WebSphere®

Loan Application

Contents

Chapter 1. Sample: Loan Application

This sample demonstrates how to build a loan approval application that customers can access from a Web page.

Learning objectives

In the sample, you will learn how to complete the following tasks:

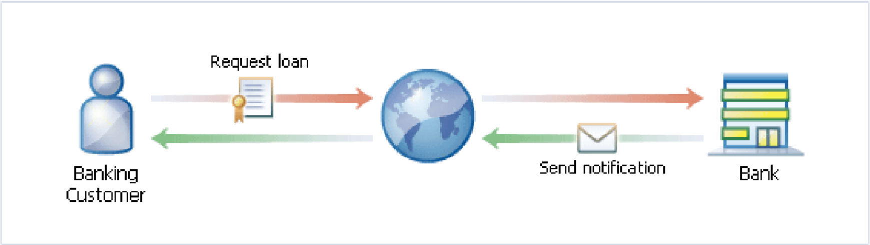
- Use an inline human task to initiate a business process.
- Use business rules to separate dynamic business logic from the business process flow.
- Invoke a separate human task to handle manual processing.
- Generate default Web interfaces for human tasks.
- Deploy and test the process using the generated Web interfaces.

Building the application sample will require approximately one hour. You will need additional time to run and test the application.

Chapter 2. Loan application sample

In this sample, a customer will apply for a loan online. After the bank receives the application, the loan will be approved automatically if the amount is less than or equal to $50000. If the amount is greater than $50000, then the application is forwarded to a loan officer for approval. The customer can view the status of the application using an online interface.

Overall Scenario



Before you start building the Loan application sample, you should understand the following concepts:

- How business objects and interfaces are created
- How business modules are created
- How business processes are built
- How business rules are defined

To refresh your understanding of these concepts, perform the Hello World 1 and Hello World 2 samples before you begin the loan application sample.

Chapter 3. Assembling the application

To build the application you first need to import some predefined artifacts into your WebSphere Integration Developer project.

To start building the sample, you need to import a set of interface and business objects definitions that are predefined for you.

To add the artifacts to your workspace, complete the following steps:

1. Start with a fresh workspace. In WebSphere Integration Developer, ensure the Welcome page is open. If not, use **Help** → **Welcome** to open it.
2. Click the **Samples/Tutorials** icon. The Samples/Tutorials page opens.
3. Under **Loan Application**, click the **Import** link. You are presented with two possible modules to import.
4. Select **Starter Artifacts**. The LoanApplicationLibrary is added to your project.

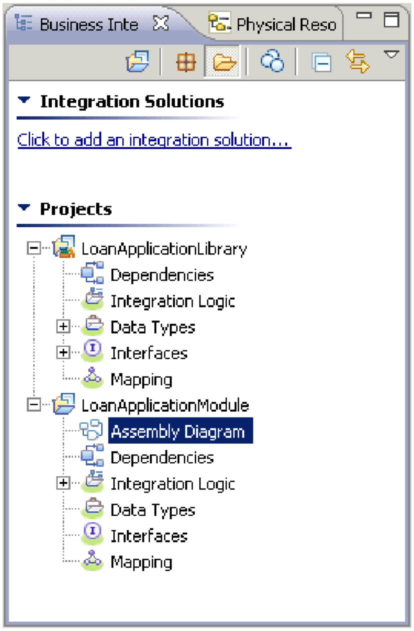
Creating the loan application module

The resources for this sample will be stored in a module project called LoanApplicationModule.

You need to create a module called LoanApplicationModule and set the LoanApplicationLibrary as a resource.

- Click **File** → **New** → **Module**.
- In the **Module name** field enter **LoanApplicationModule** and click **Next**.
- On the **Select Required Libraries** dialog ensure that the **LoanApplicationLibrary** is selected and click **Finish**.

When the module is created, you will see the following in the Business Integration view:



Creating a business process and assigning an interface

Add a process that runs the business logic, which involves receiving a loan request, checking the loan amount for automatic approval, sending a reply, and delegating manual loan approvals to a loan officer, and an interface that customers will use to send their application to the LoanApprovalProcess.

To add the process component and an interface, complete the following steps:

1. In the assembly editor palette, select **Process** and then click the canvas to add **Component1**. The exclamation mark in the lower left corner indicates that the implementation for this component has not yet been created. You will create the implementation for each component later.
2. Rename the component to LoanApprovalProcess.
3. To add an interface, right-click LoanApprovalProcess, select **Add** then **Interface**. Select **LoanApprovalInterface** and click **OK**.

Creating business objects

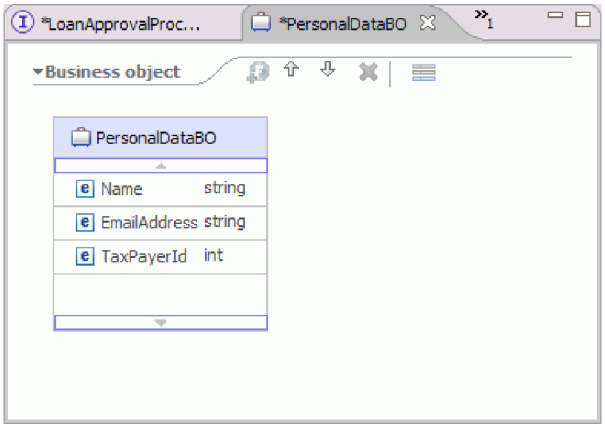
You will create business objects to pass data between components in the module, and variables which are used to hold data.

When the customer requests a loan from the bank, the customer needs to provide a name, an e-mail address and an ID. We will store these attributes in a business object, which will be sent as ApplicantInfo. The business objects act as the currency in your application and are transferred through interfaces to other components.

To organize the loan request data in a meaningful way we will create two business objects: PersonalDataBO and LoanRequestBO

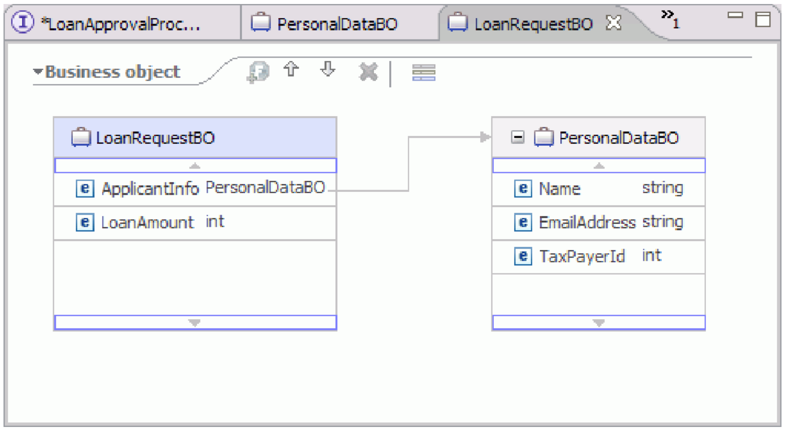
Follow these steps to create the business objects:

1. In the Business Integration view, right-click **Data Types** and click **New** → **Business Object**.
2. In the **Name** field, type `PersonalDataBO`. Leave the **Inherit from** field as `<none>` and click **Finish**. The editor opens for PersonalDataBO.
3. Click , which is the **Add a field to a business object** button and rename `field1` to `Name`. Leave the type as `string`.
4. Add a new field named `EmailAddress` with type `String`, and add another field named `TaxPayerId` Change the type of the `TaxPayerId` field to `int` by clicking the type and selecting `int` from the list. Press **Ctrl-S** to save.



PersonalDataBO	
Name	string
EmailAddress	string
TaxPayerId	int

5. Create another business object named `LoanRequestBO` with two fields: `ApplicantInfo` and `LoanAmount`. Change the type of `ApplicantInfo` to `PersonalDataBO`, and `LoanAmount` to `int`. Save the business object.



LoanRequestBO	
ApplicantInfo	PersonalDataBO
LoanAmount	int

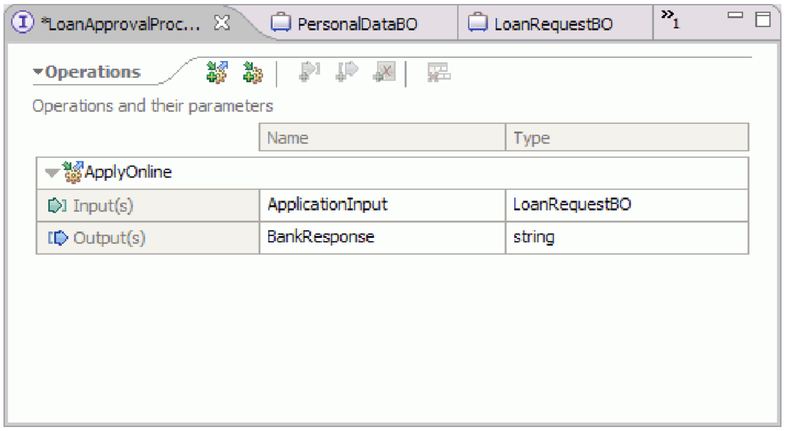
PersonalDataBO	
Name	string
EmailAddress	string
TaxPayerId	int

Update the interface

Now that we have specified the input type the process will accept, we can update the interface.

To update LoanApprovalProcessInterface, follow these steps:

1. Reopen LoanApprovalProcessInterface and change the ApplicationInput type to `LoanRequestBO`.
2. Press **Ctrl-S** to save the interface.



Operations and their parameters

	Name	Type
ApplyOnline		
Input(s)	ApplicationInput	LoanRequestBO
Output(s)	BankResponse	string

Creating components to handle the approval

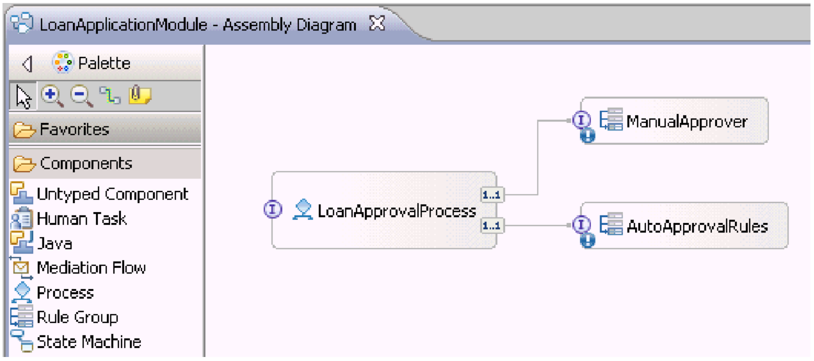
Now that you have a foundation for the process, you will create the components that the process will invoke to check the loan amount for automatic approval and handle the manual approval of the application.

If the loan amount is less than a specified value ($50 000 in our case) the bank wants to approve it automatically. Therefore, you will add a rule group to the assembly diagram. If the loan amount is more than or equal to $50 000, then the bank will want to review the application before making a decision. You will implement the bank's review using a human task.

To add the components to the assembly diagram, complete the following steps:

1. Create a rule group component and assign it the interface AutoApprovalInterface.
2. Rename the rule group component to AutoApprovalRules.
3. Create a human task component and assign the ManualApprovalInterface to it.
4. Rename the human task component to ManualApprover.
5. Create a connection between **LoanApprovalProcess** and **AutoApprovalRules**.
6. Create a connection between **LoanApprovalProcess** and **ManualApprover**.

The assembly editor canvas should be the same as the following screenshot:



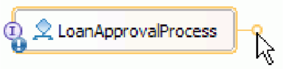
Wiring the components together

To enable components to communicate with each other through the operations defined in their interfaces, you will wire components in the assembly diagram.

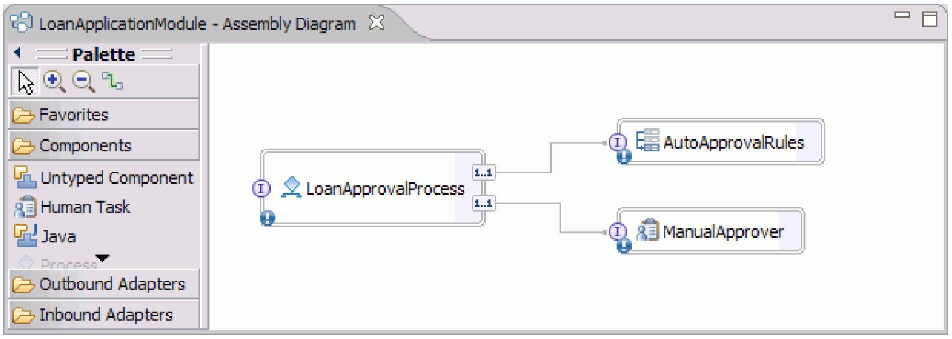
By connecting a source and a target component in the assembly diagram, the source can access the services of the target. The source components call other components via partner references shown on the right side of the component in the assembly diagram, and through operations defined by the target interface(s).

To wire the process to the rule group and human task, follow these steps:

1. Position the mouse over the border of the **LoanApprovalProcess** component until a yellow handle is displayed.



2. Click the yellow handle and drag it to connect with the **AutoApprovalRules** component. The Add Wire dialog box is displayed.
3. To create a new partner reference on **LoanApprovalProcess**, click **OK**. A wire connects the two components through the partner reference on **LoanApprovalProcess** and the interface of **AutoApprovalRules**.
4. Repeat the same steps to connect **LoanApprovalProcess** to the **ManualApprover** human task.

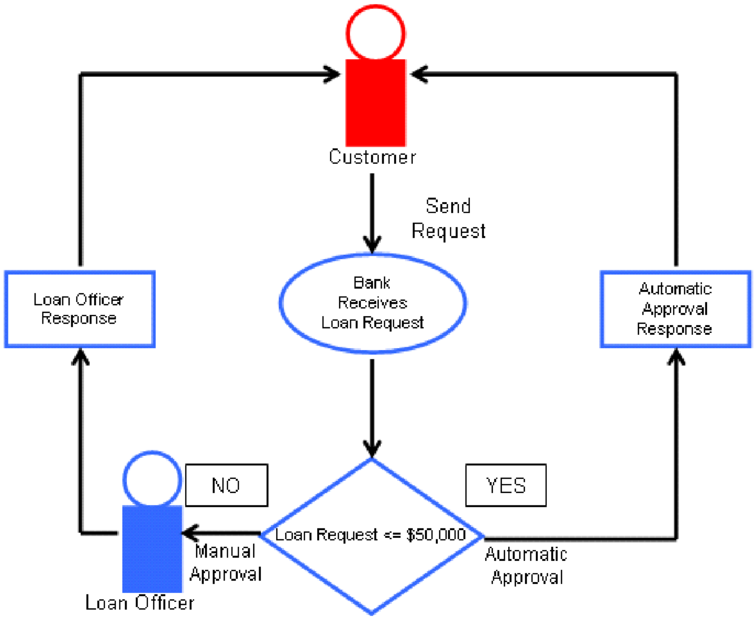


5. Press **Ctrl-S** to save.

Chapter 4. Processing the loan request

Implement the process that receives and checks the loan request.

The following diagram depicts how you will implement the business process for the loan application:



Creating the process implementation

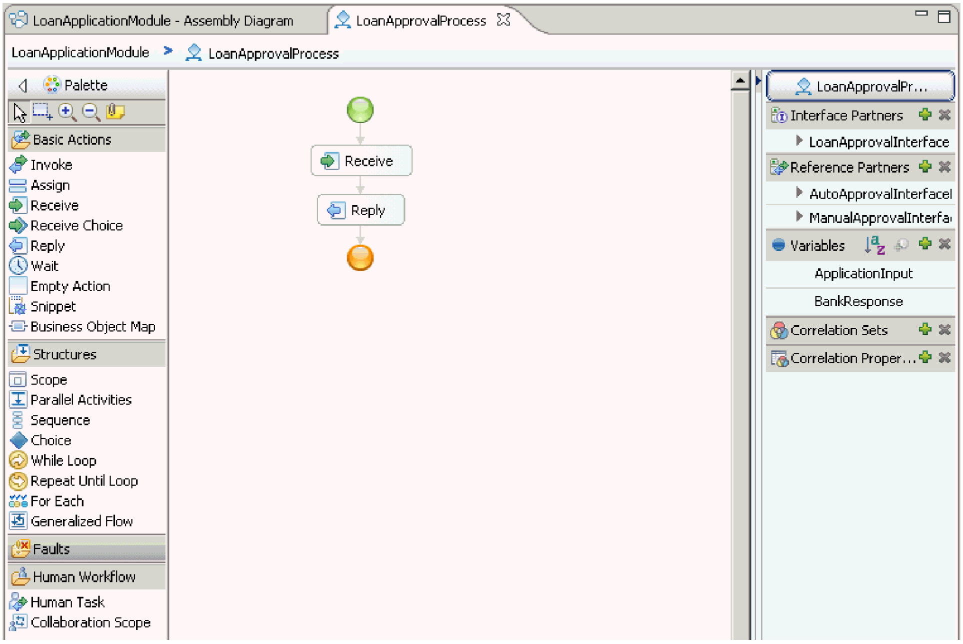
The LoanAprovalProcess is a business process component that contains the business logic for the loan application.

To create and edit the loan application business process which is represented visually as a series of activities, and modeled using Business Process Execution Language (BPEL), you will use the process editor and complete the following steps:

1. Receive the loan request
2. Perform an automatic approval request if the loan amount requested is less than $50 000
3. Manually process the request as a bank employee
4. Send a response to the bank customer

To generate the LoanApprovalProcess component, which is the process in this application, complete the following steps:

1. Right-click the **LoanApprovalProcess** component.
2. From the menu, select **Generate Implementation**. The Generate Implementation window opens.
3. In the navigation tree, click **LoanApplicationModule** and click **OK**. The process editor opens showing a basic business process.



Defining variables

Create the variables that will store the data that is exchanged between the LoanApprovalProcess component and the other components it communicates with.

The LoanApprovalProcess BPEL process uses the ApplicationInput and BankResponse variables to interact with the customer.

To define the variables used in the LoanApprovalProcess implementation, follow the steps below.

1. Under Variables, click the plus sign to add a global variable. Name the variable `AutoApprovalData` and select LoanRequestBO as the data type. Click **OK**. A variable is added to the Variables list.
2. Repeat the above steps for the following variables:

Name	Data Type	Purpose (function)
AutoApprovalResponseData	boolean	Holds the response from the bank resulting from an automatic loan approval.
ManualApprovalData	ManualApprovalBO	Holds the loan application data and instruction passed to the loan officer.

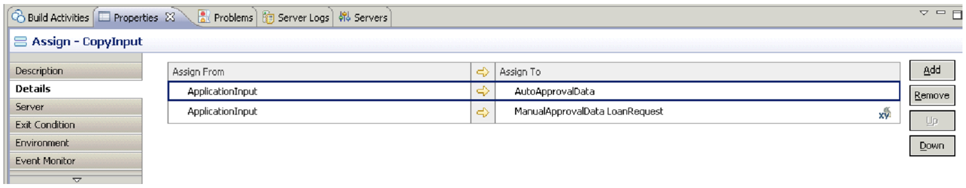
3. Save the business process.

Copying loan request information

Assign the attributes of the received ApplicationInput to variables so that you can invoke operations from within the process using the data that is stored in the variables as input.

To copy the received data into variables, you need to use an Assign activity by completing the following steps:

1. In the process editor palette, under the Basic Actions category, click **Assign**, and then click the connection between the Receive activity and the Reply activity.
2. Rename the new activity `CopyInput`.
3. In the Properties view, click the Details tab. Click **Assign From** on the left and select **ApplicationInput** from the list.
4. Click **Assign To** and select **AutoApprovalData** from the list.
5. Click **Add** to add the following mapping: **ApplicationInput** to **ManualApprovalData:LoanRequest** as shown in the following screen capture. This mapping will copy the loan request information to the loan officer.



Assign - CopyInput

Assign From		Assign To
ApplicationInput	⇨	AutoApprovalData
ApplicationInput	⇨	ManualApprovalData LoanRequest

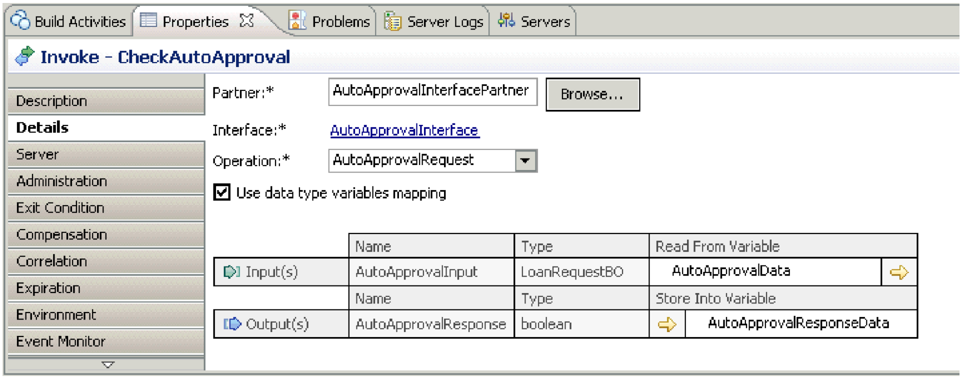
6. Save the activity.

Invoking the automatic approval rules

Define an activity in the process that will call the business rules component to check whether a loan can be approved without sending it to a loan officer.

To add the invoke activity, complete the following steps:

1. In the process editor palette, click **Invoke** and then click the connection between CopyInput and Reply. The Invoke activity is added.
2. Rename Invoke to `CheckAutoApproval`.
3. In the Properties view, click the **Details** tab and then click Browse. The Select a Partner window opens.
4. From the list, select **AutoApprovalInterfacePartner** and click **OK**. AutoApprovalRequest will be selected as the operation.
5. Select **Use Data Type Variables** and then click the top *(none)* cell. Select **AutoApprovalData**.
6. Click the second *(none)* cell and select **AutoApprovalResponseData**. The table should now look like the following screenshot:



Invoke - CheckAutoApproval

Partner:* AutoApprovalInterfacePartner

Interface:* AutoApprovalInterface

Operation:* AutoApprovalRequest

☑ Use data type variables mapping

	Name	Type	Read From Variable	
Input(s)	AutoApprovalInput	LoanRequestBO	AutoApprovalData	⇨
	Name	Type	Store Into Variable	
Output(s)	AutoApprovalResponse	boolean	⇨	AutoApprovalResponseData

7. Save the process.

You will now take the reply from the invoke activity and use it to direct the application down one of two paths in the process: automatic approval or manual approval.

Defining a case for automatically approved loan requests

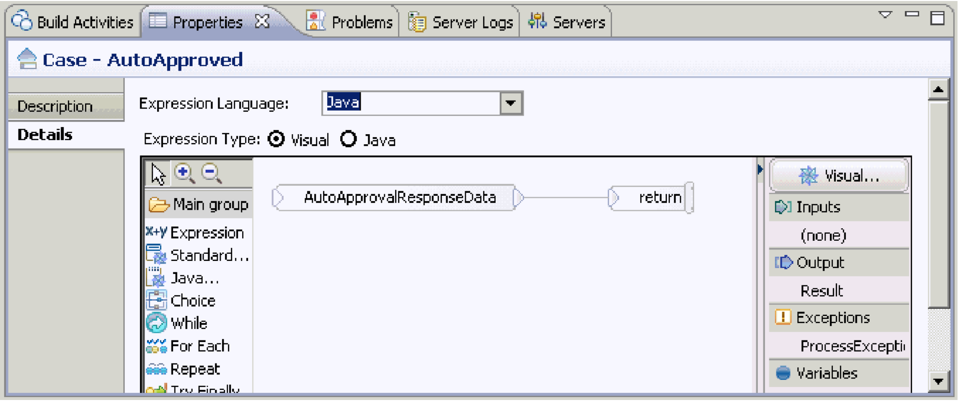
Based on the result from calling the business rules component, we will use a Choice activity to divide the process into separate paths. The loan request will follow one of two paths: automatic approval or manual approval.

To create a Choice activity, complete the following steps:

1. In the process editor palette, under the Structures category, click **Choice**.
2. Click the connection between **CheckAutoApproval** and **Reply**. The **Choice** activity is added.
3. Rename `Choice` to `AutoApprovalTest`.

To create a condition to check if the loan amount can be automatically approved, complete the following steps.

1. In the process editor, click the **Case** label.
2. In the Properties view, click the **Description** tab and type `AutoApproved` in the **Display Name** field.
3. To define the condition is checked using Visual Java, click **Details** and select **Java** from the Expression language list.
4. For the **Expression Type**, select `Visual`.
5. In the canvas, click **true** and replace it with AutoApprovalResponseData, as shown in the following screen capture. In this case, if AutoApprovalResponseData is true, we will follow this conditional path.



Case - AutoApproved

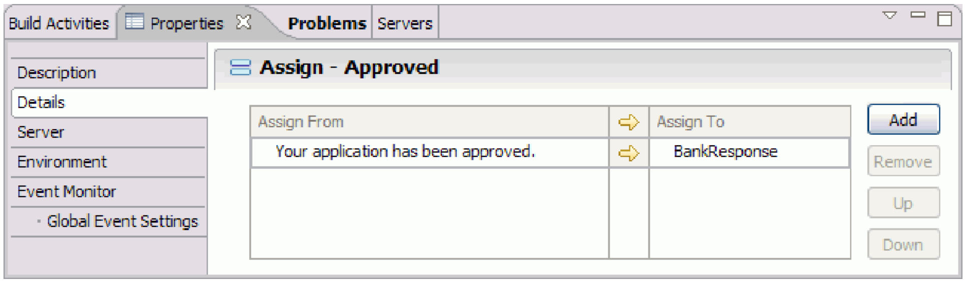
Expression Language: Java

Expression Type: ● Visual ○ Java

AutoApprovalResponseData → return

To add the approved assignment, complete the following steps.

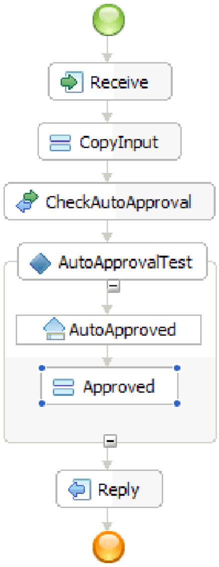
1. Add an Assign activity below AutoApproved, but in the Choice container, and rename it `Approved`.
2. Click on **Select From** and select **String**.
3. Type `Your application has been approved.`
4. Click **Select To** and select **BankReponse**. The completed Approved assign activity is shown below.



Assign - Approved

Assign From		Assign To
Your application has been approved.	⇨	BankResponse

5. The business process should now look like the following screenshot:

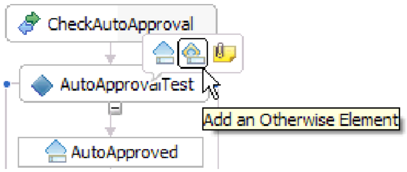


Defining a case for manually approved loan requests

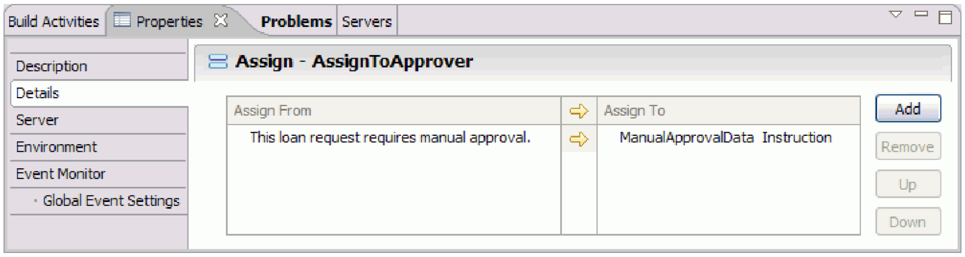
Create a case for when the loan application must be manually approved.

To create the path for manually approved requests, complete the following steps:

1. Create an Otherwise branch by clicking AutoApprovalTest and then clicking the Add an Otherwise Element icon.



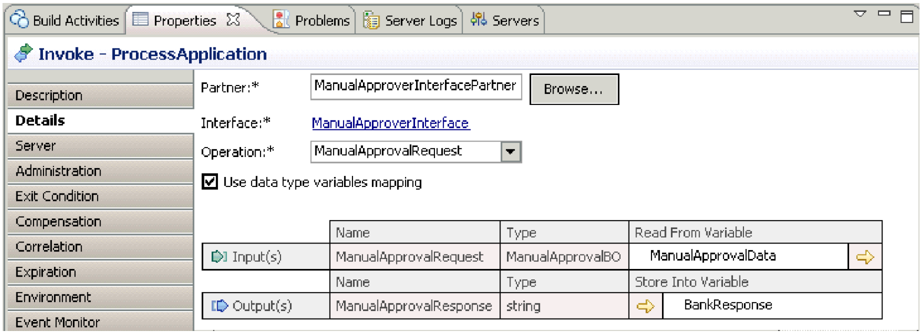
2. To transfer information about the loan request and instructions to the manual approver, add an **Assign** activity below the Otherwise case and rename it AssignToApprover.
3. In the Properties view click the Details tab, click underneath **Assign From** and select String.
4. Type `This loan request requires manual approval.`
5. Click **Assign To** and select **ManualApprovalData : Instruction** as shown in the following screen capture.



Assign - AssignToApprover

Assign From		Assign To
This loan request requires manual approval.	⇨	ManualApprovalData Instruction

6. To invoke the ManualApprover human task, complete the following steps:
   a. Add an Invoke activity below **AssignToApprover** and rename it `ProcessApplication`.
   b. In the Properties view, click **Details**.
   c. Click **Browse** and select ManualApproverInterfacePartner from the list. Click **OK**.
   d. Select **Use Data Type Variables** and click the first *(none)*.
   e. From the list, select `ManualApprovalData` and click **OK**.
   f. Click the second *(none)* button.
   g. From the list, select `BankResponse` and click **OK**. This will return a response from the loan officer to the loan applicant. The completed ProcessApplication activity is shown below.



Invoke - ProcessApplication

Partner:* ManualApproverInterfacePartner

Interface:* ManualApproverInterface

Operation:* ManualApprovalRequest

☑ Use data type variables mapping

	Name	Type	Read From Variable	
Input(s)	ManualApprovalRequest	ManualApprovalBO	ManualApprovalData	⇨
	Name	Type	Store Into Variable	
Output(s)	ManualApprovalResponse	string	⇨	BankResponse

7. Save the business process and close the Business Process editor.

Chapter 5. Implementing the approval logic and responding to the client

Implement the conditions to approve a loan and create the user interfaces.

Defining the business rules for automatic approval

Implement a set of rules that the bank will use to enforce its policy for loan applications. You will use a rule group component to specify a set of rules that, when evaluated, determine whether the loan should be automatically approved.

To implement the AutoApprovalRules business rules, complete the following steps:

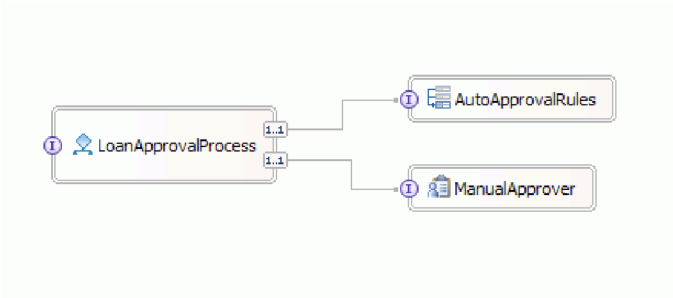
1. In the assembly editor, right-click the AutoApprovalRules component and select **Generate Implementation**. The Generate Implementation window opens.
2. In the navigation tree, click **LoanApplicationModule** and click **OK**.
3. Click **AutoApprovalRequest**.
4. Click **Enter Rule Logic** and select **New Ruleset**. Keep the default name and click **Finish**. You are now in the Rules editor.

Defining the implementation for manual approval

Create the implementation for the ManualApproval human task.

To implement the human task, follow these steps:

1. In the assembly editor, right-click the ManualApprover component and select **Generate Implementation**. Keep the defaults and click **OK**. The human task editor for ManualApprover now opens.
2. Close the implementation and save the assembly diagram. You will notice that the blue icon will disappear from each component in the assembly diagram because they have now been implemented.



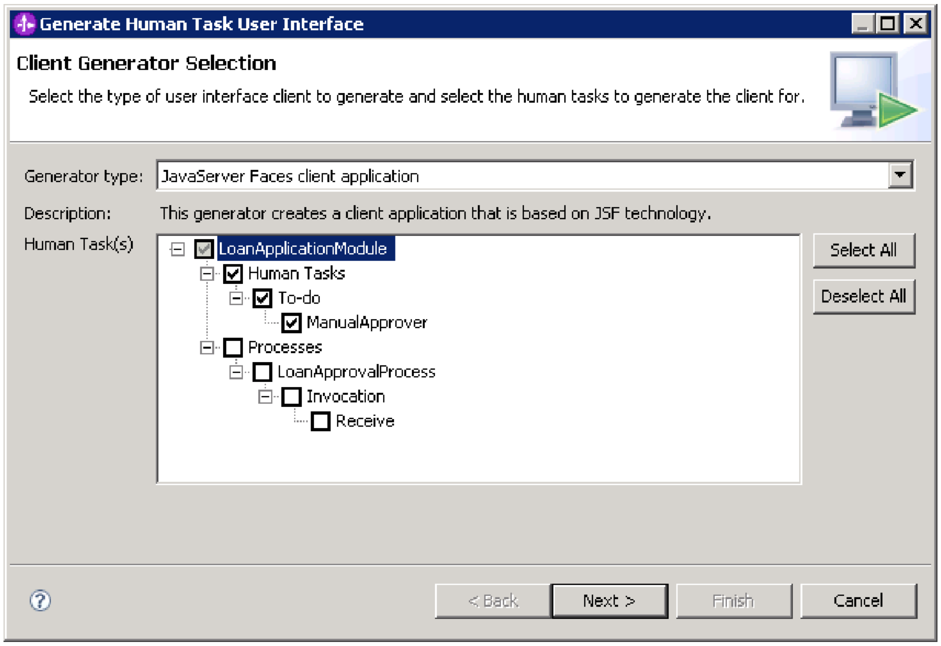
3. Close the assembly diagram.

Generating user interfaces

Create a user interface for customers to use to apply for loans and another user interface for loan officers to interact with the loan requests.

To create the user interface for the loan officer, complete the following steps:

1. In the Business Integration view, right-click **LoanApplicationModule** and select **Generate Human Task User Interfaces**. The Generate Human Task User Interface wizard opens.
2. Select **JavaServer Faces** client application as the Generator type and clear the Processes checkbox as shown in the following screen capture.



Generate Human Task User Interface

Client Generator Selection

Select the type of user interface client to generate and select the human tasks to generate the client for.

Generator type: JavaServer Faces client application

Description: This generator creates a client application that is based on JSF technology.

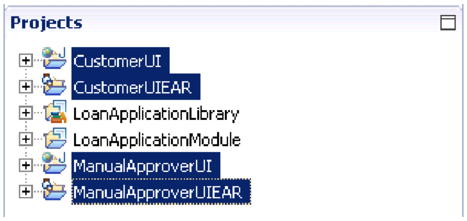
Human Task(s)

- ☑ LoanApplicationModule
  - ☑ Human Tasks
    - ☑ To-do
      - ☑ ManualApprover
  - □ Processes
    - □ LoanApprovalProcess
      - □ Invocation
        - □ Receive

3. Click **Next**. The JSF client configuration window opens. Type `ManualApproverUI` as the name of the dynamic Web project and click **Finish**.
4. A client generation completed dialog appears with how to invoke the UI. Click OK to accept the defaults.

To create the user interface for the customer, complete the following steps:

1. Generate the user interface as you did for ManualApproverUI, but in this case clear the Human Tasks checkbox. Name the dynamic Web project `CustomerUI`.
2. Enter CustomerUI as the name of the dynamic Web project and click Finish. In the Business Integration view you see four new folders:



Projects

- CustomerUI
- CustomerUIEAR
- LoanApplicationLibrary
- LoanApplicationModule
- ManualApproverUI
- ManualApproverUIEAR

Chapter 6. Testing the loan application

Test the application through custom JSF user interfaces.

There are four different use cases present in your application:

- The customer applies for a loan that is less than $50 000 and is automatically approved.
- The customer applies for a loan that is more than $50 000 and the loan officer decides to grant the loan.
- The customer applies for a loan that is more than $50 000, but the loan officer decides to reject the request.
- The loan officer wants to lower the automatic approval amount from $50 000 to $40 000. When the automatic approval amount is lowered, any customer who applies for a loan that is more than $40,000 will require manual approval.

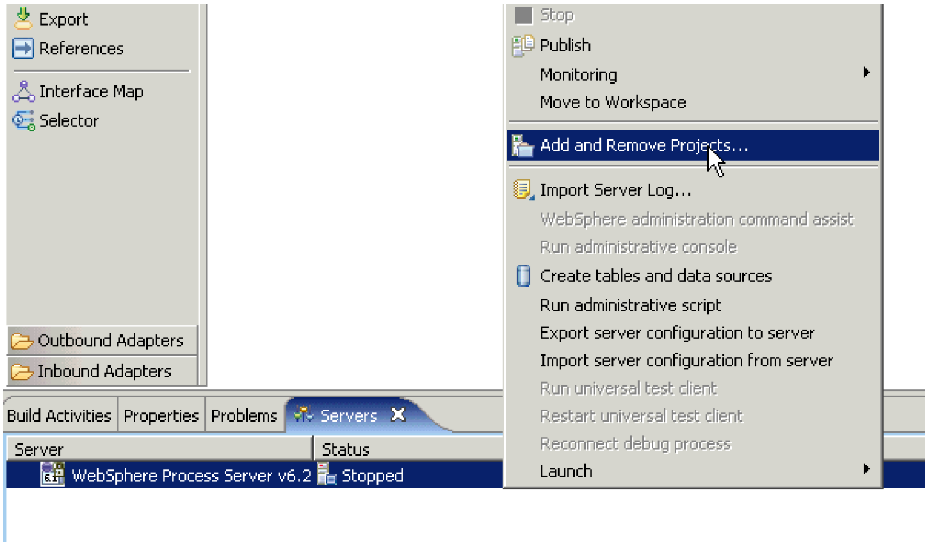
You will test each use-case in the following tasks.

Deploying the application

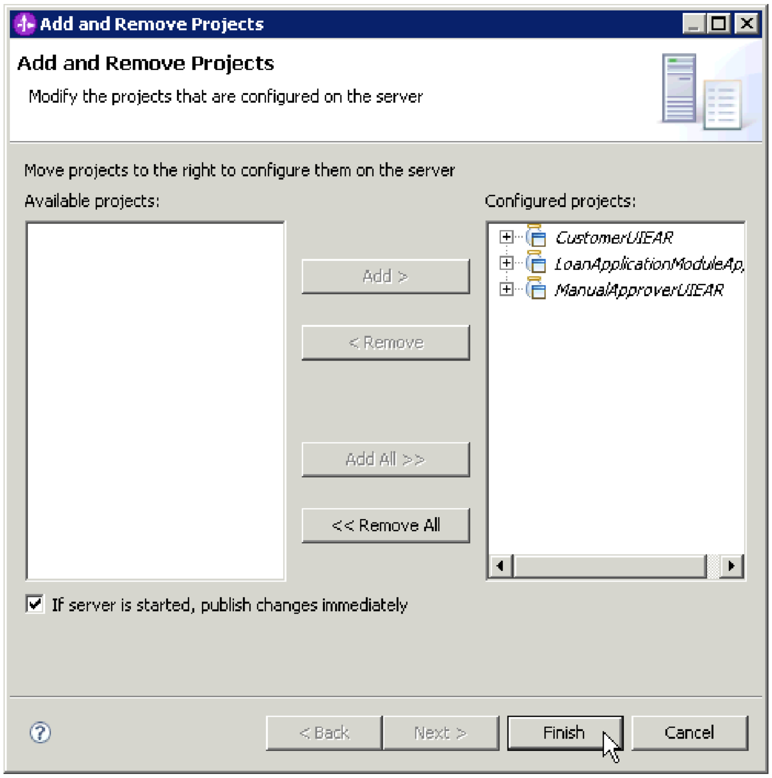
After you build the loan application, you will run it in the WebSphere® Process Server Integrated Test Client.

To deploy the loan application, complete the following steps:

1. In the Business Integration perspective click **Servers**.
2. Right-click **WebSphere Process Server** and select **Add and remove projects**. The Add and Remove Projects window opens.


Export
References
Interface Map
Selector
Outbound Adapters
Inbound Adapters
Build Activities
Properties
Problems
Servers
Server
Status
WebSphere Process Server v6.2
Stopped
Stop
Publish
Monitoring
Move to Workspace
Add and Remove Projects...
Import Server Log...
WebSphere administration command assist
Run administrative console
Create tables and data sources
Run administrative script
Export server configuration to server
Import server configuration from server
Run universal test client
Restart universal test client
Reconnect debug process
Launch

3. Click **Add all** and then click **Finish**.


Add and Remove Projects

Add and Remove Projects

Modify the projects that are configured on the server

Move projects to the right to configure them on the server

Available projects:

Configured projects:

- CustomerUIEAR
- LoanApplicationModuleAp
- ManualApproverUIEAR

☑ If server is started, publish changes immediately

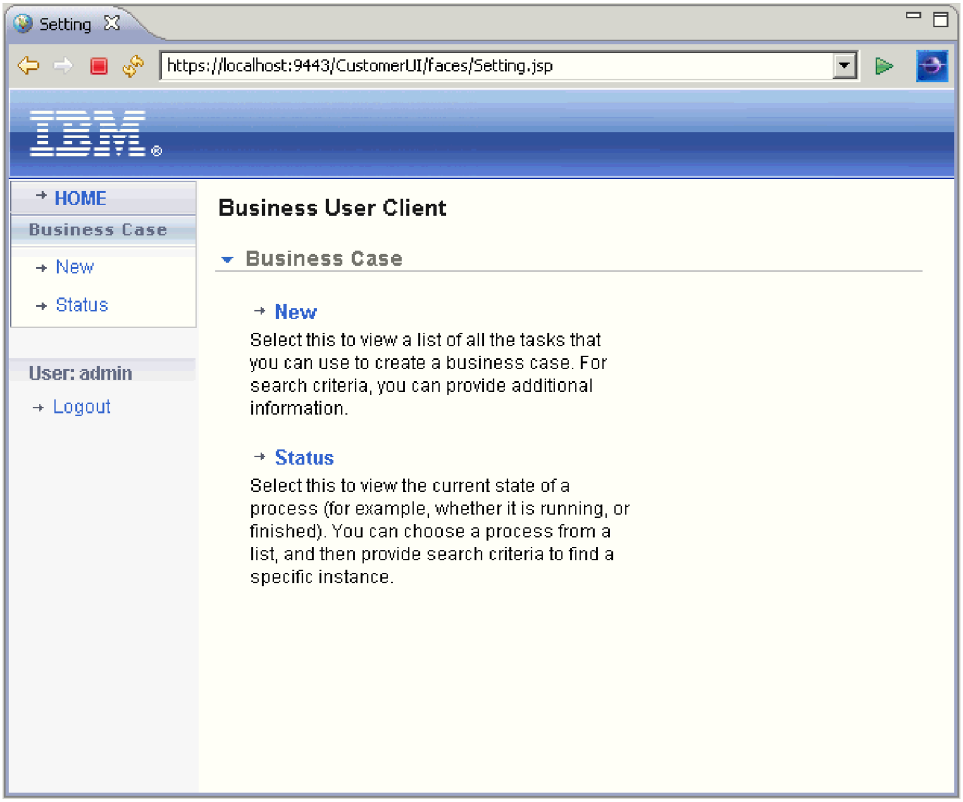
4. Right-click **WebSphere Process Server** again and select **Start**. It might take several minutes for the server to start and then publish the loan application.

Invoking the test cases

Test the loan application by interacting with it through a Web browser using two user interfaces: one for customers and one for loan officers.

To test the application, open the CustomerUI and the ManualApprovalUI user interfaces by completing the following steps:

1. In the Business Integration view, navigate to **CustomerUI -> WebContent -> Index.jsp**.
2. Right-click **Index.jsp** and select **Run As** → **1 Run on server**. The Run on Server window opens.
3. Select **WebSphere Process Server** and click **Finish**.
4. After the server is synchronized you will be asked to log in. The default username is `admin` and the default password is `admin`. After you have logged in, you see the Web page that the customer would see.



Business User Client

Business Case

New

Select this to view a list of all the tasks that you can use to create a business case. For search criteria, you can provide additional information.

Status

Select this to view the current state of a process (for example, whether it is running, or finished). You can choose a process from a list, and then provide search criteria to find a specific instance.

Alternatively, you can open a browser and enter `http://localhost:<portnumber>/CustomerUI`, where the port number can be found in the console output.

5. Similarly, open a browser window for ManualApprovalUI.

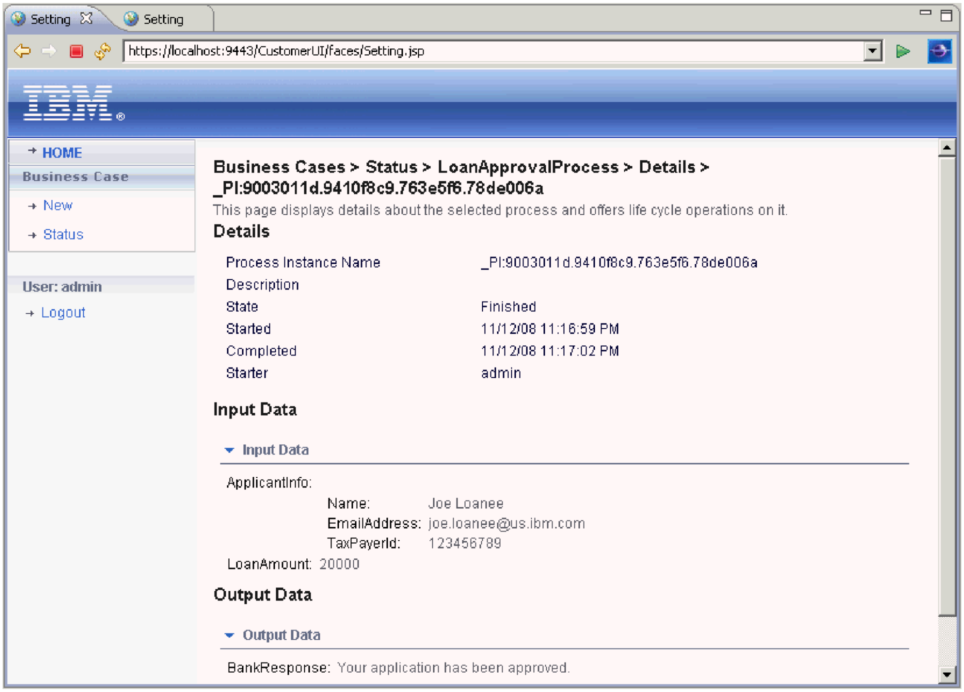
Test the first use case by invoking the automatic loan approval:

1. In the CustomerUI, click **New** then **Receive**, enter the information below, and click **Create**.

Table 1. CustomerUI information

Field	**Value**
Name	Joe Loanee
EmailAddress	joe.loanee@us.ibm.com
TaxPayerId	123456789
LoanAmount	20000

2. Check the status of the loan application by clicking **Status** then click **Status** again and then click **LoanApprovalProcess**.
3. Click the Process Instance Name to see the status of the application. You will be shown a page similar to the following image:



Business Cases > Status > LoanApprovalProcess > Details > _PI:9003011d.9410f8c9.763e5f6.78de006a

This page displays details about the selected process and offers life cycle operations on it.

Details

Process Instance Name	_PI:9003011d.9410f8c9.763e5f6.78de006a
Description	
State	Finished
Started	11/12/08 11:16:59 PM
Completed	11/12/08 11:17:02 PM
Starter	admin

Input Data

Input Data

ApplicantInfo:
Name: Joe Loanee
EmailAddress: joe.loanee@us.ibm.com
TaxPayerId: 123456789
LoanAmount: 20000

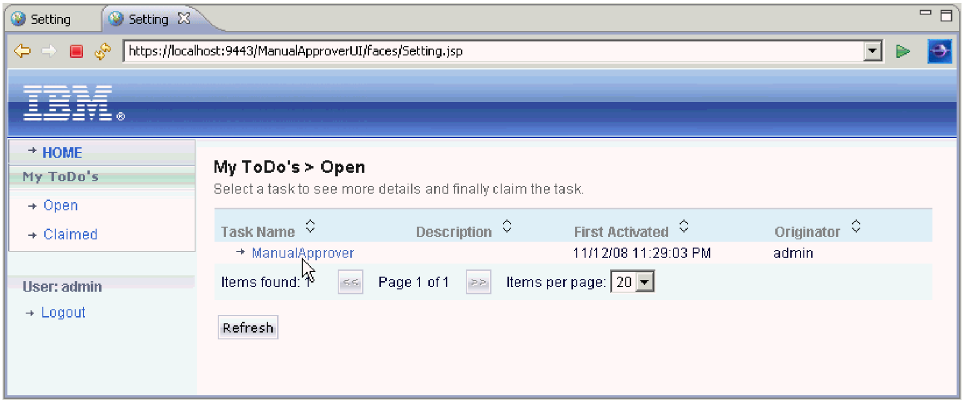
Output Data

Output Data

BankResponse: Your application has been approved.

To test the second and third use cases, complete the following steps:

1. Repeat the steps that you completed for the first test case, but in this case enter an amount greater than $50 000 and click **Create**.
2. Return to the ManualApproverUI and click **Open**. You see that there is a task waiting for the loan officer to investigate.



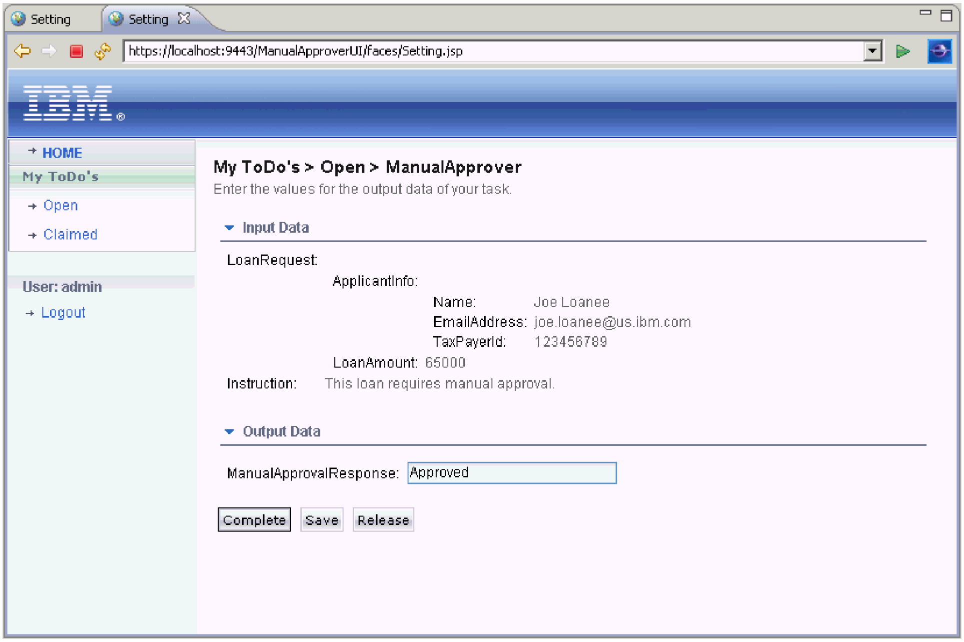
My ToDo's > Open

Select a task to see more details and finally claim the task.

Task Name	Description	First Activated	Originator
ManualApprover		11/12/08 11:29:03 PM	admin

Items found: 1 Page 1 of 1 Items per page: 20

3. Click the task name to see more details about the task. To process the task, click **Claim** in the detailed view.
4. In the ManualApprovalResponse, type `Approved` and click Complete. The customer can view the response by going through the steps mentioned previously.



My ToDo's > Open > ManualApprover

Enter the values for the output data of your task.

Input Data

LoanRequest:
ApplicantInfo:
Name: Joe Loanee
EmailAddress: joe.loanee@us.ibm.com
TaxPayerId: 123456789
LoanAmount: 65000
Instruction: This loan requires manual approval.

Output Data

ManualApprovalResponse: Approved

To test the case when the loan officer rejects the loan, complete the same steps as you did for the second and third use cases but replace the response with `Rejected`.

Chapter 7. Import

A complete ready-made version of the Loan Application sample is available for you to download.

To download the ready-made sample:

1. In WebSphere Integration Developer, select **Help > Welcome**. The Welcome opens.
2. Click the **Samples / Tutorials** icon. The Samples / Tutorials page opens.
3. Under **Loan Application**, click the **Import** link. You are presented with two possible modules to import.
4. Select **Complete Artifacts**.

Instructions for running the sample are found in the topic ″Testing the loan application″.

Chapter 8. PDF version

The Loan Application documentation is also available in PDF.

This document is also available as a PDF file

You will need Adobe Acrobat to view it. A free version of this software is available from www.adobe.com.