Oracle® BPEL Process Manager

Quick Start Guide

10*g* Release 2 (10.1.2)

B15604-02

December 2005

ORACLE®

Oracle BPEL Process Manager Quick Start Guide, 10*g* Release 2 (10.1.2)

B15604-02

Primary Author: Mark Kennedy

Contributor: Guru Balse, Prashant Nema, Dave Shaffer, Raj Venkatesan, Shirley Yen

Contents

3 Loan Process Tutorial

Index

Preface

This guide is the primary source of introduction and usage information for Oracle BPEL Process Manager.

Audience

This manual is intended for users who want to install and use Oracle BPEL Process Manager.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

`http://www.oracle.com/accessibility/`

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see the following Oracle resources:

- *Oracle BPEL Process Manager Order Booking Tutorial*
- *Oracle BPEL Process Manager Developer's Guide*

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

To download free release notes, installation documentation, white papers, or other collateral, visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

`http://www.oracle.com/technology/membership/`

To download Oracle BPEL Process Manager documentation, technical notes, or other collateral, visit the Oracle BPEL Process Manager site at Oracle Technology Network (OTN):

`http://www.oracle.com/technology/bpel/`

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

`http://www.oracle.com/technology/documentation/`

See the *Business Process Execution Language for Web Services Specification*, available at the following URL:

`http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbizspec/html/bpel1-1.asp`

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Getting Started with the Oracle BPEL Process Manager

This chapter provides an overview of how the business process execution language (BPEL) and Oracle BPEL Process Manager enable you to design service-oriented architecture (SOA)-based applications by developing synchronous and asynchronous services into end-to-end BPEL process flows. This chapter also describes how to install and start the Oracle BPEL Process Manager, Oracle BPEL Server, and other key components.

This chapter contains the following topics:

BPEL: The Cornerstone of SOA

Many companies are looking at Web services and SOA as a method for addressing the integration requirements involved in building connected applications. While SOA has existed for over a decade, there was confusion about which interfaces to adopt. BPEL and Web service standards solve this dilemma by addressing common application requirements in an open, portable, and standard way. SOA optimizes business performance by using existing resources and minimizing the cost of deploying new applications. Enterprises adopting these standards and architectural approaches achieve a significant return on investment (ROI) from using the same standards-based approach to building connected applications that they used for building Web applications with Java/Java 2 Platform, Enterprise Edition (J2EE).

Making Web services work is a two-step process:

1. Publish the services.

   Publishing a service involves taking a function within an existing application or system and making it available in a standard way.

2. Compose, or orchestrate, the services into business flows.

Orchestration involves composing multiple services into an end-to-end business process.

Web services standards, including web services description language (WSDL), extensible markup language (XML), and simple object access protocol (SOAP), have emerged as an effective and highly interoperable platform for publishing services. In addition, high performance binding frameworks enable enterprises to access legacy systems and native Java code without having to wrap them in a SOAP interface.

BPEL Concepts

BPEL is emerging as the clear standard for composing multiple synchronous and asynchronous services into collaborative and transactional process flows. BPEL benefits from over 15 years of research that improves upon its predecessor languages of XLANG and WSFL. BPEL provides the following features:

- Web services/WSDL as component model
- XML as data model (data loose-coupling)
- Synchronous and asynchronous message exchange patterns
- Deterministic and nondeterministic flow coordination
- Hierarchical exception management
- Long-running unit of work/compensation

Since the BPEL specification was submitted to the Organization for the Advancement of Structured Information Standards (OASIS) in March 2003, it has gained the support of nearly every major industry vendor. This provides a great benefit to enterprises that can now implement their business processes in a standard and portable way, avoiding vendor-specific rules to a degree not previously possible.

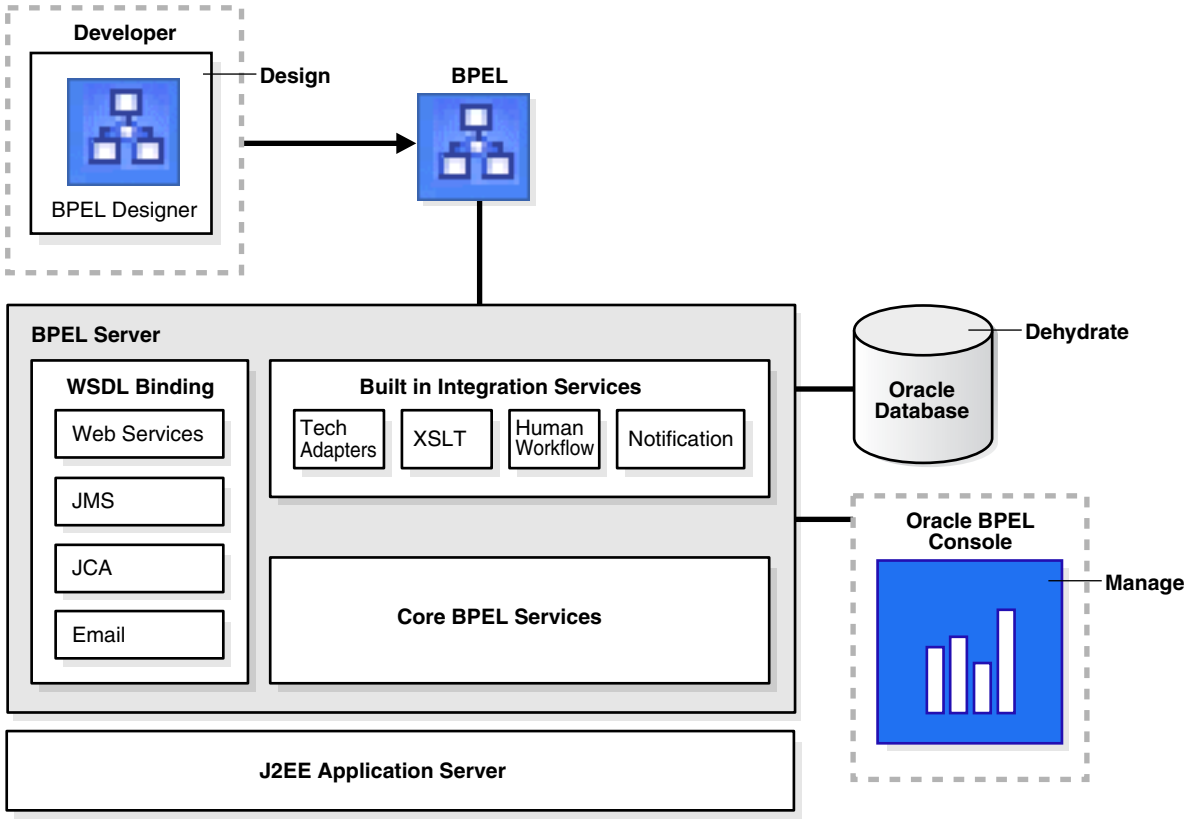
Oracle BPEL Process Manager

This section contains these topics:

Introduction to Oracle BPEL Process Manager

Oracle BPEL Process Manager provides a user-friendly and reliable solution for designing, deploying, and managing BPEL business processes. Oracle BPEL Process Manager consists of the key components shown in Figure 1–1.

Figure 1–1 Oracle BPEL Process Manager



Oracle BPEL Process Manager includes JDeveloper BPEL Designer, which extends the functionality of Oracle JDeveloper to enable you to model, edit, and design business processes using BPEL. JDeveloper BPEL Designer provides a graphical and user-friendly way to build BPEL processes. What is unique about JDeveloper BPEL Designer is that it uses BPEL as its native format. This means that processes built with JDeveloper BPEL Designer are 100% portable. JDeveloper BPEL Designer also enables you to view and modify the BPEL source without decreasing the usefulness of the tool. The core BPEL engine provides the most mature, scalable, and robust implementation of a BPEL server available today. Oracle BPEL Process Manager executes standard BPEL processes. Oracle BPEL Process Manager also provides a dehydration capability that enables the states of long-running flows to be automatically maintained in a database, thus enabling clustering for both fail over and scalability. Oracle BPEL Server uses an underlying J2EE application server, with support for most major commercial application servers and a bundled version available.

The built-in integration services enable you to use advanced connectivity and transformation capabilities of standard BPEL processes. These capabilities include support for XSLT and XQuery transformation, and bindings to hundreds of legacy systems through Java connector architecture (JCA) adapters and native protocols. A user task service is provided as a built-in BPEL service to enable the integration of people and manual tasks into BPEL flows.

The extensible WSDL binding framework enables connectivity to protocols and message formats other than SOAP. Bindings are available for JMS, e-mail, JCA, HTTP GET, HTTP POST, and many other protocols enabling simple connectivity to hundreds of back-end systems.

Oracle BPEL Console provides a mature Web-based interface for management, administration, and debugging of processes deployed to Oracle BPEL Server. Audit trails and process history and reporting information are automatically maintained and available through both Oracle BPEL Console and a Java API.

How Is Oracle BPEL Process Manager Different?

Oracle BPEL Process Manager provides the following key differences:

- The power of an open standard

  By capturing your business processes in BPEL, you protect your intellectual property and investments while avoiding vendor lock-in. BPEL is to business process management what SQL is to data management.

- Unparalleled visibility and administration

  Oracle BPEL Console reduces the cost and complexity of deploying and managing your business processes. Visually monitor the execution of each BPEL process, drill down into the audit trail and view the details of each conversation, or debug a running flow against its BPEL implementation.

- Open and flexible binding framework

  You can orchestrate XML Web services, and also Java/J2EE components, portals, JCA interfaces, and Java message service (JMS) destinations. You can tie into back-end systems. You can use your Java skills and application server investments.

Oracle BPEL Process Manager Feature Summary

Oracle BPEL Process Manager includes the following features:

- JDeveloper BPEL Designer
  - Native BPEL support
  - Drag-and-drop process modeler
  - Universal description, discover, and integration (UDDI) and Web services inspection language (WSIL) service browser
  - Visual XPath editor
  - One-click build and deploy
- Oracle BPEL Console
  - Visual monitoring
  - Auditing
  - BPEL debugging
  - Process versioning
  - In-flight administration
  - Process performance monitoring statistics
  - Partitioning/domains
- Built-in integration services
  - Java embedding
  - E-mail and JMS messaging services

  - Extensible style sheet language transformation (XSLT) and XQuery transformation services
  - Workflow tasks and portal integration
  - Extensible WSIF binding framework
- Oracle BPEL Server
  - Comprehensive BPEL version 1.1
  - Synchronous and asynchronous messaging
  - Context dehydration
  - Advanced exception management
  - Side-by-side versioning
  - Large XML documents

Installing Oracle BPEL Process Manager for Developers

Follow these instructions to install Oracle BPEL Process Manager for Developers. After completing installation, Oracle recommends that you visit the following URL to determine if any patches also require installation:

`http://www.oracle.com/technology/bpel`

1. Ensure that you meet the following requirements:

Element	Requirement
Operating system	Windows XP, Windows 2000 with Service Pack 3, Windows 2003 with Service Pack 1, Sun SPARC Solaris version 8 and 9, Red Hat Enterprise Linux AS/ES 2.1 and 3.0, or SUSE Linux Enterprise Server 8 and 9
Disk space	1 GB
Memory	512 MB RAM minimum (1 GB preferred)
Swap space	1535 MB minimum
Web browser	Internet Explorer 6.0 or Mozilla Firefox 1.0.4
Monitor	Configured to display at least 256 colors
Large display settings	If you configure your Windows environment to use display settings of large fonts and large size DPI settings (120 DPI), you must configure JDeveloper BPEL Designer after installation. This ensures that all BPEL elements correctly display in JDeveloper BPEL Designer. Perform the following procedures: 1. Right-click in JDeveloper BPEL Designer. 2. Select **Diagram Properties**. 3. Disable **Use Inline Editors**.

2. If installing on Solaris or Linux, see the appropriate *Oracle Application Server Installation Guide* for each operating system for instructions on:
   - Setting the mount point for the CD-ROM
   - Starting Oracle Universal Installer
3. Insert the CD-ROM.

4. Start Oracle Universal Installer from the `bpel` directory at the root of the CD-ROM:

On...	Do This...
Solaris or Linux	Enter the following command at the operating system prompt: `./runInstaller`
Windows	Click `setup.exe`.

The Welcome screen appears.

5. Click **Next**.

The Specify File Locations screen appears.

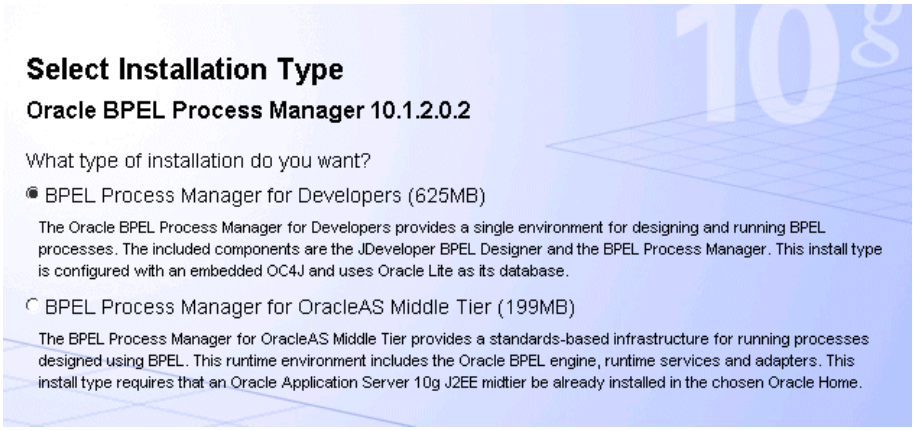
6. Create a new Oracle home name and directory path in which to install Oracle components in the **Destination** fields. Do not use an existing Oracle home name and directory path. For example, enter the following:

```
Name: OraBPELPM
Path: c:\OraBPELPM
```

Do not change the directory path in the **Source** field. This is the location of installation files.

7. Click **Next**.

The Select Installation Type screen appears.



The following installation types are available:

Product	Description
BPEL Process Manager for Developers	Installs an Oracle JDeveloper-based environment for modeling, editing, designing, and running business processes using BPEL. Oracle BPEL Server and Oracle BPEL Console are also installed.
BPEL Process Manager for OracleAS Middle Tier	Installs a BPEL server infrastructure for running BPEL processes that you design. This type does not install JDeveloper BPEL Designer. This type requires an Oracle Application Server 10*g* J2EE middle tier be installed in the chosen Oracle home directory. **See Also:** *Oracle Application Server Integration Installation Guide* for instructions on installing this type

8. Select BPEL Process Manager for Developers.

9. Click Next.

   The Specify Outgoing HTTP Proxy Information window appears.

10. If your host is located behind a fire wall, you may need to route outgoing HTTP connections through a proxy server. Enter the following information. If you are directly connected to the Internet or do not want to configure proxy information now, click **Next**.

Field	Description	Example
HTTP Proxy Host	Enter the name of the proxy server host.	`www-proxy.us.acme.com`
HTTP Proxy Port	Enter the port number of the proxy server host.	`80`
Bypass proxy for addresses	Enter an address that bypasses the proxy.	`*.us.acme.com;us.acme.com<local>`

11. Review specific details, including the space requirements to ensure that you have sufficient disk space.
12. Click Install.
13. Click Exit when installation completes and confirm when prompted.
14. Visit the following URL to determine if any patches also require installation:

    `http://www.oracle.com/technology/bpel`

Directory Structure

After completing installation, the directory structure shown in Table 1–1 is created:

Table 1–1 Directory Structure

Directory	Description
`cfgtoollogs`	Contains postinstallation configuration tool log files.
`diagnostics`	Contains an Oracle Universal Installer `.xml` file.
`integration`	Contains the following subdirectories:
▪ `jdev`	▪ Contains JDeveloper BPEL Designer files and directories, including the workspace and project directories (under `jdev\jdev\mywork`) in which your BPEL processes are created and designed.
▪ `orabpel`	▪ Contains Oracle BPEL Process Manager files and directories, including the `samples\demos` and `samples\utils` directories from which you start Web services for the tutorials in this guide. The samples\tutorials directory also provides additional tutorials.
`inventory`	Contains Oracle Universal Installer installation inventory files.
`jdk`	Contains the required Java Developer's Kit version.
`jre`	Contains the required Java Runtime environment files.
`lib`	Contains library files.
`Opatch`	Contains path files.
`oui`	Contains Oracle Universal Installer files.

Table 1–1 (Cont.) Directory Structure

Directory	Description
`perl`	Contains required Perl script files.

Starting Oracle BPEL Process Manager Components

Follow the instructions in Table 1–2 to start and stop Oracle BPEL Process Manager components.

Table 1–2 Starting and Stopping Oracle BPEL Process Manager Components

To Access The...	On Windows...	On UNIX...

Oracle BPEL Server	To start Oracle BPEL Server:
	Select **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **Start BPEL PM Server**
	To stop Oracle BPEL Server:
	Select **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **Stop BPEL PM Server**
		To start Oracle BPEL Server:
		From `$ORACLE_HOME/integration/orabpel/bin`:
		`startorabpel.sh`
		To stop Oracle BPEL Server:
		From `$ORACLE_HOME/integration/orabpel/bin`:
		`shutdownorabpel.sh`

JDeveloper BPEL Designer	Select **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **JDeveloper BPEL Designer** to start JDeveloper BPEL Designer, or use the shortcut icon that is placed on your desktop.	`$ORACLE_HOME/integration/jdev/jdev/bin/jdev`
Oracle BPEL Console	You must first start Oracle BPEL Server. To start Oracle BPEL Console: 1. Select **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console** You can also start Oracle BPEL Console using the URL for your installation, which can found in `bpelsetupinfo.txt`.	First start Oracle BPEL Server. To start Oracle BPEL Console: - Log on to the URL for your installation, which can found in `bpelsetupinfo.txt`.

Table 1–2 (Cont.) Starting and Stopping Oracle BPEL Process Manager Components

To Access The...	On Windows...	On UNIX...

Developer Prompt	Select **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **Developer Prompt** to open up a command prompt at the *Oracle_Home*\integration\orabpel\samples directory. This enables you to easily access demonstrations and start any required Web services.	Setting Developer Prompt in Bourne shell:
		`$ ORACLE_HOME=/home/oracle/installs/midtier`
		`$ export ORACLE_HOME`
		`$ PATH=$ORACLE_HOME/integration/orabpel/bin:$PATH`
		`$ export PATH`

Oracle BPEL Process Manager Samples and Tutorials	For details about BPEL samples and additional tutorials available for use: Select **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **Getting Started with Samples**	Log into the following URL: `$ORACLE_HOME/integration/orabpel/samples/sampleshome.html`
Sample Worklist Application	To access the login window for Oracle BPEL Worklist Application: Select **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **Sample Worklist Application** You may also start Oracle BPEL Worklist Application using the URL for your installation, which is found in `bpelsetupinfo.txt`.	First start Oracle BPEL Server. To start Oracle BPEL Worklist Application: ▪ Log on to the URL for your installation, which is found in `bpelsetupinfo.txt`.

See Also: *Oracle BPEL Process Manager Order Booking Tutorial* for a tutorial that uses Oracle BPEL Worklist Application

Connecting to Oracle BPEL Server

When you start JDeveloper BPEL Designer for the first time, a connection named **LocalBPELServer** is automatically created. This connection enables you to deploy your BPEL process from JDeveloper BPEL Designer to Oracle BPEL Server. The **LocalBPELServer** connection is sufficient for completing all tutorials in this guide.

If you want to create additional connections, follow these instructions.

1. Select **Connection Navigator** from the **View** main menu in JDeveloper BPEL Designer.
2. Right-click **BPEL Process Manager Server**.
3. Select **New BPEL Process Manager Connection**.
4. Click **Next** on the **Welcome** page.
5. Provide a meaningful name for connecting to the server.
6. Click **Next**.
7. Enter the following details:

Field	Value
Host Name	`localhost` (default value)
Port	`9700` (default value)

8. Click **Next**.
9. Test the connection by clicking **Test Connection**. If the connection is successful, the following message appears:

   ```
   Success.
   ```

10. Click **Finish**.

Setting the Hostname in Your Web Browser Preferences

Add the hostname of your computer to the Oracle JDeveloper preference settings. If you do not do this, you can receive parsing errors when selecting a WSDL file on the WSDL Chooser window while creating a partner link.

1. Select **Preferences** from the **Tools** main menu.
2. Click **Web Browser and Proxy**.
3. Enter your hostname in the **Exceptions** field. For example, if your hostname is `myhost-pc`:

   ```
   us.acme.com|*.us.acme.com|localhost|127.0.0.1|myhost-pc
   ```

4. Ensure also that `localhost` appears in the **Exceptions** field.
5. Click **OK**.

   You are now ready to design your BPEL process.

Understanding Namespaces and Namespace URIs

As you use JDeveloper BPEL Designer, namespace values such as **ns1:**, **ns2:**, or **client:** can display in an **Expression** field or **Variables** navigation tree when creating copy rules in **Assign** activities, or in fields of other activities.

Oracle BPEL Process Manager attempts to reduce your interaction with namespaces. When a namespace appears in an **Expression** field or **Variables** navigation tree, accept the namespace prefixes that display when filling in the parts of your query or expression.

XML namespaces provide a method for distinguishing between duplicate element type and attribute names. Such duplication can occur, for example, in an XSLT style sheet or document that contains element types and attributes from two different DTDs. A `<refuse>` element means something entirely different for a sanitation agency than a `<refuse>` element for a loan processing agency. For this reason, they are distinguished by different namespace prefixes, such as **ns1:**, **ns2:**, or **client:**.

The URIs provide a system for creating unique identifiers.

Eclipse BPEL Designer on Eclipse Platform

A production version of Eclipse BPEL Designer is also available as a plug-in on the Eclipse Platform version 3.0GA. The Eclipse Platform is designed for building integrated development environments (IDEs). You can use IDEs to create applications such as Web sites, embedded Java programs, C++ programs, and Enterprise JavaBeans.

For additional details, including Eclipse BPEL Designer and Oracle BPEL Process Manager software downloads, see the following URL:

http://www.oracle.com/technology/bpel

2

Credit Flow Tutorial

This tutorial describes how to use JDeveloper BPEL Designer to design, deploy, and test your first BPEL process.

This tutorial contains these topics:

Introduction

This tutorial teaches you how to use JDeveloper BPEL Designer to build, deploy, and test your first BPEL process. The process is a flow that you create to call a Credit Rating service. When you run this process, you enter your social security number into an HTML user interface. The Credit Rating service takes your number and returns a credit rating. Creating this process is intended to be the first step toward building a more sophisticated application (like that shown in Chapter 3, "Loan Process Tutorial").

Using the Tutorial

This tutorial contains these topics:

Starting Oracle BPEL Server and JDeveloper BPEL Designer

Ensure that JDeveloper BPEL Designer (which provides extensions to Oracle JDeveloper) and Oracle BPEL Server are started. See "Starting Oracle BPEL Process Manager Components" on page 1-8 for instructions.

Starting and Testing Your Service

During this tutorial, the BPEL process that you design communicates with a Credit Rating service described in "Introduction" on page 2-1. You must first start the service and test that it is running.

1. Select **Start** > **All Programs** > **Oracle -** ***Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **Developer Prompt** to open up a command prompt at the *Oracle_Home*\integration\orabpel\samples directory.
2. Change directories to the utils\CreditRatingService subdirectory:

   ```
   cd utils\CreditRatingService
   ```

3. Enter the following command:

   ```
   obant
   ```

   This deploys and starts the Credit Rating service for using this tutorial. If successful, a message similar to the following appears at the end:

   ```
   BUILD SUCCESSFUL
   Total time: 13 seconds
   C:\oraBPEL\integration\orabpel\samples\utils\CreditRatingService>ENDLOCAL
   ```

4. Log into Oracle BPEL Console by selecting **Start** > **All Programs** > **Oracle -** ***Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console**.
5. Enter **bpel** as the password when prompted.

   All services are running if the **CreditFlow**, **TaskActionHandler**, and **TaskManager** services display in the Dashboard tab.

Creating a Workspace and a Project

You must create a workspace and a project to begin. The project automatically creates the basics for your BPEL process, including:

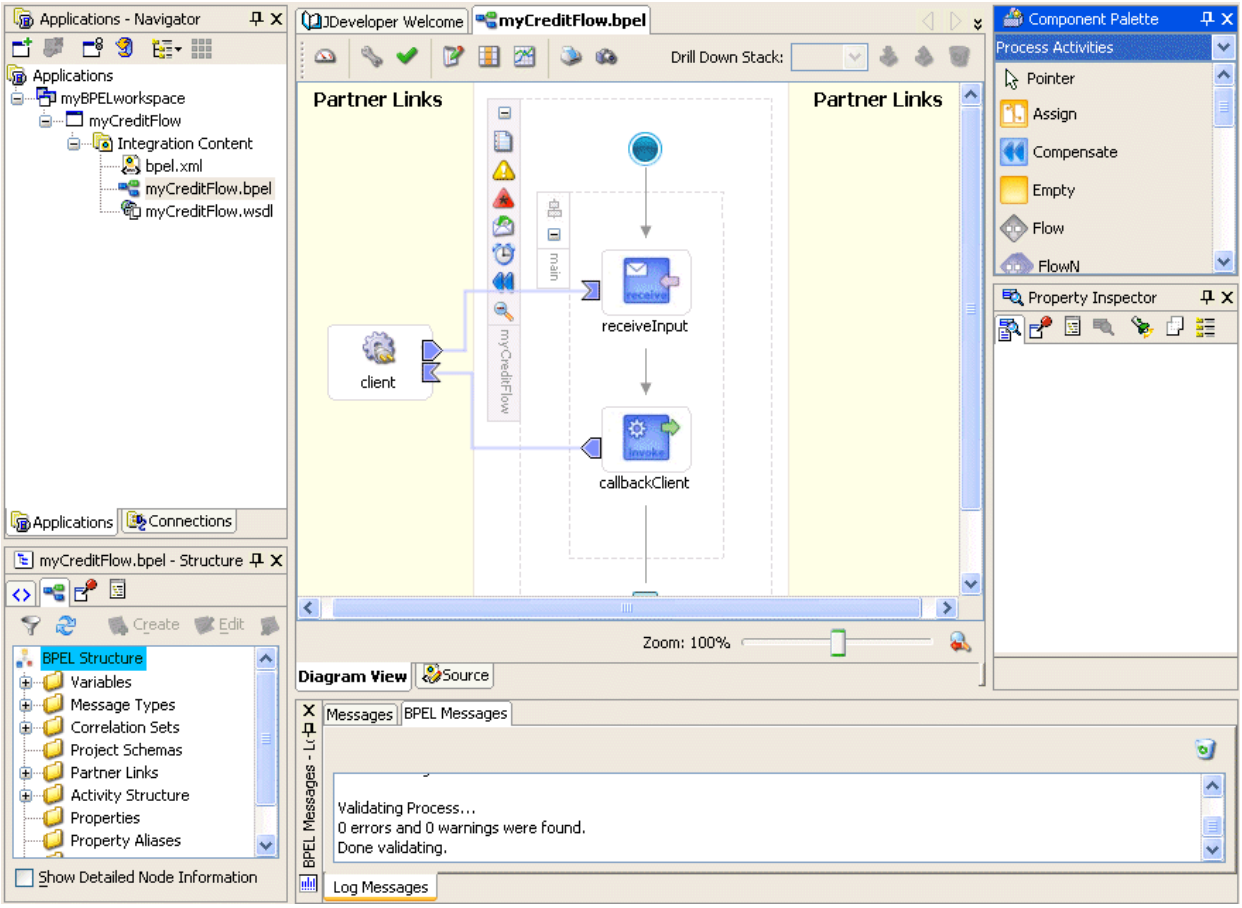
- BPEL process source (*projectname*.bpel)
- Web services description language (WSDL) client interface (*projectname*.wsdl)
- BPEL process deployment descriptor (bpel.xml)

Follow these instructions to create a new Credit Flow project.

> **Caution:** Do not include any special characters in the project name (such as periods). If you do include special characters, errors appear when you attempt to compile your project.

1. Return to JDeveloper BPEL Designer.
2. Select **File** > **New** from the main menu.
3. Double-click **Workspace** in the **Items** window to display the Create Workspace window.
4. Enter **myBPELworkspace** in the **Workspace Name** field and accept the default path in the **Directory Name** field.
5. Deselect the **Add a New Empty Project** check box.
6. Click **OK**.

7. Right-click **myBPELworkspace** in the **Applications Navigator** section.
8. Select New **Project** to define a new BPEL process project.
9. Double-click **BPEL Process Project** in the **Items** window to display the BPEL Process Project window.
10. Enter **myCreditFlow** in the **BPEL Process Name** field. All other fields default to the correct values for creating an asynchronous BPEL process.
11. Click **OK**. The BPEL process files are created in the *Oracle_Home*\integration\jdev\jdev\mywork\myBPELworkspace\myCreditFlow directory.
12. The following sections appear. If they do not appear, click **Diagram View** in JDeveloper BPEL Designer and double-click **myCreditFlow.bpel** in the **Applications Navigator** section. See the *Oracle BPEL Process Manager Developer's Guide* for a description of JDeveloper BPEL Designer sections.



Viewing the WSDL File Source Code

You now review the sections of the WSDL file.

1. Double-click **myCreditFlow.wsdl** in the **Applications Navigator** section.
2. View the following section of code. A `myCreditFlowProcessRequest` element is accepted as input. A `myCreditFlowProcessResponse` element is returned as output.

```
<element name="myCreditFlowProcessRequest">
```

```
   <complexType>
      <sequence>
         <element name="input" type="string"/>
      </sequence>
   </complexType>
</element>
<element name="myCreditFlowProcessResponse">
   <complexType>
      <sequence>
         <element name="result" type="string"/>
      </sequence>
   </complexType>
</element>
```

3. View the following section of code. Two port types are defined, each with a one-way operation. One operation initiates the asynchronous process. The other operation calls back the client with the asynchronous response.

```
<portType name="myCreditFlow">
   <operation name="initiate">
      <input message="client:myCreditFlowRequestMessage"/>
   </operation>
</portType>

<!-- portType implemented by the requester of myCreditFlow BPEL process
for asynchronous callback purposes
-->
<portType name="myCreditFlowCallback">
   <operation name="onResult">
      <input message="client:myCreditFlowResponseMessage"/>
   </operation>
</portType>
```

4. View the following section of code. The `partnerLinkType` for this asynchronous process has two roles. One role is for the service provider, and the other is for the requester.

```
<plnk:partnerLinkType name="myCreditFlow">
   <plnk:role name="myCreditFlowProvider">
      <plnk:portType name="client:myCreditFlow"/>
   </plnk:role>
   <plnk:role name="myCreditFlowRequester">
      <plnk:portType name="client:myCreditFlowCallback"/>
   </plnk:role>
</plnk:partnerLinkType>
```

Editing the WSDL File Source Code

You now edit the input and output messages of the WSDL file. This WSDL file represents the user interface with which you interact when you run your BPEL process and request a credit rating in "Running the Credit Flow Process" on page 2-12.

1. Change `input` to `ssn`.
2. Change `result` to `creditRating` and `string` to `int` as the output to be returned.

```
<element name="CreditFlowProcessRequest">
   <complexType>
      <sequence>
         <element name="ssn" type="string"/>
```

```
      </sequence>
   </complexType>
</element>
<element name="CreditFlowProcessResponse">
   <complexType>
      <sequence>
         <element name="creditRating" type="int"/>
      </sequence>
   </complexType>
</element>
```

3. Select **Save** from the **File** main menu.
4. Close the WSDL window by clicking the **x** in **myCreditFlow.wsdl** at the top of the designer window.
5. Double-click **myCreditFlow.bpel** in the **Applications Navigator** section.
6. Go to "Viewing the BPEL File Source Code" on page 2-5.

Viewing the BPEL File Source Code

You now review the sections of the BPEL file.

1. Click **Source** at the bottom of the designer window.
2. View the following section of code. The `partnerLink` created for the client interface includes two roles, `myRole` and `partnerRole`. An asynchronous BPEL process typically has two roles for the client interface: one for the flow itself, which exposes an input operation, and one for the client, which gets called back asynchronously.

```
<partnerLinks>
   <partnerLink name="client" partnerLinkType="client:myCreditFlow"
   myRole="myCreditFlowProvider" partnerRole="myCreditFlowRequester"/>
</partnerLinks>
```

3. View the following section of code. The `<receive>` activity in the main body of the process is followed by an `<invoke>` activity to perform an asynchronous callback to the requester. Note the difference between this and a synchronous process, which uses a `<reply>` activity to respond synchronously to the caller.

```
  <sequence name="main">

    <!-- Receive input from requestor.
    Note: This maps to operation defined in myCreditFlow.wsdl
    -->
    <receive name="receiveInput" partnerLink="client"
 portType="client:myCreditFlow" operation="initiate" variable="inputVariable"
 createInstance="yes"/>

    <!-- Asynchronous callback to the requester.
     Note: the callback location and correlation id is transparently handled
     using WS-addressing.
     -->
    <invoke name="callbackClient" partnerLink="client"
 portType="client:myCreditFlowCallback" operation="onResult"
 inputVariable="outputVariable"/>
  </sequence>
</process>
```

4. Click **Diagram View**. You are now ready to design your BPEL process.

Creating and Configuring a Partner Link for the Credit Rating Service

You now create and configure a partner link for the synchronous Credit Rating service.

This section contains these tasks:

- Creating a Credit Rating Service Partner Link
- Creating a Scope Activity
- Creating an Invoke Activity Inside the Scope Activity
- Creating an Initial Assign Activity Inside the Scope Activity
- Creating a Second Assign Activity Inside the Scope Activity

> **Note:** As you create and open activities such as Scope and Assign for the first time, the message `Invalid Settings` can appear at the top. This is because you have not yet entered details. You can ignore this message. After you enter and apply your details, the message disappears.

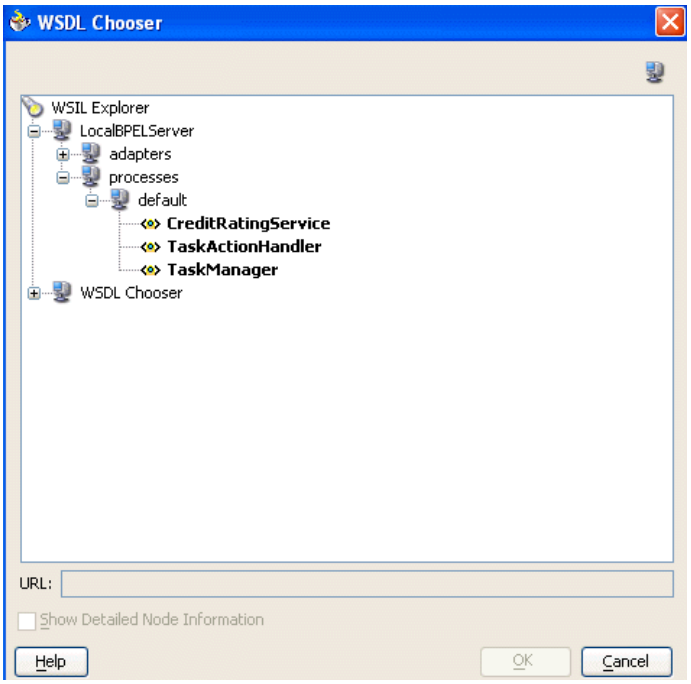
Creating a Credit Rating Service Partner Link

> **Summary:** Partner links define the external services with which your BPEL process interacts. You must create a partner link for the Credit Rating service.

1. Ensure that **Process Activities** is selected in the drop-down list of the **Component Palette** section in the upper right part of JDeveloper BPEL Designer.
2. Drag and drop a **PartnerLink** activity onto the right side of JDeveloper BPEL Designer.

   The Create Partner Link window appears.
3. Enter the following values to create a partner link for the Credit Rating service:

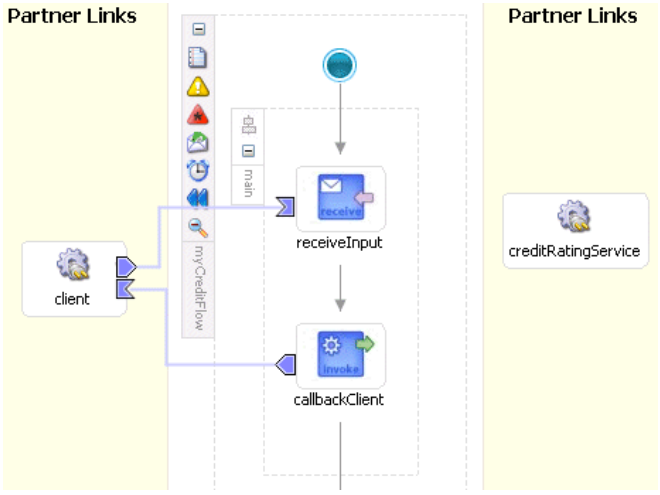
> **Note:** For the **WSDL File** field below, click the **flashlight** (the second icon from the left named **WSIL Browser**) to access the WSDL Chooser window shown below for automatically selecting the Credit Rating service deployed in "Starting and Testing Your Service" on page 2-2.



Field	Value
Name	**creditRatingService**
WSDL File	Access this URL by clicking the WSIL Browser **flashlight** icon and expanding and selecting **LocalBPELServer** > **processes** > **default** > **CreditRatingService**. **http://localhost:9700/orabpel/default/CreditRatingService/CreditRatingService?wsdl** **See Also:** "Setting the Hostname in Your Web Browser Preferences" on page 1-10 if you receive a parsing error when attempting to add a WSDL file in the WSDL Chooser window.
Partner Link Type	**CreditRatingService**
My Role	Leave unspecified. Because this is a synchronous partner link, no role is required.
Partner Role	**CreditRatingServiceProvider**

4. Click **OK**.

   The **creditRatingService** partner link appears on the right side of the design window.



5. Select **Save** from the **File** main menu.

Creating a Scope Activity

> **Summary:** You first create a **Scope** activity in this section. A **Scope** activity consists of a collection of activities that can have their own local variables, fault handlers, and so on. A **Scope** activity is analogous to a `{ }` block in a programming language.
>
> Within this **Scope** activity, the client's credit rating background is identified.

1. Go to the **Component Palette** section.
2. Drag and drop a **Scope** activity between the **receiveInput** and **callbackClient** activities.
3. Double-click the **Scope** activity to display the Scope window.
4. Enter **GetCreditRating** in the **Name** field of the **General** tab.
5. Click **OK**.
6. Select **Save** from the **File** main menu.

Creating an Invoke Activity Inside the Scope Activity

> **Summary:** You create an **Invoke** activity in this section. An **Invoke** activity enables you to specify the operation you want to invoke for the service (identified by its partner link).
>
> This **Invoke** activity provides the initial interaction operation between the client and the Credit Rating service.

1. Click the **+** sign to expand the **GetCreditRating Scope** activity.
2. Drag and drop an **Invoke** activity from the **Component Palette** section into the **GetCreditRating Scope** activity.
3. Double-click the **Invoke** icon to display the Invoke window.
4. Enter the following details:

Field	Value
Name	**invokeCRS**
Partner Link	**creditRatingService**

The **Operation** (**process**) field is automatically filled in.

5. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.

6. Click **OK** on the Create Variable window that appears.

   A variable named **invokeCRS_process_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **CreditRatingServiceRequestMessage**.

7. Click the first icon to the right of the **Output Variable** field.

8. Click **OK** on the Create Variable window that appears.

   A variable named **invokeCRS_process_OutputVariable** is automatically created in the **Output Variable** field. This variable is automatically assigned a message type of **CreditRatingServiceResponseMessage**.

9. Click **OK**.

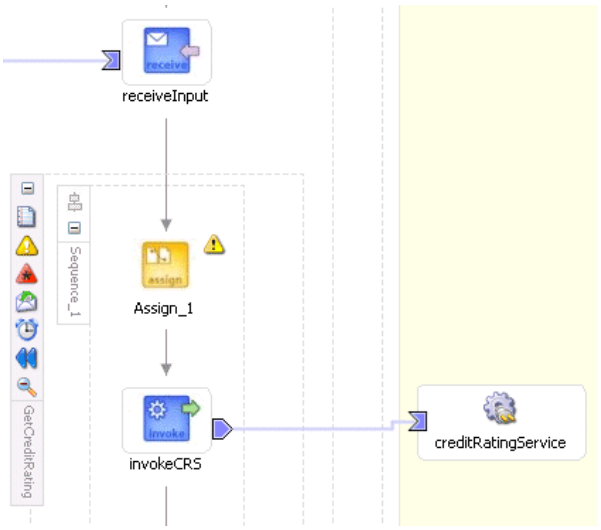
10. Select **Save** from the **File** main menu.

Creating an Initial Assign Activity Inside the Scope Activity

Summary: In this section you create the first of two **Assign** activities in this tutorial. An **Assign** activity provides a method for simple data manipulation, such as copying the contents of one variable to another.

This **Assign** activity takes the client's social security number as input and contacts the Credit Rating service to assign a credit rating.

1. Drag and drop an **Assign** activity from the **Component Palette** section to above the **invokeCRS Invoke** activity.



2. Double-click the **Assign** icon to display the Assign window.
3. Enter **assignSSN** in the **Name** field of the **General** tab.
4. Click **Apply**.
5. Click the **Copy Rules** tab.
6. Click **Create** to display the Create Copy Rule window.
7. Enter the following values:

Field	Value
From	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **inputVariable** > **payload** > client:**myCreditFlowProcessRequest** > **client:ssn** **Note:** The namespace number values (for example, **ns1**, **ns2**) can vary. Use the namespace values that automatically appear.
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **invokeCRS_process_InputVariable** > **payload** > **ns1:ssn**

8. Click **OK** to close the Create Copy Rule window and the Assign window.
9. Select **Save** from the **File** main menu.

Creating a Second Assign Activity Inside the Scope Activity

Summary: This **Assign** activity returns details about the client's assigned credit rating.

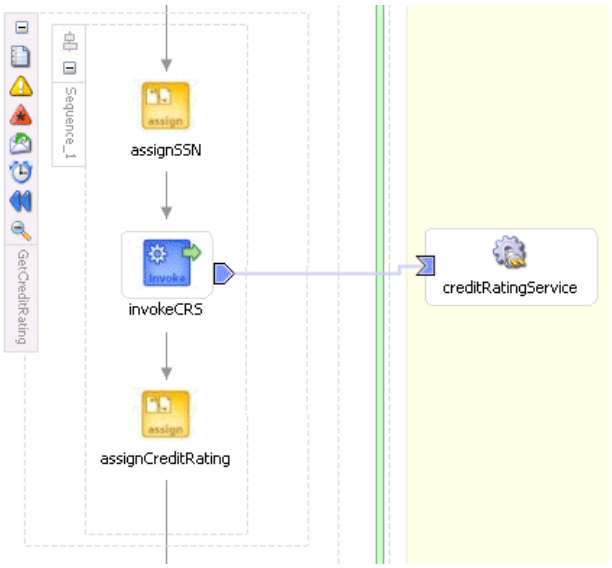
1. Drag and drop a second **Assign** activity from the **Component Palette** section to below the **invokeCRS Invoke** activity within the **GetCreditRating Scope** activity.
2. Double-click the **Assign** icon to display the Assign window.

3. Enter **assignCreditRating** in the **Name** field of the **General** tab.
4. Click **Apply**.
5. Click the **Copy Rules** tab.
6. Click **Create** to display the Create Copy Rule window.
7. Enter the following values:

Field	Value
From	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables > invokeCRS_process_OutputVariable > payload** > **ns1:rating** **Note:** The namespace number values (for example, **ns1**, **ns2**) can vary. Use the namespace values that automatically appear.
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **outputVariable** > **payload** > **client:myCreditFlowProcessResponse** > **client:creditRating**

8. Click **OK** to close the Create Copy Rule window and the Assign window.

   The **Scope** activity looks as follows:



9. Select **Save** from the **File** main menu.

   You have now completed the Credit Rating service design and are ready to proceed to the validation, compilation, and deployment tasks described in the following section.

Validating, Compiling, and Deploying the Credit Flow Process

You are now ready to deploy your BPEL process.

1. Go to the **Applications Navigator** section.

2. Right-click **myCreditFlow**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter **bpel** when prompted for the domain password.
5. Click **OK**.

   This compiles the BPEL process. Review the bottom of the window for any errors. If there are no errors, the following message appears:

   ```
   [3:43:31 PM] Successful compilation: 0 errors, 0 warnings.
   Deploying to http://localhost:9700 domain: default. Please wait ...
               bpel_myCreditFlow_1.0.jar deployed successfully.
   ```

6. If there are errors, click **BPEL Validation Errors** to display details about the type and location of the error.
7. Make corrections and deploy again.

Running the Credit Flow Process

You are now ready to begin the process of applying for and receiving a loan offer.

1. Log into Oracle BPEL Console by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console**.
2. Enter **bpel** when prompted for the password.

   The **Dashboard** tab of Oracle BPEL Console appears.
3. Click **myCreditFlow** in the **Deployed BPEL Processes** list.
4. Enter a nine-digit integer value that does *not* begin with zero in the **ssn** field of the HTML Form and click **Post XML Message**. Note that the name **ssn** that displays in the user interface is the value you entered in "Editing the WSDL File Source Code" on page 2-4.

   The **BPEL Processes** tab displays a message similar to the following:

   ```
   Test Instance Initiated

   Instance '39e706a46ad531be:858bf1:fcc240f310:-7ffc' is being processed
   asynchronously.
   ```

5. Click **Visual Flow** to view a visual audit trail of the current state of the process instance.

   An audit trail displaying the current state of the process appears. It indicates that you have successfully invoked the Credit Rating service.
6. Click **callbackClient** in the audit trail to see the results of the loan offer, including the credit rating returned (**560**) by the Credit Rating service to the client.

7. Click the **Audit** link at the top to observe additional information.

3

Loan Process Tutorial

This tutorial describes how to design and execute a more sophisticated BPEL process than the one you created in Chapter 2, "Credit Flow Tutorial". Fault-handling, interaction with asynchronous services, parallel flows of execution, and decision making logic are all key requirements for many business flows. This tutorial enables you to design, test, debug, and manage a BPEL process that implements all of these requirements.

This tutorial contains these topics:

- Loan Process Introduction
- Using the Loan Process Tutorial

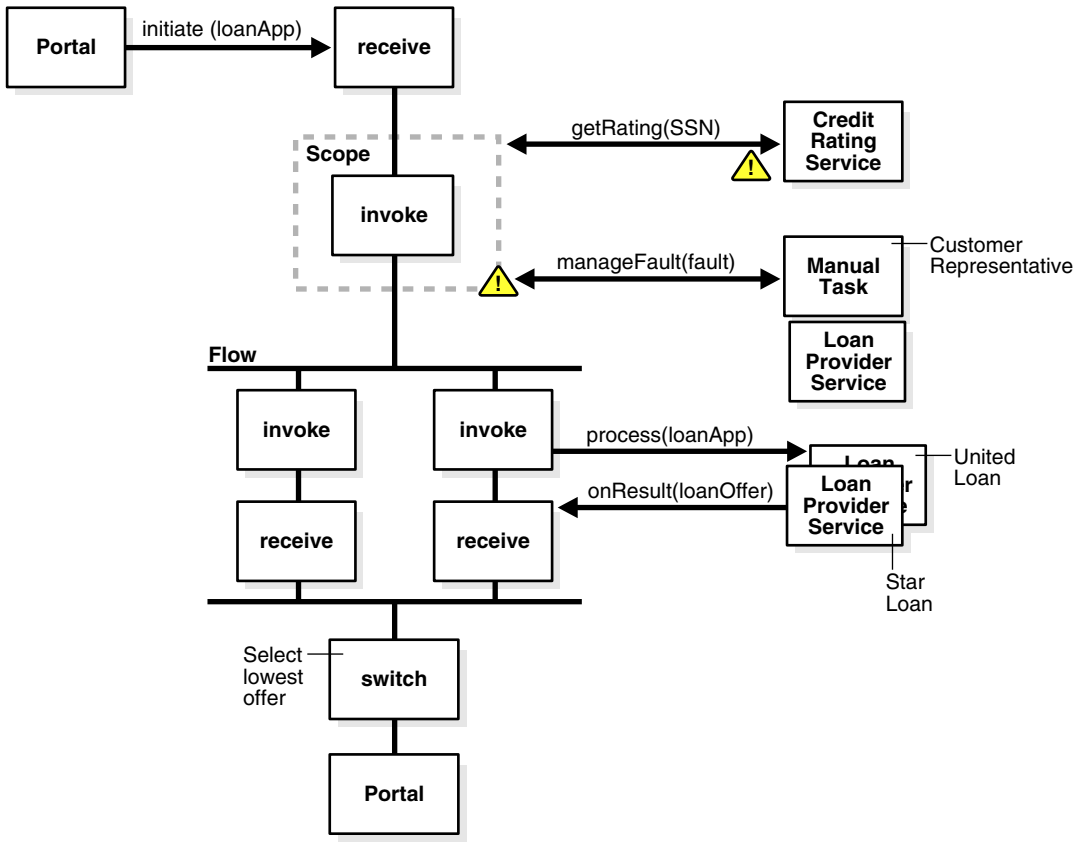
Loan Process Introduction

In this tutorial, you design a process that takes a loan application document from a client and returns a selected and approved loan offer. When you run this process, it retrieves a social security number that you enter into an HTML user interface and contacts the Credit Rating service to request a credit rating for the client. If a credit rating is supplied, the process sends the completed loan application to two loan providers (Star Loan and United Loan). Each provider can take some time before returning loan offers. The flow waits for both loan offers to be received before selecting the offer with the lowest interest rate to send to the client.

The process can also be designed to handle negative credit exceptions. These exceptions are made known by the credit rating service if a negative credit event is identified for the client (for example, a bankruptcy).

Figure 3–1 provides an overview of this communication:

Figure 3–1 Loan Process Tutorial Overview



Using the Loan Process Tutorial

This tutorial contains these topics:

- Starting Oracle BPEL Server and JDeveloper BPEL Designer
- Starting and Testing Your Services
- Creating a Loan Process Project
- Creating the Loan Process Schema File
- Importing the Loan Process Schema File
- Creating and Configuring a Partner Link for the Credit Rating Service
- Creating and Configuring Partner Links for the Asynchronous United Loan and Star Loan Services
- Adding Offer Decision Making Logic to the Loan Process
- Validating, Compiling, and Deploying the Loan Process
- Running the Loan Process
- Adding Error Handling Capabilities to the Loan Process (Optional)

Note: Save frequently to ensure that you do not lose any changes.

Starting Oracle BPEL Server and JDeveloper BPEL Designer

Ensure that Oracle BPEL Server and JDeveloper BPEL Designer are started. See "Starting Oracle BPEL Server and JDeveloper BPEL Designer" on page 2-1 for instructions.

Starting and Testing Your Services

During this tutorial, the BPEL process that you design communicates with the Credit Rating service and the Star Loan and United Loan provider services described in "Loan Process Introduction" on page 3-1. You must first start these services and test that they are running.

1. Select **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **Developer Prompt** to open up a command prompt at the *Oracle_Home*`\integration\orabpel\samples` directory.
2. Change directories to the `demos\LoanDemo` subdirectory:

   ```
   cd demos\LoanDemo
   ```

3. Enter the following command:

   ```
   obant
   ```

   This deploys and starts the required services for using this tutorial. If successful, a message similar to the following appears at the end:

   ```
   deployLoanFlowUI:

   all:

   BUILD SUCCESSFUL
   Total time: 1 minute 25 seconds
   C:\oranew\integration\orabpel\samples\demos\LoanDemo>ENDLOCAL
   ```

4. Log into Oracle BPEL Console by selecting **Start** > **All Programs** > **Oracle -** *Oracle_Home* > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console**.
5. Enter **bpel** as the password when prompted.

   All services for this tutorial are running if the following names display in the **Dashboard** tab.

   - **CreditRatingService**
   - **LoanFlow**
   - **StarLoan**
   - **TaskActionHandler**
   - **TaskManager**
   - **UnitedLoan**

   See Also: "Connecting to Oracle BPEL Server" on page 1-9 for instructions on creating a connection

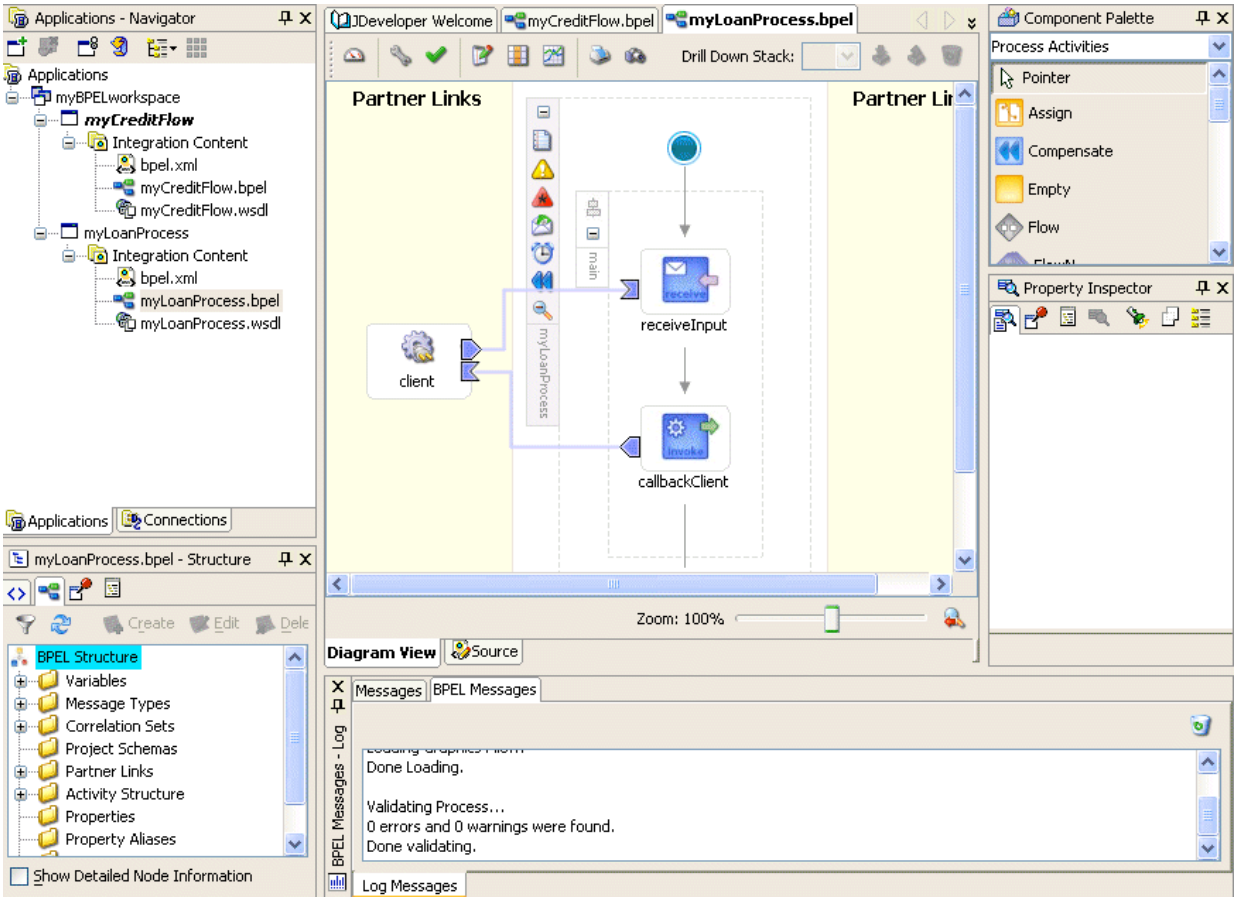
Creating a Loan Process Project

You must create a project to begin. Follow these instructions to create a new Loan Process project inside the same **myBPELworkspace** workspace you created in

"Creating a Workspace and a Project" on page 2-2. If you have not yet created a workspace, see that section for instructions.

> **Caution:** Do not include any special characters in the project name (such as periods) or in any activity or element names. If you do include special characters, errors appear when you attempt to compile your project.

1. Right-click **myBPELworkspace** in the **Applications Navigator** section of JDeveloper BPEL Designer.
2. Select **New Project**.
3. Double-click **BPEL Process Project** in the **Items** window to display the BPEL Process Project window.
4. Enter **myLoanProcess** in the **BPEL Process Name** field. All other fields default to the correct values for creating an asynchronous BPEL process.
5. Click **OK**.
6. The following sections appear. If these do not appear, click **Diagram View** and double-click **myLoanProcess.bpel** in the **Applications Navigator** section. See the *Oracle BPEL Process Manager Developer's Guide* for a description of JDeveloper BPEL Designer sections.



Creating the Loan Process Schema File

1. Use an ASCII text editor to copy and paste the following syntax into a file:

```
<?xml version="1.0"?>
    <schema attributeFormDefault="qualified" elementFormDefault="qualified"
 targetNamespace="http://www.autoloan.com/ns/autoloan"
 xmlns="http://www.w3.org/2001/XMLSchema"
 xmlns:s1="http://www.autoloan.com/ns/autoloan">
         <element name="loanApplication" type="s1:LoanApplicationType"/>
         <element name="loanOffer" type="s1:LoanOfferType"/>

         <complexType name="LoanOfferType">
             <sequence>
               <element name="providerName" type="string"/>
               <element name="selected" type="boolean"/>
               <element name="approved" type="boolean"/>
               <element name="APR" type="double"/>
             </sequence>

         </complexType>
         <complexType name="LoanApplicationType">
             <sequence>
               <element name="SSN" type="string"/>
               <element name="email" type="string"/>
               <element name="customerName" type="string"/>
               <element name="loanAmount" type="double"/>
               <element name="carModel" type="string"/>
               <element name="carYear" type="string"/>
               <element name="creditRating" type="int"/>
             </sequence>
         </complexType>
    </schema>
```

2. Save the file to *Oracle_Home*\integration\jdev\jdev\mywork\myBPELworkspace\myLoanProcess\Loanflow.xsd.

Importing the Loan Process Schema File

1. Go to the Structure section on the lower left side of JDeveloper BPEL Designer. If this section does not appear, double-click **myLoanProcess.bpel** in the **Applications Navigator**.
2. Right-click **Project Schemas**.
3. Select **Import Schema** to display the Import Schema window.
4. Click the **flashlight** icon to access the Open window.
5. Select **Loanflow.xsd** from the directory in which you saved it in Step 2 on page 3-5 and click **Open**.

   The file is added to the **URL** field of the Import Schema window.
6. Click **OK**.

   The **Loanflow.xsd** file now displays in both the **Applications Navigator** section (under **myLoanProcess** > **Web Content** > **Miscellaneous Files**) and the **Structure** section (under **Project Schemas**).
7. Select and right-click **Message Types** in the **Structure** tree.

8. Select **Expand All Child Nodes**.
9. Select **myLoanProcessRequestMessage** > **payload**.
10. Right-click **payload** and select **Edit Message Part**.
11. Click **Element** and click the **flashlight** icon to display the Type Chooser window.
12. Select **Project Schema Files** > **Loanflow.xsd** > **loanApplication**.
13. Click **OK** to close the Type Chooser window and Edit Message Part window.
14. Select **myLoanProcessResponseMessage** > **payload**.
15. Right-click **payload** and select **Edit Message Part**.
16. Click **Element** and click the **flashlight** icon to display the Type Chooser window.
17. Select **Project Schema Files** > **Loanflow.xsd** > **loanOffer**.
18. Click **OK** to close the Type Chooser window and Edit Message Part window.
19. Select **Save** from the **File** main menu.

Creating and Configuring a Partner Link for the Credit Rating Service

You now create and configure a partner link for the synchronous Credit Rating service.

This section contains these tasks:

- Creating a Credit Rating Service Partner Link
- Creating a Scope Activity
- Creating an Invoke Activity Inside the Scope Activity
- Creating an Initial Assign Activity Inside the Scope Activity
- Creating a Second Assign Activity Inside the Scope Activity

---

Note: As you create and open activities such as **Scope** and **Assign** for the first time, the message `Invalid Settings` can appear at the top. This is because you have not yet entered details. You can ignore this message. After you enter and apply your details, the message disappears.

Note: As you drag and drop activities, create partner links, create variables, and so on, the project files that display in the **Applications Navigator** section are also updated. If you want, you can click **Source** for the selected project file to view the syntax changes. However, do not manually edit these files.

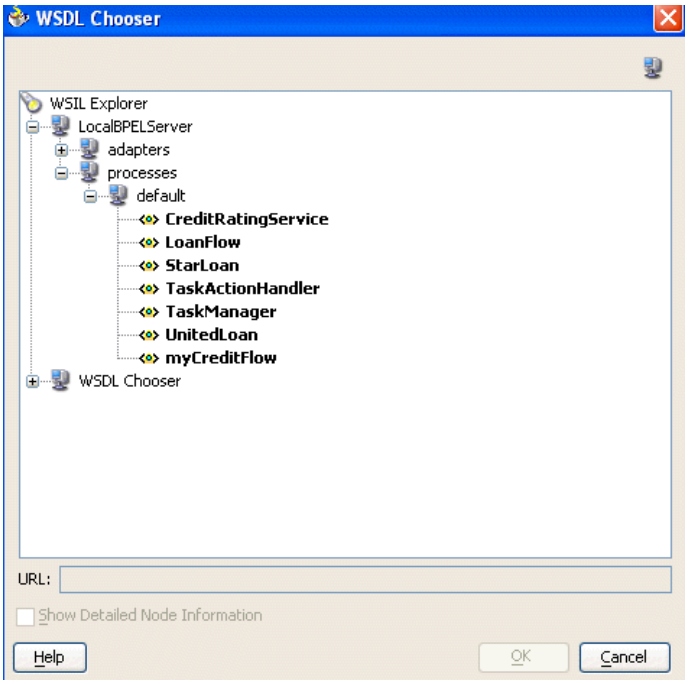
Creating a Credit Rating Service Partner Link

Summary: Partner links define the external services with which your BPEL process interacts. You must create a partner link for the Credit Rating service.

1. Ensure that **Process Activities** is selected in the drop-down list of the **Component Palette** section.

2. Drag and drop a **PartnerLink** activity onto the right side of JDeveloper BPEL Designer.
3. Enter the following values to create a partner link for the Credit Rating service:

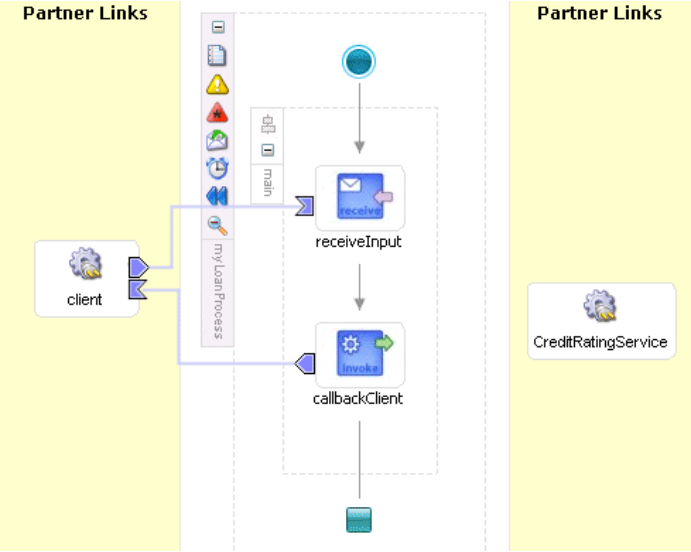
> **Note:** For the **WSDL File** field below, click the **flashlight** (the second icon from the left named **WSIL Browser**) to access the WSDL Chooser window shown below for automatically selecting the Credit Rating service deployed in "Starting and Testing Your Services" on page 3-3.



Field	Value
Name	**CreditRatingService**
WSDL File	Access this URL by clicking the **WSIL Browser flashlight** icon and expanding and selecting **LocalBPELServer > processes > default > CreditRatingService**. **http://localhost:9700/orabpel/default/CreditRatingService/CreditRatingService?wsdl** **See Also:** "Setting the Hostname in Your Web Browser Preferences" on page 1-10 if you receive a parsing error when attempting to add a WSDL file in the WSDL Chooser window.
Partner Link Type	**CreditRatingService**
My Role	Leave unspecified.
Partner Role	**CreditRatingServiceProvider**

4. Click **OK**.

   The **CreditRatingService** partner link now appears on the right side of JDeveloper BPEL Designer.



Creating a Scope Activity

Summary: You now create a **Scope** activity. A **Scope** activity consists of a collection of activities that can have their own local variables, fault handlers, and so on. A **Scope** activity is analogous to a { } block in a programming language.

Within this **Scope** activity, the client's credit rating background is identified.

1. Go to the **Component Palette** section in the upper right part of JDeveloper BPEL Designer.
2. Drag and drop a **Scope** activity between the **receiveInput** and **callbackClient** activities that were automatically created with your BPEL project.
3. Double-click the **Scope** activity to display the Scope window.
4. Enter **GetCreditRating** in the **Name** field of the **General** tab.
5. Click **OK**.
6. Click the **+** sign to expand the GetCreditRating **Scope** activity.
7. Select **Save** from the **File** main menu.

Creating an Invoke Activity Inside the Scope Activity

Summary: You create an **Invoke** activity in this section. An **Invoke** activity enables you to specify the operation you want to invoke for the service (identified by its partner link). You can also create variables in an **Invoke** activity.

This **Invoke** activity provides the initial interaction operation between the client and the Credit Rating service.

1. Drag and drop an **Invoke** activity from the **Component Palette** section into the **GetCreditRating Scope** activity.
2. Double-click the **Invoke** icon to display the Invoke window.

3. Enter the following details:

Field	Value
Name	**invokeCR**
Partner Link	**CreditRatingService**

The **Operation** (**process**) field is automatically filled in.

4. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.

5. Click **OK** on the Create Variable window that appears.

   A variable named **invokeCR_process_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **CreditRatingServiceRequestMessage**.

6. Click the first icon to the right of the **Output Variable** field.

7. Click **OK** on the Create Variable window that appears.

   A variable named **invokeCR_process_OutputVariable** is automatically created in the **Output Variable** field. This variable is automatically assigned a message type of **CreditRatingServiceResponseMessage**.

8. Click **OK**.

9. Select **Save** from the **File** main menu.

Creating an Initial Assign Activity Inside the Scope Activity

Summary: In this section you create the first of several **Assign** activities in this tutorial. An **Assign** activity provides a method for simple data manipulation, such as copying the contents of one variable to another.

This **Assign** activity takes the client's social security number as input and contacts the Credit Rating service to assign a credit rating.

1. Drag and drop an **Assign** activity from the **Component Palette** section to above the **invokeCR Invoke** activity and inside the **Scope** activity.

2. Double-click the **Assign** icon to display the Assign window.

3. Enter **assignSSN** in the **Name** field of the **General** tab.

4. Click **Apply**.

5. Click the **Copy Rules** tab.

6. Click **Create** to display the Create Copy Rule window.

7. Enter the following details:

Field	Value
From	

Field	Value
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **inputVariable** > **payload** > **ns1:loanApplication** > **ns1:SSN** **Note:** The namespace number values (for example, **ns1**, **ns2**) can vary. Use the namespace values that automatically appear.
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **invokeCR_process_InputVariable** > **payload** > **ns2:ssn**

8. Click **OK** to close the Create Copy Rule window and the Assign window.
9. Select **Save** from the **File** main menu.

Creating a Second Assign Activity Inside the Scope Activity

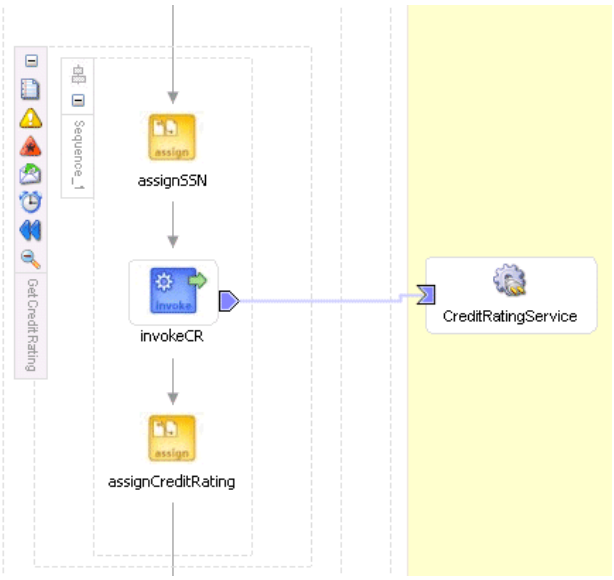
Summary: This **Assign** activity returns details about the client's assigned credit rating. If the credit rating query did not identify a bad credit history (such as a bankruptcy), the client's loan application request document is ready to be sent to the Star Loan and United Loan service providers.

1. Drag and drop a second **Assign** activity from the **Component Palette** section to below the **invokeCR Invoke** activity and inside the **Scope** activity.
2. Double-click the **Assign** icon to display the Assign window.
3. Enter **assignCreditRating** in the **Name** field of the **General** tab.
4. Click **Apply**.
5. Click the **Copy Rules** tab.
6. Click **Create** to display the Create Copy Rule window.
7. Enter the following details:

Field	Value
From	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **InvokeCR_process_OutputVariable** > **payload** > **ns2:rating** **Note:** The namespace number values (for example, **ns1**, **ns2**) can vary. Use the namespace values that automatically appear.
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **inputVariable** > **payload** > **ns1:loanApplication** > **ns1:creditRating**

8. Click **OK** to close the Create Copy Rule window and the Assign window.

   The **Scope** activity (when expanded) now displays the following details:



9. Click the **-** sign to close the **GetCreditRating Scope** activity.
10. Select **Save** from the **File** main menu.

Creating and Configuring Partner Links for the Asynchronous United Loan and Star Loan Services

You now create and configure partner links for the asynchronous Star Loan and United Loan Rating service loan providers.

This section contains these tasks:

- Creating Partner Links for the United Loan and Star Loan Services
- Creating a Second Scope Activity
- Adding a Flow Activity to the Second Scope
- Creating Invoke and Receive Activities for the Star Loan Service
- Creating Invoke and Receive Activities for the United Loan Service
- Creating an Assign Activity

Caution: Do not include any special characters in activity or element names (such as periods). If you do include special characters, errors appear when you attempt to compile your project.

Creating Partner Links for the United Loan and Star Loan Services

Summary: You now create partner links for the United Loan service and Star Loan service loan providers. Star Loan and United Loan are asynchronous services that support a one-way initiate operation and an asynchronous onResult callback that is invoked when a loan offer is available from each service. This can take anywhere from a few seconds to a few days or more to process. Since the loan providers can be long-running, they are implemented as asynchronous services. The BPEL process invokes them in parallel.

1. Drag and drop a **PartnerLink** activity from the **Component Palette** section onto the *right* side of JDeveloper BPEL Designer.
2. Enter the following values to create a partner link for the United Loan provider:

> **Note:** For the **WSDL File** field below, click the **flashlight** (the second icon from the left named **WSIL Browser**) to access the WSDL Chooser window for automatically selecting the United Loan service deployed in "Starting and Testing Your Services" on page 3-3.

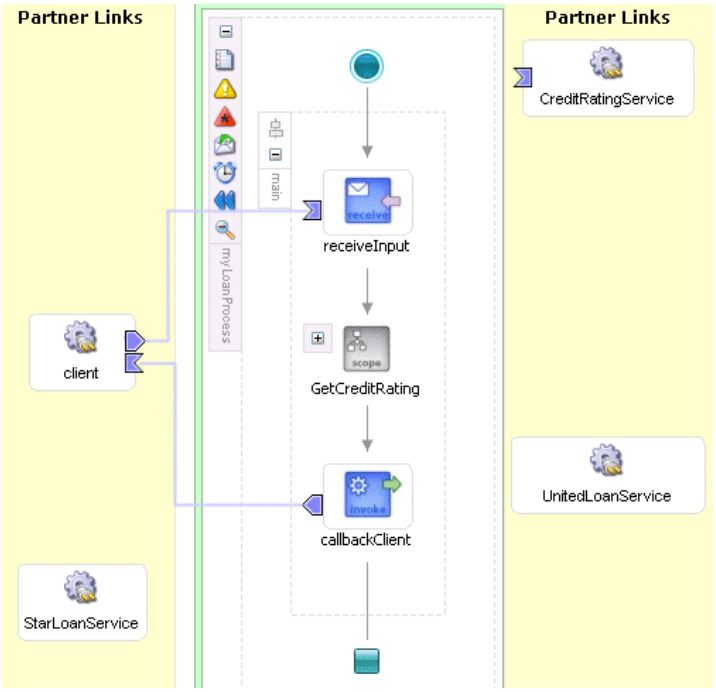
Field	Value
Name	**UnitedLoanService**
WSDL File	Access this URL by clicking the **WSIL Browser flashlight** icon and expanding and selecting **LocalBPELServer > processes > default > UnitedLoan**. **http://localhost:9700/orabpel/default/UnitedLoan/UnitedLoan?wsdl**
Partner Link Type	**LoanService**
My Role	**LoanServiceRequester**
Partner Role	**LoanServiceProvider**

3. Click **OK**.
4. Drag another **PartnerLink** activity from the **Component Palette** section. However, drop it onto the *left* side of JDeveloper BPEL Designer this time. This is not a requirement. However, it makes it easier to view the BPEL process interaction with the three partner links in JDeveloper BPEL Designer when you are done.
5. Enter the following values to create a partner link for the Star Loan provider:

Field	Value
Name	**StarLoanService**
WSDL File	Access this URL by clicking the **WSIL Browser flashlight** icon and expanding and selecting **LocalBPELServer** > **processes** > **default** > **StarLoan**. **http://localhost:9700/orabpel/default/StarLoan/StarLoan?wsdl**
Partner Link Type	**LoanService**
My Role	**LoanServiceRequester**
Partner Role	**LoanServiceProvider**

6. Click **OK**.

   The three partner links now appear as follows in JDeveloper BPEL Designer.



7. Select **Save** from the **File** main menu.

> **Summary:** If you accidently created the Star Loan and United Loan partner links on the same side of JDeveloper BPEL Designer, right-click the **StarLoanService** partner link, and select **Move To Opposite Swim Lane**.

Creating a Second Scope Activity

> **Summary:** Within this **Scope** activity, the client's loan application document is submitted to the Star Loan and United Loan service providers. Both loan providers submit offers, of which the best offer is returned to the client.

1. Drag and drop a second **Scope** activity from the **Component Palette** section to below the **GetCreditRating Scope** activity.
2. Double-click the **Scope** activity.
3. Enter **GetLoanOffers** in the **Name** field of the **General** tab.
4. Click **OK**.
5. Select **Save** from the **File** main menu.

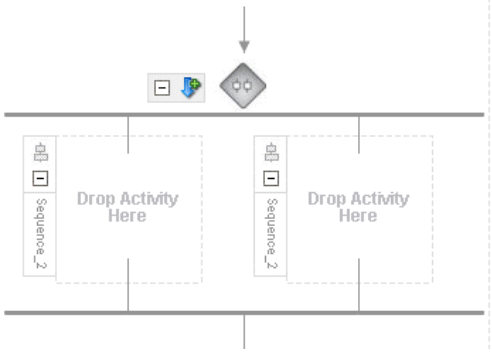
Adding a Flow Activity to the Second Scope

Summary: You now create a **Flow** activity into which you place **Invoke** and **Receive** activities for both the Star Loan and United Loan providers. A **Flow** activity is required to invoke the two loan providers in parallel. This is because they can each take an arbitrary amount of time to complete their loan processes. A **Flow** activity enables several actions to be taken in parallel. In this case, the **Flow** activity contains two parallel activities – the sequence to invoke the United Loan service and the sequence to invoke the Star Loan service.

1. Click the **+** sign to expand the **GetLoanOffers Scope** activity.
2. Drag and drop a **Flow** activity from the **Component Palette** section into the **GetLoanOffers Scope** activity.
3. Double-click the diamond to display the Flow window.

4. Enter **FlowInvokeProviders**.
5. Click **OK**.

   JDeveloper BPEL Designer now looks as follows:



   Note that the **FlowInvokeProviders** name does not appear when the **Flow** activity is expanded. You must click the - sign to display the **FlowInvokeProviders** name.

6. Select **Save** from the **File** main menu.

Creating Invoke and Receive Activities for the Star Loan Service

Summary: This **Invoke** activity invokes the operation between the client's loan application document and the Star Loan service loan provider.

This **Receive** activity returns the loan offer results from the Star Loan service loan provider.

1. Drag and drop an **Invoke** activity into the *left* side of the **Flow** activity.
2. Double-click the **Invoke** icon to display the Invoke window.
3. Enter the following details:

Field	Value
Name	**invokeStarLoan**
Partner Link	**StarLoanService**

The **Operation** (**initiate**) field is automatically filled in.

> **Note:** The **Output Variable** field is disabled. This is because only a single input variable is received.

4. Click the first icon to the right of the **Input Variable** field. This is the automatic variable creation icon.
5. Click **OK** on the Create Variable window that appears.

   A variable named **invokeStarLoan_initiate_InputVariable** is automatically created in the **Input Variable** field. This variable is automatically assigned a message type of **LoanServiceRequestMessage**.
6. Click the **flashlight** (the second icon) to the right of the **Input Variable** field to display the Variable Chooser window.
7. Right-click **invokeStarLoan_initiate_InputVariable**.
8. Select **Edit Variable**.
9. Rename it to **loanApplication**. You perform this task because the **loanApplication** variable is passed to **Receive** activities for both the Star Loan and United Loan services. The best offer is determined and returned to the client.
10. Click **OK** to close the Edit Variable window, Variable Chooser window, and Invoke window.
11. Drag and drop a **Receive** activity below the **invokeStarLoan Invoke** activity you just created.
12. Double-click the **Receive** icon to display the Receive window.
13. Enter the following details:

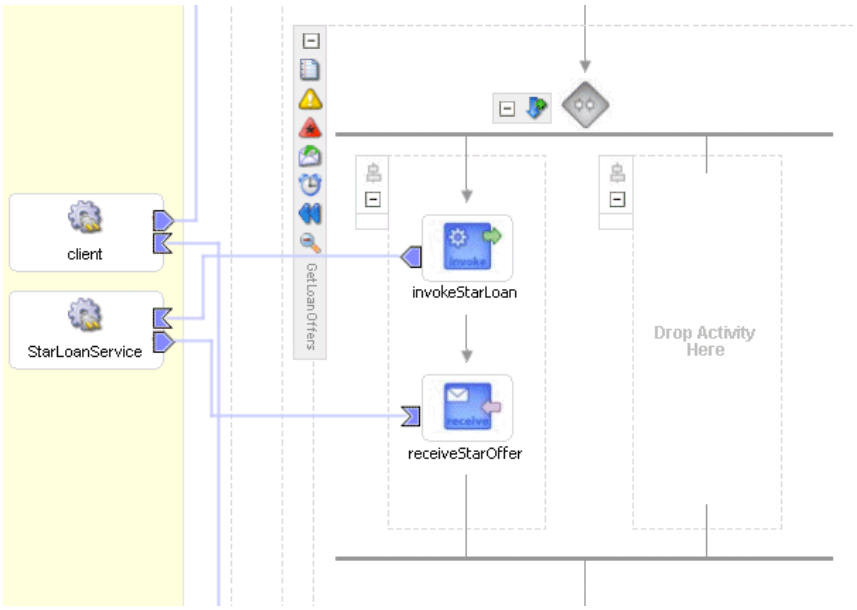
Field	Value
Name	**receiveStarOffer**
Partner Link	**StarLoanService**
Create Instance	Leave this box unchecked.

The **Operation** (**onResult**) field is automatically filled in.

14. Click the first icon to the right of the Variable field.
15. Click **OK** on the Create Variable window that appears.

    A variable named **receiveStarOffer_onResult_InputVariable** is automatically created in the **Variable** field. This variable is automatically assigned a message type of **LoanServiceResultMessage**.
16. Click **OK**.
17. Select **Save** from the **File** main menu.

The Flow activity now looks as follows:



Creating Invoke and Receive Activities for the United Loan Service

Summary: This **Invoke** activity invokes the operation between the client's loan application document and the United Loan service loan provider.

This **Receive** activity returns the loan offer results from the United Loan service loan provider.

1. Drag and drop an **Invoke** activity into the *right* side of the **Flow** activity where it says **Drop Activity Here**.
2. Double-click the **Invoke** icon to display the Invoke window.
3. Enter the following details:

Field	Value
Name	**invokeUnitedLoan**
Partner Link	**UnitedLoanService**

The **Operation** (**initiate**) field is automatically filled in.

Note: The **Output Variable** field is disabled. This is because only a single variable is received.

4. Click the **flashlight** icon (the second icon) to the right of the **Input Variable** field. This displays the Variable Chooser window. We reuse the **loanApplication** variable for this activity.
5. Select **loanApplication**.
6. Click **OK** to close the Variable Chooser window and Invoke window.

7. Drag and drop a **Receive** activity below the invokeUnitedLoan **Invoke** activity you just created.
8. Double-click the **Receive** icon to display the Receive window.
9. Enter the following details:

Field	Value
Name	**receiveUnitedOffer**
Partner Link	**UnitedLoanService**
Create Instance	Leave this box unchecked.

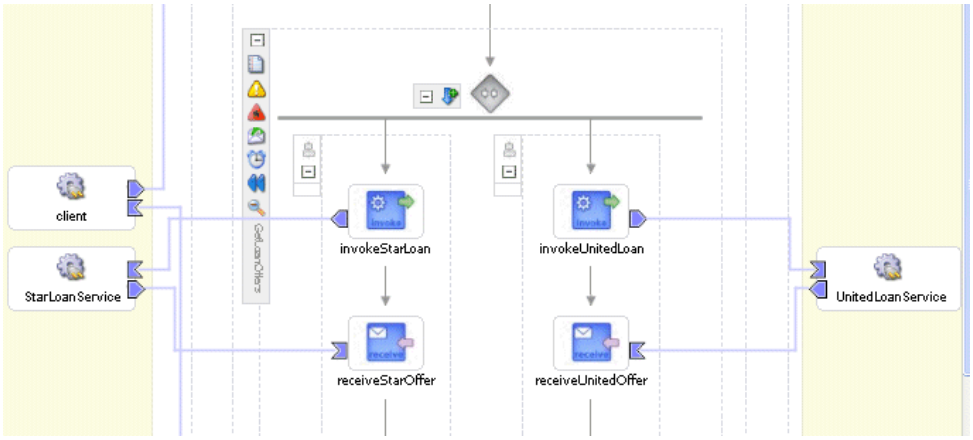
The **Operation** (**onResult**) field is automatically filled in.

10. Click the first icon to the right of the **Variable** field.
11. Click **OK** on the Create Variable window that appears.

    A variable named **receiveUnitedOffer_onResult_InputVariable** is automatically created in the **Variable** field. This variable is automatically assigned a message type of **LoanServiceResultMessage**.

12. Click **OK**.
13. Select **Save** from the **File** main menu.

    The **Flow** activity now displays the following details. This activity passes the loan application document as an input variable to the initiate operations for the United Loan and Star Loan services. These initiate operations return immediately, but the next activity, the **Receive** activity for the **onResult** callback, waits until the service has called back with a loan offer.



Summary: A dehydration point is established between the **Invoke** and **Receive** activities. Dehydration enables long-running, asynchronous flows to be automatically maintained in a database while they wait for asynchronous callbacks. This enables flows to be reliably and persistently stored in a database, along with their current state information, whenever they are waiting for asynchronous events.

BPEL also enables support for correlating asynchronous messages so that asynchronous callbacks can locate the appropriate waiting process instance. By default, WS-Addressing, which uses simple object access protocol (SOAP) message headers for correlation of asynchronous messages, is used for message content correlation with the correct instance.

Creating an Assign Activity

Summary: This **Assign** activity takes the client's loan application document and sends it to the Star Loan and United Loan service loan providers.

1. Drag and drop an **Assign** activity from the **Component Palette** section inside the GetLoanOffers **Scope** activity and above the **Flow** activity (above the diamond).
2. Double-click the **Assign** activity.
3. Enter **initializeLoanApplication** in the **Name** field of the **General** tab.
4. Click **Apply**.
5. Click the **Copy Rules** tab.
6. Click **Create** to display the Create Copy Rule window.
7. Enter the following details:

Field	Value
From	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **inputVariable** > **payload**
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **loanApplication** > **payload**

8. Click **OK** to close the Create Copy Rule window and the Assign window.
9. Click all **-** signs to close the **Flow** activity and **Scope** activity. When all are closed, your BPEL process flow displays the following details:

10. Select **Save** from the **File** main menu.

Adding Offer Decision Making Logic to the Loan Process

Summary: The **Flow** activity enables your flow to gather both loan offers at the same time, but does not compare any of the offers. To compare the two offers and make decisions based on that comparison, the BPEL flow requires a **Switch** activity. A **Switch** activity includes two branches: **<case>** and **<otherwise>**. The first branch is executed if a defined condition (inside the **<case>** branch) is met. If it is not met, the **<otherwise>** branch is executed.

You now add a **Switch** activity to the **GetLoanOffers Scope** activity.

1. Click the + sign to expand the **GetLoanOffers Scope** activity.
2. Drag and drop a **Switch** activity from the **Component Palette** section into the **GetLoanOffers Scope** activity and below the **FlowInvokeProviders Flow** activity.
3. Double-click the question mark for the **Switch** activity to display the Switch window.
4. Enter **GetBestOffer** in the **Name** field.
5. Click **OK**.
6. Double-click **<case>** to display the Switch Case window.
7. Press **Ctrl** and then the space bar in the **Expression** field to display a list for selecting (double-clicking) the following syntax. Edit as necessary. As you enter information, a trailing slash can appear. This means you are being prompted for additional information. Either enter additional information, or press the **Esc** key and delete the trailing slash to complete the input of information.

```
bpws:getVariableData('receiveStarOffer_onResult_
InputVariable','payload','/ns1:loanOffer/ns1:APR') >
bpws:getVariableData('receiveUnitedOffer_onResult_
InputVariable','payload','/ns1:loanOffer/ns1:APR')
```

8. Click **OK**.

Note that the syntax you added does not explicitly display when you view the <case> icon in the flow. (See Step 24 on page 3-21.) If you want to view this syntax, double-click **<case>**. The syntax you entered displays in the Expression field.

9. Drag and drop an **Assign** activity from the **Component Palette** section into the **<case>** section of this Switch activity.

   Summary: This **Assign** activity takes the United Loan offer and, if it is the best offer, submits it to the client. The syntax you defined in the **Expression** field in Step 7 on page 3-19 determines the best offer.

10. Double-click the **Assign** activity.
11. Enter **PickUnited** in the **Name** field of the **General** tab.
12. Click **Apply**.
13. Click the **Copy Rules** tab.
14. Click **Create** to display the Create Copy Rule window.
15. Enter the following details:

Field	Value
From	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **receiveUnitedOffer_onResult_ InputVariable** > **payload**
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **outputVariable** > **payload**

16. Click **OK** to close the Create Copy Rule window and the Assign window.
17. Drag and drop an **Assign** activity from the **Component Palette** section into the **<otherwise>** section of this **Switch** activity.

    Summary: This **Assign** activity takes the Star Loan offer and, if it is the best offer, submits it to the client. The syntax you defined in the **Expression** field in Step 7 on page 3-19 determines the best offer.

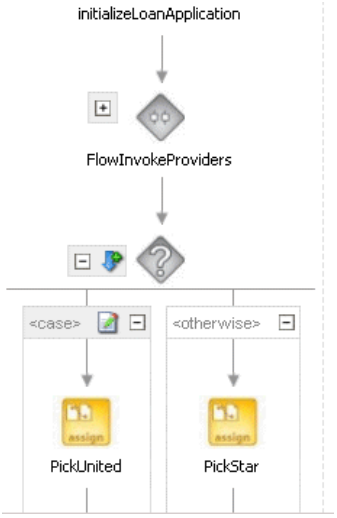
18. Double-click the **Assign** activity.
19. Enter **PickStar** in the **Name** field of the **General** tab.
20. Click **Apply**.
21. Click the **Copy Rules** tab.
22. Click **Create** to display the Create Copy Rule window.
23. Enter the following details:

Field	Value
From	
▪ **Type**	**Variable**

Field	Value
▪ **Variables**	Expand and select **Variables** > **receiveStarOffer_onResult_ InputVariable** > **payload**
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **outputVariable** > **payload**

24. Click **OK** to close the Create Copy Rule window and the Assign window.

    The **Switch** activity displays the following details:



25. Select **Save** from the **File** main menu.

    You have now completed the loan process design and are ready to proceed to the validation, compilation, and deployment tasks described in the following section.

Validating, Compiling, and Deploying the Loan Process

You are now ready to deploy your BPEL process.

1. Go to the **Applications Navigator** section.
2. Right-click **myLoanProcess**.
3. Select **Deploy** > **LocalBPELServer** > **Deploy to default domain**.
4. Enter **bpel** when prompted for the domain password.
5. Click **OK**.

   This compiles the BPEL process. Review the bottom of the window for any errors. If there are no errors, the following message appears:

```
[3:43:31 PM] Successful compilation: 0 errors, 0 warnings.
Deploying to http://localhost:9700 domain: default. Please wait ...
            bpel_myLoanProcess_1.0.jar deployed successfully.
```

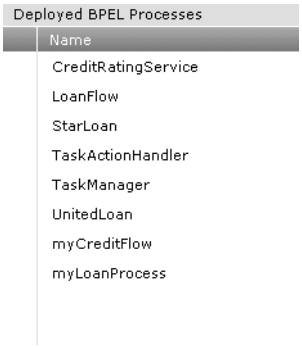
6. If there are errors, click **BPEL Validation Errors** to display details about the type and location of the error.
7. Make corrections and deploy again.

Running the Loan Process

You are now ready to begin the process of applying for and receiving a loan offer.

1. Log into Oracle BPEL Console by selecting **Start** > **All Programs** > **Oracle - *Oracle_Home*** > **Oracle BPEL Process Manager 10.1.2** > **BPEL Console**.
2. Enter **bpel** as the password when prompted.

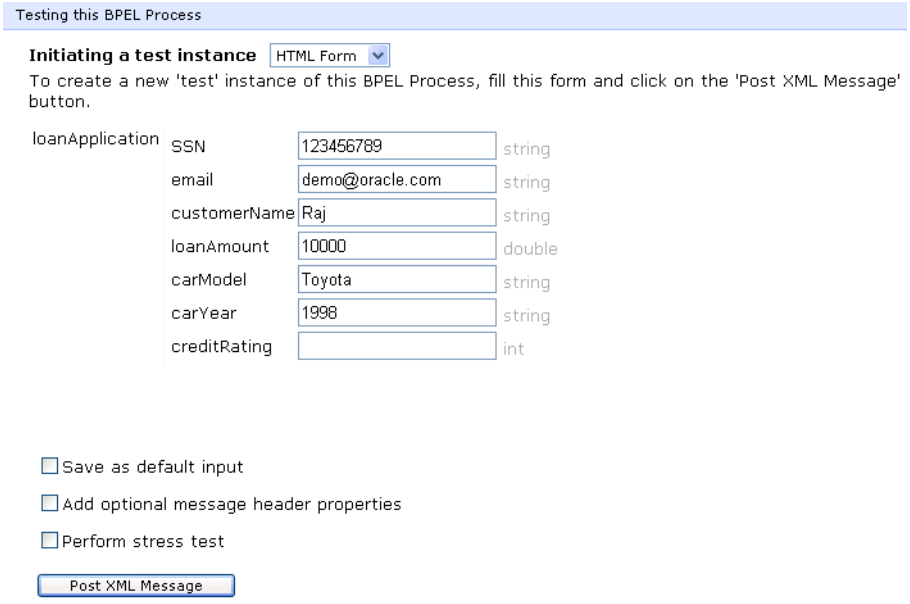
   The **Dashboard** tab of Oracle BPEL Console appears. These are the services that you started and verified in Step 5 of "Starting and Testing Your Services" on page 3-3. Note that your BPEL process, **myLoanProcess**, now appears in the **Deployed BPEL Processes** list.



3. Click **myLoanProcess** in the **Deployed BPEL Processes** list.

   The loan application form appears. Note that the form includes a social security number field. At the beginning of the execution of this process, the social security number you enter is retrieved and the Credit Rating service is contacted to request a credit rating for you.

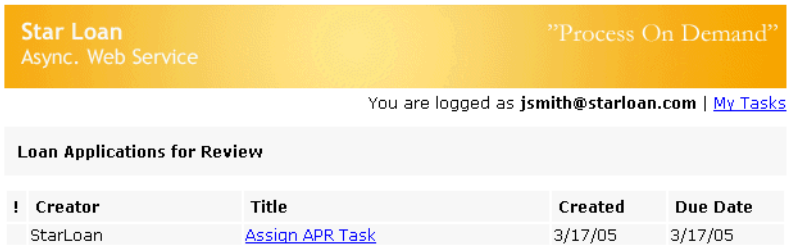
4. Complete the loan application form. Ensure that you enter a social security number that does *not* begin with zero in the **SSN** field of the HTML Form and click **Post XML Message**.



5. Go to the following URL for the Star Loan service loan provider to enter an APR value:

```
http://localhost:9700/StarLoanUI
```

The loan approval typically takes several minutes to display. Refresh as necessary.



6. Click **Assign APR Task**.
7. Enter an APR value that is more than 5.75% or less than 5.7% in the **APR** field.
8. Click **Approve**.

   The following message appears:

   ```
   Assign APR Task has been completed
   ```

9. Return to Oracle BPEL Console.
10. Click the **Instances** tab to access details about this running process instance.
11. Click the latest instance of **myLoanProcess** in the **Instance** column.

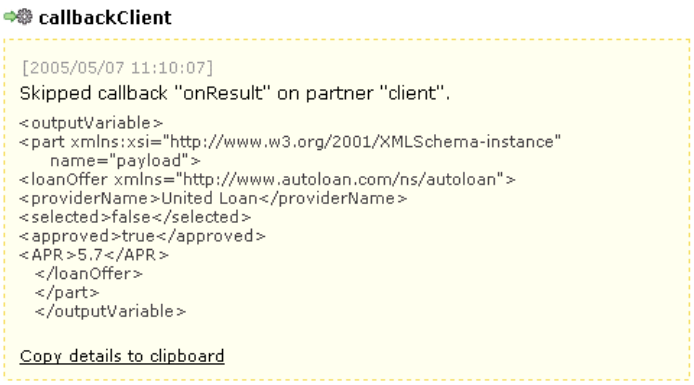
   A message indicates that the state of the instance is completed.

12. Click the **Flow** link.

    A visual representation of the history of this process instance flow appears. This represents a view of the current state and history of the execution of this process flow instance. You can select activities to view details.

13. Scroll through the flow.

14. Continue scrolling and click the **callbackClient** activity at the end of the flow to see the results of the loan offer, including the best credit rating returned by the Credit Rating service to the client.



You have now completed running the loan process. You can continue to explore the features of Oracle BPEL Console.

Adding Error Handling Capabilities to the Loan Process (Optional)

Now that you have designed, validated, compiled, deployed, and run a process, you can add optional error handling functionality to handle rejections (for example, due to a bankruptcy).

This section contains these tasks:

- Creating a Fault Handling Error Variable
- Creating a Catch Branch in the GetCreditRating Scope Activity
- Creating an Assign Activity Inside the Catch Branch of the Scope Activity
- Creating an Invoke Activity Inside the Catch Branch of the Scope Activity
- Creating a Terminate Activity Inside the Catch Branch of the Scope Activity

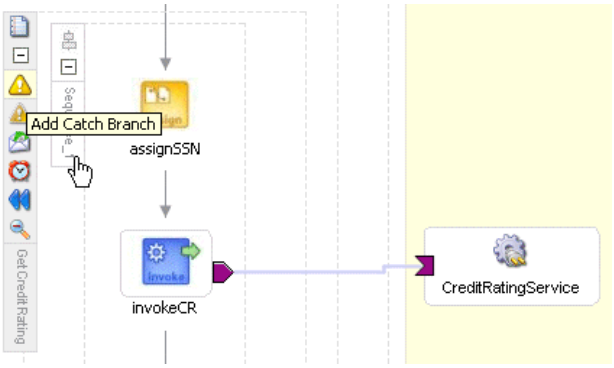
Creating a Fault Handling Error Variable

Summary: You create fault handling to catch and manage exceptions identified by the credit rating service. Fault handlers are associated with a particular **Scope** activity and faults encountered within a scope are passed up to this enclosing **Scope** activity.

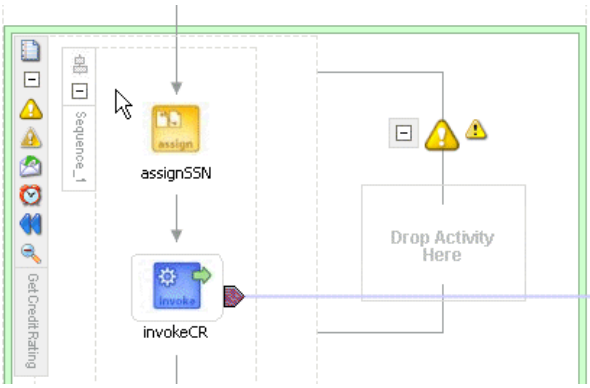
1. Go to the **Structure** section of JDeveloper BPEL Designer.
2. Select **Variables** > **Process** > **Variables**.
3. Right-click **Variables** and select **Create Variable** to display the Create Variable window.
4. Enter **crError** in the **Name** field.
5. Select **Message Type** and click the **flashlight** to display the Type Chooser window.
6. Select **Message Types** > **Partner Links** > **CreditRatingService** > **CreditRatingService** > **Message Types** > **CreditRatingServiceFaultMessage**.
7. Click **OK** to close the Type Chooser window and Create Variable window.
8. Select **Save** from the **File** main menu.

Creating a Catch Branch in the GetCreditRating Scope Activity

1. Click the **+** sign to expand the **GetCreditRating Scope** activity.
2. Click **Add Catch Branch** near the top of the icons (fourth icon) on the left side of **GetCreditRating**.



A new box displays to the right with an exclamation point and the words **Drop Activity Here**.



3. Double-click the exclamation point inside the new box to display the Catch window.
4. Enter the following details. Note that you assign the **crError** variable that you previously created.

Field	Description
Namespace URI	**http://services.otn.com** Note: This URI is the Credit Rating service's namespace. If you want to see where this namespace URI comes from, perform the following steps: 1. Copy the URI for the Credit Rating service **WSDL File** field entered in Step 3 on page 3-7. 2. Paste this value into a Web browser. 3. View the **targetnamespace** element value in the URI that appears.
Local Part	**NegativeCredit** **Note:** This is the **fault name** element value that displays in the above URI.
Fault Variable	**crError**

5. Click **OK**.
6. Select **Save** from the **File** main menu.

Creating an Assign Activity Inside the Catch Branch of the Scope Activity

> **Summary:** The client initially provided a social security number to the Credit Rating service for a credit check to occur. If a bad credit history is identified or the social security number is invalid, this **Assign** activity notifies the client of the loan offer rejection. The entire loan application process is terminated, and the client's loan application document is never submitted to the Star Loan and United Loan service loan providers.

1. Drag and drop an **Assign** activity from the Component Palette section into the **ns2:NegativeCredit Catch** branch you just created. Note that **ns2:** automatically appears.
2. Double-click the **Assign** icon to display the Assign window.
3. Enter **RejectApplication** in the **Name** field of the **General** tab.
4. Click **Apply**.
5. Click the **Copy Rules** tab.
6. Click **Create** to display the Create Copy Rule window. You now create three fault handling rules.
7. Enter the following values:

Note: If you are entering an expression, you can press **Ctrl** and then the space bar in the **Expression** field instead of manually entering details. Scroll through the list of values that appears and double-click the value you want. Edit the value as necessary. As you enter information, a trailing slash can appear. This means you are being prompted for additional information. Either enter additional information, or press the Esc key and delete the trailing slash to complete the input of information.

Field	Value
From	
▪ **Type**	**Expression**
▪ **Expression**	**string('None - rejected because of bad credit')**
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **outputVariable** > **payload** > **ns1:loanOffer** > **ns1:providerName** **Note:** The namespace number values (for example, **ns1**, **ns2**) can vary. Use the namespace values that automatically appear.

8. Click **OK**.
9. Repeat Step 6 through Step 8 to add a second copy expression for the **Assign** activity with the following values:

Field	Value
From	
▪ **Type**	**Expression**
▪ **Expression**	**false()**
To	
▪ **Type**	**Variable**
▪ **Variables**	Expand and select **Variables** > **outputVariable** > **payload** > **ns1:loanOffer** > **ns1:selected** **Note:** The namespace number values (for example, **ns1**, **ns2**) can vary. Use the namespace values that automatically appear.

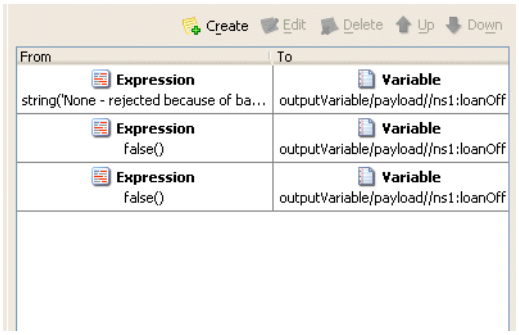
10. Click **OK**.
11. Repeat Step 6 through Step 8 to add a third copy expression to the **Assign** activity with the following values:

Field	Value
From	
▪ **Type**	**Expression**
▪ **Expression**	**false()**
To	

Field	Value
▪ **Type**	**Variable**
▪ **Variables**	**outputVariable** > **payload** > **ns1:loanOffer** > **ns1:approved** **Note:** The namespace number values (for example, **ns1**, **ns2**) can vary. Use the namespace values that automatically appear.

12. Click **OK** to close the window.

    The **Assign** activity now displays the copy rules.



13. Click **OK**.
14. Select **Save** from the **File** main menu.

Creating an Invoke Activity Inside the Catch Branch of the Scope Activity

1. Drag and drop an **Invoke** activity from the **Component Palette** section to below the **RejectApplication Assign** activity you just created.
2. Double-click the **Invoke** icon to display the Invoke window.
3. Enter the following details:

Field	Value
Name	**BadCreditInvokeClient**
Partner Link	**client**
Input Variable	**outputVariable** Note: Click the **flashlight** icon (the second icon to the right of this field) to access the Variable Chooser window for selecting this variable. Then, click **OK**.

The **Operation** (**onResult**) field is automatically filled in.

> **Note:** The **Output Variable** field is disabled. This is because only a single input variable is received.

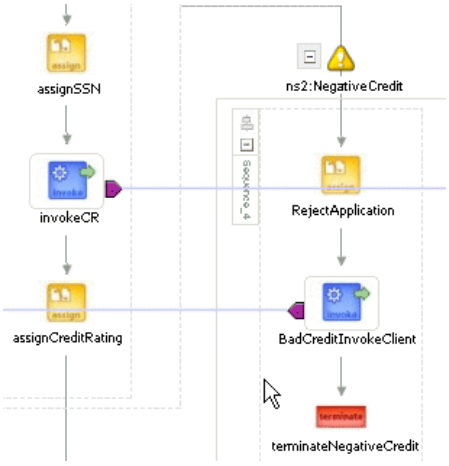
4. Click **OK**.
5. Select **Save** from the **File** main menu.

Creating a Terminate Activity Inside the Catch Branch of the Scope Activity

Summary: A **Terminate** activity enables you to end the fault handling tasks in the **Catch** branch. If a bad credit history is identified or the social security number is invalid, the entire loan application process is terminated, and the client's loan application document is never submitted to the Star Loan and United Loan service loan providers. Entering a social security number that begins with zero in the **SSN** field of the HTML Form executes this bad credit condition.

1. Drag and drop a **Terminate** activity from the **Component Palette** section to below the **BadCreditInvokeClient Invoke** activity you just created.
2. Double-click the **Terminate** activity to display the Terminate window.
3. Name it **terminateNegativeCredit**.
4. Click **OK**.

   The **Catch** branch now displays the following activities.



5. Click the **-** sign to close the **ns2:NegativeCredit Catch** branch.
6. Select **Save** from the **File** main menu.
7. See section "Validating, Compiling, and Deploying the Loan Process" on page 3-21 to run the loan process again with error handling functionality. Note that since you are deploying **myLoanProcess** again, the Deploy Properties window appears. Increment the version number to start a new version of **myLoanProcess** (for example, enter **1.1**). This means you have two **myLoanProcess** versions running: the one you previous deployed and ran, and this new one that tests the fault handling logic.

   When you complete the loan application form, ensure that you enter a social security number *beginning* with zero in the **SSN** field of the HTML Form and click **Post XML Message**. This action activates the fault handling logic.

Index

V

W

X