ActiveBPEL® Development Server User's Guide

Version 4.1

Revised September 2007

ActiveBPEL Development Server User Guide

Copyright © 2004-2007 Active Endpoints, Inc.

Printed in the United States of America

ActiveBPEL is a registered trademark of Active Endpoints, Inc.

Contents

ActiveBPEL®
Development Server
User’s Guide 1

1 ActiveBPEL® Engine Administration Console Help

The ActiveBPEL Engine Administration Console allows you to manage and configure the ActiveBPEL engine and the artifacts that are deployed into it.

The Administration Console provides several ways to select and inspect process instances and related deployment logs, descriptors and resources.

You can do the following from the Administration Console:

- Start and stop the engine. See Administration Home Page.
- Update engine configuration. See Engine.
- Tune engine performance by setting a WSDL cache size and other values. See *Configuration* in the Engine topic.
- Make endpoint references more flexible and dynamic by providing a map between an URN and an URL. See *Configuration* in the Engine topic.
- View each deployed process in graphic, tree, or text mode. View details of each deployed process and associated WSDL files. See Deployment Status.
- Suspend or terminate a running process. See Process Status.
- Analyze execution steps and diagnose problems by viewing active processes. See Process ID and Process Details.
- View and download variable attachments. See Working with Variable Attachments.
- Enable logging and then download a copy of the execution log for a running or completed process. See Process Status.

Note: If you are running the open source version of the ActiveBPEL engine, you can also configure the engine for persistent storage. You can then purge completed processes on the Storage page. For details, see Persistence Storage Setup.

Administration Home Page

The Home page displays the following details.

Item	Description
Date Started	Engine start date
Deployed Processes	Number of business process archive files (.bpr files) currently stored in *ActiveBPEL_Tomcat\bpr*. This folder is scanned for new and modified processes every 20 seconds by default. If errors or warnings are generated, ActiveBPEL creates a deployment log. Select **Deployment Log** from the ActiveBPEL navigation menu to view details.
Description	Engine configuration. The configuration for the ActiveBPEL engine that is installed with ActiveBPEL Designer is In-Memory. In this mode, all processes stop if the ActiveBPEL engine stops. The other configuration available for the ActiveBPEL engine is Persistence, which is supported only through ActiveBPEL.org forums. In Persistence mode, process states are saved if the ActiveBPEL engine stops. For more information see Persistence Storage Setup.
Status	Statuses for the ActiveBPEL engine are Running and Stopped. Additional messages are included for an engine connecting to a database for persistent storage. Select Storage to see more detailed information regarding the database.
Version	Engine version number

Stopping and Starting the Engine

When the ActiveBPEL engine is in In-Memory mode, the engine starts when you start Tomcat and stops when you shut down Tomcat. Select **Stop Engine** when you need to stop all running processes. Running processes are saved in Persistence mode, but not in In-Memory mode.

Select **Start Engine** to change the engine status to Running.

Back to ActiveBPEL® Engine Administration Console Help

Engine

Engine information includes Configuration, Storage, and Version Detail.

Configuration

The Configuration page is divided into tabs for Engine Properties and URN Mappings

Engine Properties

You can make engine property changes without stopping and restarting the engine. When you make a change and select **Update** the changes are in effect immediately as well as persisted to the engine configuration file.

View and update engine configuration settings as shown in the following table. Note that some of these properties are the same as the ActiveBPEL Designer Simulation preferences.

Property Name	Description
Auto create target path for Copy/To (BPEL 1.1 only)	Applies only to processes that are validated against the BPEL4WS 1.1 specification. For WS-BPEL 2.0 processes, this property can be added as an extension on a per process basis. Refer to Extensions help topic in *ActiveBPEL Designer Online Help*. Determines if the ActiveBPEL engine is allowed to create a location path for a non-existent node in a complex variable in a process instance document. When an assignment refers to a non-existent node (or to more than one node), the standard BPEL fault, bpws:selectionFailure, must be thrown, according to the BPEL specification. Enabling this option allows selections to be created on-the-fly. This means an assign copy TO operation can refer to a non-existent node and assign a value to it. This option is disabled by default.

Property Name	Description
Disable bpws:selectionFailure fault (BPEL 1.1 only)	Applies only to processes that are validated against the BPEL4WS 1.1 specification. For WS-BPEL 2.0 processes, this property can be added as an extension on a per process basis. Refer to Extensions help topic in *ActiveBPEL Designer Online Help*. Enabling this option allows a null value to be returned from a function or assignment that contains an XPath query string. You can enable this to override XPath behavior, for cases that handle data samples with optional elements. By default, this option is not enabled, and if the query string returns an empty selection from an assign copy FROM, the process throws a bpws:selectionFailure fault, which is the standard response described in the BPEL4WS specification.
Logging Level	By default the ActiveBPEL engine does not generate an execution log for running processes. Logging is turned off to enhance engine performance. You can enable this setting, and then view or download an execution log for a running or completed process. An execution log provides start/end times for activity execution and helps you troubleshoot faulted processes. The logging levels are: • **None**. (default) • **Full**. All execution statements are logged, including the *Will Not Execute* statements for deadpath activities. For example, all fault handling statements that are not executed are logged. • **Execution**. All execution statements are logged, except for *Will Not Execute* statements. Using this setting can greatly decrease the size of the log file.

Property Name	Description
Replace existing resources on deployment	Overwrites the current WSDL or schema definition, or other resource. By default, resources are not replaced. However, if you enable this setting, ActiveBPEL allows you to replace a WSDL definition or schema file currently in cache without restarting the server. You can deploy a new version of a BPR file containing updated resources. BPEL developers who are testing and modifying processes may find this option useful. Note that you can add this setting to an individual BPR file in the ActiveBPEL Designer Export Wizard. The Deployment Log shows whether or not the resources are being replaced.
Validate input/output messages against schema	Validates the data loaded into process variables against the WSDL schema. Enable this option to validate data before execution starts. Disable this option for faster execution. This option is enabled by default.
Resource Cache	The number of WSDL and other resource files in stored cache. The default is 100. Modifying the cache size may improve engine performance. A value of -1 means unlimited caching, but is not recommended.
Unmatched Correlated Receive Timeout	Set the amount of time to wait (in seconds) for a correlated message to be matched to a receive, in the case that the message arrives before the receive becomes active. If this value is exceeded, a message is discarded so that the process can complete normally. The default is 300. Specifying 0 indicates that unmatched correlated messages are immediately discarded.
Web Service Timeout	For performance reasons, a reply activity matching a receive, as well as synchronous invokes, are timed out if they do not execute within 10 minutes. If you are receiving timeout errors, you can specify a greater amount of time to wait before a process is timed out due to a reply or synchronous invoke activity not executing within 10 minutes. The default is 600 seconds.

Property Name	Description
Work Manager Thread Pool Min	Set the minimum number of execution threads the engine allocates for its Work Manager. The default is 10.
Work Manager Thread Pool Max	Set the maximum number of execution threads the engine can spawn simultaneously. The default is 50. A value of -1 means that there is no maximum number of threads.
Work Manager Threads Per Alarms Max	Set the maximum number of threads the engine will use from the work manager to dispatch work scheduled by an alarm in a process. If there are 100's of alarms firing concurrently, it is possible that all of the threads in the work manager could be used just to dispatch the alarms. If you experience performance issues or deadlocks due to all of the threads being used by the alarm manager, you can increase this value. The default is 5.
Work Manager Threads Per Process Max	Set the maximum number of execution threads the engine can spawn simultaneously for an individual process. The default is 10.

URN Mappings

URN mappings provide a flexible and dynamic way to define target endpoint references. Use URN mappings to specify the physical address of a partner link endpoint reference in the Administration Console instead of specifying an address in a process deployment descriptor (.pdd) file or WSDL file. By mapping an URN to an URL, you do not have to rely on invoking a statically defined endpoint address. URN mappings give you flexibility, for example, to deploy the same BPR files for testing and production environments.

Instead of using the default invocation, you can specify a logical or physical address for a static endpoint reference in the .pdd. If you specify a logical address, or URN, you can then map the URN to the physical address in the URN Mappings page of the Administration Console. If you specify an URL, you can replace the URL by mapping it to a different URL.

The following example illustrates one type of URN to URL mapping:

```
urn:localhost = http://localhost:8080/active-bpel/services/
${urn.3}
```

This mapping might be used when a process is deployed with the following partner link address information:

```
<partnerLink name="assessor">
   <partnerRole endpointReference="static"
     invokeHandler="default:Address">
      <wsa:EndpointReference xmlns:assessor="http://
       tempuri.org/services/loanassessor">
      <wsa:Address>urn:localhost:AssessRisk</wsa:Address>
      <wsa:ServiceName PortName=
        "SOAPPort">assessor:LoanAssessor</wsa:ServiceName>
   </partnerRole>
</partnerLink>
```

The ActiveBPEL invocation framework resolves the URN as follows:

urn:localhost:AssessRisk = http://localhost:8080/active-bpel/services/AssessRisk

Here are some ways you can map URNs to URLs. Note that each segment of the URN is separated by a colon. This means you can use a variable, such as ${urn.4} shown in the third example below, to indicate a replacable token in the fourth segment.

URN	URL
urnSegment1:urnSegment2	http://localhost:8080/active-bpel/services/MyService
http://ServerA:8080/active-bpel/services/MyService	http://ServerB:8081/active-bpel/services/MyService
urn:localhost:service	http://localhost:${AE-NODE1-PORT}/activebpel/services/${urn.4}

The last example in the table above shows how you can use variable substitution in an URL.

The URL values can optionally contain variables. The variables can be environment variables accessible through java.lang.System.getProperties() or a segment from the URN itself. The Apache Ant style variable declaration of `${property}` is used to identify a property within the URL. Segments from the input URN value can be referenced by using a special property naming convention of `${urn.offset}` where *offset* is a one-based offset identifying the segment from the input URN value to use for substitution.

The URL in the mapping above contains two variables. The `${AE-NODE1-PORT}` variable pulls the port number from an environment variable. This variable would need to be set as a -D parameter on the Java runtime environment (e.g., `java -D`

`${AE-NODE1-PORT}=8080 ...`) or populated externally to the ActiveBPEL server.

The `${urn.4}` variable in the above mapping references the fourth segment from the input URN value. Notice that the URN contains only three segments. The URN in the .pdd file should contain at least one other segment. A sample URN might be:

`urn:localhost:service:StoreService`.

The value of the fourth segment of this URN is `StoreService`. The resulting URL is:

`http://localhost:8080/activebpel/services/StoreService/`.

To update an URN mapping, select the URN. The URN and URL values appear in the text boxes where you can edit them and select **Update**. Editing the URN results in a new URN mapping, it does not update the existing one. Only the URL can be updated.

To delete a mapping, select the check box next to the mapping and select **Delete**.

Storage

By default the ActiveBPEL engine is not configured for persistent storage. If you set up database storage, you can view storage configuration settings as well as delete old processes that are stored. The engine supports both relational databases and XML databases.

See Persistence Storage Setup.

For relational databases, the following storage configuration settings are displayed.

JNDI	Java Naming and Directory Interface (JNDI) context that specifies where to look for the database. For example, jdbc/ActiveBPEL
Database Type	The type such as `mysql`
User Name	Username and password, if they are required for database access

For XML databases, the following configuration settings are displayed.

Database Type	The type such as `tamino`

Tamino URL	The path to the tamino database `http://host/tamino` where `host` is the name of the computer on which the Tamino Server resides. You can specify `localhost` for your local computer, or a path such as *yourpc.ourcompany.com* for a remote computer.
Database Name	Name provided in the ActiveBPEL persistence setup file
Collection	For tamino databases, this is the name that matches the collection name specified in the database schema file (.tsd file). If the collection name was modified in the ActiveBPEL persistence setup file, it must also be modified in the schema file that ActiveBPEL provides for database set up.
Connection Pool Size	Number of simultaneous connections allowed to the database. The default size is 30.
Domain	Network domain for host computer
Username	Username if required for ActiveBPEL's access to the database
Password	Password associated with the Username

You can delete completed processes stored in the database. Select a **Completed By** date and select **Delete**.

Version Detail

The Version Detail page shows the build number and date of the ActiveBPEL engine libraries. This information may be useful for troubleshooting purposes.

Back to ActiveBPEL® Engine Administration Console Help

Deployment Status

Deployment Status includes Deployment Log, Deployed Processes, Deployed Services, Partner Definitions, and Resource Catalog.

Deployment Log

The deployment log shows messages that are generated when new and modified business process archive (.bpr) files are added to the *ActiveBPEL_Tomcat\bpr* folder. Messages include successful deployments, loading or replacing of WSDL files, errors and warnings.

Each time you start the ActiveBPEL Administration Console, a new deployment log is created. Deployment logs are not saved.

Deployed Processes

The Deployed Processes page lists all business process files that have been deployed into the *ActiveBPEL_Tomcat\bpr* folder.

Select a process to view details. To view running, faulted, or complete processes, see Process ID and Process Details.

Deployed Process Detail

The Deployed Process Detail page shows the deployment descriptor (.pdd) and process (.bpel) XML source code that have been deployed to the ActiveBPEL engine in a business process archive (.bpr) file.

Select View Process Graph to see the process in Outline view and Graph view. For details on using these views, see Process ID and Process Details.

Partner Definitions

A partner definition file contains the service information for a partner link that has been deployed designated as a *principal* endpoint reference in the process deployment descriptor (.pdd) file. For details, see the `docs` folder of the ActiveBPEL engine installation or see the *ActiveBPEL Designer Online Help*.

If any principals have been defined, you can select one to view details.

Deployed Services

A deployed process contains at least one My Role partner link, and this partner link is assigned a service name in the Process Deployment Descriptor (PDD) file. The service name identifies the WSDL that the ActiveBPEL engine generates during deployment, and adds to the Services page. The WSDL includes the messages, operations, service, and binding details for the Web Service exposed by the process' My Role partner link.

The process receives messages at the Web Service address, which is shown in the following example:

```
http://localhost:8080/active-bpel/services/[servicename]?wsdl
```

Note: Some services are deployed as *external*, indicating they are not exposed as Web Services. An example is a Retry Policy service, which is a process deployed to tell the engine how many times to retry a non-communicating service. This type

of process is not intended for outside consumption. External services are listed, but are not linked, on the Deployed Services page.

The following details and links are included on the Deployed Services page.

Name	Service name assigned to a My Role partner link in the PDD file. Select the name to link to the WSDL generated for this partner link. The WSDL is the Web Service that receives inbound messages. The Web Service address is in the form of: *http://localhost:8080/active-bpel/services/[servicename]?wsdl*
Process Name	Process containing the My Role partner link associated with this service
Binding	Standard SOAP binding styles indicating how to format inbound messages for the service. Can be one of: • MSG (Document Literal) • RPC Literal • RPC Encoded • External • Policy Driven
Partner Link	Name assigned to a My Role partner link that is exposing the service. Select the name to link to the Deployed Process Version Detail page.

Partner Definition Details

The following details are displayed for the selected principal.

Partner link type	The partner link type used in the partner definition for the principal
Role	Role defined for the partner link type

Select a partner link type to view the namespace and endpoint reference details for the partner definition.

Resource Catalog

The Resource Catalog is the centralized cross reference for all WSDL, schema and other resource files referenced in the .pdd files deployed to the ActiveBPEL engine.

Any resource in the catalog can be accessed by any deployed BPEL process and only one copy is maintained. There are no restrictions based on the deployment context.

Item	Description
Total Reads	The number of reads to retrieve resource file information during process execution (in cache or not)

Item	Description
Disk Reads (%)	The number of reads made to resource files not in the cache expressed as an absolute number and percentage of Total Reads
Cache Size	The number of resource files in stored cache. The default is 100. You can set cache size on the Configuration page. Modifying the cache size may improve engine performance See the Engine topic.

The Deployed Resources list shows the type and namespace for the resources.

You can do the following from this page:

- Select a resource to view details.
- Select one or more options from the **Selection Filter** option list to view a selection of resources. Narrow the list of displayed resources by selecting a resource type (such as WSDL, XSD, or XSL), a resource name, and/or a target namespace.

Resource Detail Page

The Resource Detail page shows the same information that is on the Resource Catalog for each resource and also displays the XML source code.

Type	The resource type, such as WSDL or XSD
Location	The actual physical location where the resource is loaded from. This helps to uniquely define the location when the BPR was created and can be used to have multiple resource files of the same name deployed to the engine. The resource location is referenced in the .pdd file
Namespace	Target namespace in the resource

Back to ActiveBPEL® Engine Administration Console Help

Process Status

Learn more about Process Status by selecting Active Processes, Alarm Queue, and Receive Queue.

Active Processes

The Active Processes page shows a list of process instances that have been or are executing in the ActiveBPEL engine. States can be *running*, *completed*, *compen-*

satable, or *faulted*. Select a Process Name to view details and an execution log of the process instance. Select a number of **records per page** to set the list display.

Use the **Selection Filter** settings to view All, Running, Completed, Compensatable (for a subprocess), or Faulted process instances. Narrow your selections by entering process instance creation/completion dates and/or process names.

When you select a Process Name, the **Process Detail** page appears. See Process ID and Process Details.

Alarm Queue

View a list of active On Alarm process activities.

Select one or more options from the **Selection Filter** option list to narrow your view of active alarms.

Deadline Between	Beginning and Ending date and time for alarm
Process Id	Process instance Id. You can find the Id of any process instance on the Active Processes page.
Process Name	Local part of the process qualified name (qname)

Receive Queue

View a list of active receive and onMessage activities. These activities are queued for incoming messages.

You can do the following from this page:

- Select a receive and then select a partner link to view details. A window opens where you can see the BPEL process location in which the receive activity executes. You can also see the correlation property alias and data, if any, associated with this receive activity.
- Select one or more options from the **Selection Filter** option list to view a selection of active receives. You can find this information on the Deployed Process Detail page, which shows the BPEL source code.

You do not need to enter the fully qualified name for the operation.

Back to ActiveBPEL® Engine Administration Console Help

Process ID and Process Details

You can enter a process instance ID in the **Process ID** text box, select **Go**, and then view the Process Details window, a comprehensive snapshot of a running, faulted,

compensatable, or completed process instance. The list of process IDs to select from is on the Active Processes page.

The Process Details page helps you analyze the execution state of the process instance.

You can do the following from Process Details page:

- View process and activity-level properties and values
- View the execution state of each activity
- Inspect the current value of variables, activity links, partner links, correlation sets, fault, compensation, and event handlers
- **Refresh** a running process to view an updated snapshot of the execution state
- **Suspend**, **Resume**, or **Terminate** a running process. These buttons appear at the top of the outline view.
- View and download the execution log to your computer from the process log, if you enabled logging. Select the **View Process Log** button at the top of the outline view.

Note: If the Process Log box displays "Log file not available," it means logging was not enabled when this process instance ran. For future process instances, you can enable logging from the Configuration page of the Administration Console.

See also:

- Using the Process Details Page
- Using the Process Details Graphic View
- Using the Process Details Outline View
- Inspecting Where and Why an Activity Faulted
- Working with Variable Attachments

Back to ActiveBPEL® Engine Administration Console Help

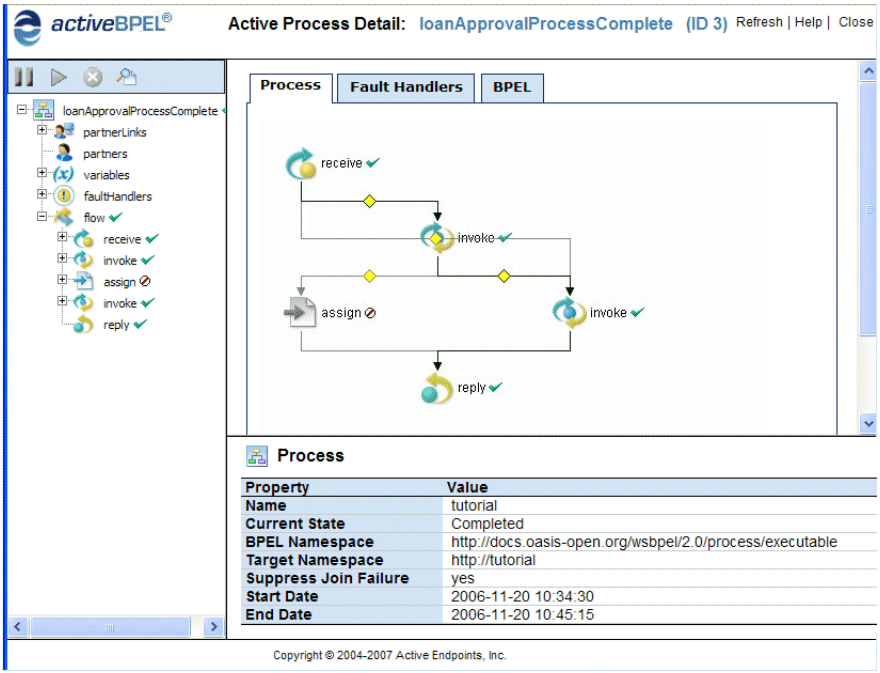
Using the Process Details Page

The Process Details page presents many details about a process instance:

- An **Outline** view shows the structural elements of a BPEL process and the current process execution state of each activity. You can select an element to view its properties and values.

- A **Graphic** view shows the main process flow. If the process has event handlers, fault handlers, and compensation handlers, you can view them by selecting a tab, such as Fault Handlers shown below. You can also select an activity to view its properties.

The following illustration shows an example of the Process Details page.



For details, see:

- Using the Process Details Graphic View
- Using the Process Details Outline View

Back to ActiveBPEL® Engine Administration Console Help

Using the Process Details Graphic View

The Graphic view of Process Details shows the main process flow and the execution path through the process. You can also view the process fault, event, and compensation handlers, if the process definition includes these process-level handlers. The handlers have their own tabs in the view.

In the upper-right panel of the page, you see the main flow of a BPEL process. The process diagram reflects the layout rendering that is part of Active Endpoints BPEL design tool, *ActiveBPEL Designer*.

Each process activity has an icon, a label, and an execution state indicator, as shown in the following illustration.

1 Activity icon. Activity icons are the same as those supplied with ActiveBPEL Designer.

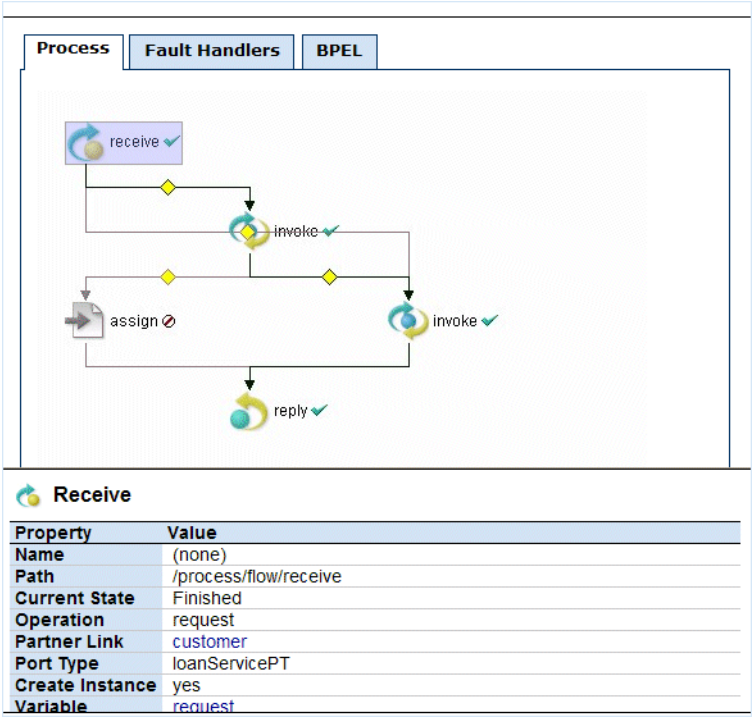
2 Activity label, which can be the activity type, name, type:name, or custom text

3 Execution state indicator. For a description of each indicator, see Using the Process Details Outline View.

Activities may appear in different colors, to indicate different execution states, as the following table describes.

Activity Color	**Execution State**
full color	Executed
muted color	Ready to execute or inactive
gray	Dead path

The Graphic view looks similar to the following example.



To view details, do the following:

- Select an activity from the diagram to view its properties
- Select an activity from the diagram to put the activity in focus in the Outline

To print the diagram, select **Print Picture** from the right-mouse menu. The diagram prints with the same caption that appears in the graph view. The timestamp indicates when the Process Details page was opened or refreshed.

Printing Tips

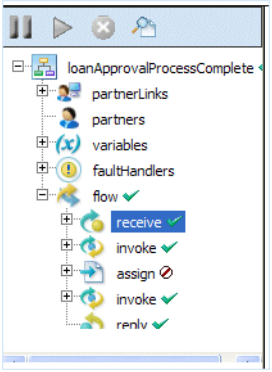
- To print a large diagram, select appropriate scaling options in your Printer options dialog, such as Fit to Page or print as x% of Normal Size
- Save the diagram as an image file to print later

See also Using the Process Details Outline View.

Back to ActiveBPEL® Engine Administration Console Help

Using the Process Details Outline View

The following illustration shows a sample Outline view of a process instance.



Outline View Menu Bar

1 **Suspend** running process

2 **Resume** suspended process

3 **Terminate** running or suspended process

4 **View Process Log**. Display the execution details for the process instance. The logging level must be set to Full or Execution, as described on the *Configuration* page of the Engine topic.

Structural elements of a BPEL process

- **Process name** is the local part of the process qualified name
- **Partner links** represent the Web services that are invoked
- **Variables** contain the message or other data received, manipulated, and sent from the process
- **Correlation sets**, if present, contain the message properties that track different conversations carried on by the process
- **Fault handlers**, if present, catch faults thrown by process activities

- **Event handlers**, if present, run concurrently with a process scope and invoke an activity based on an alarm or event received
- **Activities** carry out the processing steps

To view details about a process element, select it. The following table describes each element.

Process Element in Outline	Details Displayed
Process name	Current state, Start/end time of process instance, and deployment details for the process. For details, see *Configuration* in the Engine topic. Fault details may also be displayed.
Partner links	The type(s) of partner links: partner role and/or my role. The endpoint reference of the partner link service. This reference is defined in one of the following ways: • Static assignment. The <service> section of the WSDL file is defined in the process deployment descriptor. • A My Role partner link may have an externally defined service. For more information on external services, see *ActiveBPEL Designer User's Guide*. • Dynamic assignment. The partner role address may be assigned within a Copy operation of an Assign activity within the process.
Variables	The variable type: message, simple, or schema. The current value of the variable. For a running process, the value is current as of the time you opened or refreshed the Process Details window.
Correlation sets	The message property definition and current value. A correlation set contains a message property to ensure that each process conversation is uniquely identified.
Fault handlers	Name, state, and details of fault handling activity at the process level. Scopes can have their own local fault handlers.
Event handlers	Name, state, and details of event handling activity at the process level. Scopes can have their own local event handlers.

Process Element in Outline	Details Displayed
Activities	The activities section of the Outline begins with a flow activity that represents the main container for the whole process. Within the flow, there is a list of all process activities. The activities are in the same order as in the BPEL XML code. If the process was designed in ActiveBPEL Designer, the order matches the Outline view order. **Note:** The activity list shown is not necessarily in execution order. For each activity, you can view the current execution state and activity definition.
Links	If an activity is the source of a link, the link is displayed below the activity node. Link properties are displayed, including link status (whether or not the link executed), the transition condition, if it exists, and the link's target activity.

Activity States

You can determine the execution status of each activity by looking at the indicator on the right side of the activity.


Executing


Ready to Execute


Finished


Faulted. Occurs when a fault is thrown during the execution of an activity.


Terminated. Occurs when the process is manually terminated.


Dead Path


Suspended

(none) Inactive (the initial state of an activity) or Terminated.

For a running process, the icon next to an activity may change if you refresh the Process Details window.

Process States

The process can have the following execution states:

Completed	Normal completion
Faulted	Completed with a fault or termination
Running	Snapshot of the executing process when you open the Process Details window. The process continues to run, but the Process Details window is not updated unless you select **Refresh**.
Compensatable	A completed subprocess. A subprocess is invoked by another process and is eligible for compensation within the enclosing scope of the calling process.
Suspended	The process stops running when you select Suspend from the Process Details window. It can also be suspended by a BPEL Suspend activity.

See also:

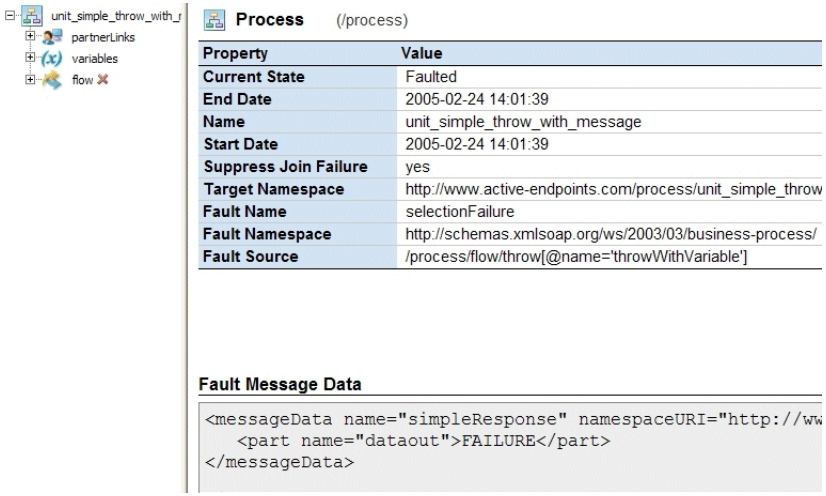
- Using the Process Details Graphic View
- Inspecting Where and Why an Activity Faulted
- Working with Variable Attachments

Back to ActiveBPEL® Engine Administration Console Help

Inspecting Where and Why an Activity Faulted

In the Process Details window, the Outline view shows a list of process activities. A red X appears next to an activity that faulted.

You can select the process name to view details about the fault, as the following illustration shows.



Fault information includes:

Fault Name	Standard BPEL or engine fault name
Fault Namespace	Standard BPEL or engine fault namespace
Fault Source	Process activity that threw the fault
Fault Message Data	Data in the throw or catch fault variable

You can get further information about faults:

- Select the faulted activity to view the Fault Name. For details about BPEL faults, see *BPEL Standard Faults* in the ActiveBPEL Designer Online Help or in the WS-BPEL 2.0 specification. For engine faults see ActiveBPEL Custom Faults.
- Select **View Process Log** in the Outline view toolbar. The process log shows the execution path leading to the faulted activity.

Note: If the Process Log is not visible, you must enable logging on the Configuration page.

See also Using the Process Details Graphic View and Working with Variable Attachments.

Back to ActiveBPEL® Engine Administration Console Help

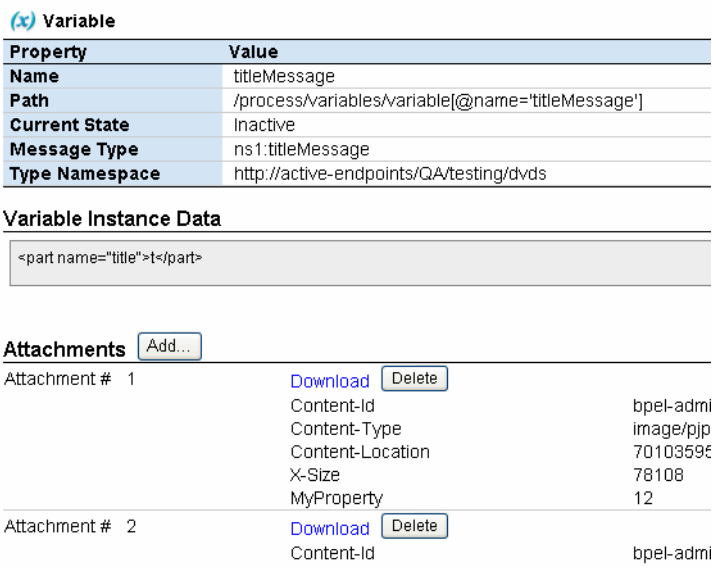
Working with Variable Attachments

You can view and download attachments for a variable, whether a process is running, faulted, or completed.

If a process is running, you can also add and remove variable attachments.

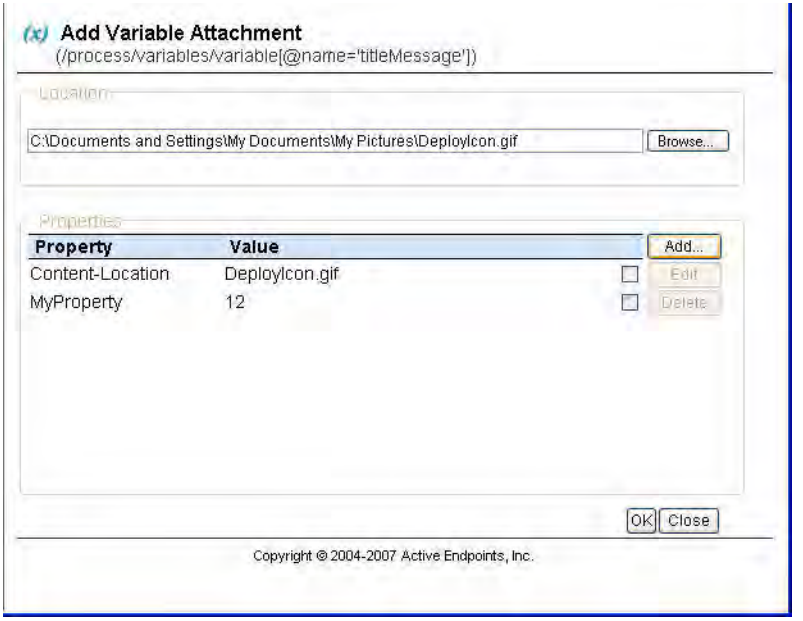
You can do the following to view, download, add, and remove process variable attachments:

1. Select a process from the Active Processes list.
2. In the Active Process Detail window's Outline View, expand the list of variables.
3. Select a variable from the list to view the variable instance data and attachments, as shown in the following illustration for a running process.



Adding an Attachment

While a process is running, you can add one or more attachments to a variable. In the Active Process Detail page, expand a variable to display the Attachments header. Select **Add** to open the Add Variable Attachment window, as shown in the example.



You can browse to locate a file to attach. You can also add or edit attachment properties and values.

Deleting an Attachment

To delete an attachment for a running process, select **Delete**, next to the Download link.

Viewing or Downloading an Attachment

For a running, faulted, or completed process, select the Download link to open a dialog that asks where to open or save the file.

Back to ActiveBPEL® Engine Administration Console Help

Persistence Storage Setup

If you are running the open source version of ActiveBPEL engine that you downloaded from www.active-endpoints.com, you can configure the engine to connect to a database for persistent storage. In persistence mode, running processes are not terminated when the engine is stopped and completed processes are stored in a database. You can select the type of persistence to use, ranging from none to full.

The requirements, necessary files, and instructions for configuring database settings and the ActiveBPEL engine are included in the `docs` folder of the ActiveBPEL engine installation folder.

Back to ActiveBPEL® Engine Administration Console Help

A ActiveBPEL Custom Faults

The following list specifies the custom faults defined for the ActiveBPEL engine. All these faults are in the namespace *http://www.active-endpoints.com/2004/06/bpel/extensions/*.

Table 1:

Fault name	Description
systemError	Unrecoverable system error
badProcess	Invalid BPEL
validationError	Error in message variable data. Validation errors are reported only if the configuration option “Validate input/output messages against schema” is enabled.
xpathFunctionError	Error in executing XPath function
invalidTransitionCondition	Non-Boolean return from an XPath evaluation of a transition condition
xpathDateParseError	error in parsing an xsd:date or xsd:datetime
xpathDurationFormatError	error in parsing an xsd:duration

Back to ActiveBPEL® Engine Administration Console Help