewing DB2 diagnostic information

Use a text editor to view the DB2® diagnostic log file on the machine where you suspect a problem to have occurred. The most recent events recorded are the furthest down the file.

## About this task

Review DB2 diagnostic information when your systems are not working well. This is a way to see if the log files are full.

**Procedure**

## Procedure

On Unix type the following command: tail -f /home/db2inst1/sqllib/db2dump/db2diag.log
If the database is unresponsive, you will see something similar to the following:

```
2008-04-03-11.57.18.988249-300 I1247882009G504    LEVEL: Error
PID     : 16020                TID  : 3086133792  PROC : db2agent (WPRCSDB) 0
INSTANCE: db2inst1             NODE : 000         DB   : WPRCSDB
APPHDL  : 0-658                APPID: 9.5.99.208.24960.080403084643
AUTHID  : DB2INST1
FUNCTION: DB2 UDB, data protection services, sqlpWriteLR, probe:6680
RETCODE : ZRC=0x85100009=-2062548983=SQLP_NOSPACE
          "Log File has reached its saturation point"
          DIA8309C Log file was full.

2008-04-03-11.57.18.994572-300 E1247882514G540    LEVEL: Error
PID     : 16020                TID  : 3086133792  PROC : db2agent (WPRCSDB) 0
INSTANCE: db2inst1             NODE : 000         DB   : WPRCSDB
APPHDL  : 0-658                APPID: 9.5.99.208.24960.080403084643
AUTHID  : DB2INST1
FUNCTION: DB2 UDB, data protection services, sqlpgResSpace, probe:2860
MESSAGE : ADM1823E  The active log is full and is held by application handle
          "274".  Terminate this application by COMMIT, ROLLBACK or FORCE
          APPLICATION.
```

In the preceding example, looking at the DB line, you can see that the WPRCSDB is experiencing full transaction logs.
Another way of viewing the db2diag logs is to log in as the DB2 user and run db2diag:

```
su -l db2inst1
 db2diag | less
```

**Related information**

Interpreting diagnostic log file entries

# Process recovery troubleshooting tips

Using Business Process Choreographer Explorer can facilitate process recovery efforts.

The Business Process Choreographer Explorer provides a user interface for administrators to manage business processes and human tasks.

You can use the Business Process Choreographer Explorer to check the status of the Business Process Choreographer database (BPEDB). If you are unable to retrieve database information through the Business Process Choreographer Explorer, or if the Business Process Choreographer is slow to return database information, it might be an indication of a problem with the database.

Attempting to retrieve thousands of process instances or tasks is not wise if performance or database problems are suspected. Selecting a view which does not retrieve considerable data, such as “My Process Templates”, or limiting the amount of data retrieved for another view would be better options.

**Related concepts**

Business Process Choreographer Explorer overview
Business Process Choreographer Explorer is a Web application that implements a generic Web user interface for interacting with business processes and human tasks.

**Related tasks**

Repairing processes and activities
If the process runs into problems, you can analyze the process and then repair the activities.

Configuring Business Process Choreographer Explorer
You can either run a script or use the administrative console to configure Business Process Choreographer Explorer.

Starting Business Process Choreographer Explorer
Business Process Choreographer Explorer is a Web application that can be installed as part of the configuration of the business process container. Before you can start using Business Process Choreographer Explorer from a Web browser, you must have installed the business process container, human task container, and the Business Process Choreographer Explorer application, and the application must be running. The event collector application must be installed and running in order to use the reporting function.

Tuning Business Process Choreographer Explorer
The following suggestions provide various ways to improve the performance of the Business Process Choreographer Explorer.

## About recovering the messaging subsystem

If the messaging system experiences problems you may need to recover the underlying messaging subsystem.

Typically this involves checking the state of various queues but can also include analyzing the integration bus infrastructure.

Detailed information on recovering the messaging subsystem can be found in the WebSphere Application Server information center.

**Related information**

Troubleshooting service integration message problems

Enterprise service bus messaging infrastructure

## IBM Support Assistant

The IBM Support Assistant is a tool that helps you use various IBM Support resources.

**Note:** The IBM Support Assistant is supported on Microsoft® Windows and Linux systems.

The IBM Support Assistant offers four components to help you with software questions:

- a Search component, which helps you access pertinent Support information in multiple locations.
- a Product Information component, which helps you find the right IBM site for your product questions.
- a Tools component, which provides specialized analysis tools to investigate product problems.
- a Service component, which helps you submit an enhanced problem report that includes key system data to IBM.

Using the IBM Support Assistant with WebSphere Process Server, requires installing IBM Support Assistant and then installing plug-ins for WebSphere Process Server. The plug-ins for WebSphere Process Server include an automated way to gather information about a problem and send it to IBM, and tools that help you set trace levels.

For more information and to install the latest version of IBM Support Assistant, see the IBM Support Assistant Web page.

IBM Support Assistant also is included on the *WebSphere Application Server Network Deployment Supplements V6.1* disk that is included with WebSphere Process Server and can be installed from the WebSphere Process Server launchpad. For more information, see the "Options on the launchpad" topic link in the Related Topics section at the bottom of this page.

After the IBM Support Assistant is installed, you can start it with the **Start** menu option on Windows operating systems or with the `startisa.sh` shell script on all other platforms. On Windows operating systems, the IBM Support Assistant opens in its own window. On all other platforms, it opens in a Web browser.

When you have IBM Support Assistant open, you can view available plug-ins for WebSphere Process Server by clicking **Updater**, clicking **New Plug-ins** and then expanding **WebSphere**. When you select the check box for the WebSphere Process Server plug-in, and click **Install**, the download page opens.

To learn more about how to use the IBM Support Assistant, click **Help** in the IBM Support Assistant window.

**Related tasks**

Chapter 15, “Getting fixes,” on page 111
A product fix might be available to resolve your problem.

Chapter 13, “Searching knowledge bases,” on page 107
You can often find solutions to problems by searching IBM knowledge bases. Optimize your results by using available resources, support tools, and search methods.

Chapter 16, “Contacting IBM Software Support,” on page 113
IBM Software Support provides assistance with product defects.

**Related reference**

 Options on the launchpad

Chapter 16, “Contacting IBM Software Support,” on page 113
IBM Software Support provides assistance with product defects.

 IBM Support Assistant

# Chapter 13. Searching knowledge bases

You can often find solutions to problems by searching IBM knowledge bases. Optimize your results by using available resources, support tools, and search methods.

## About this task

To search for solutions to your problems in IBM knowledge bases, perform the following steps.

**Procedure**

## Procedure

1. Search with IBM Support Assistant. IBM Support Assistant (ISA) is a free software serviceability workbench that helps you resolve questions and problems with IBM software products. The ISA tool can search multiple knowledge bases simultaneously

   To search multiple Internet resources for your product, open the ISA and click **Search**. From this page, you can search a variety of resources including:

   - IBM Software Support Documents
   - IBM developerWorks®
   - IBM newsgroups and forums
   - Google
   - IBM product information centers

   **Note:** These free newsgroups and forums do not offer any formal IBM product support. They are intended for user-to-user communication. IBM will not be actively participating in these discussions. However, IBM does review these newsgroups periodically to maintain a free flow of accurate information. You may also want to browse the following resources individually.
2. Search with the IBM Software Support Toolbar. IBM Software Support Toolbar is a browser plug-in that provides you with a mechanism to easily search IBM support sites.
3. Search the information center.

   IBM provides extensive documentation in the form of online information centers. An information center can be installed on your local machine or on a local intranet. An information center can also be viewed on the IBM Web site. You can use the powerful search function of the information center to query conceptual and reference information and detailed instructions for completing tasks.
4. Search available technical resources. In addition to this information center, the following technical resources are available to help you answer questions and resolve problems:

   - WebSphere Process Server technotes
   - WebSphere Process Server Authorized Program Analysis Reports (APARs)
   - WebSphere Process Server support Web site
   - WebSphere Redbooks Domain
   - IBM Education Assistant

- WebSphere Process Server forums and newsgroups

## What to do next

**Tip:**

The following resources describe how to optimize your search results:

- Searching the IBM Support Web site
- Using the Google search engine
- IBM Software Support RSS feeds
- My Support e-mail updates

**Related reference**

IBM WebSphere Process Server technotes

IBM WebSphere Process Server Authorized Program Analysis Reports (APARs)

IBM WebSphere Process Server support Web site

IBM WebSphere Redbooks Domain

IBM Education Assistant

WebSphere Process Server forums and newsgroups

"IBM Support Assistant" on page 105
The IBM Support Assistant is a tool that helps you use various IBM Support resources.

IBM Software Support Toolbar

# Chapter 14. IBM Support Assistant

The IBM Support Assistant is a tool that helps you use various IBM Support resources.

**Note:** The IBM Support Assistant is supported on Microsoft Windows and Linux systems.

The IBM Support Assistant offers four components to help you with software questions:

- a Search component, which helps you access pertinent Support information in multiple locations.
- a Product Information component, which helps you find the right IBM site for your product questions.
- a Tools component, which provides specialized analysis tools to investigate product problems.
- a Service component, which helps you submit an enhanced problem report that includes key system data to IBM.

Using the IBM Support Assistant with WebSphere Process Server, requires installing IBM Support Assistant and then installing plug-ins for WebSphere Process Server. The plug-ins for WebSphere Process Server include an automated way to gather information about a problem and send it to IBM, and tools that help you set trace levels.

For more information and to install the latest version of IBM Support Assistant, see the IBM Support Assistant Web page.

IBM Support Assistant also is included on the *WebSphere Application Server Network Deployment Supplements V6.1* disk that is included with WebSphere Process Server and can be installed from the WebSphere Process Server launchpad. For more information, see the "Options on the launchpad" topic link in the Related Topics section at the bottom of this page.

After the IBM Support Assistant is installed, you can start it with the **Start** menu option on Windows operating systems or with the `startisa.sh` shell script on all other platforms. On Windows operating systems, the IBM Support Assistant opens in its own window. On all other platforms, it opens in a Web browser.

When you have IBM Support Assistant open, you can view available plug-ins for WebSphere Process Server by clicking **Updater**, clicking **New Plug-ins** and then expanding **WebSphere**. When you select the check box for the WebSphere Process Server plug-in, and click **Install**, the download page opens.

To learn more about how to use the IBM Support Assistant, click **Help** in the IBM Support Assistant window.

**Related tasks**

Chapter 15, “Getting fixes,” on page 111
A product fix might be available to resolve your problem.

Chapter 13, “Searching knowledge bases,” on page 107
You can often find solutions to problems by searching IBM knowledge bases. Optimize your results by using available resources, support tools, and search methods.

Chapter 16, “Contacting IBM Software Support,” on page 113
IBM Software Support provides assistance with product defects.

**Related reference**


Options on the launchpad

Chapter 16, “Contacting IBM Software Support,” on page 113
IBM Software Support provides assistance with product defects.


IBM Support Assistant

# Chapter 15. Getting fixes

A product fix might be available to resolve your problem.

## About this task

To get product fixes, perform the following steps.

**Procedure**

## Procedure

1. Obtain the tools required to get the fix. See "Installing the Update Installer for WebSphere Software".
2. Determine which fix you need. Check the list of WebSphere Process Server recommended fixes to confirm that your software is at the latest maintenance level. Check the list of problems fixed in the IBM WebSphere Process Server fix readme documentation that is available for each listed fix pack and refresh pack to see if IBM has already published an individual fix to resolve your problem. To determine what fixes are available using IBM Support Assistant, run a query on `fix` from the search page.

   Individual fixes are published as often as necessary to resolve defects in WebSphere Process Server. In addition, two kinds of cumulative collections of fixes, called fix packs and refresh packs, are published periodically for WebSphere Process Server, in order to bring users up to the latest maintenance level. You should install these update packages as early as possible in order to prevent problems.

   **Note:** Fixes specific to the underlying WebSphere Application Server product may also be obtained from the WebSphere Application Server Support Site or from the WebSphere Application Server Support team. Fixes for individual APARs for WebSphere Application Server generally can be applied without affecting WebSphere Process Server. However, consult with the software requirements page before updating WebSphere Application Server with cumulative collections of fixes (fix packs). First check to see that the cumulative fix has passed certification, or contact the Support team for verification.
3. Download the fix. Open the download document and follow the link in the **Download package** section. When downloading the file, ensure the name of the maintenance file is not changed. This includes both intentional changes and inadvertent changes caused by certain web browsers or download utilities.
4. Apply the fix. Follow the instructions in the **Installation Instructions** section of the download document. For more information, see the "Installing fix packs and refresh packs with the Update Installer" topic in the Installing WebSphere Process Server documentation.
5. Optional: To receive weekly notification of fixes and updates, subscribe to My Support e-mail updates.

**Related reference**

"IBM Support Assistant" on page 105
The IBM Support Assistant is a tool that helps you use various IBM Support resources.

Installing fix packs and refresh packs with the Update Installer

Installing the Update Installer for WebSphere Software

Subscribe to My Support e-mail updates

Recommended Fixes for WebSphere Process Server

# Chapter 16. Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

## Before you begin

To take advantage of unique Support features, see the WebSphere Process Server support page. The Support Page contains the latest information on fixes and downloads, educational resources, and commonly encountered problems and their solutions.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have. For information about the types of maintenance contracts available, see "Enhanced Support" in the *Software Support Handbook* site listed in the Related Topics section.

To contact IBM Software Support with a problem, perform the following steps.

**Procedure**

## Procedure

1. Define the problem, gather background information, and determine the severity of the problem. For help, see the "Contacting IBM" in the *Software Support Handbook*.
2. Gather diagnostic information. When explaining a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. For information that IBM Support needs in order to help you solve a problem, see the WebSphere Process Server MustGather technote.

   **Tip:** You can use the WebSphere Process Server plug-in for the IBM Support Assistant to capture data and send it to IBM.

   **Note:** If you are able to determine that the problem is purely with underlying WebSphere Application Server functionality, consider requesting assistance specifically from the WebSphere Application Server Support team rather than the WebSphere Process Server team. For information that IBM Support needs in order to help you solve a WebSphere Application Server problem, see the WebSphere Application Server MustGather Technote.
3. Submit your problem to IBM Software Support in one of the following ways:
   - Using IBM Support Assistant: See the "IBM Support Assistant" topic.
   - Online: Open a service request on the IBM Software Support site using the Electronic Service Request (ESR) tool.
   - By telephone: For the telephone number to call in your country or region, go to the contacts page of the IBM Software Support Handbook on the Web and click the name of your geographic region.

## What to do next

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved. Support will work and communicate with you on the progress and deliver the fix once it is completed. Additionally, once completed, IBM will also publish the resolved APARs on the Software Support Web site, so that other users who experience the same problem can benefit from the same resolution.

**Related tasks**

"IBM Support Assistant" on page 105
The IBM Support Assistant is a tool that helps you use various IBM Support resources.

**Related reference**

 WebSphere Process Server Support

 Software Support Handbook

 MustGather: Read first for WebSphere Process Server for Version 6

 MustGather: Read first for all WebSphere Application Server products

"IBM Support Assistant" on page 105
The IBM Support Assistant is a tool that helps you use various IBM Support resources.

 IBM Software Support site

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing*
*IBM Corporation*
*North Castle Drive*
*Armonk, NY 10504-1785*
*U.S.A.*

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

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

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

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

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

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

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

IBM Corporation
1001 Hillsdale Blvd., Suite 400
Foster City, CA 94404
U.S.A.

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

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

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

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

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

**Programming interface information**

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

**Warning:** Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

**Trademarks and service marks**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol ($^{R}$ or $^{TM}$), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

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

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

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

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

This product includes software developed by the Eclipse Project (http://www.eclipse.org).


IBM WebSphere Process Server for Multiplatforms, Version 6.2

Printed in USA