

Orchestration de Web Services

Module 5

Exercice Pratique à l'usage de l'environnement de développement NetBeans

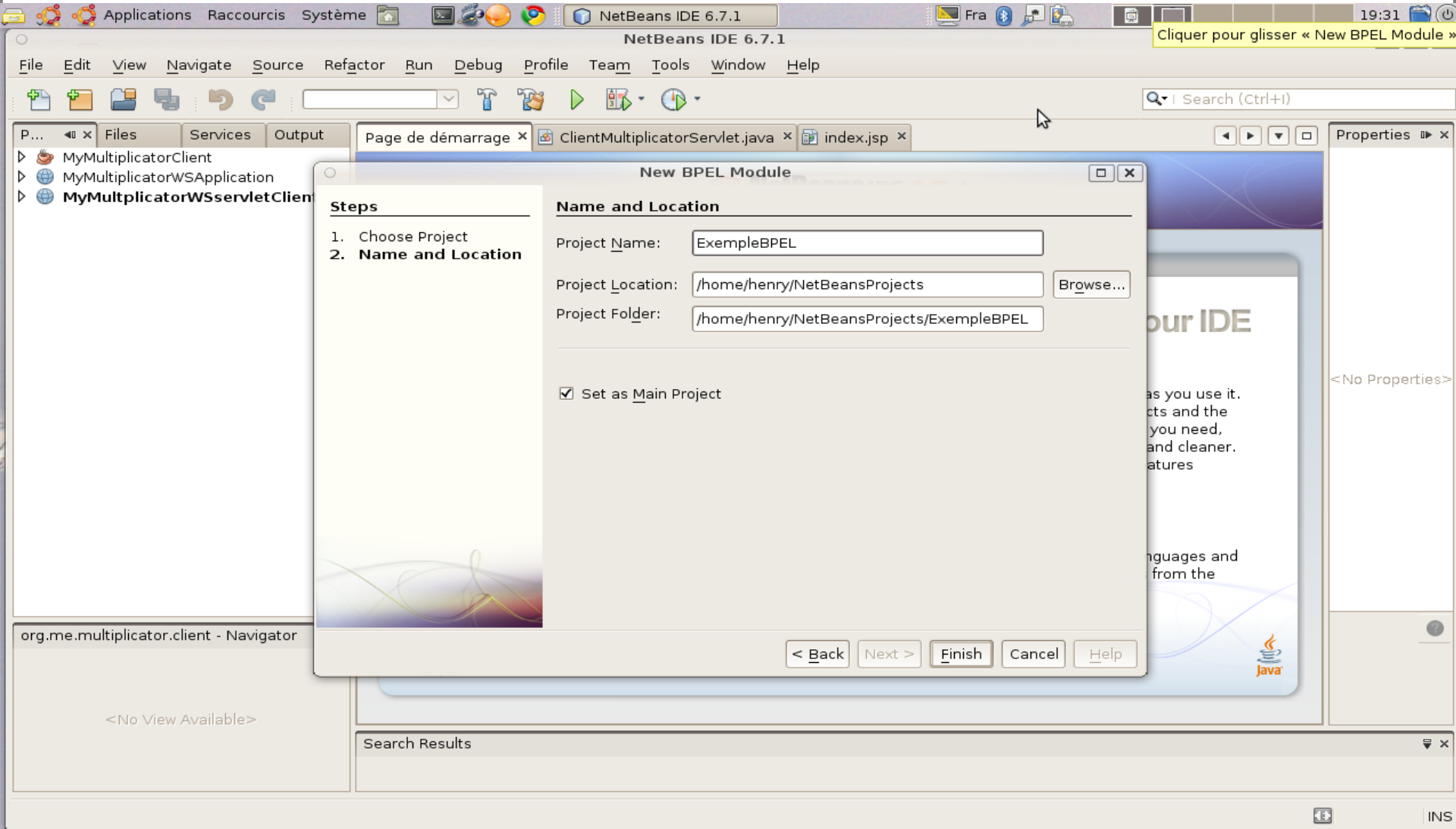
Développement en BPEL (Business Process Execution Language)

de l'organisation de l'appel à des services Web

Création d'un module de projet BPEL

The screenshot displays the NetBeans IDE 6.7.1 interface. The 'New Project' dialog is open, showing the 'Choose Project' step. The 'Categories' list includes Java, JavaFX, Java Web, Java EE, Java ME, Maven, PHP, Ruby, Groovy, C/C++, SOA, and NetBeans Modules. The 'SOA' category is selected, and the 'Projects' list shows Composite Application, BPEL Module, SQL Module, and XSLT Module. The 'BPEL Module' project type is selected. The description for the BPEL Module is: 'Creates an empty BPEL Module, which may contain multiple BPEL processes.' The dialog has buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'. The background shows the IDE interface with a project tree on the left and a 'Properties' panel on the right.

Création de l'exemple



Création de la ressource WSDL

ExempleBPEL - NetBeans IDE 6.7.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Process Files

ExempleBPEL

MyMultiplicatorClient

MyMultiplicatorWSApplication

MyMultiplicatorWSservletClient

Page de démarrage x ClientMultiplicatorServlet.java x index.jsp x

New WSDL Document

Steps

1. Choose File Type
2. **Name and Location**
3. Abstract Configuration

Name and Location

File Name: newWSDL4exempleBPEL

Project: ExempleBPEL

Folder: src Browse...

Created File: tBeansProjects/ExempleBPEL/src/newWSDL4exempleBPEL.wsdl

Target Namespace: atbeans.org/wsdl/ExempleBPEL/newWSDL4exempleBPEL

WSDL Type:

- Abstract WSDL Document
- Concrete WSDL Document

< Back Next > Finish Cancel Help

org.me.multiplicator.client - Navigator

<No View Available>

Search Results

Process ...

INS

Configuration de la ressource WSDL par défaut (à éditer)

Steps

1. Choose File Type
2. Name and Location
3. **Abstract Configuration**

Abstract Configuration

Port Type Name:

Operation Name:

Operation Type:

Input:

Message Part Name	Element Or Type
part1	xsd:string

Output:

Message Part Name	Element Or Type
part1	xsd:string

Fault:

Message Part Name	Element Or Type
-------------------	-----------------

Generate partnerlinktype automatically.

< Back Next > Finish Cancel Help

Vue du source XML de la ressource WSDL

The screenshot displays the NetBeans IDE interface with the following components:

- Project Explorer:** Shows the project structure for 'ExempleBPEL', including 'Process Files' and sub-projects like 'MyMultiplierClient' and 'MyMultiplierServletClient'.
- Source Editor:** Displays the XML source code for 'newWSDL4exempleBPEL.wsdl'. The code includes:
 - XML declaration: `<?xml version="1.0" encoding="UTF-8" ?>`
 - Definitions block with namespaces for soap, wsdl, and xsd.
 - Two messages: 'newWSDL4exempleBPELOperationRequest' and 'newWSDL4exempleBPELOperationResponse', both with a single part named 'part1' of type 'xsd:string'.
 - A port type 'newWSDL4exempleBPELPortType' containing an operation 'newWSDL4exempleBPELOperation' with input 'input1' and output 'output1'.
 - A partner link type 'newWSDL4exempleBPEL' with a role 'newWSDL4exempleBPELPortTypeRole'.
- Properties Window:** Shows properties for the selected WSDL file, including Name, Target Namespace, and Prefixes (plnk, tns, wsdl, xsd).
- Search Results:** A panel at the bottom showing the results of a search operation.

La vue 'partenaire' de l'éditeur de WSDL

The screenshot shows the NetBeans IDE 6.7.1 interface for editing a WSDL file. The main editor area is in the 'Partner' view, showing the following structure:

- Partner Link Types (1)**: A single entry 'newWSDL4exempleBPEL' is listed.
- Messages (2)**: Two message types are defined:
 - newWSDL4exempleBPELOperationRequest (1 part)**:

Part Name	Part Element or Type
part1	xsd:string
 - newWSDL4exempleBPELOperationResponse (1 part)**:

Part Name	Part Element or Type
part1	xsd:string

The interface also includes a left-hand project explorer, a top toolbar, and a right-hand palette and properties panel.

La vue 'partenaire' de l'éditeur de WSDL

drag & drop pour ajouter un message

Partner Link Types (1)

- newWSDL4exempleBPEL

Add Message

newWSDL4exempleBPELOperationRequest (1 part)

Part Name	Part Element or Type
part1	xsd:string

Add Part Remove Part

newWSDL4exempleBPELOperationResponse (1 part)

Part Name	Part Element or Type
part1	xsd:string

Add Part Remove Part

newWSDL4exempleBPEL.wsd - Navig...

- Types
- Imports
- Messages
 - newWSDL4exempleBPELOperationf
 - newWSDL4exempleBPELOperationf

Ajouter des messages

The screenshot shows the NetBeans IDE 6.1 interface. The main editor displays the WSDL file 'newWSDL4ExempleBPEL.wsdl' in the 'Partner' tab. The editor is divided into two sections: 'Partner Link Types (0)' and 'Messages (1)'. The 'Messages (1)' section contains a table with one row:

Part Name	Part Element or Type
part1	<Undefined>

Buttons for 'Add Part' and 'Remove Part' are located below the table. The right sidebar shows the 'WSDL Components' palette with 'Message' selected, and the 'message1 [Message] - Properties' window showing the 'Name' property set to 'message1'. The bottom output window displays the error message: 'Per WS-I Basic Profile Rule R2204, a document-literal binding must only use wsdl:part element(s) that have been defined using the element attribute.'

Définition du type (XSD) d'une partie du message

The screenshot displays the NetBeans IDE 6.1 interface for editing a WSDL file named 'newWSDL4ExempleBPEL.wsdl'. The main editor area shows the 'Partner Link Types' section with the message: 'The WSDL file does not contain any partner link types.' Below this, the 'Messages' section contains one message, 'message1', which has a single part named 'inputType' with an undefined type. A dialog box titled 'Select Element Or Type' is open, showing a list of XSD types. The type 'string (restricts :anySimpleType)' is currently selected. The 'WSDL Components' palette on the right side of the IDE shows options for 'Message' (One Way) and 'Partner Link Type' (Request-Response). The bottom of the IDE shows the 'Output - Standard output' window with the text: 'Per WS-I Basic Profile Rule R2204, a document-literal binding must only use wsdl:part element(s)'.

Applications Raccourcis Système

ExempleBPEL - NetBeans IDE 6.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

Start Page x newWSDL4ExempleBPEL.wsdl x

Source WSDL Partner 100%

Partner Link Types (0) Add Partner Link Type

The WSDL file does not contain any partner link types.

Messages (1)

Part Name	Part Element or Type
inputType	<Undefined>

Add Part Remove Part

Select Element Or Type

- IDREFS
- int (restricts :long)
- integer (restricts :decimal)
- language (restricts :token)
- long (restricts :integer)
- Name (restricts :token)
- NCName (restricts :Name)
- negativeInteger (restricts :nonPositiveInteger)
- NMTOKEN (restricts :token)
- NMTOKENS
- nonNegativeInteger (restricts :integer)
- nonPositiveInteger (restricts :integer)
- normalizedString (restricts :string)
- NOTATION (restricts :anySimpleType)
- positiveInteger (restricts :nonNegativeInteger)
- QName (restricts :anySimpleType)
- short (restricts :int)
- string (restricts :anySimpleType)
- time (restricts :anySimpleType)
- token (restricts :normalizedString)
- unsignedByte (restricts :unsignedShort)
- unsignedInt (restricts :unsignedLong)

OK Cancel Help

WSDL Components

- Message One Way
- Partner Link Type Request-Response

Output - Standard output

Per WS-I Basic Profile Rule R2204, a document-literal binding must only use wsdl:part element(s)

ExempleBPEL - NetB...

Ajout du message de réponse

The screenshot displays the IBM WebSphere Studio interface for editing a BPEL file named `newWSDL4exempleBPEL.wsdl`. The main workspace is divided into several sections:

- Partner Link Types (1):** A single entry for `newWSDL4exempleBPEL` is shown.
- Messages (2):** Two message definitions are listed:
 - `newWSDL4exempleBPELOperationRequest` (1 part):

Part Name	Part Element or Type
inputType	xsd:string
 - `newWSDL4exempleBPELOperationResponse` (1 part):

Part Name	Part Element or Type
resultType	xsd:string

The `resultType` part in the response message is highlighted with an orange border. The right-hand side of the interface features a **Palette** with **WSDL Components** (Message, One Way, Partner Link Type, Request-Response) and a **Properties** window for the selected `resultType` part, showing its name and type as `xsd:string`.

Ajout d'un modèle d'association (partnerLinkType)

The screenshot shows the Eclipse IDE interface for editing a WSDL file named 'newWSDL4exempleBPEL.wsdl'. The main editor displays the 'Partner Link Types' section, which contains two entries: 'newWSDL4exempleBPEL' and 'partnerlinktype1'. The 'partnerlinktype1' entry is highlighted with an orange border and contains two roles: '[External Partner]' and 'role1'. A callout bubble points to the 'Add Partner Link Type' button with the text 'obtenu en cliquant ici'. Below the Partner Link Types section, the 'Messages (2)' section shows a message named 'newWSDL4exempleBPELOperationRequest' with one part named 'inputType' of type 'xsd:string'. The left sidebar shows the project structure, and the right sidebar shows the 'WSDL Components' palette.

Partner Link Types (2)

- newWSDL4exempleBPEL
- partnerlinktype1
 - [External Partner]
 - role1

Messages (2)

- newWSDL4exempleBPELOperationRequest (1 part)

Part Name	Part Element or Type
inputType	xsd:string

Définition d'un rôle, question/réponse

The screenshot displays an IDE interface for defining a BPEL role and message. The main workspace shows a diagram titled "Partner Link Types (1)" for the file "newWSDL4exempleBPEL.wsdl". The diagram illustrates a role definition for "newWSDL4exempleBPEL" involving an "External Partner" and an "External Port Type". The role is defined by a "newWSDL4exempleBPELPortTypeRole" which contains a "newWSDL4exempleBPELPortType". A "newWSDL4exempleBPELOperation" is shown with a request message "newWSDL4exempleBPELOperationRequest" and a response message "newWSDL4exempleBPELOperationResponse".

Below the diagram, the "Messages (2)" section shows a table for "newWSDL4exempleBPELOperationRequest (1 part)":

Part Name	Part Element or Type

The interface also includes a "Palette" on the right with "WSDL Components" such as "Message", "One Way", "Partner Link Type", and "Request-Response". The "Properties" window shows the name "newWSD...".

Association par sélection des messages

The screenshot shows an IDE window titled "newWSDL4exempleBPEL.wsdl" with a "Partner" tab selected. The main diagram area displays "Partner Link Types (1)" for "newWSDL4exempleBPEL". It features an "External Partner" and an "External Port Type" (both in dashed boxes) connected to a "newWSDL4exempleBPELPortTypeRole" and "newWSDL4exempleBPELPortType" respectively. A message association is shown between the external port type and the role, with a solid arrow for "newWSDL4exempleBPELOperationRequest" and a dashed arrow for "newWSDL4exempleBPELOperationResponse". A blue callout bubble labeled "question" points to the association line. Below the diagram, the "Messages (2)" section shows a table with one entry: "newWSDL4exempleBPELOperationRequest (1 part)".

Part Name	Part Element or Type
newWSDL4exempleBPELOperationRequest (1 part)	

Association par sélection des messages

The screenshot displays an IDE interface for editing a BPEL file named `newWSDL4exempleBPEL.wsdl`. The main workspace shows a **Partner Link Types** diagram with a single link type named `newWSDL4exempleBPEL`. This link type is associated with an **External Partner** and an **External Port Type**. The diagram illustrates a message exchange between the external partner and the local port type. A solid arrow labeled `newWSDL4exempleBPELOperationRequest` points from the external partner to the local port type. A dashed arrow labeled `newWSDL4exempleBPELOperationResponse` points from the local port type back to the external partner. A blue callout bubble with the word `réponse` (response) points to the dashed arrow. Below the diagram, the **Messages** section shows a message named `newWSDL4exempleBPELOperationRequest` with one part. The **WSDL Components** palette on the right lists `Message`, `One Way`, `Partner Link Type`, and `Request-Response`. The **Properties** window shows the name of the selected component as `newWSDL4exempleBPELOperationRequest`.

Création d'un processus BPEL

The screenshot shows an IDE interface for creating a BPEL process. The main window displays a diagram titled "Partner Link Types (1)" for "newWSDL4exempleBPEL". The diagram shows an "External Partner" connected to a "newWSDL4exempleBPELPortTypeRole". A message exchange is shown between the external partner and the port type role, with a message type "newWSDL4exempleBPELOperationRequest" and a response "newWSDL4exempleBPELOperationResponse". A "New" dialog is open, and a "BPEL Process..." option is highlighted. The "Messages (2)" section below the diagram shows a message type "newWSDL4exempleBPELOperationRequest (1 part)".

nommage

ExempleBPEL - NetBeans IDE 6.7.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Process Files - Navigator

Process Files - Properties

WSDL Components

Message

One Way

Partner Link Type

Request-Response

Process Files - Properties

Properties

Name src

Sorting

Sort Mode By Name ...

Process Files

newWSDL4exempleBPELOperationRequest (1 part)

Search Results

1 | 1 INS

New BPEL Process

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name: ExempleBPEL_Process

Project: ExempleBPEL

Folder: src

Created File: NetBeansProjects/ExempleBPEL/src/ExempleBPEL_Process.bpel

Target Namespace: tp://enterprise.netbeans.org/bpel/ExempleBPEL/Exemple

L'éditeur de processus

The screenshot displays a software interface for editing Business Process Execution Language (BPEL) processes. The main window shows a diagram titled "ExempleBPEL_Process" with a vertical flow from "Process Start" to "Process End". A central activity node is visible, containing a circle with a dot. The interface includes a menu bar (File, Edit, View, etc.), a toolbar, and several panels: a left-hand file explorer, a right-hand palette with "Web Service" and "Basic Activities" categories, and a bottom search results panel.

Sélection « drag & drop » de PartnerLink à partir de la Palette

The screenshot displays an IDE interface for developing BPEL processes. The main workspace shows a diagram titled "ExempleBPEL_Process" with a "Process Start" node and a connector leading to a "Partner Link" node. The "Partner Link" node is highlighted with a red circle and a blue callout bubble labeled "Drag & drop".

On the right side, the "Palette" is visible, containing several categories of activities:

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Empty
 - Wait

The "Partner Link" activity is highlighted with a blue callout bubble labeled "Drag & drop".

The bottom status bar shows "Search Results" and "1 | 1 INS".

Édition de type d'association (Partnerlink)

Applications Raccourcis Système ExempleBPEL - NetB... 14:44

ExempleBPEL - NetBeans IDE 6.7.1

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

ExempleBPEL

- Process Files
 - ExempleBPEL_Process.bpel
 - newWSDL4exempleBPEL.wsdl
 - MyMultiplierClient
 - MyMultiplierWSApplication
 - MyMultiplierWServletClient

ExempleBPEL_Process [Process] - Na...
BPEL Logical View
ExempleBPEL_Process

- Variables
- Correlation Sets
- Sequence
- Imports

Search Results

1 | 1 INS

Create New Partner Link

Name: PartnerLink1

WSDL File: [ExempleBPEL] newWSDL4exempleBPEL.wsdl

Use Existing Partner Link Type

Partner Link Type: newWSDL4exempleBPEL

My Role: newWSDL4exempleBPELPortTypeRole

Partner Role: -----

Swap Roles

Use a Newly Created Partner Link Type

Create in File: /newWSDL4exempleBPELWrapper Browse

Partner Link Type Name: _____

Process will implement (My Role)

Role Name: _____

Port Type: newWSDL4exempleBPELPortType

Partner service will implement (Partner Role)

Role Name: _____

Port Type: newWSDL4exempleBPELPortType

Ok Cancel Help

Web Service

- Invoke
- Receive
- Reply
- Partner Link

Basic Activities

- Assign
- Empty
- Wait

ExempleBPEL_Process ...

Main

Name ExempleB...

Target Namesp http://ente...

Atomic

Documentation ...

ExempleBPEL_Proces...

Le partenariat à éditer

The screenshot displays an IDE interface for editing a BPEL process. The main workspace shows a diagram for 'ExempleBPEL_Process' with a vertical flow from 'Process Start' to 'Process End'. A 'PartnerLink' activity is positioned between them, containing a 'Receive' sub-activity. The 'PartnerLink' is highlighted in the 'PaLETTE' on the right. The 'PaLETTE' also shows 'Web Service' activities (Invoke, Receive, Reply) and 'Basic Activities' (Assign, Empty, Wait). The 'PartnerLink1 [Partner...]' section in the 'PaLETTE' shows a 'Main' section with 'Property Edit' and various fields like 'Name', 'WSDL File', 'Partner Link T', 'My Role', 'Partner Role', and 'Documentati...'. The 'PartnerLink1 [Partn...]' section is also visible at the bottom right. The 'Search Results' section is empty at the bottom.

Sélection de l'activité de réponse insérée par « drag&drop »

The screenshot displays the NetBeans IDE 6.1 interface for editing a BPEL process. The main workspace shows a diagram for 'ExempleBPEL' with a 'Process Start' node, a 'Receive1' activity, and a 'Process End' node. The 'Receive1' activity is highlighted with a red border and a red 'x' icon, indicating it is the selected element. The 'Web Service' palette on the right lists various activities, with 'Receive' selected. The 'Properties' window at the bottom right shows the configuration for the selected 'Receive1' activity, including its name, creation status, and message details.

Applications Raccourcis Système

ExempleBPEL - NetBeans IDE 6.1

File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help

Start Page x newWSDL4ExempleBPEL.wsdl * x ExempleBPEL.bpel * x

Source Design Mapper Logging

100%

ExempleBPEL

Process Start

Process End

Receive1 [Receiv...]

PartnerL...

operatio...

Web Service

- Invoke
- Receive
- Reply
- Partner Link

Basic Activities

Structured Activities

Receive1 [Receive] - ...

Main

Property Edit

Name Receive1

Create Instar

Documentati

Message

Partner Link

Port Type

Receive1 [Receive]

ExempleBPEL - NetB...

Sélection de l'association

The screenshot displays the NetBeans IDE 6.7.1 interface. The main window shows the 'ExempleBPEL - NetBeans IDE 6.7.1' project. The 'Receive1 [Receive] - Property Editor' dialog is open, showing the following configuration:

- Name: Receive1
- Partner Link: (dropdown menu)
- Operation: (dropdown menu)
- Input Variable: (text field) with 'Create ...' and 'Browse ...' buttons
- Create Instance

The dialog has 'Ok', 'Cancel', and 'Help' buttons. The background shows the BPEL Logical View with a 'Process End' activity selected. The 'Receive1 [Receive] - Navigator' window shows the BPEL Logical View structure:

- ExempleBPEL_Process
- Variables
- Correlation Sets
- Sequence
- Receive1
- PartnerLink1
- Imports

The 'Web Service' palette on the right shows the 'Receive' activity selected. The 'Search Results' window at the bottom is empty.

Variable d'entrée

The screenshot displays the NetBeans IDE 6.7.1 interface for configuring a BPEL process. The main window shows the 'ExempleBPEL_Process.bpel' file in the Design view. A 'Receive1 [Receive] - Property Editor' dialog is open, showing the following configuration:

- Name: Receive1
- Partner Link: PartnerLink1
- Operation: newWSDL4exempleBPELOperation
- Input Variable: (empty)
- Create Instance:

A 'New Input Variable' dialog is also open, showing the configuration for a new variable:

- Name: NewWSDL4exempleBPELOperationIn
- Type: wWSDL4exempleBPELOperationRequest
- Scope: ExempleBPEL_Process

The background shows the BPEL process diagram with a 'Process End' activity. The left sidebar shows the project structure, and the right sidebar shows the Palette and the Receive1 [Receive] activity details.

L'association représentée

The screenshot shows an IDE window for editing a BPEL process. The main canvas displays a diagram for 'ExempleBPEL_Process' with a 'Process Start' node, a 'Receive' activity (represented by an envelope icon), and a 'Process End' node. A 'PartnerLink' component is visible on the left side of the canvas. A blue arrow points from the 'PartnerLink' component to the 'Receive' activity, indicating a drag-and-drop operation. A blue callout bubble with the text 'Drag & drop' is positioned at the end of this arrow. The right-hand side of the IDE features a 'Palette' with 'Web Service' and 'Basic Activities' categories. The 'Web Service' category is expanded, showing 'Invoke' and 'Receive' activities. The 'Basic Activities' category is also expanded, showing 'Assign', 'Empty', and 'Wait' activities. The bottom of the IDE window shows a 'Search Results' panel.

Insertion de l'activité de réponse dans le processus (drag&drop)

The screenshot displays a BPMN editor interface. The main canvas shows a process flow for 'ExempleBPEL_Process' starting with 'Process Start', followed by a 'Receive1' activity, and then a 'Reply1' activity. A 'PartnerLink1' activity is shown on the left, with a blue arrow indicating its connection to the 'Reply1' activity. The 'Reply1' activity is currently selected, and a blue arrow points from the 'Reply' icon in the 'Web Service' section of the 'Palette' to it. The 'Palette' also shows 'Basic Activities' like 'Assign', 'Empty', and 'Wait'. The 'Reply1 [Reply] - Prop...' panel on the right shows the properties of the selected activity, including 'Name: Reply1', 'Documentati...', and 'Partner Link'. The 'Search Results' panel at the bottom is empty.

Association au partenaire

The screenshot displays the NetBeans IDE 6.7.1 interface for developing a BPEL process. The main workspace shows a BPEL diagram for 'ExempleBPEL_Process' in Design mode. The process starts with 'Process Start', followed by a 'PartnerLink' (newWS...) and a 'Reply1' activity. The 'Reply1' activity is currently selected, and its 'Property Editor' is open. The 'Property Editor' shows the following configuration:

- Name:** Reply1
- Partner Link:** PartnerLink1
- Operation:** newWSDL4exempleBPELOperation
- Response Type:** Normal Response
- Output Variable:** (empty field) [Create ...] [Browse ...]
- Response Type:** Fault Response
- Fault Name:** (empty field) [Choose ...]
- Fault Variable:** (empty field) [Create ...] [Browse ...]

The 'Reply1 [Reply] - Property Editor' dialog has 'Ok', 'Cancel', and 'Help' buttons at the bottom.

Variable à retourner

The screenshot displays the NetBeans IDE 6.7.1 interface for editing a BPEL process. The main workspace shows a diagram of the `ExempleBPEL_Process` with a `Reply1` activity. A `PartnerLink` is connected to the `Reply1` activity. Two dialog boxes are open:

- Reply1 [Reply] - Property Editor**: Shows the configuration for the `Reply1` activity. The `Name` is `Reply1`, `Partner Link` is `PartnerLink1`, and `Operation` is `newWSDL4exempleBPELOperation`. The `Normal Response` radio button is selected. The `Output Variable` field is empty, with `Create ...` and `Browse ...` buttons next to it.
- New Output Variable**: A dialog for creating a new output variable. The `Name` is `NewWSDL4exempleBPELOperationOut`, the `Type` is `WSDL4exempleBPELOperationResponse`, and the `Scope` is `ExempleBPEL_Process`. It also has `Ok` and `Cancel` buttons.

The left sidebar shows the project structure for `ExempleBPEL`, including `Process Files` and `MyMultiplicatorClient`. The bottom-left pane shows the `Reply1 [Reply] - Navigator` with the `BPEL Logical View` selected, showing the `Reply1` activity highlighted.

La représentation du lien

The screenshot displays a software interface for editing a BPEL process. The main workspace shows a diagram titled "ExempleBPEL_Process" with a flow from "Process Start" to "Process End". A "Reply1" activity is highlighted, connected to a "PartnerLink1" component. The "Reply1" activity is shown in a "Logical View" in the bottom-left pane, which lists its properties: Name (Reply1), Partner Link (PartnerLink1), Port Type (newWSDL4...), and Operation (newWSDL4...). The right-hand side features a "Palette" with "Web Service" and "Basic Activities" categories, and a "Reply1 [Reply] - Prop..." window showing its configuration. The bottom status bar indicates "13 | 11 INS".

Insertion d'une activité d'assignation de valeur (drag&drop) entre réception et réponse

The screenshot displays a BPMN editor interface. The main canvas shows a process diagram for 'ExempleBPEL_Process'. The flow starts with 'Process Start', followed by a 'Receive1' activity (represented by an envelope icon), then an 'Assign1' activity (represented by a circle icon), and finally a 'Reply1' activity (represented by a curved arrow icon), ending at 'Process End'. A 'PartnerLink1' activity is also visible on the left side of the diagram.

The 'Palette' on the right side of the editor contains various activities, including 'Web Service' (Invoke, Receive, Reply, Partner Link) and 'Basic Activities' (Assign, Empty, Wait). The 'Assign' activity is highlighted in the palette. A blue callout bubble with the text 'Drag & drop' points to the 'Assign' activity in the palette and the 'Assign1' activity in the diagram.

The 'Assign1 [Assign] - Navigator' window on the left shows the 'BPEL Logical View' of the process, listing elements such as 'Variables', 'Correlation Sets', 'Sequence', 'Receive1', 'Assign1', 'Reply1', 'PartnerLink1', and 'Imports'. The 'Assign1' element is currently selected.

Sélection de l'associé (« mapper »)

The screenshot displays the Eclipse IDE interface for BPEL development. The main window is titled 'ExempleBPEL' and shows the 'Mapper' tab selected. The workspace is divided into several panes:

- Left Pane (Project Explorer):** Shows the project structure under 'ExempleBPEL', including 'Process Files' and 'MyMultiplicatorClient'.
- Top Center Pane (Mapper):** Contains a toolbar with icons for 'Operator', 'Boolean', 'String', 'Node', 'Number', 'Date & Time', and 'BPEL'. Below the toolbar are 'All' and 'Output' buttons. The main area is divided into 'Variables' and 'Partner Links' sections, with a mouse cursor hovering over the 'Variables' section.
- Right Pane (Properties):** Shows the properties for the selected 'Assign1' element, currently displaying '<No Properties>'. A 'Partner Links' section is also visible.
- Bottom Left Pane (Navigator):** Shows the 'Assign1 [Assign] - Navigator' with a 'BPEL Logical View' dropdown. The tree view includes 'ExempleBPEL_Process', 'Variables', 'NewWSDL4exempleBPELOperatio', 'Correlation Sets', 'Sequence', 'Receive1', 'Assign1', 'Reply1', 'PartnerLink1', and 'Imports'.
- Bottom Center Pane (Search Results):** A search results pane is visible at the bottom.

A blue callout bubble points to the 'Mapper' tab in the top center pane.

Association graphique

Page de démarrage × newWSDL4exempleBPEL.wsdl × ExempleBPEL_Process.bpel * ×

Source Design Mapper Logging

Operator Boolean String Node Number Date & Time BPEL

All Output

Input All

Variables

- NewWSDL4exempleBPELOperationOut
 - resultType
- NewWSDL4exempleBPELOperationIn
 - inputType

Partner Links

- PartnerLink1
 - myRole

Variables

- NewWSDL4exempleBPELOperationOut
 - resultType
- NewWSDL4exempleBPELOperationIn
 - inputType

Partner Links

<No Properties>

Assign1 ...

Assign1 [Assign] - Navigator

BPEL Logical View

ExempleBPEL_Process

Variables

- NewWSDL4exempleBPELOperatic
- NewWSDL4exempleBPELOperatic

Correlation Sets

Sequence

- Receive1
- Assign1
- NewWSDL4exempleBPELOpera
- NewWSDL4exempleBPELOpera

Reply1

PartnerLink1

Search Results

13 | 11 | INS

L'expression en XML/BPEL du processus défini

The screenshot displays an IDE window with the following components:

- File Explorer (Left):** Shows the project structure for 'ExempleBPEL', including 'Process Files' and 'ExempleBPEL_Process.bpel'.
- Navigator (Bottom Left):** Shows the 'BPEL Logical View' for 'ExempleBPEL_Process', detailing 'Variables' (NewWSDL4exempleBPELC), 'Correlation Sets', 'Sequence' (Receive1, Assign1), and 'Reply1'.
- Code Editor (Center):** Displays the XML BPEL code for 'ExempleBPEL_Process'. The code includes namespace declarations, partner links, variable declarations, and a sequence of operations: a receive operation, an assign operation with two copy elements, and a reply operation.
- Search Results (Bottom):** A search results pane is visible at the bottom of the IDE.

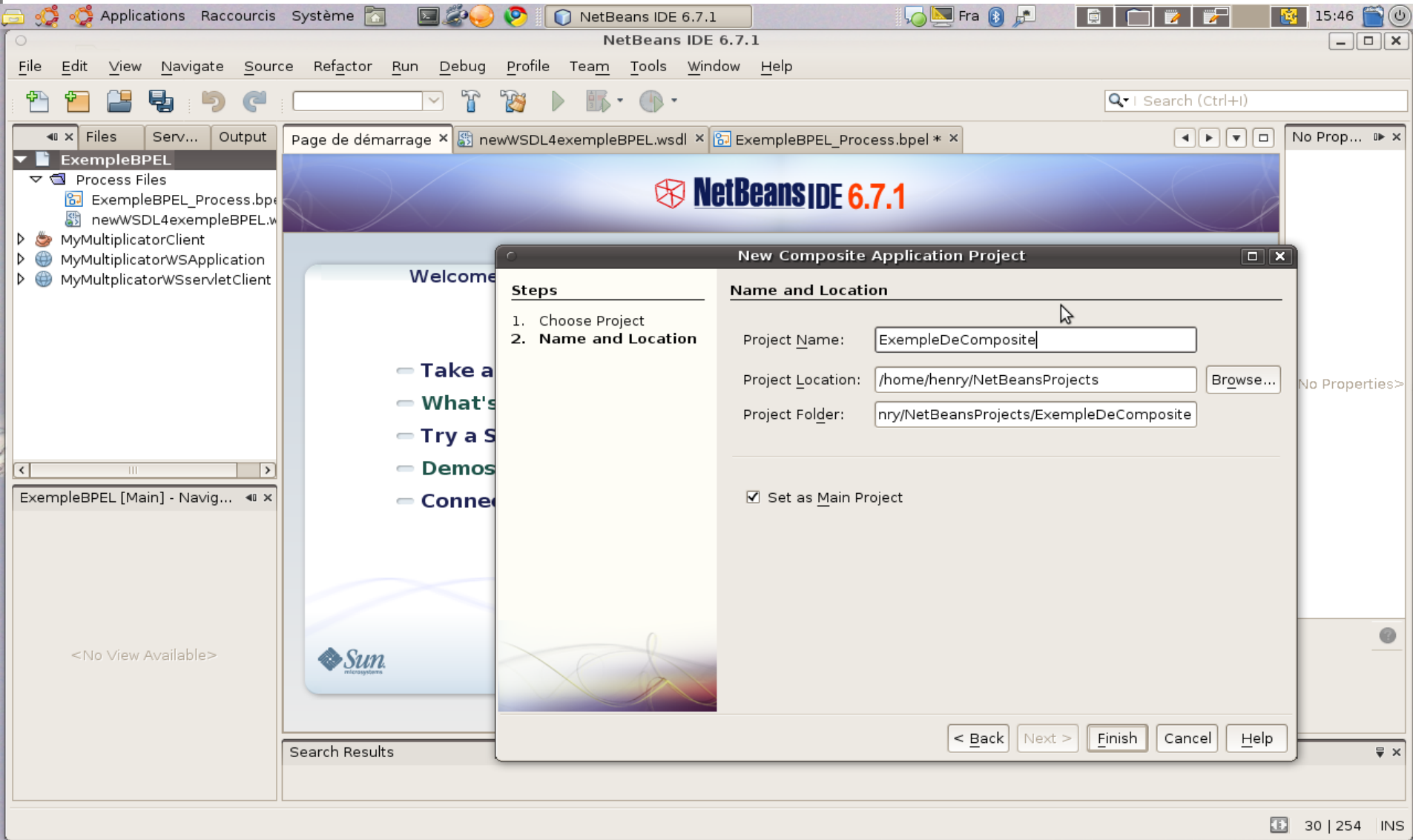
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <process
3   name="ExempleBPEL_Process"
4   targetNamespace="http://enterprise.netbeans.org/bpel/ExempleBPEL/ExempleBPEL_Process"
5   xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
6   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
7   xmlns:sxt="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Trace"
8   xmlns:sxed="http://www.sun.com/wsbpel/2.0/process/executable/SUNExtension/Editor"
9   xmlns:tns="http://enterprise.netbeans.org/bpel/ExempleBPEL/ExempleBPEL_Process">
10  <import namespace="http://j2ee.netbeans.org/wsd/ExempleBPEL/newWSDL4exempleBPEL" location="newWSDL
11  <partnerLinks>
12    <partnerLink name="PartnerLink1" xmlns:tns="http://j2ee.netbeans.org/wsd/ExempleBPEL/newWSDL4ex
13  </partnerLinks>
14  <variables>
15    <variable name="NewWSDL4exempleBPELOperationOut" xmlns:tns="http://j2ee.netbeans.org/wsd/Exempl
16    <variable name="NewWSDL4exempleBPELOperationIn" xmlns:tns="http://j2ee.netbeans.org/wsd/Exemple
17  </variables>
18  <sequence>
19    <receive name="Receive1" createInstance="yes" partnerLink="PartnerLink1" operation="newWSDL4exem
20    <assign name="Assign1">
21      <copy>
22        <from variable="NewWSDL4exempleBPELOperationIn" part="inputType" />
23        <to variable="NewWSDL4exempleBPELOperationIn" />
24      </copy>
25      <copy>
26        <from variable="NewWSDL4exempleBPELOperationOut" part="resultType" />
27        <to variable="NewWSDL4exempleBPELOperationOut" />
28      </copy>
29    </assign>

```

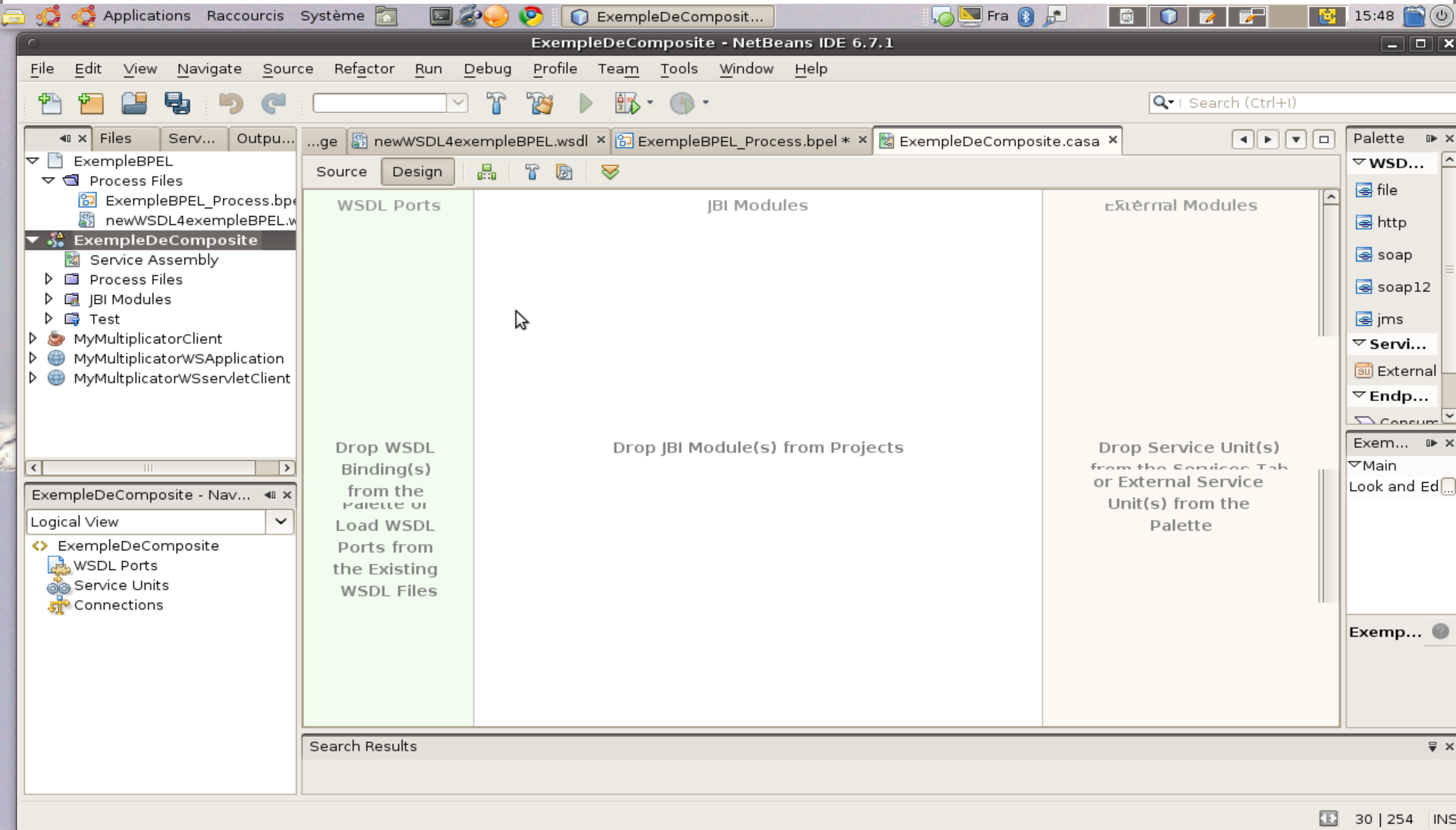
Création de l'application composite qui hébergera le processus

The screenshot displays the NetBeans IDE 6.7.1 interface. The main window shows a project named 'ExempleBPEL' with a file tree containing 'Process Files', 'ExempleBPEL_Process.bpel', and 'newWSDL4exempleBPEL.wsd'. A 'New Project' dialog box is open in the foreground, showing the 'Choose Project' step. The 'Categories' list includes Java, JavaFX, Java Web, Java EE, Java ME, Maven, PHP, Ruby, Groovy, C/C++, SOA, and NetBeans Modules. The 'Projects' list shows 'Composite Application', 'BPEL Module', 'SQL Module', and 'XSLT Module'. The 'Description' field contains the text: 'modules. Note: For more information, press the **Help** button and see the on-line help section on **About Composite Application Projects**.' The dialog has buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

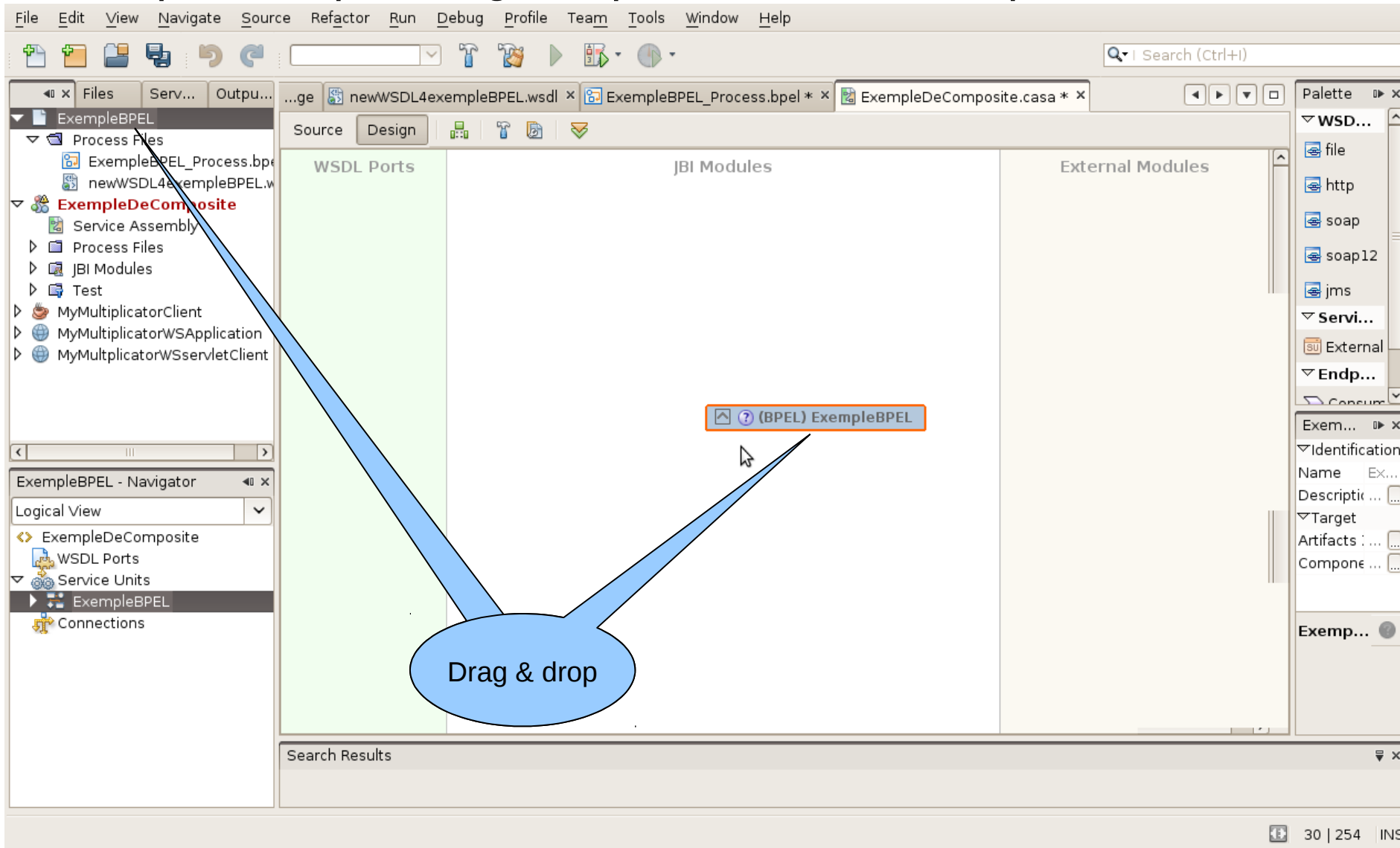
nommage de l'application composite



L'interface 'casa' de conception d'application composite



Composition par drag&drop du module ExempleBPEL



Après compilation (Build) sur les 2 modules

The screenshot displays an IDE interface with the following components:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Includes icons for file operations, search, and execution.
- Project Explorer:** Shows a project named 'ExempleBPEL' with a sub-project 'XML check' containing 'build.xml (jbi-build)'. The console output indicates a successful build: 'BUILD SUCCESSFUL (total...)'.
- Design View:** The main workspace shows a BPEL process diagram with three panels: 'WSDL Ports', 'JBI Modules', and 'External Modules'. A box labeled '(BPEL) ExempleBPEL' contains 'ExempleBPEL_Process' and 'PartnerLink1'.
- Palette:** Lists various service types: file, http, soap, soap12, jms, Servi..., External, Endp..., and Consum...
- Search Results:** Located at the bottom of the IDE.

Création d'un point d'accès WSDL drag&drop de l'option d'échange par fichier

The screenshot shows an IDE interface with the following components:

- File Explorer:** Shows a project structure with folders like 'ExempleBPEL', 'Process Files', and 'ExempleDeComposite'. The 'ExempleDeComposite' folder is expanded to show 'Service Assembly', 'Process Files', 'JBI Modules', and 'Test'.
- Source Editor:** Displays the 'WSDL Ports' section of a BPEL file. A port named 'casaPort1' is visible, with a 'FILE' icon and a double-headed arrow next to it. A tooltip for '(BPEL) ExempleBPEL' is shown over the 'ExempleBPEL_Process' element, with a 'PartnerLink1' label below it.
- Palette:** Located on the right, it contains 'WSDL Bindings' with options like 'file', 'http', 'soap', 'jms', and 'External'. Below it are 'Service Units' and 'Endpoints'.
- Properties View:** Shows the 'Properties' for 'casaPort1', including 'Interface Name', 'Endpoint Name' (casaPort1), and 'Component Name' (sun-file-binding).
- Search Results:** A search bar at the bottom right with the text 'Search Results'.

A blue callout bubble with the text 'Drag & drop' points to the 'FILE' icon in the 'WSDL Ports' section, indicating the action of dragging the file binding option to the port.

Association graphique

The screenshot displays a software development environment with the following components:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Includes icons for file operations (new, open, save, print), navigation (back, forward), and execution (run, debug, stop).
- Search Bar:** Search (Ctrl+I).
- Left Panel (Project Explorer):** Shows a tree view of the project structure, including 'ExempleBPEL', 'Process Files', and 'ExempleDeComposite' with its sub-components like 'Service Assembly', 'Process Files', 'JBI Modules', and 'Test'.
- Center Panel (Design View):** Displays a graphical association between a 'WSDL Ports' container (containing 'casaPort1') and a 'JBI Modules' container (containing '(BPEL) ExempleBPEL' and 'ExempleBPEL_Process'). An orange arrow points from 'casaPort1' to 'PartnerLink1' in the BPEL module.
- Right Panel (Palette):** Lists various components for WSDL Bindings, Service Units, and Endpoints. The 'Endpoints' section shows 'Consume' and 'Provide' options.
- Bottom Panel (Search Results):** Currently empty.

Adaptation du fichier d'entrée (-->input.txt)

The screenshot displays the IBM Business Process Manager (BPM) Studio interface. The main workspace shows a design view with two columns: 'WSDL Ports' and 'JBI Modules'. A 'file' binding is connected to a 'PartnerLink1' in the 'ExempleBPEL_Process' module. The left sidebar shows a project tree with 'ExempleDeComposite' selected. The bottom-left pane shows the 'Logical View' of 'casaPort1', highlighting the 'file:message' component. The bottom-right pane shows the 'Properties' for 'file:message', including 'use: literal', 'encodingStyle', 'fileName: test.xml', and 'pollingInterval: 1000'.

Adaptation semblable du fichier de sortie

The screenshot displays the IBM Business Architect interface. The main workspace shows a BPEL process diagram with a 'WSDL Ports' section containing a 'casaPort1' port and a 'JBI Modules' section containing an '(BPEL) ExempleBPEL' module. A connection is established between the 'casaPort1' port and the 'PartnerLink1' of the 'ExempleBPEL_Process' module. The 'Logical View' pane on the left shows the structure of the 'file:message' component, including 'output1' and 'file:message' sub-elements. The 'Properties' pane on the right shows the configuration for the 'file:message' component, with the 'fileName' property set to 'output.txt'. The 'WSDL Bindings' pane shows the 'file' binding type. The 'Service Units' pane shows the 'External' service unit. The 'Endpoints' pane shows the 'Consume' and 'Provide' endpoints. The 'Search Results' pane is empty.

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Palette

- WSDL Bindings
 - file
 - http
 - soap
 - soap12
 - jms
- Service Units
 - External
- Endpoints
 - Consume
 - Provide

file:message - Properties

- Properties
 - use: literal
 - encodingStyle
 - fileType
 - fileName: output.txt
 - fileNamesPattern
 - pollingInterval: 1000
 - part
 - removeEOL
 - addEOL
 - multipleRecordsPerF
- fileName: 4.%{. where %{ denotes an integer number in input file name or a one up sequence number persisted in a sequence file by name if it is for output file.

Search Results

30 | 254 | INS

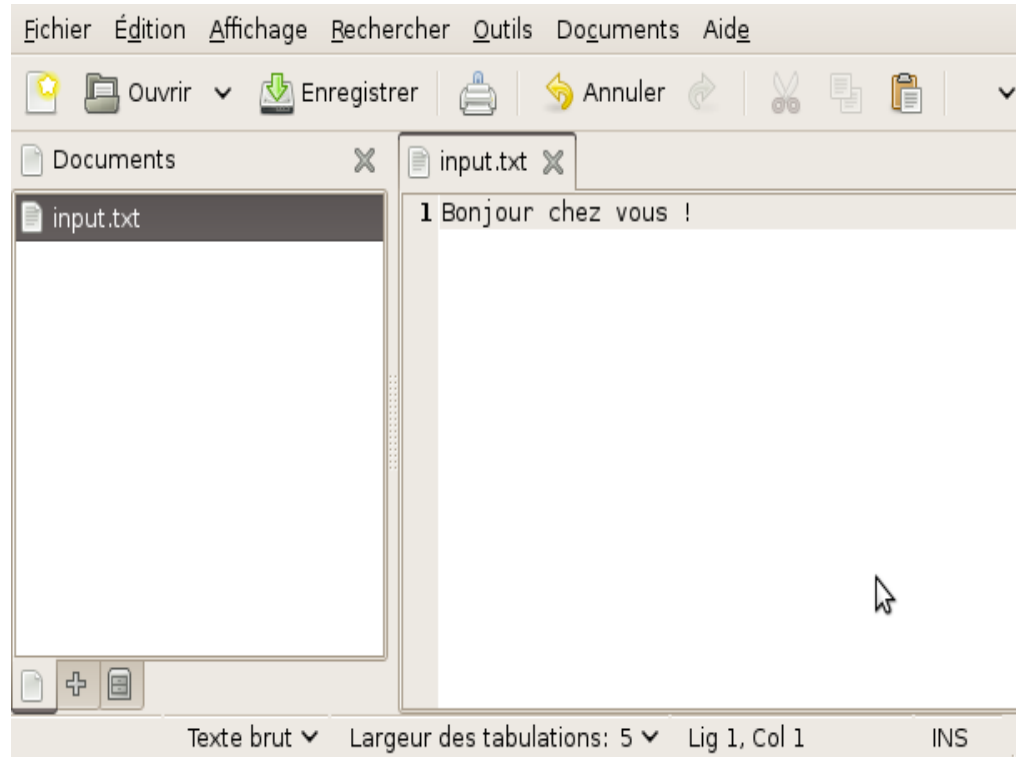
emplacement des fichiers de tests

The screenshot displays the IBM Business Architect interface for configuring a BPEL process. The main workspace shows a diagram with a 'FILE' icon connected to a 'PartnerLink1' in a '(BPEL) ExempleBPEL' module. The left sidebar shows a project tree with 'ExempleDeComposite' and 'Test' folders. The bottom-left pane shows the 'Logical View' of 'casaPort1' with a 'file:address' binding. The right pane shows the 'Properties' for 'file:address', including 'fileDirectory' set to '/home/henry/Bureau'.

file:address - Properties	
Properties	
fileDirectory	/home/henry/Bureau
relativePath	
pathRelativeTo	
lockName	filebc.lck
workArea	filebc_tmp
seqName	filebc.seq

fileDirectory
Defines the directory name where WSDL provisioner should read input files from, and where client should write files to.

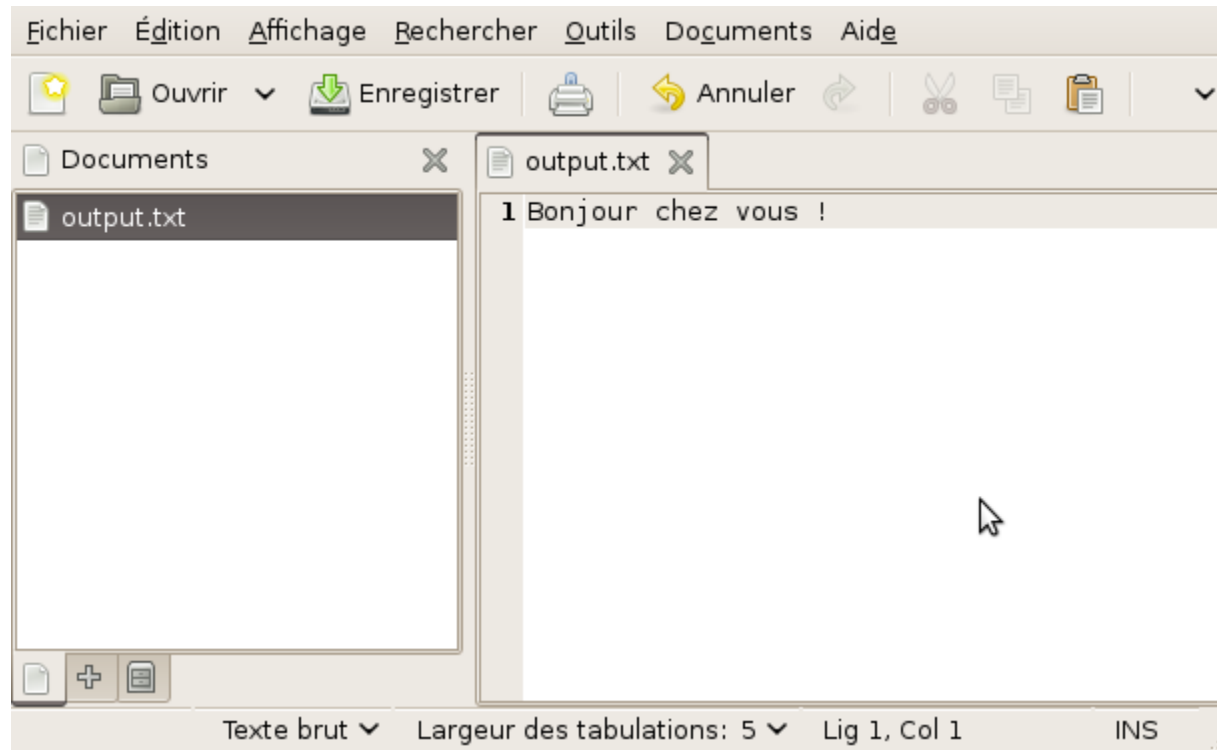
Un fichier simple en entrée



Si tout se passe bien...

À l'exécution de l'application, celle-ci a simplement
copié le contenu du fichier
input.txt
dans un nouveau fichier
output.txt

résultat



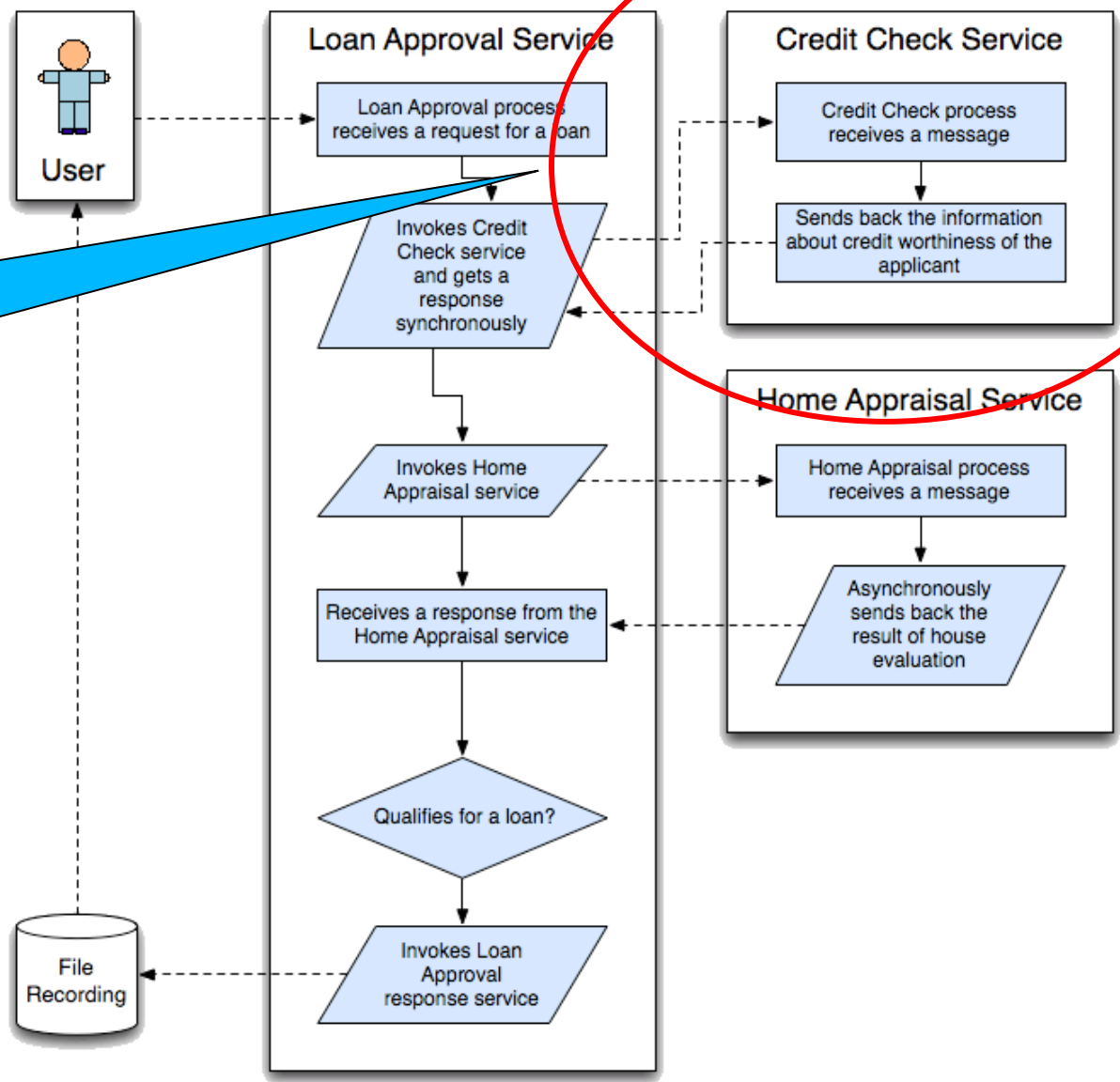
Orchestration de Web Services

Utilisation de NetBeans

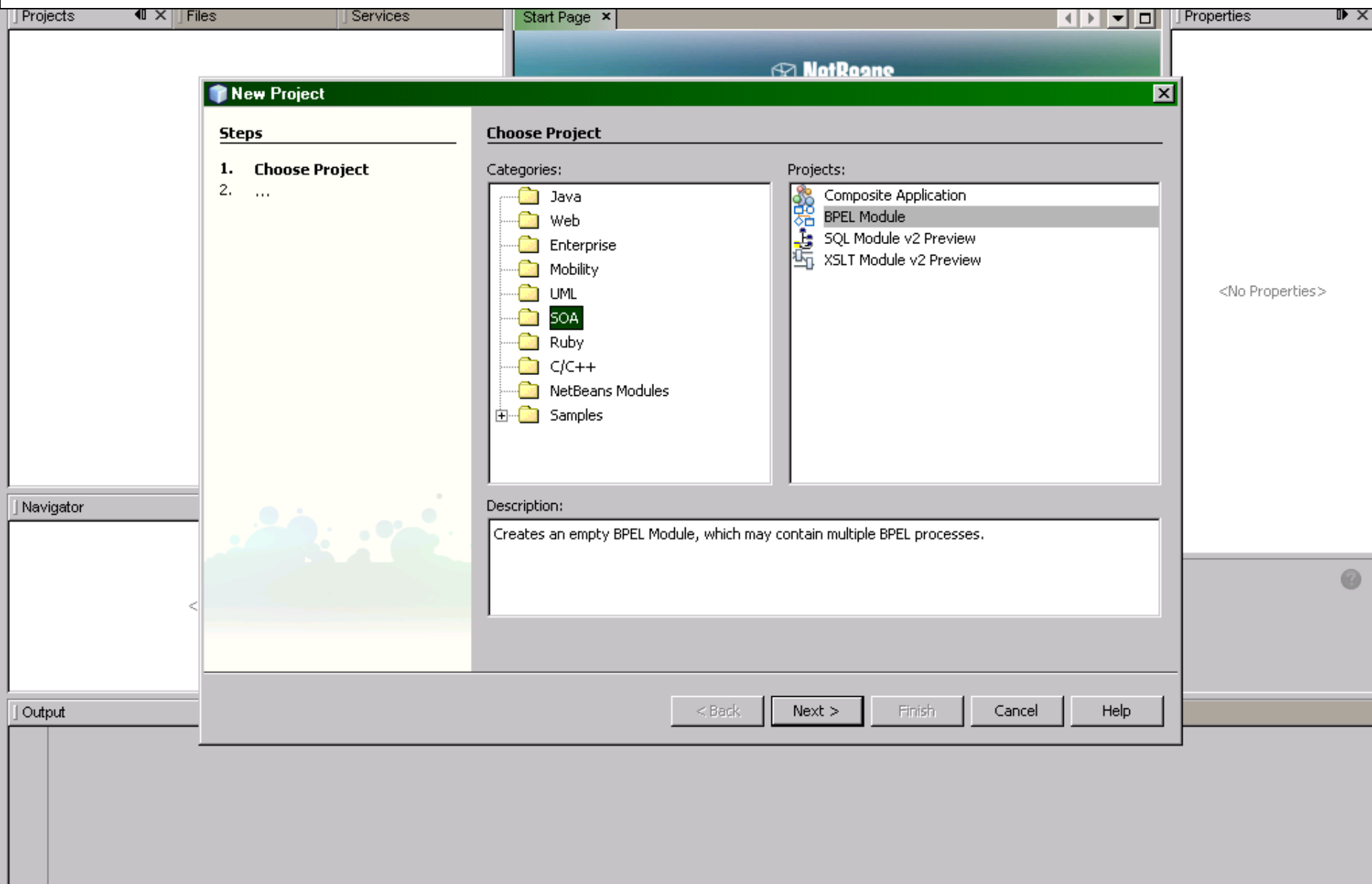
pour le développement d'un service de demande de prêt

Soit le processus suivant:

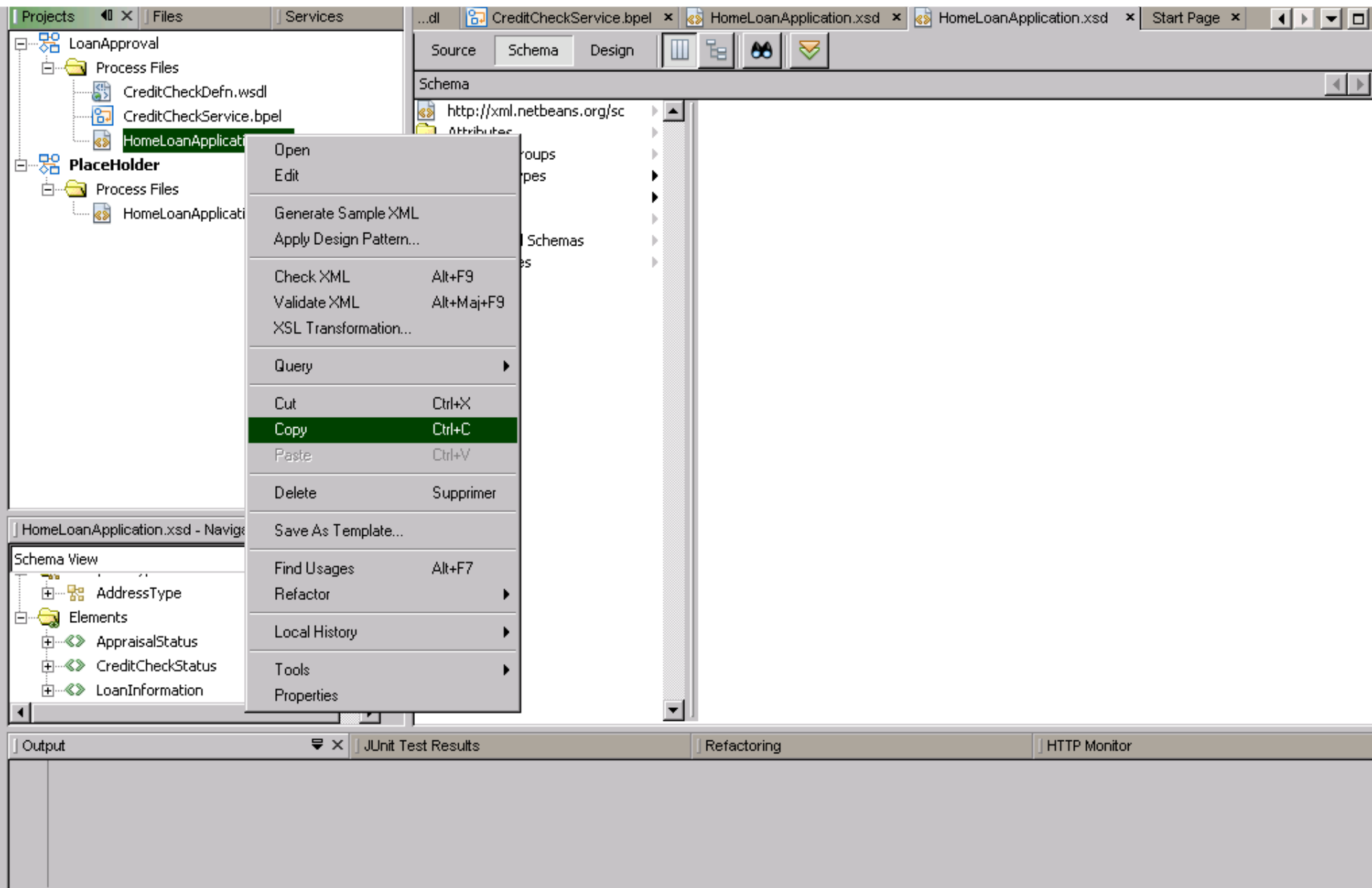
dont on commencera par implémenter cette partie



Création du module principal d'orchestration (dit BPEL)



Récupération du schéma des données



Partage du schéma de données avec le WS partenaire

The screenshot displays a software interface for configuring a WSDL document. The main window is titled "New WSDL Document" and features a "Steps" pane on the left with the following steps:

1. Choose File Type
2. Name and Location
3. **Abstract Configuration**
4. Concrete Configuration

The "Abstract Configuration" section includes the following fields:

- Port Type Name: CreditCheckDefn.wsdPortType
- Operation Name: CreditCheckDefn.wsdOperation
- Operation Type: Request-Response Operation

The "Input:" section contains a table with columns "Message Part Name" and "Element Or Type". The table has one row with "part1" in the "Message Part Name" column. Below the table are "Add" and "Remove" buttons.

The "Output:" section contains a table with columns "Message Part Name" and "Element Or Type". The table has one row with "part1" in the "Message Part Name" column. Below the table are "Add" and "Remove" buttons.

The "Fault:" section contains a table with columns "Message Part Name" and "Element Or Type". Below the table are "Add" and "Remove" buttons.

At the bottom of the configuration area, there is a checkbox labeled "Generate partnerlinktype automatically." which is checked.

A "Select Element Or Type" dialog box is open, showing a tree view of the schema structure. The tree is expanded to "src/HomeLoanApplication.xsd" and shows the following elements:

- Complex Types
- Elements
 - AppraisalStatus
 - CreditCheckStatus
 - LoanInformation** (highlighted)
- Simple Types

The "Built-In Schema Types" folder is also visible at the bottom of the tree.

The background interface shows a "Projects" pane with a tree view containing "LoanApproval", "Process Files", and "HomeLoanA". The "Design View" pane shows a list of elements: "LoanInformation", "AppraisalStatus", "CreditCheckStatus", and "AddressType". The "Output" pane is currently empty.

Définition des messages

The screenshot shows the 'New WSDL Document' dialog box in the 'Abstract Configuration' step. The 'Steps' pane on the left lists: 1. Choose File Type, 2. Name and Location, 3. **Abstract Configuration**, and 4. Concrete Configuration. The main area contains the following configuration:

- Port Type Name: CreditCheckDefn.wsdlPortType
- Operation Name: CreditCheckDefn.wsdlOperation
- Operation Type: Request-Response Operation
- Input: A table with columns 'Message Part Name' and 'Element Or Type'. It contains one entry: 'part1' with 'ns:LoanInformation'.
- Output: A table with columns 'Message Part Name' and 'Element Or Type'. It contains one entry: 'part1' with 'ns:CreditCheckStatus'.
- Fault: An empty table with columns 'Message Part Name' and 'Element Or Type'.
- Checkbox: Generate partnerlinktype automatically.

Navigation buttons at the bottom: < Back, Next >, Finish, Cancel, Help.

Création du processus lui-même

New BPEL Process

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name:

Project:

Folder:

Created File:

Target Namespace:

< Back Next > **Finish** Cancel Help

Association du partenaire fournisseur de service

The screenshot displays the BPEL editor interface. In the top-left pane, the project tree shows a folder labeled 'WSDL' circled in red. A red arrow points from this folder to a 'CreditCh...' activity icon in the main workspace. The 'Receive1 [Receive] - Property Editor' dialog is open, showing the following configuration:

- Name: Receive1
- Partner Link: CreditCheckPartner
- Operation: CreditCheckDefn.wsdlOperation
- Input Variable: (empty)
- Create Instance

The right-hand pane shows the 'Palette' with 'Web Service' and 'Basic Activities' sections. The 'Receive1 [Receive]' activity is highlighted in the palette. The bottom status bar shows 'Output', 'Refactoring', and 'HTTP Monitor' tabs.

- Choix d'opération

Choix d'opération

- Idem : Réponse, et définition de variable échangée

Reply1 [Reply] - Property Editor

Main | Correlations

Name: Reply1

Partner Link: CreditCheckPartner

Operation: CreditCheckDefn.wsdlOperation

Normal Response

Output Variable: CreditCheckDefn.wsdlOperationOut

Fault Response

Fault Name:

Fault Variable:

Le processus tel qu'il se dessine

The screenshot displays a BPEL editor interface for a process named **CreditCheckService**. The main workspace shows a flow starting with **Process Start**, leading to a **Reply1** activity. This activity is connected to an external service **CreditCh...** via two message flows: a blue arrow pointing to the service and a green arrow pointing from the service back to the **Reply1** activity. The **Reply1** activity is also connected to **Process End**.

The interface includes several panels:

- Left Panel:** A tree view showing the project structure with folders for **LoanApproval** and **Placeholder**, each containing **Process Files** and **HomeLoanApplic**.
- Bottom-Left Panel:** **BPEL Logical View** showing **Variables** (Variable2, Variable1), **Correlation Sets**, **Sequence**, and **Receive1**.
- Right Panel:** **Palette** with categories **Web Service** (Invoke, Receive, Reply, Partner Link) and **Basic Activities** (Assign, Empty, Wait, Throw, ReThrow, Exit, Compensate). A detailed view for **Reply1 [Reply]** is also visible, showing **Property Edit**, **Name** (Rep...), **Documentatio**, and **Message**.
- Bottom Panel:** **Output**, **Refactoring**, and **HTTP Monitor** tabs.

Insertion de condition

The screenshot displays a BPMN editor interface for a BPEL process named 'CreditCheckService'. The main workspace shows a process flow starting with 'Process Start', followed by a 'Receive' activity, an 'If' activity (labeled 'If1'), a 'Reply' activity (labeled 'Reply1'), and ending at 'Process End'. A green box highlights the 'If1' activity, and a red circle highlights the 'If' activity in the 'Structured Acti...' section of the Palette on the right. A red arrow points from the 'If' activity in the palette to the 'If1' activity in the diagram. The 'If1' activity is currently empty. The Palette also shows 'Basic Activities' such as Assign, Empty, Wait, Throw, ReThrow, Exit, Compensate, and CompensateScope. The 'If1 [If] - Proper...' window is open at the bottom right, showing the 'If1 [If]' activity. The 'If1 [If] - Navigator' window on the left shows the BPEL Logical View with 'If1' selected. The 'Output' window at the bottom shows 'Refactoring' and 'HTTP Monitor' tabs.

Assigination selon le cas

The screenshot displays a BPMN editor interface for a BPEL process named **CreditCheckService**. The process flow is as follows:

- Process Start** leads to a **Receive1** activity.
- Receive1** leads to an **If1** decision diamond.
- The **If1** diamond branches into two paths, each leading to an **Assign** activity (**Assign1** and **Assign2**).
- Both **Assign** activities merge back into the **If1** diamond.
- The **If1** diamond then leads to a **Reply1** activity.
- Reply1** leads to **Process End**.

The **Assign** activity is highlighted in the **Basic Activities** section of the **Palette** on the right. A red oval and arrow point to this activity. The **Palette** also shows other activities like **Invoke**, **Receive**, **Reply**, **Partner Link**, **Empty**, **Wait**, **Throw**, **ReThrow**, **Exit**, **Compensate**, **CompensateScope**, **Structured Acti...**, **If**, **While**, and **Repeat Until**.

The **Navigator** on the left shows the **BPEL Logical View** of the process, including **Receive1**, **If1**, **Assign1**, and **Else**.

The **Output** window at the bottom shows **Refactoring** and **HTTP Monitor** tabs.

Pour chaque assignation son Mapping et son traitement

The screenshot displays the IBM Business Process Manager (BPM) interface for configuring a BPEL process. The main window is titled "CreditCheckService.bpel" and is in the "Mapper" tab. A context menu is open over the "Starts With" operator, listing various string manipulation functions such as "Contains", "Normalize Space", "String", "String Length", "Substring", "Substring Before", "Substring After", "Translate", "Concat", and "String Literal".

The "Navigator" window on the left shows the "BPEL Logical View" for the "CreditCheckService" process. The "If1" activity is selected, and its sub-activities, "Assign1" and "Else", are visible. The "Output" window at the bottom shows the "Refactoring" and "HTTP Monitor" tabs.

The "If1 [If] - Navigator" window displays the following structure:

- BPEL Logical View
- CreditCheckService
- Variables
- CreditCheckDefnOperationOut tns:CreditCheckDefn.wsdlOperationOut
- CreditCheckDefnOperationIn tns:CreditCheckDefn.wsdlOperationIn
- Correlation Sets
- Sequence
- Receive1
- If1
- Assign1
- Else
- Reply1
- CreditCheckPartner

The "If1 [If] - Main" window displays the following structure:

- Main
- Name If1
- Condition ;st...
- Documentatic...



Files | Serv...

- LoanApproval
 - Process Files
 - CreditCheckDefn.w
 - CreditCheckService
 - HomeLoanApplicatio
- Placeholder
 - Process Files
 - HomeLoanApplicatio

If1 [If] - Navigator

BPEL Logical View

- CreditCheckService
 - Variables
 - CreditCheckDefnOperationO
 - CreditCheckDefnOperationIr
 - Correlation Sets
 - Sequence
 - Receive1
 - If1
 - If1
 - Assign1
 - Else
 - Reply1

Start Page | CreditCheckDefn.wsdl | CreditCheckService.bpel *

Source | Design | Mapper | Logging

Operator | Boolean | String | Node | Number | Date & Time | BPEL

All | Output

Input | All

Variables

- CreditCheckDefnOperatio
 - part1
 - Status
 - SSN
 - CreditCheckDefnOperatio
 - part1
 - BorrowerInforma
 - Name
 - SSN
 - Address
 - Addr1
 - CityState
 - Zip
 - Telephon
 - LoanDetails
 - LoanAmount
 - HomePrice
 - HomeAddress
 - Addr1
 - CityState
 - Zip
 - Telephon

Diagram: A flowchart showing a variable assignment of the string "6" to a 'Starts ...' activity. The 'Starts ...' activity has two 'String' outputs and one 'Boolean' output. The 'Boolean' output is connected to a 'Boolean Condition' activity, which is part of an 'If' block.

If1 [If] - ...

Main

Name	If1
Condition	st...
Documentatic	...

If1 [If] ?

Output | Refactoring | HTTP Monitor



Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - Placeholder
 - Process Files
 - HomeLoanApplication.xsd

Assign1 [Assign] - Navigator

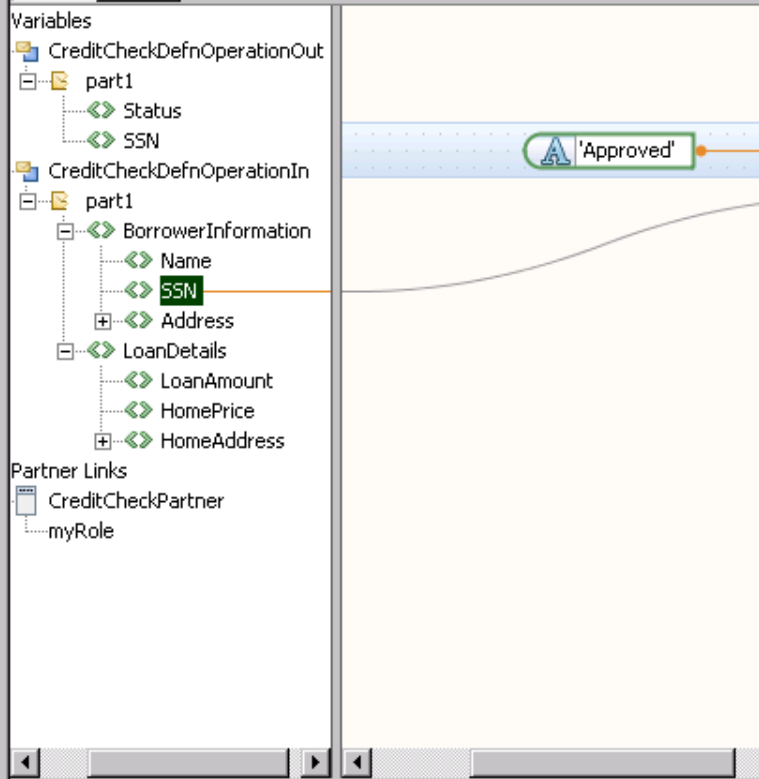
BPEL Logical View

- CreditCheckService
 - Variables
 - CreditCheckDefnOperationOut
 - part1
 - Status
 - SSN
 - CreditCheckDefnOperationIn
 - part1
 - BorrowerInformation
 - Name
 - SSN
 - Address
 - LoanDetails
 - LoanAmount
 - HomePrice
 - HomeAddress
 - Partner Links
 - CreditCheckPartner
 - myRole

Start Page x CreditCheckDefn.wsdl x CreditCheckService.bpel * x

Source Design Mapper Logging

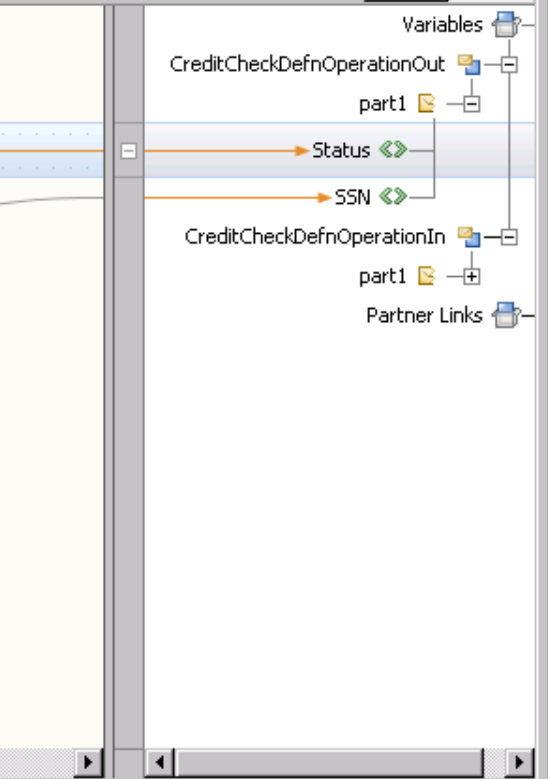
Operator Boolean String Node Number Date & Time BPEL



Assign1 [Assign] x

Main

Name As...
Assignments 2
Documentat...



Output Refactoring HTTP Monitor

Output Refactoring HTTP Monitor



Intégration dans une application composite

New Project

Steps

1. Choose Project
2. ...

Choose Project

Categories:

- Java
- Web
- Enterprise
- Mobility
- UML
- SOA
- Ruby
- C/C++
- NetBeans Modules
- Samples

Projects:

- Composite Application
- BPEL Module
- SQL Module v2 Preview
- XSLT Module v2 Preview

Description:

Creates an empty Composite Application project, which may include multiple BPEL Modules and other types of Java Business Integration (JBI) modules.

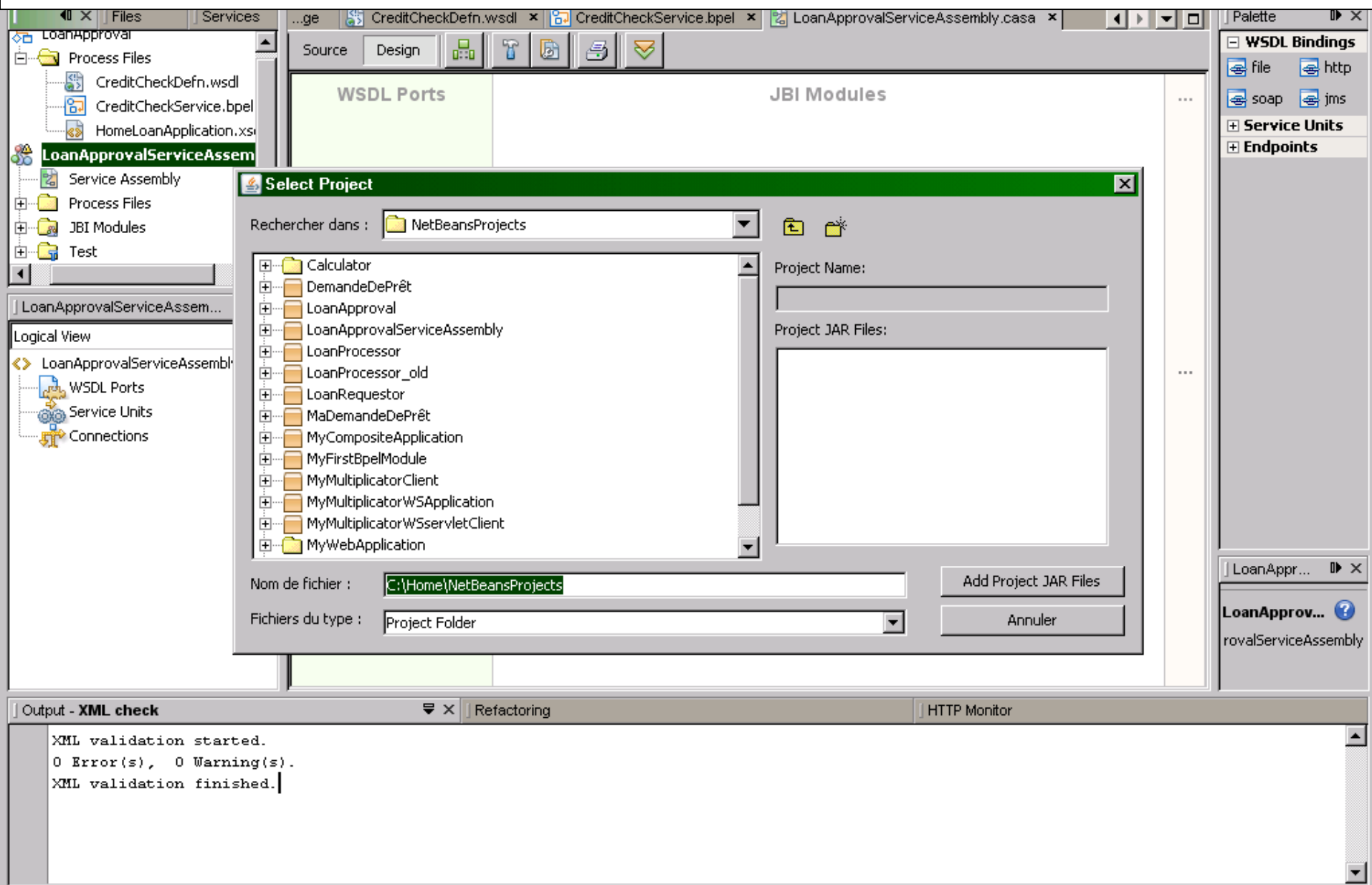
*Note: For more information, press the the **Help** button and see the on-line help section on **About Composite Application Projects**.*

< Back Next > Finish Cancel Help

Output - XML check

```
XML validation finished.  
0 Error(s), 0 Warning(s).  
XML validation finished.
```

Choix du module PBEL à intégrer

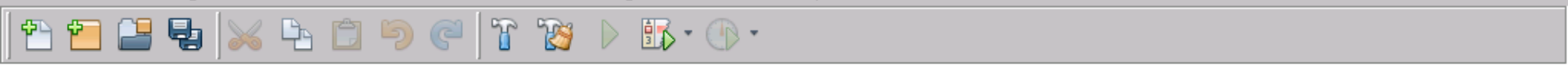


The screenshot displays the NetBeans IDE interface. The main window shows the 'LoanApprovalServiceAssembly' project with tabs for 'WSDL Ports' and 'JBI Modules'. A 'Select Project' dialog box is open, showing a list of projects in the 'NetBeansProjects' folder. The 'LoanApprovalServiceAssembly' project is highlighted. The dialog also includes fields for 'Project Name', 'Project JAR Files', and 'Nom de fichier' (set to 'C:\Home\NetBeansProjects').

The 'Select Project' dialog box contains the following elements:

- Rechercher dans : NetBeansProjects
- Project Name: [Empty text field]
- Project JAR Files: [Empty list area]
- Nom de fichier : C:\Home\NetBeansProjects
- Fichiers du type : Project Folder
- Buttons: Add Project JAR Files, Annuler

The background IDE interface shows the 'Logical View' of the 'LoanApprovalServiceAssembly' project, which includes 'WSDL Ports', 'Service Units', and 'Connections'. The bottom status bar indicates 'XML validation finished.'



Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApprovalServiceAssem**
 - Service Assembly
 - Process Files
 - JBIC Modules
 - Test

LoanApprovalServiceAssem...

Logical View

- LoanApprovalServiceAssembly
 - WSDL Ports
 - Service Units
 - LoanApproval
 - Connections

CreditCheckDefn.wsdl x CreditCheckService.bpel x LoanApprovalServiceAssembly.casa * x

Source Design

WSDL Ports

JBIC Modules

(BPEL) LoanApproval

Palette

- WSDL Bindings
 - file http
 - soap jms
- Service Units
- Endpoints

LoanAppr... ?

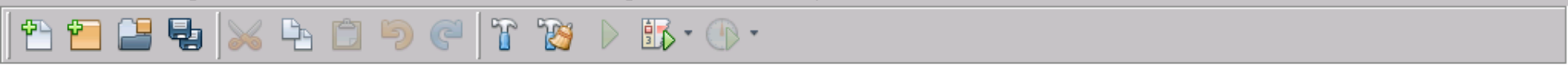
LoanApprov... ?

rovalServiceAssembly

```
XML validation started.  
0 Error(s), 0 Warning(s).  
XML validation finished.
```

Construction en cours...

The screenshot displays a Java IDE interface for developing a service assembly. The main workspace is in Design View, showing a diagram with two main sections: **WSDL Ports** and **JB Modules**. In the WSDL Ports section, there is a **SOAP** port icon connected to a **WSDLPORTTYPEBINDINGPORT** component. A blue arrow points from this port to a **(BPEL) LoanApproval** module in the JB Modules section. This module contains a **CreditCheckService** and a **CreditCheckPartner** component. A **Reloading .** dialog box is overlaid on the diagram, indicating that the service assembly is being updated. The left sidebar shows a project tree with **LoanApprovalServiceAssem** selected. The bottom status bar shows **Start Server GlassFish V2**.



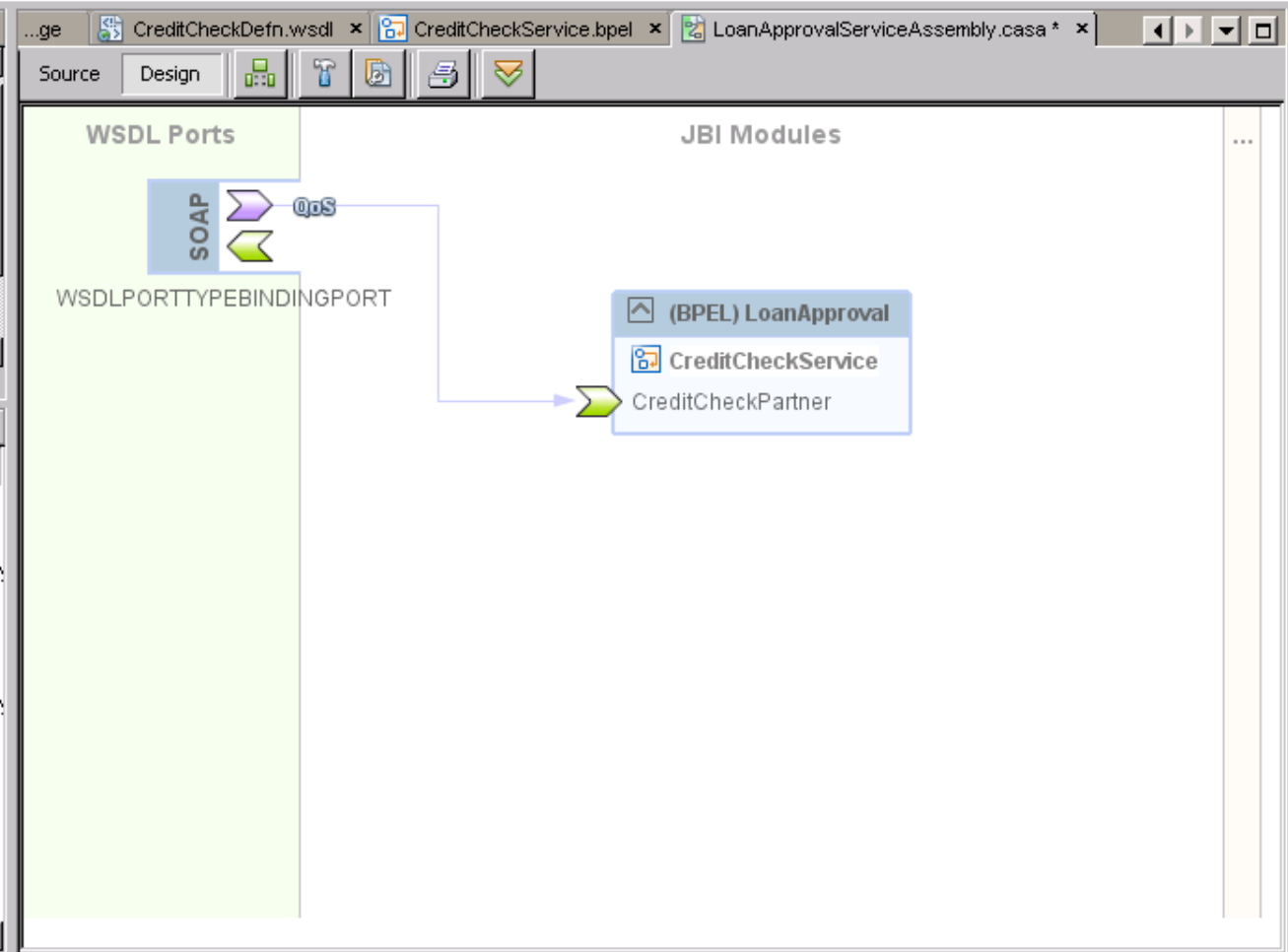
Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApprovalServiceAssem**
 - Service Assembly
 - Process Files
 - JBIC Modules
 - Test

LoanApprovalServiceAssem...

Logical View

- LoanApprovalServiceAssembly
 - WSDL Ports
 - CreditCheckDefn.wsdlPortT...
 - Service Units
 - LoanApproval
 - Connections
 - CreditCheckDefn.wsdlPortT...



Palette

- WSDL Bindings
 - file
 - http
 - soap
 - jms
- Service Units
- Endpoints

LoanAppr...

LoanApprov... ?

rovalServiceAssembly

```
XML check x build.xml (run) x Java DB Database Process x GlassFish V2 x
host=localhost
port=4848
name=LoanApprovalServiceAssembly
run:
BUILD SUCCESSFUL (total time: 3 minutes 52 seconds)
```

définition de test

The screenshot displays an IDE interface for configuring a test case in a BPEL environment. The main workspace shows a diagram with two sections: "WSDL Ports" and "JBIC Modules". A "SOAP" port is connected to a "CreditCheckPartner" module. The left sidebar shows a project tree with "LoanApprovalServiceAssembly" and a "Test" folder containing a "New Test Case" button. The bottom pane shows a "BPEL Logical View" for "CreditCheckService" with a sequence of "Receive1", "If1", and "Assign1" activities. The bottom status bar shows a successful build message: "BUILD SUCCESSFUL (total time: 3 minutes 52 seconds)".



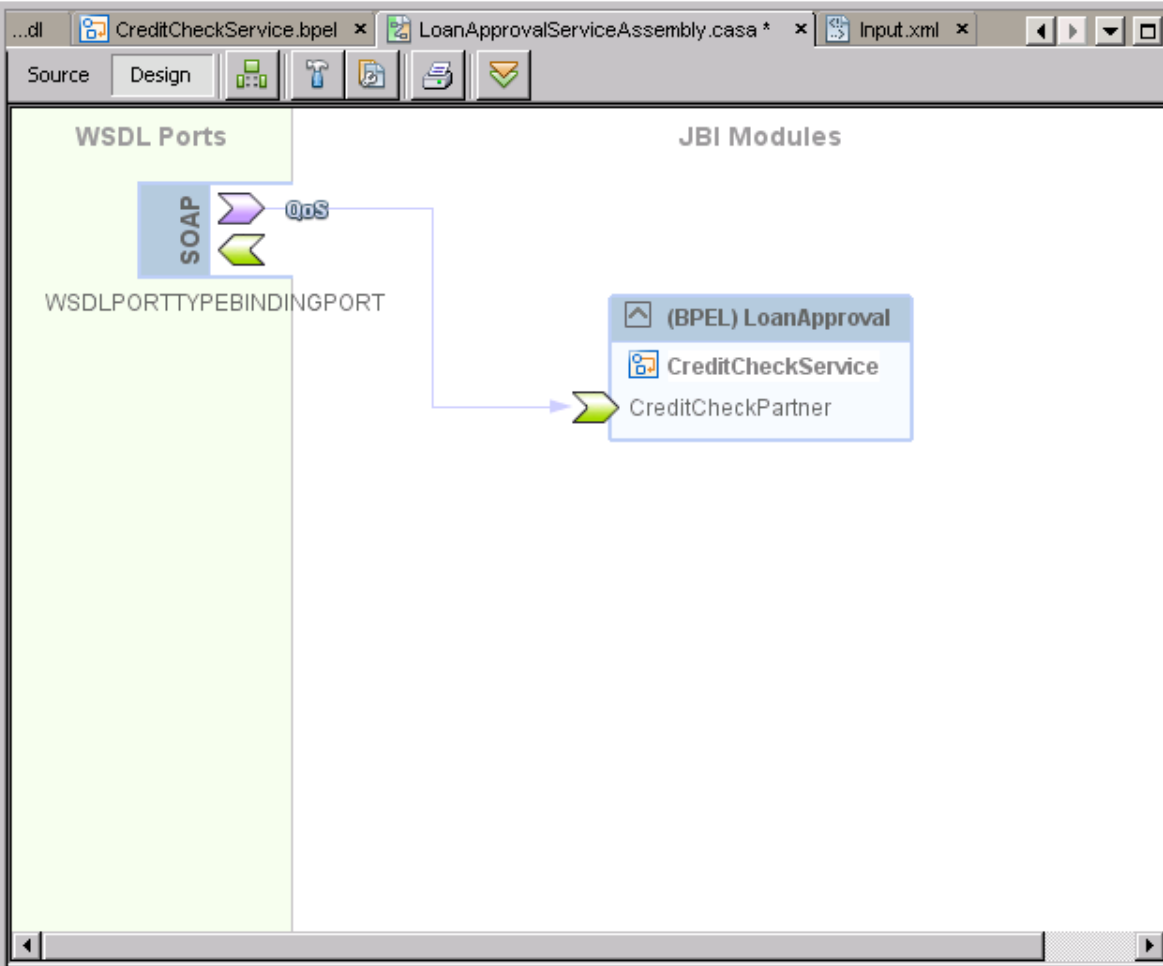
Project Explorer showing the structure of the LoanApprovalServiceAssembly project. The tree view includes:

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
- LoanApprovalServiceAssembly
 - Service Assembly
 - Process Files
 - JBIC Modules
 - Test

Context menu options: New Test Case, Delete All Results

Logical View

- LoanApprovalServiceAssembly
 - WSDL Ports
 - CreditCheckDefn.wsdlPortTypeBindingPor
 - Service Units
 - LoanApproval
 - Connections
 - CreditCheckDefn.wsdlPortTypeBindingPor



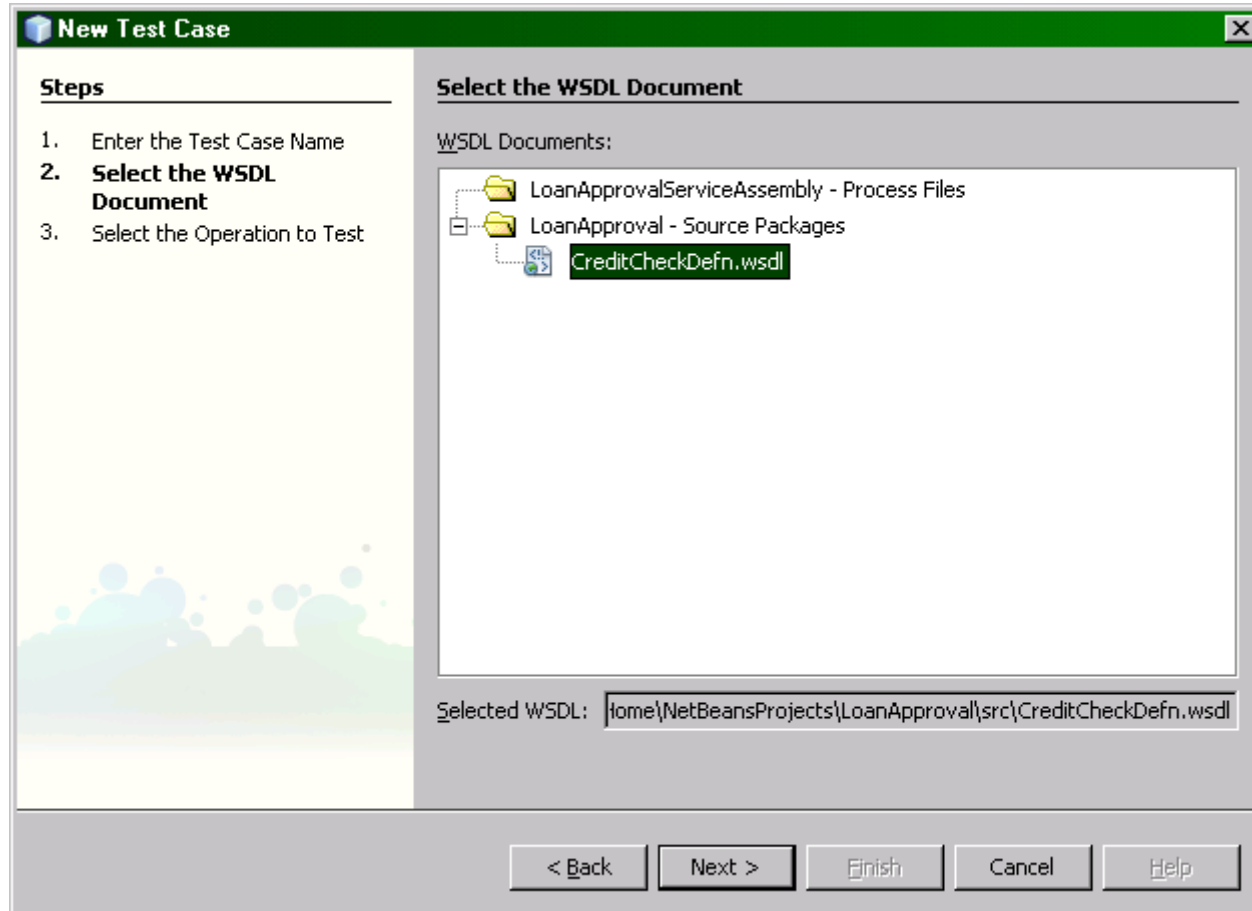
Palette showing WSDL Bindings, Service Units, and Endpoints. The 'WSDL Bindings' section includes file, http, soap, and jms. The 'Service Units' and 'Endpoints' sections are currently empty.

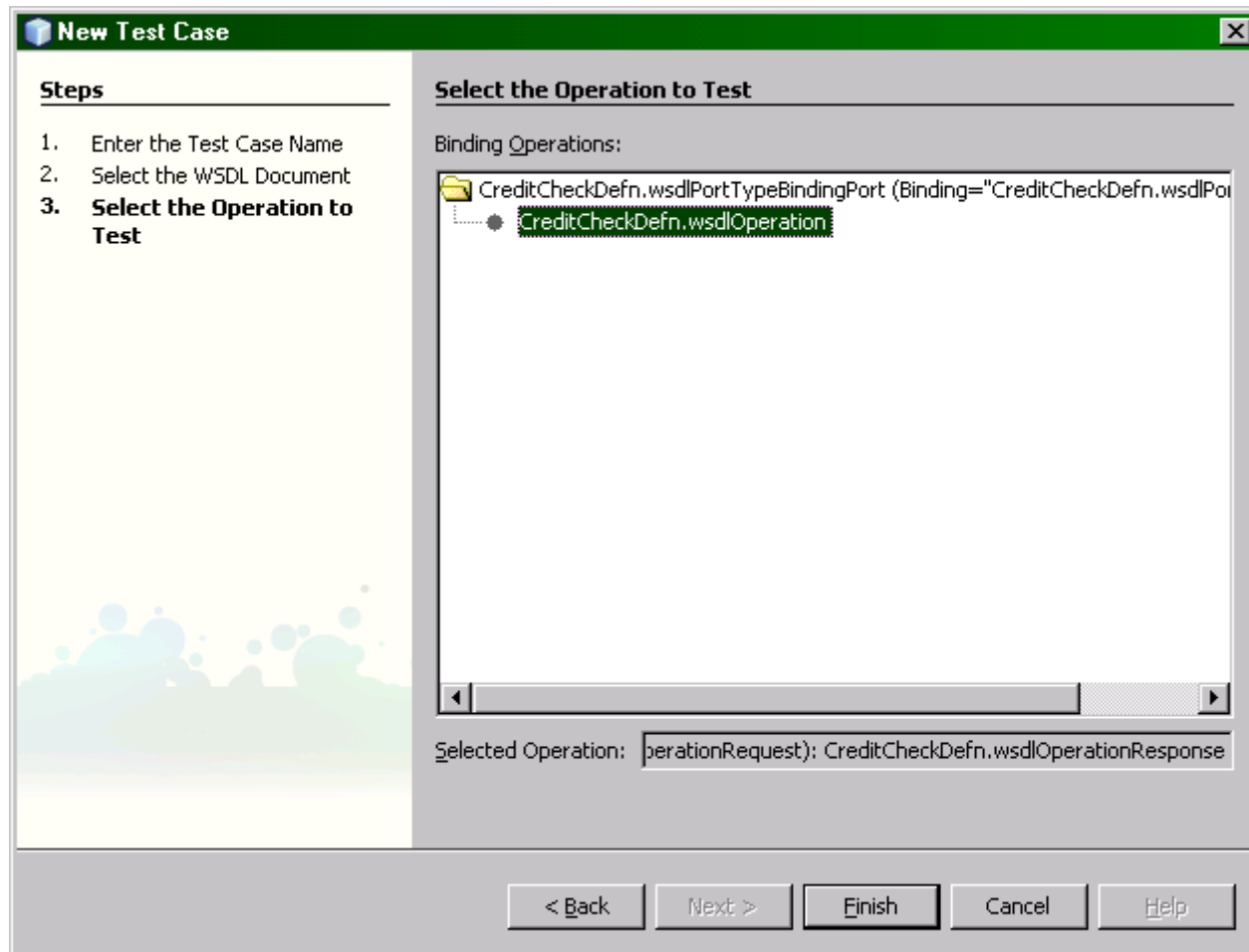
Test - Pro... [Play]

Test [?] test

```
XML check x build.xml (run) x Java DB Database Process x GlassFish V2 x
host=localhost
port=4848
name=LoanApprovalServiceAssembly
run:
BUILD SUCCESSFUL (total time: 3 minutes 52 seconds)
```


Choix du service à tester





Et ça plante !

The screenshot displays the NetBeans IDE interface. The top toolbar shows the 'Design' tab selected. The main workspace contains a BPEL process diagram with the following elements:

- Process Start
- Receive1 (Message In)
- If1 (Exclusive Split)
- Assign1 (Assign)
- Assign2 (Assign)
- Reply1 (Message Out)

The 'BPEL Logical View' on the left shows the structure of the process, including the 'Assign2' element. The 'Output' window at the bottom shows the following error message:

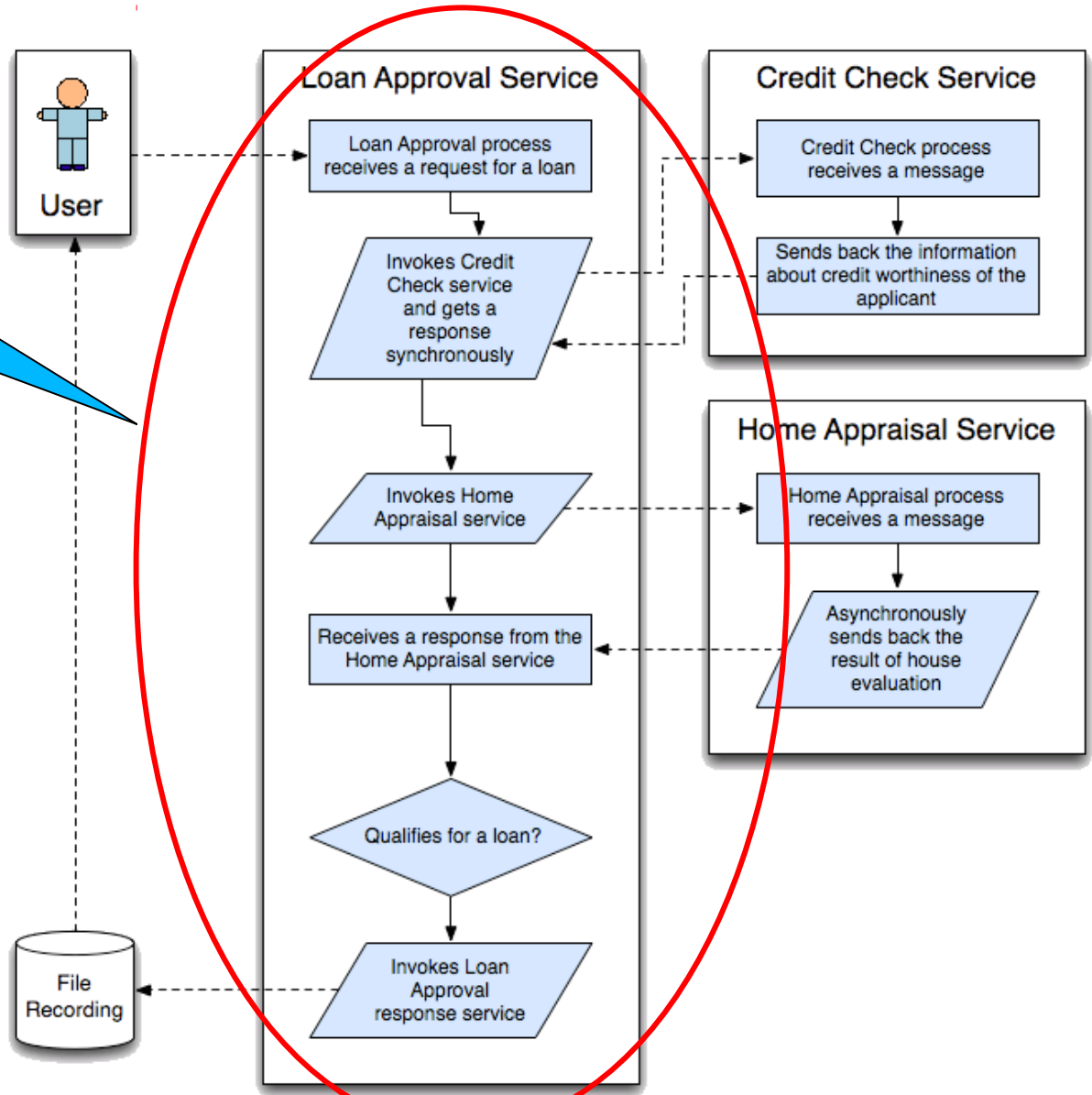
```
No test passed, 1 test caused an error.  
org.netbeans.modules.compapp.catd.ConfiguredTest FAILED  
TestCreditCheckService caused an ERROR (8,933 s)  
Unknown attribute indent-number  
java.lang.IllegalArgumentException  
at net.sf.saxon.Configuration.setConfigurationProperty(Configuration.java:2211)  
at net.sf.saxon.TransformerFactoryImpl.setAttribute(TransformerFactoryImpl.java:  
at org.netbeans.modules.compapp.catd.ConfiguredTest.replyAsByteArrayOS(Conf  
at org.netbeans.modules.compapp.catd.ConfiguredTest.checkExpectedOutput(Coi  
at org.netbeans.modules.compapp.catd.ConfiguredTest.testConcurrentSOAPRequ
```

The 'Refactoring' window on the right shows the following code snippet:

```
{java.runtime.name=Java(TM) SE Runtime Environment, sun.boot.library.  
, java.vm.specification.vendor=Sun Microsystems Inc., user.variant=  
Test TestCreditCheckService\Concurrent.properties  
Replace '${HttpDefaultPort}' in WSDL soap location by '9080' defined  
Failed. (Destination: http://localhost:${HttpDefaultPort}/CreditChec
```

Soit le processus suivant:

Suite du sujet





Projects Files Services

LoanApproval

- Process
 - Cred
 - Cred
 - Hom
- PlaceHolder
 - Process I
 - Hom

New

- BPEL Process...
- WSDL Document...
- XML Schema...
- Other...

Local History

Find... Ctrl+F

Paste Ctrl+V

Tools

HomeLoanApplication.xsd - Navigator

Design View

- LoanInformation
- AppraisalStatus
- CreditCheckStatus
- AddressType

Output

JUnit Test Results

Refactoring

HTTP Monitor

New WSDL Document [X]

Steps

1. Choose File Type
- 2. Name and Location**
3. Abstract Configuration
4. Concrete Configuration

Name and Location

File Name:

Project:

Folder:

Created File:

Target Namespace:

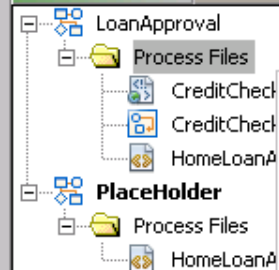
Import XML Schema File(s)

XML Schema(s):

< Back Next > Finish Cancel Help



Projects Files Services



New WSDL Document

Steps

1. Choose File Type
2. Name and Location
3. **Abstract Configuration**
4. Concrete Configuration

Abstract Configuration

Port Type Name:

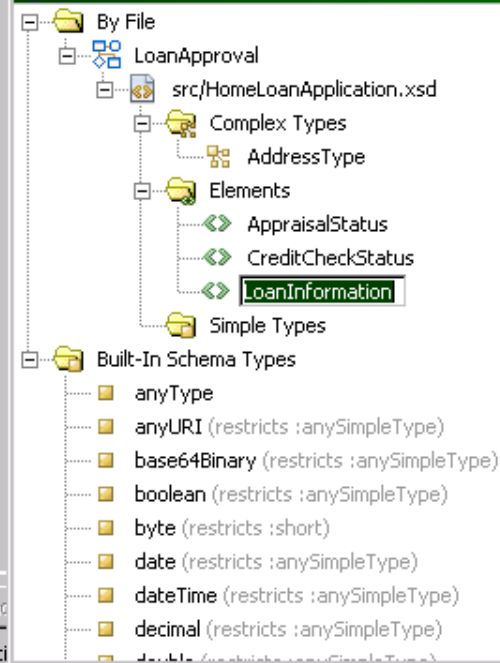
Operation Name:

Operation Type:

Input:

Message Part Name	Element Or Type
part1	xsd:string

Select Element Or Type



Add

Remove

Generate partnerlinktype automatically

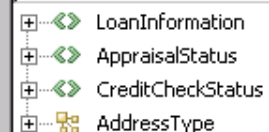
OK

Cancel

Help

HomeLoanApplication.xsd

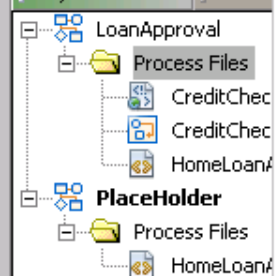
Design View



Output



Projects Files



HomeLoanApplication.xsd

Design View



Output - Standard output

WS-I Basic Prof

New WSDL Document

Steps

1. Choose File Type
2. Name and Location
3. Abstract Configuration
4. **Concrete Configuration**

Concrete Configuration

Binding Name: Binding Type: Binding Subtype: RPC Literal Document Literal RPC EncodedService Name: Port Name:

< Back

Next >

Finish

Cancel

Help



Projects | Files | Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - Placeholder
 - Process Files
 - HomeLoanApplication.xsd

LoanApproveDefn.wsdl x

Source WSDL Partner

- http://j2ee.netbeans.org/wsdl/LoanApproveDefn
 - Types
 - http://j2ee.netbeans.org/wsdl/LoanApproveDefn
 - Imports
 - Messages
 - LoanApproveDefnOperationRequest
 - Port Types
 - LoanApproveDefnPortType
 - Bindings
 - LoanApproveDefnBinding PortType="LoanApproveDefnPortType"
 - Services
 - LoanApproveDefnService
 - Extensibility Elements
 - LoanApproveDefn

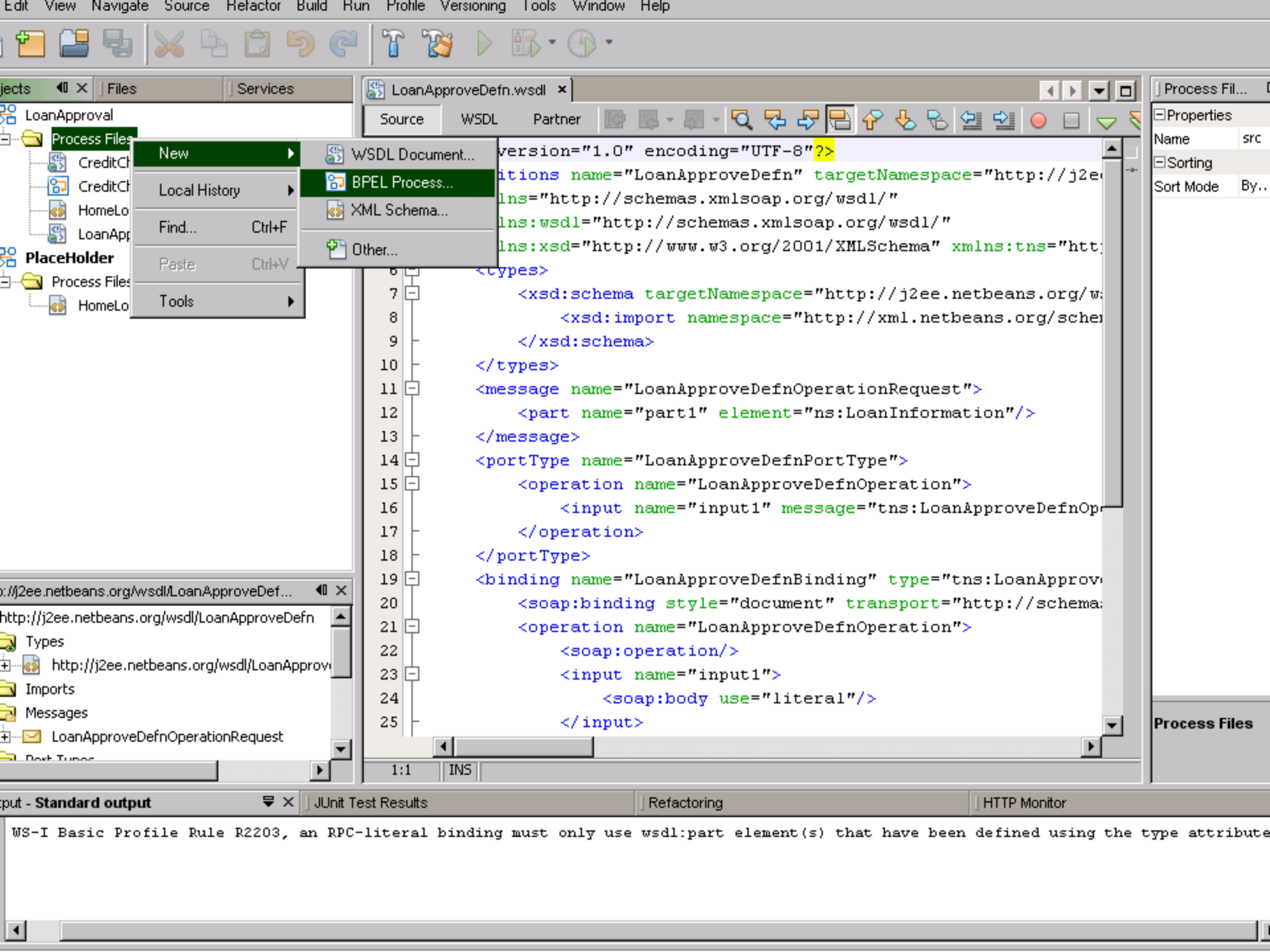
No Proper...

<No Properties>

http://j2ee.netbeans.org/wsdl/LoanApproveDefn...

- LoanApproveDefnPortType
- Bindings
 - LoanApproveDefnBinding PortType="LoanA"
- Services
 - LoanApproveDefnService
- Extensibility Elements
 - LoanApproveDefn

```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```



New BPEL Process

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name:

Project:

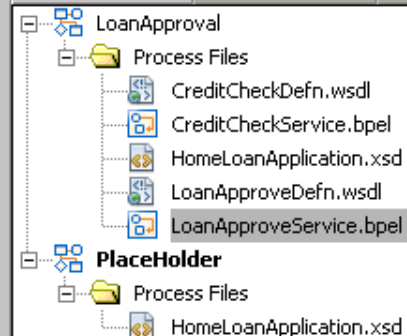
Folder:

Created File:

Target Namespace:

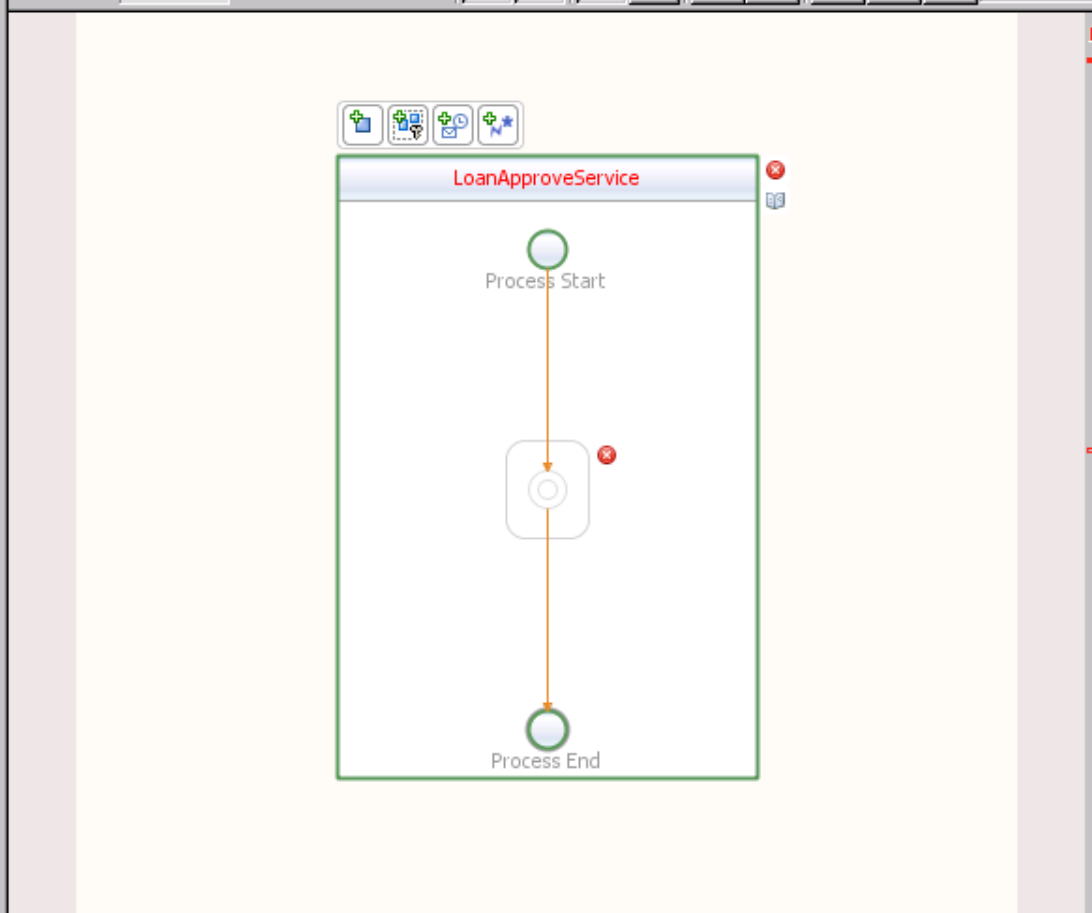


Projects Files Services



LoanApproveDefn.wsdl LoanApproveService.bpel

Source Design Mapper Logging



Palette

Web Service

- Invoke
- Receive
- Reply
- Partner Link

Basic Activi...

- Assign
- Empty
- Wait
- Throw
- ReThrow
- Exit
- Compensate
- CompensateSc...

Structured ...

- If
- While
- RepeatUntil

LoanAppro...

Main

LoanApprove...

Output - Standard output

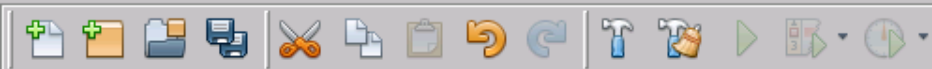
JUnit Test Results

Refactoring

HTTP Monitor

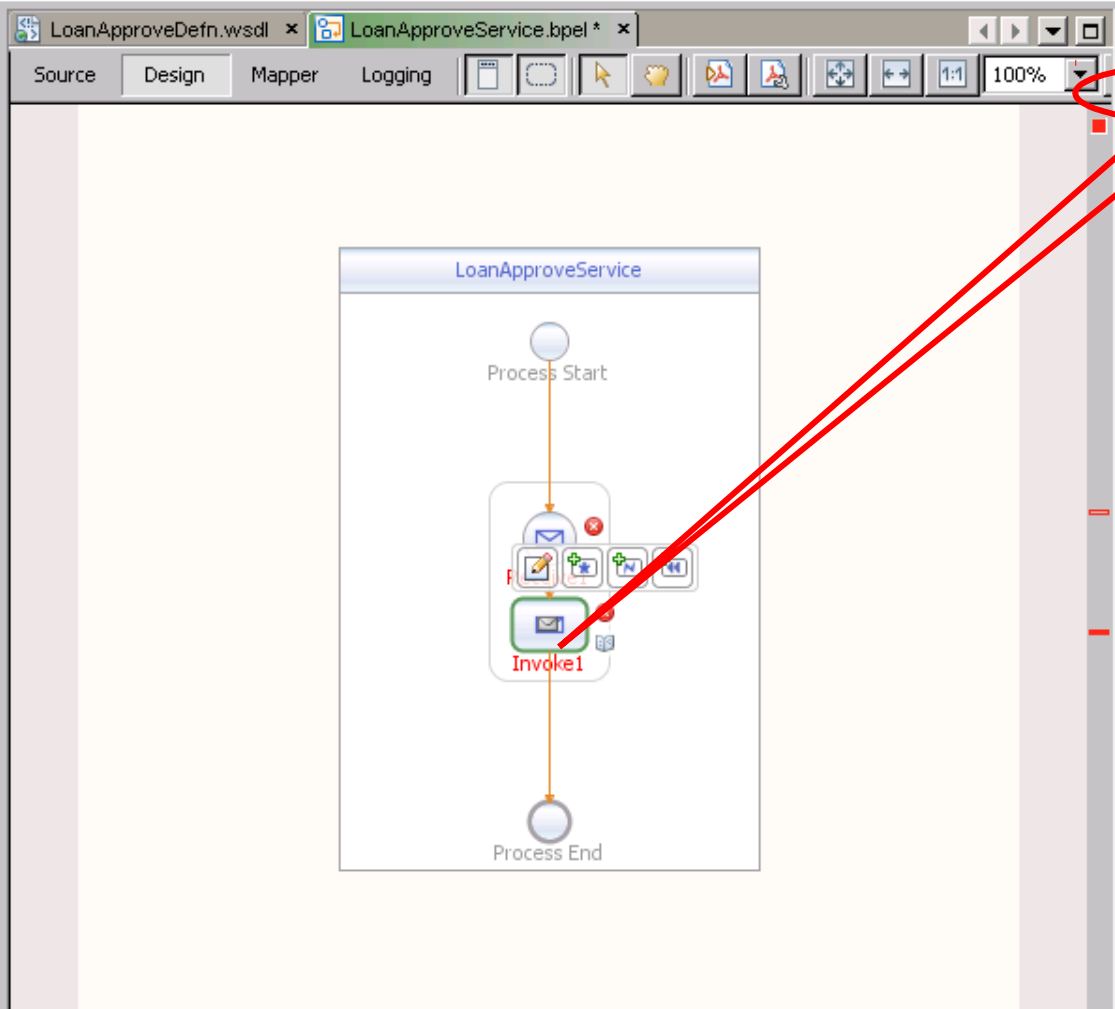
```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```

The screenshot displays the NetBeans IDE 6.1 interface for developing a BPEL process. The main workspace shows a BPEL diagram for `LoanApproveService` with a flow from `Process Start` to `Receive1` and then to `Process End`. A red circle highlights the `Receive` activity in the `Web Service` category of the Palette on the right, with a red arrow pointing to the `Receive1` activity in the diagram. The Palette also shows `Basic Activi...` and `Structured ...` categories. The Navigator on the left shows the `BPEL Logical View` for `LoanApproveService`, including `Variables`, `Correlation Sets`, `Sequence`, `Receive1`, and `Imports`. The Output window at the bottom shows a warning message: `WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.`



Projects | Files | Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- Placeholder
 - Process Files
 - HomeLoanApplication.xsd



Palette

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activi...
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateSc...
- Structured ...
 - If
 - While
 - RepeatUntil

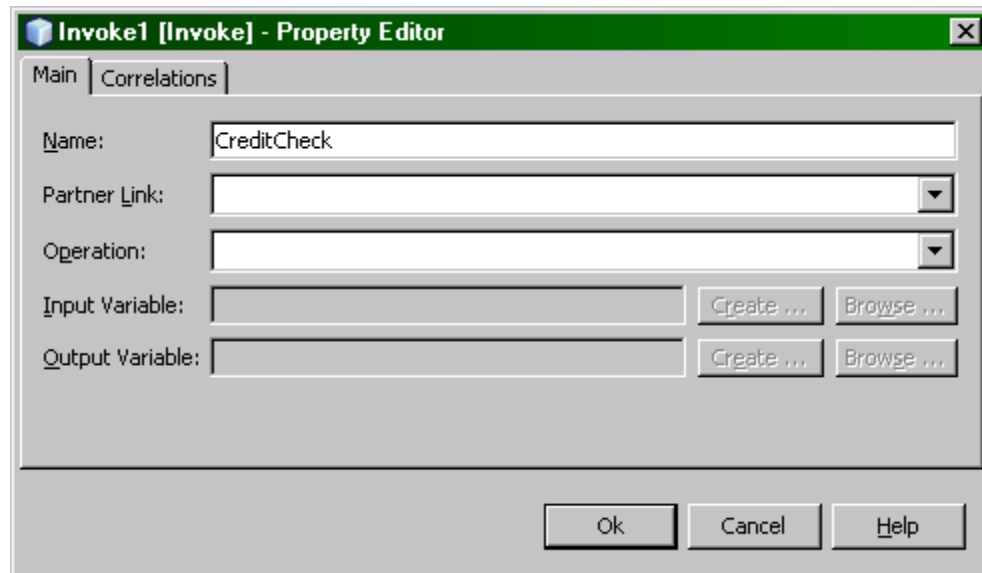
Invoke1 [In... | Main | Invoke1 [Invo... ?

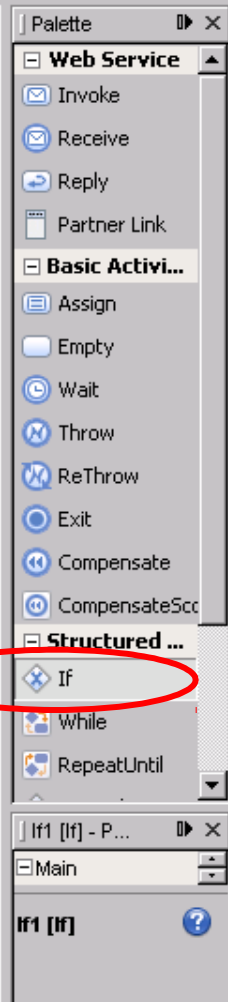
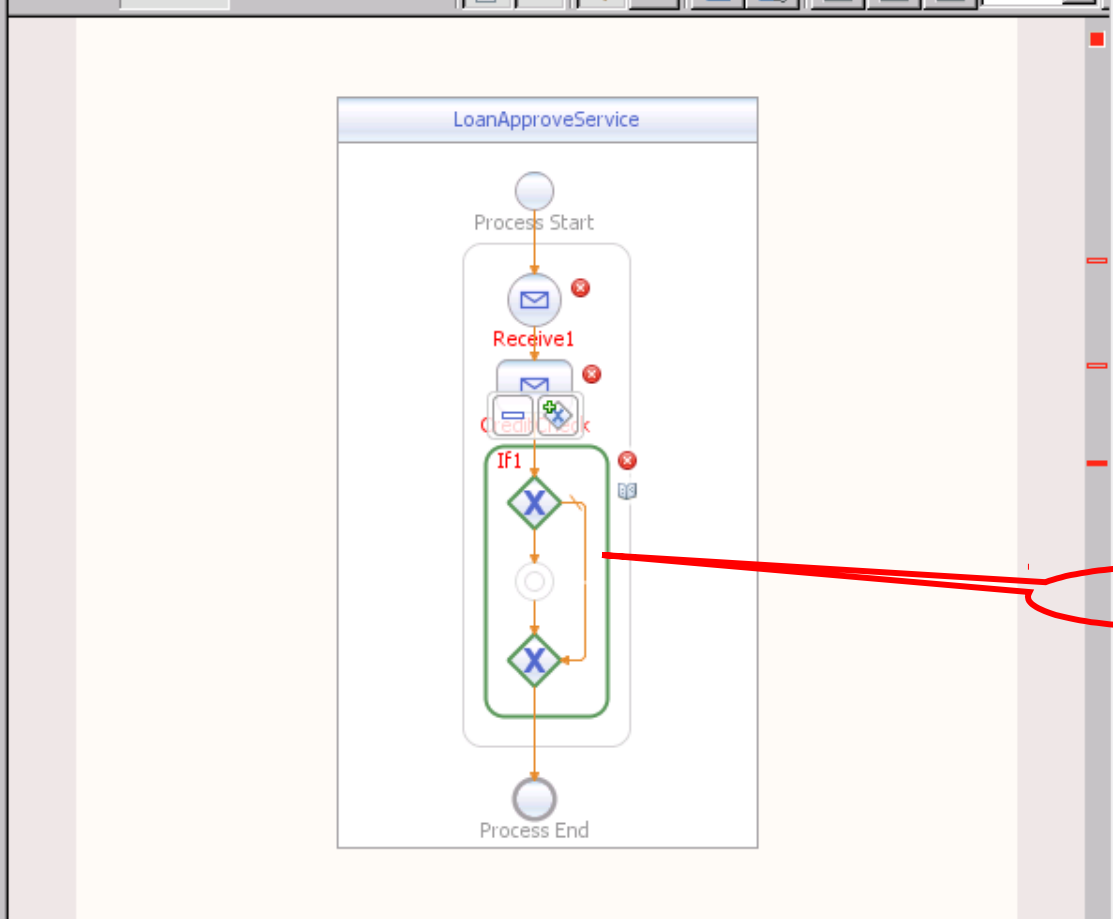
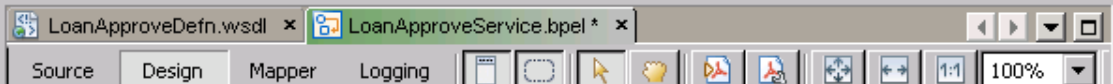
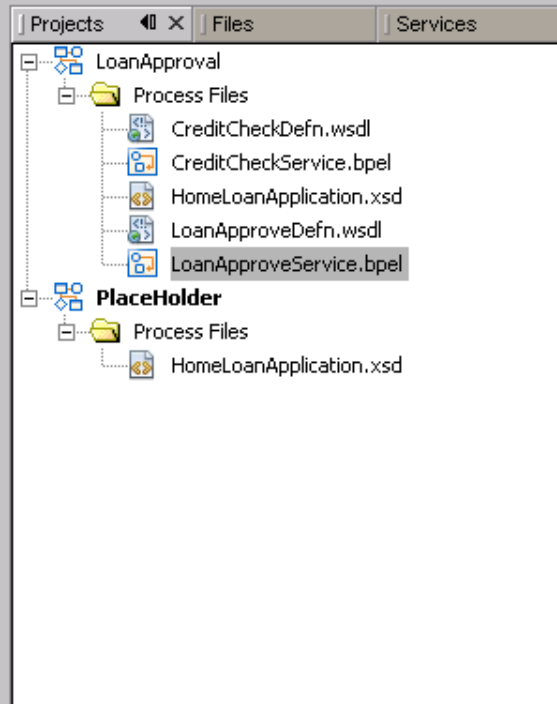
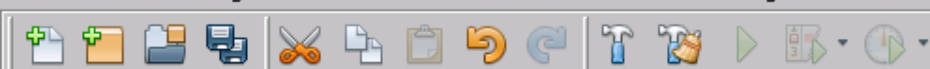
Invoke1 [Invoke] - Navigator

BPEL Logical View

- LoanApproveService
 - Variables
 - Correlation Sets
 - Sequence
 - Receive1
 - Invoke1

```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```





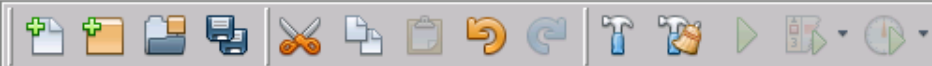
Output - Standard output

JUnit Test Results

Refactoring

HTTP Monitor

```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```

Projects | Files | Services

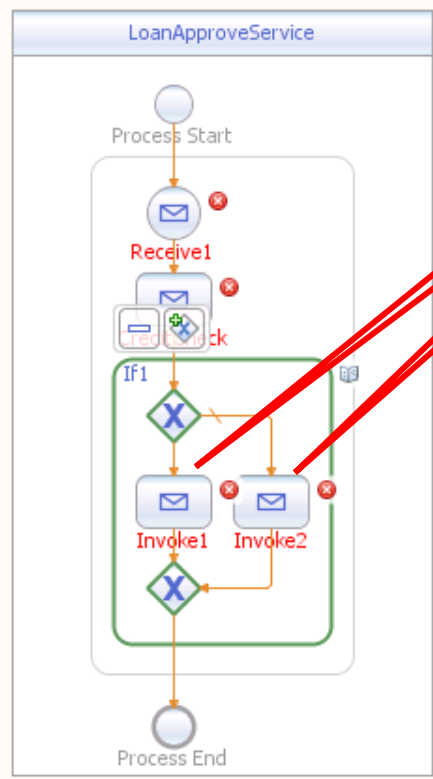
- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - Placeholder
 - Process Files
 - HomeLoanApplication.xsd

If1 [If] - Navigator

BPEL Logical View

- Receive1
- CreditCheck
- If1
 - If1
 - Invoke1
 - Else

LoanApproveDefn.wsdl | LoanApproveService.bpel * | Source | Design | Mapper | Logging | 100%



Palette

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activi...
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateSc...
- Structured ...
 - If
 - While
 - RepeatUntil

If1 [If] - P... | Main | If1 [If]

Output - Standard output | JUnit Test Results | Refactoring | HTTP Monitor

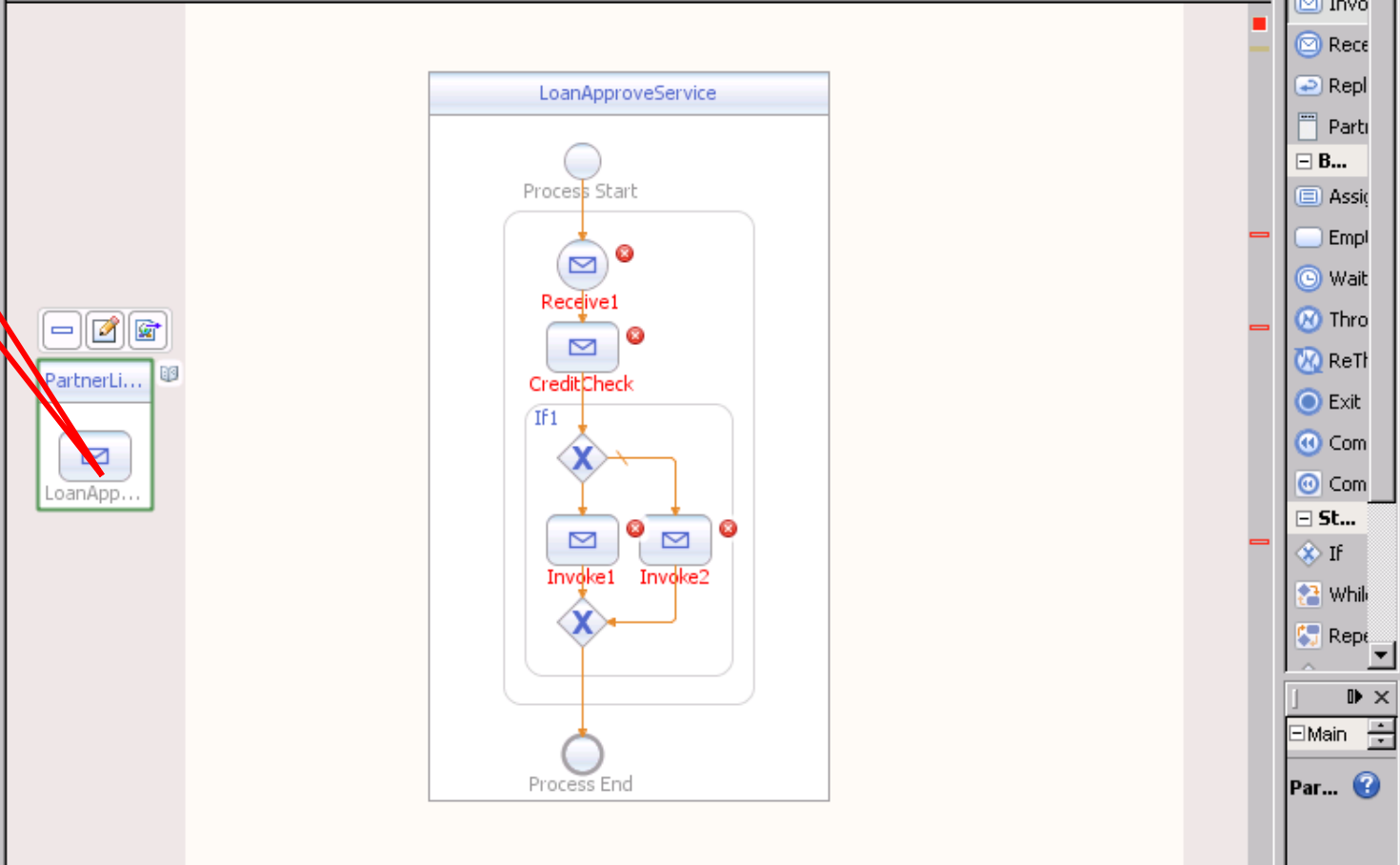
```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```



Pr... | Files | Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl**
 - LoanApproveService.bpel
 - Placeholder
 - Process Files
 - HomeLoanApplication.xsd

LoanApproveDefn.wsdl | LoanApproveService.bpel * | Source | Design | Mapper | Logging | 100%



PartnerLink1 [Partner Link] - Navig...

BPEL Logical View

- CreditCheck
- If1
 - If1
 - Invoke1
 - Else
- PartnerLink1

Output - Standard output | JUnit Test Results | Refactoring | HTTP Monitor

WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.

LoanApprovePartner [Partner Link] - Property Editor

Name:

WSDL File:

Use Existing Partner Link Type

Partner Link Type:

My Role:

Partner Role:

Use a Newly Created Partner Link Type

Create in File:

Partner Link Type Name:

Process will implement (My Role)

Role Name:

Port Type:

Partner service will implement (Partner Role)

Role Name:

Port Type:



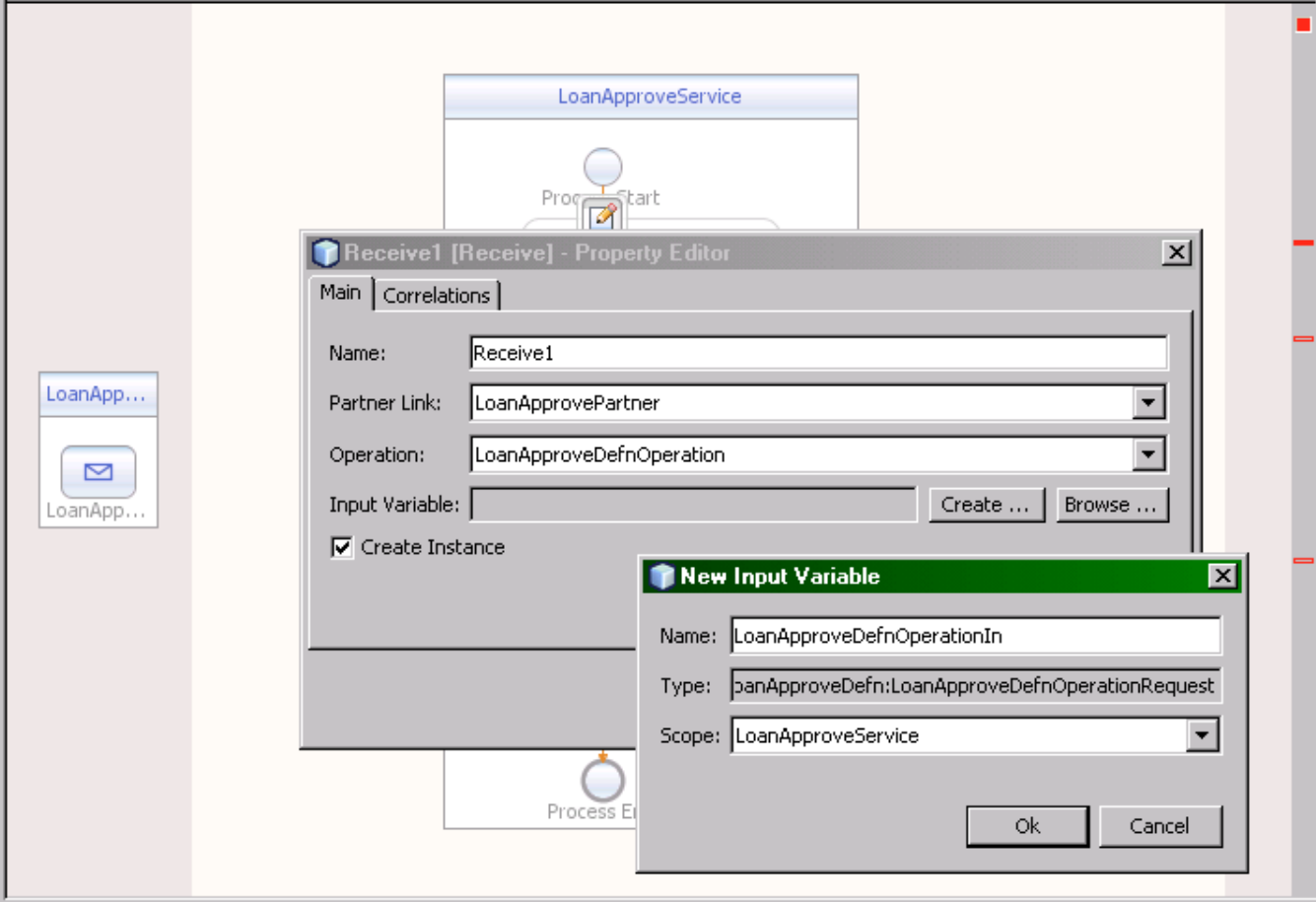
Pr... Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - Placeholder
 - Process Files
 - HomeLoanApplication.xsd

LoanApproveDefn.wsdl LoanApproveService.bpel * x

Source Design Mapper Logging

100%



Receive1 [Receive] - Property Editor

Main | Correlations

Name:

Partner Link:

Operation:

Input Variable:

Create Instance

New Input Variable

Name:

Type:

Scope:

Receive1 [Receive] - Navigator

BPEL Logical View

- Receive1
- CreditCheck
- If1
 - If1
 - Invoke1
- Else

```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```



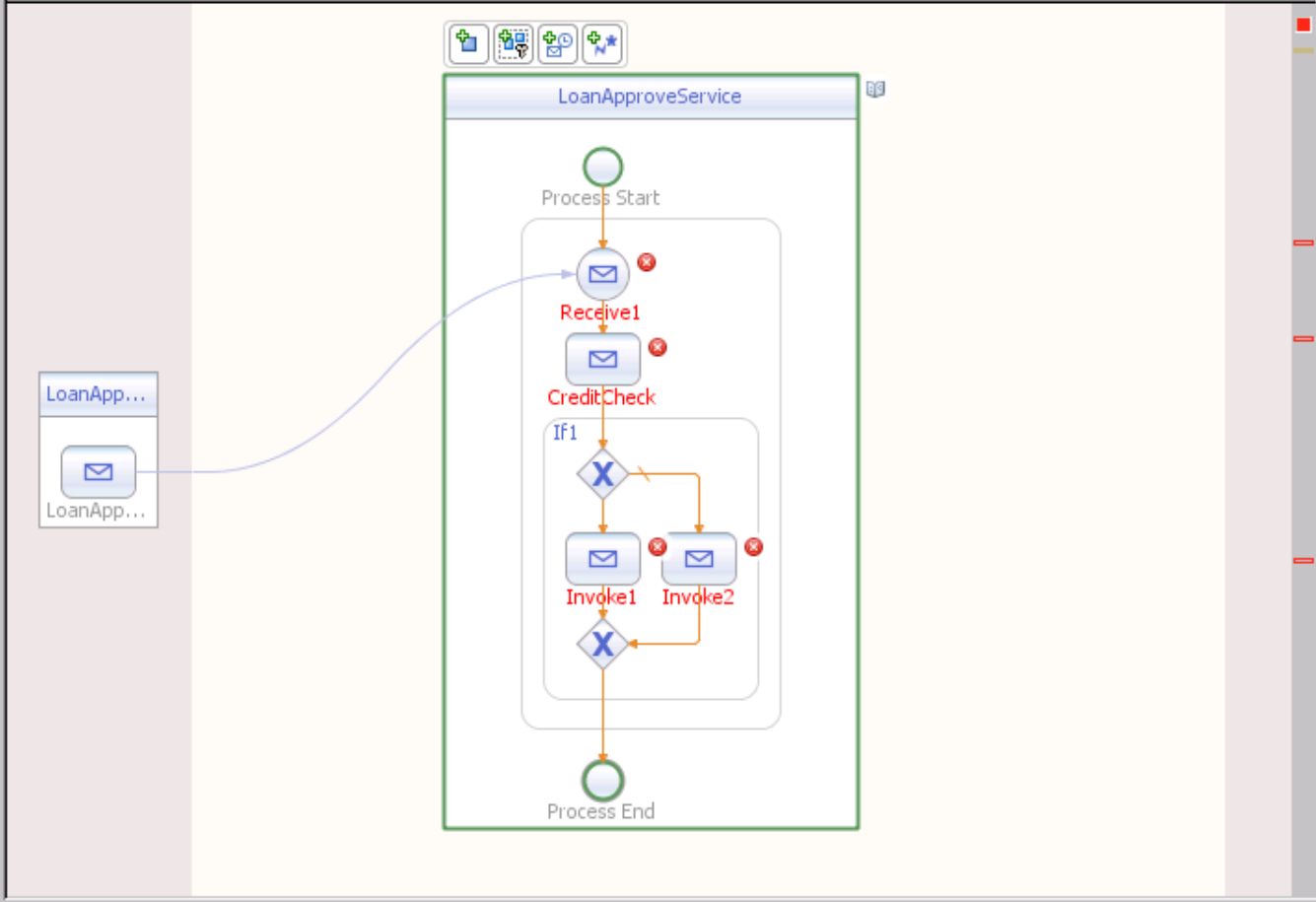
Pr... Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- Placeholder
 - Process Files
 - HomeLoanApplication.xsd

LoanApproveDefn.wsdl x LoanApproveService.bpel * x

Source Design Mapper Logging

100%



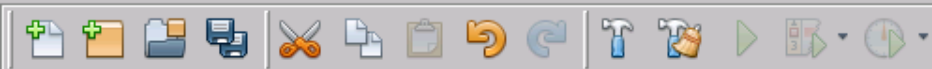
W... | Invo | Rece | Repl | Part | B... | Assi | Empl | Wait | Thro | ReTh | Exit | Com | Com | St... | If | Whil | Rept

Main

Loa...

Output - Standard output | JUnit Test Results | Refactoring | HTTP Monitor

WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.



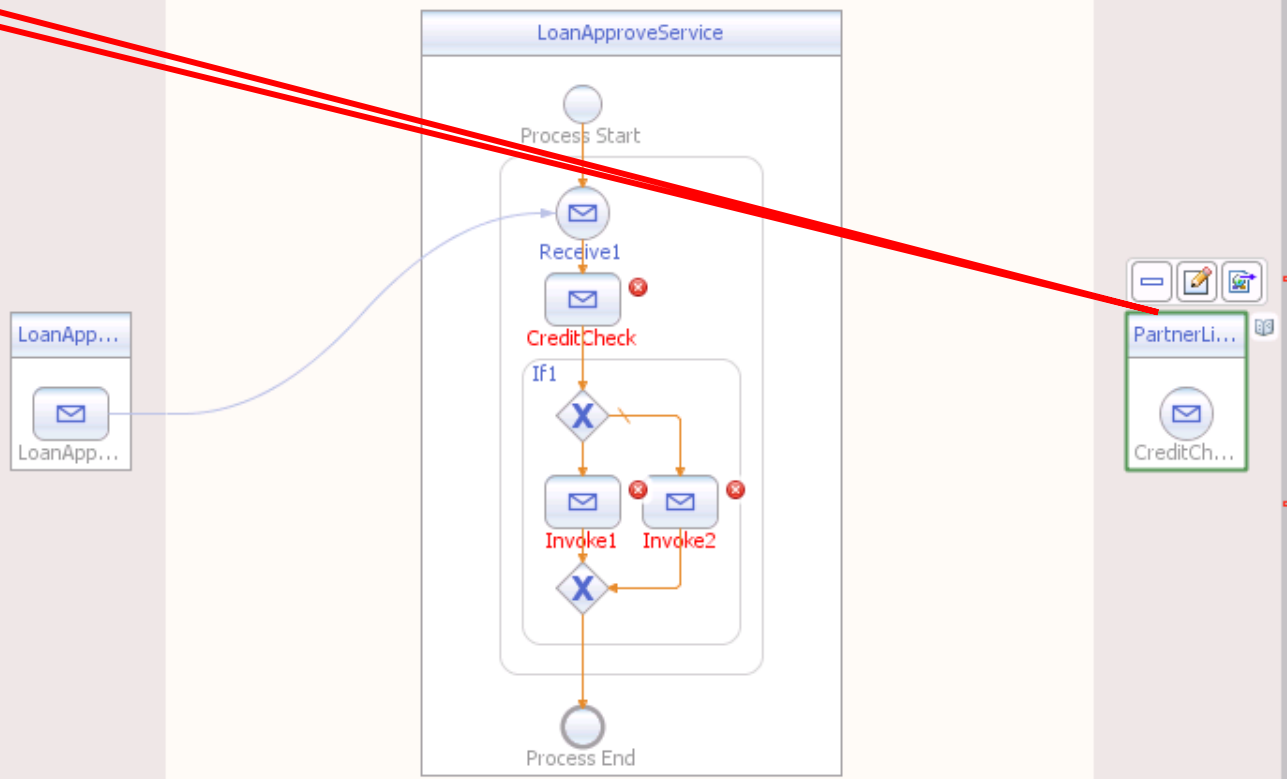
Pr... Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl**
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - Placeholder
 - Process Files
 - HomeLoanApplication.xsd

LoanApproveDefn.wsdl x LoanApproveService.bpel * x

Source Design Mapper Logging

100%



PartnerLink1 [Partner Link] - Navig...

BPEL Logical View

- If1
 - If1
 - Invoke1
 - Else
- PartnerLink1

Output - Standard output | JUnit Test Results | Refactoring | HTTP Monitor

```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```

PartnerLink1 [Partner Link] - Property Editor

Name:

WSDL File:

Use Existing Partner Link Type

Partner Link Type:

My Role:

Partner Role:

Use a Newly Created Partner Link Type

Create in File:

Partner Link Type Name:

Process will implement (My Role)

Role Name:

Port Type:

Partner service will implement (Partner Role)

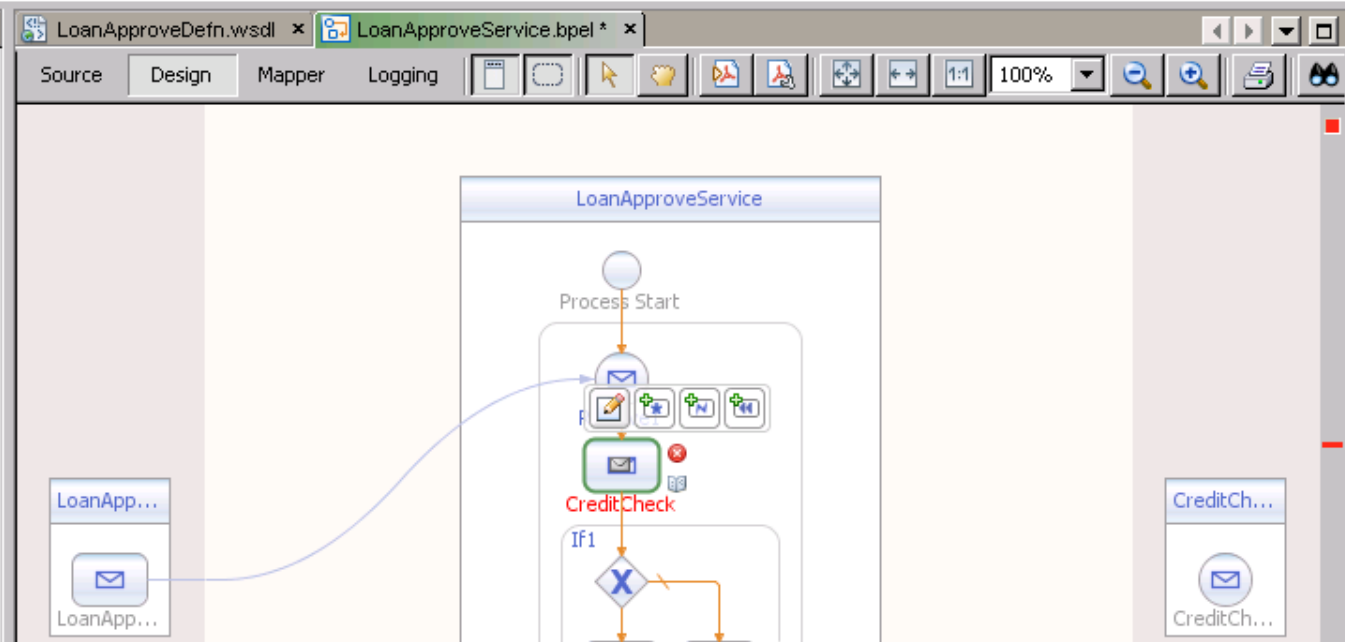
Role Name:

Port Type:



Pr... Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- Placeholder
 - Process Files
 - HomeLoanApplication.xsd



W... | Invo | Rece | Repl | Part | B... | Assi | Empl | Wait | Thro | ReTh | Exit | Com | Com | St... | If | Whil | Rept

CreditCheck [Invoke] - Navigator

BPEL Logical View

- CreditCheck
- If1
- If1
- Invoke

CreditCheck [Invoke] - Property Editor

Main Correlations

Name: CreditCheck

Partner Link: CreditCheckServicePartner

Operation: CreditCheckDefn.wsdlOperation

Create ... Browse ...

Create ... Browse ...

Ok Cancel Help

New Input Variable

Name: CreditCheckDefn.wsdlOperationIn

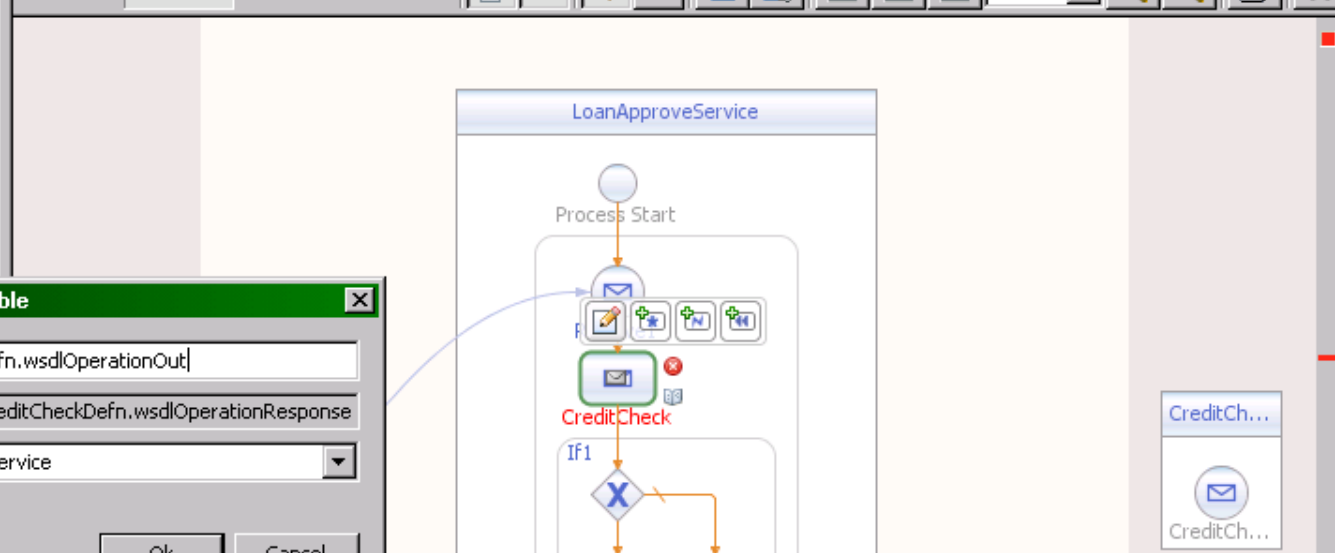
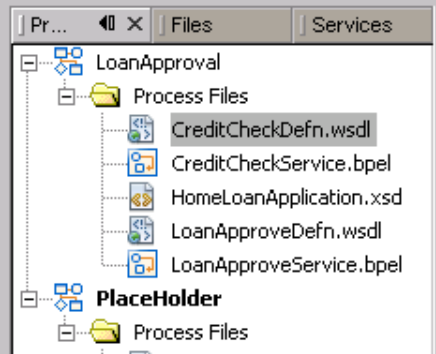
Type: jckDefn.wsdl:CreditCheckDefn.wsdlOperationRequest

Scope: LoanApproveService

Ok Cancel

TTP Monitor

defined using the type attribute.



New Output Variable

Name:

Type:

Scope:

Ok Cancel

Property Editor

Main Correlations

Name:

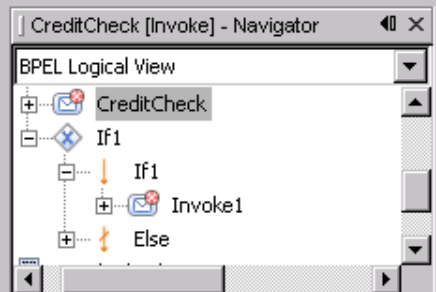
Partner Link:

Operation:

Input Variable: Create ... Browse ...

Output Variable: Create ... Browse ...

Ok Cancel Help



Output - Standard output

WS-I Basic Profile Rule R2203, an RI

TTP Monitor

defined using the type attribute.



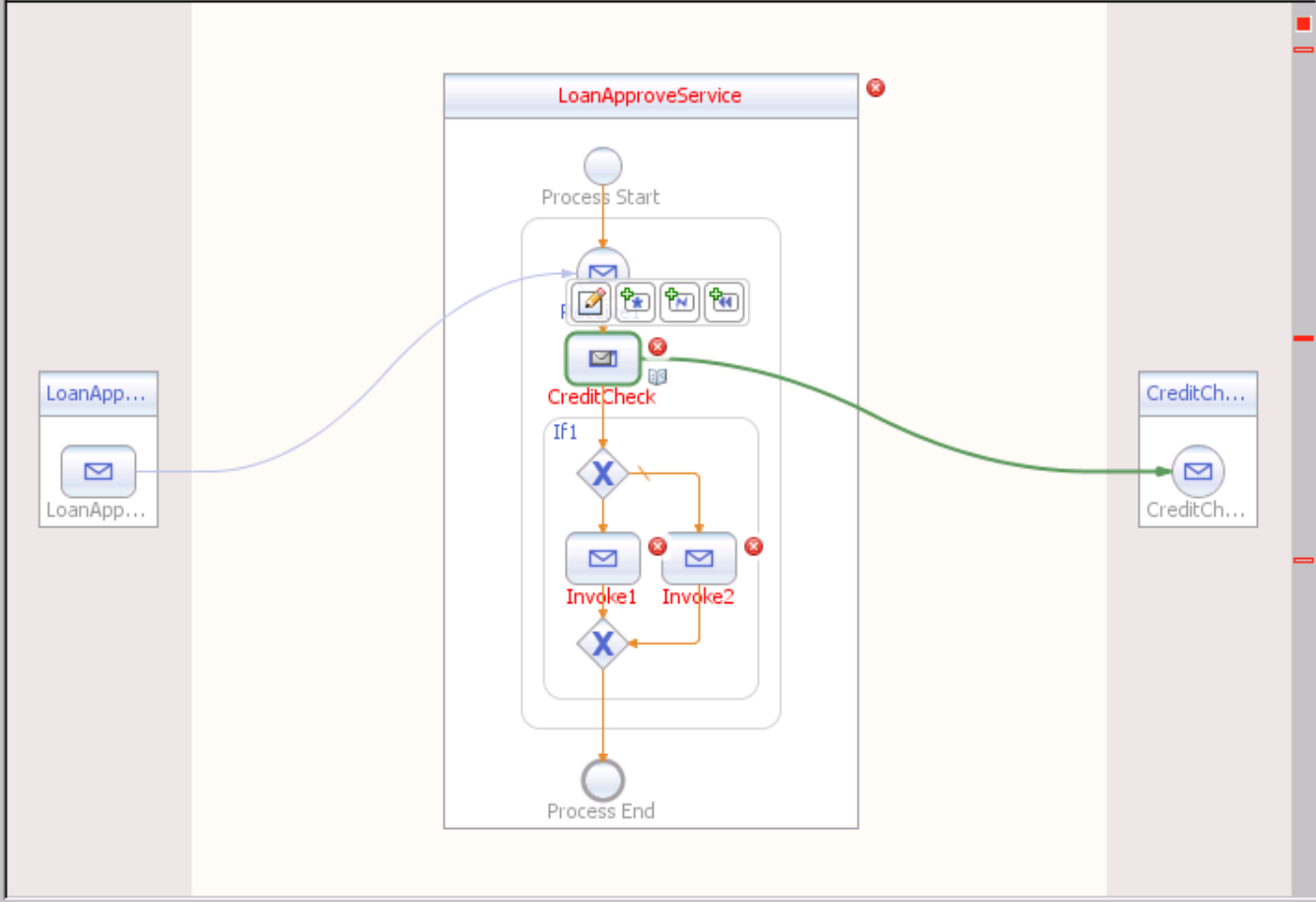
Pr... Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- Placeholder
 - Process Files
 - HomeLoanApplication.xsd

LoanApproveDefn.wsdl * LoanApproveService.bpel * x

Source Design Mapper Logging

100%



W... | Invo | Rece | Repl | Part | B... | Assi | Empl | Wait | Thro | ReTl | Exit | Com | Com | St... | If | Whil | Rept

Main | Cre... ?

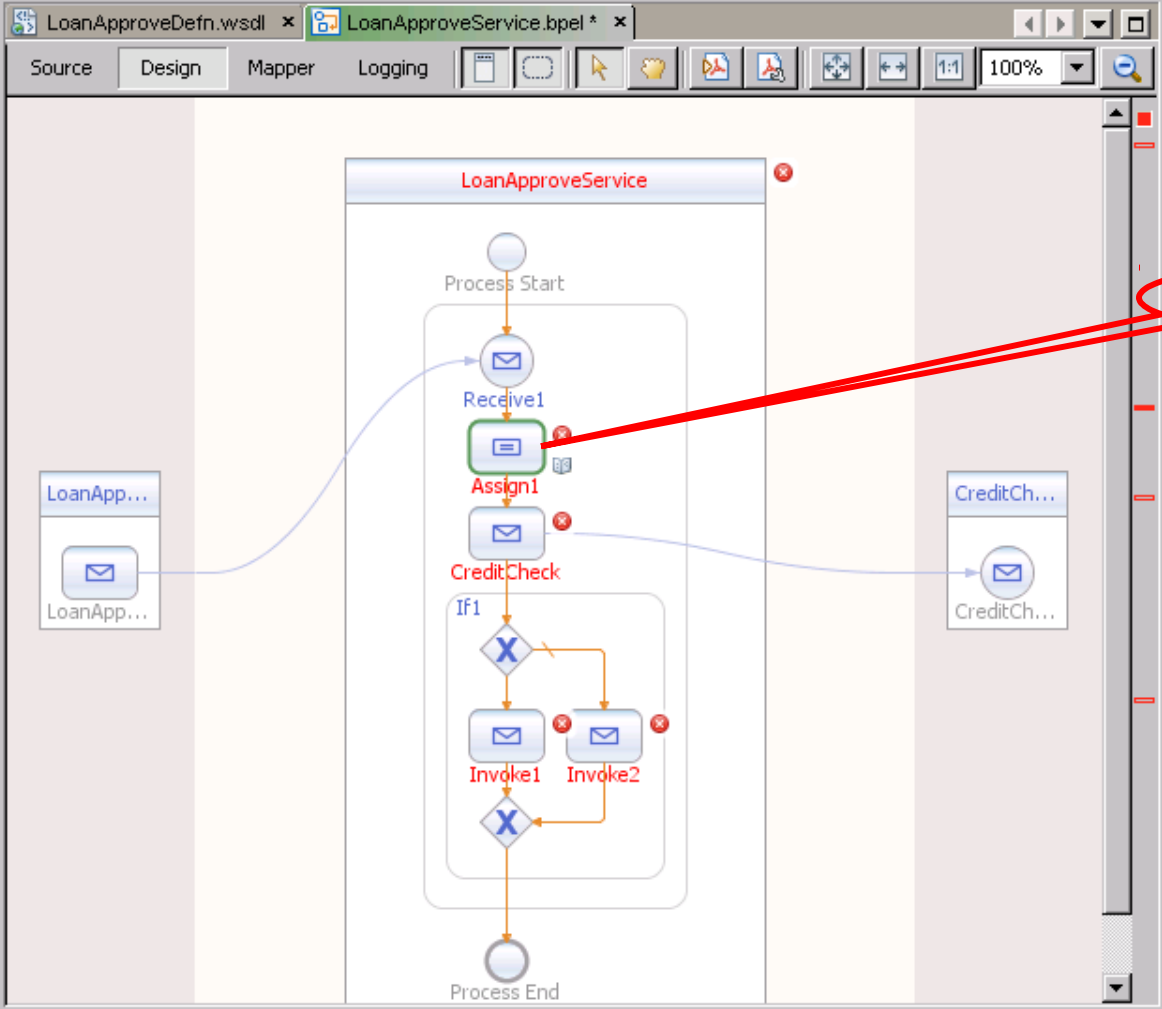
Output - Standard output | JUnit Test Results | Refactoring | HTTP Monitor

WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.



Pr... Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- Placeholder
 - Process Files
 - HomeLoanApplication.xsd



Palette

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign**
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities
 - If
 - While
 - RepeatUntil

Assign1 [Assign] - ...

Main

Assign1 [Assign]

Assign1 [Assign] - Navigator

BPEL Logical View

- LoanApproveDefnOperationIn tns
- Correlation Sets
- Sequence
 - Receive1
 - Assign1

```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```



Pr... Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - Placeholder
 - Process Files
 - HomeLoanApplication.xsd

LoanApproveDefn.wsdl x LoanApproveService.bpel * x

Source Design Mapper Logging

Operator Boolean String Node Number Date & Time BPEL

All Output

Input All

Variables

- Variable1
- LoanApproveDefnOperationIn
 - part1
- Partner Links

Variables

- Variable1
- LoanApproveDefnOperationIn
 - part1
- BorrowerInformation
- LoanDetails
- Partner Links
 - CreditCheckServicePartner

No Properties

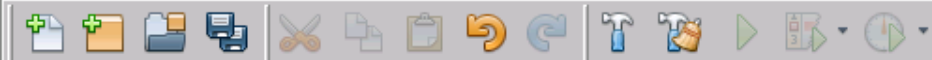
Assign1 [Assign] - Navigator

BPEL Logical View

- relation Sets
- sequence
- Receive1
- Assign1
- LoanApproveDefnOperationIn=LoanA

Output - Standard output JUnit Test Results Refactoring HTTP Monitor

```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```



Pr... | Files | Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - Placeholder
 - Process Files
 - HomeLoanApplication.xsd

LoanApproveDefn.wsdl * LoanApproveService.bpel * x

Source Design Mapper Logging

Operator Boolean String Node Number Date & Time BPEL

All Output

- Contains
- Normalize Space
- String
- Starts With
- String Length
- Substring
- Substring Before
- Substring After
- Translate
- Concat
- String Literal**

Variables

- Variable1
- LoanApproveDefn
 - part1
- Partner Links

Input All

Variables

- Variable1
- LoanApproveDefnOperationIn
- part1
- BorrowerInformation
- LoanDetails
- Partner Links
- CreditCheckServicePartner

Assign1 [Assign] - Navigator

BPEL Logical View

- Sequence
 - Receive1
 - Assign1
 - CreditCheck
 - If1

Output - Standard output | JUnit Test Results | Refactoring | HTTP Monitor

```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```

Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- Placeholder
 - Process Files
 - HomeLoanApplication.xsd

[If] - Navigator

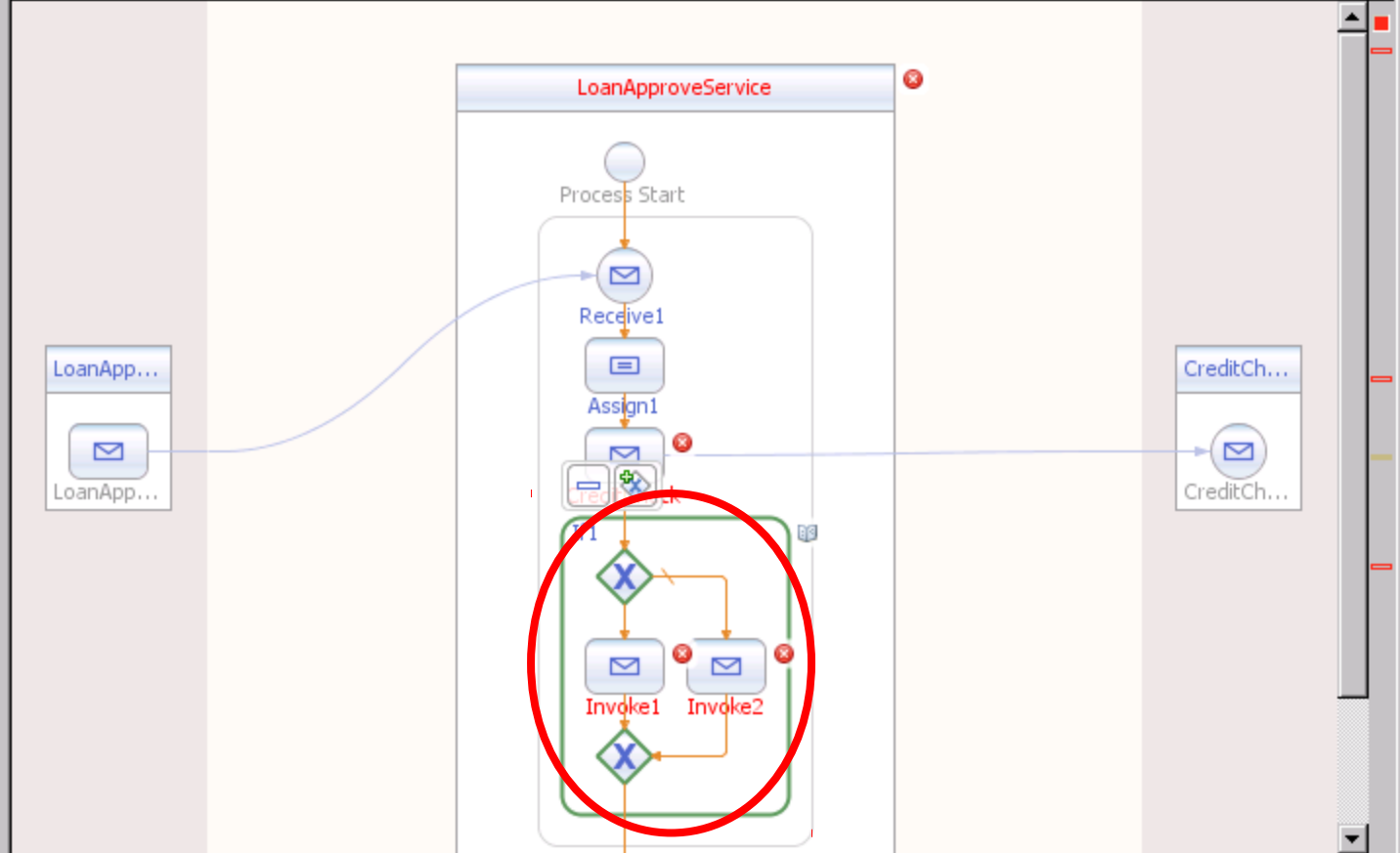
Logical View

- Receive1
- Assign1
- LoanApproveDefnOperati
- CreditCheck
- If1

LoanApproveDefn.wsdl x LoanApproveService.bpel * x

Source Design Mapper Logging

100%



WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute



Files | Ser...

- LoanApproval
 - Process Files
 - CreditCheckDe
 - CreditCheckSe
 - HomeLoanApp
 - LoanApproveD
 - LoanApproveS
 - Placeholder
 - Process Files
 - HomeLoanApp

LoanApproveDefn.wsdl x LoanApproveService.bpel x

Source Design Mapper Logging

Operator Boolean String Node Number Date & Time BPEL

All Output

- Variables
 - CreditCheckDefnc
 - part1
 - Status
 - SSN
 - CreditCheckDefnc
 - part1
 - Borrower
 - LoanData
 - LoanApproveDefn
 - part1
 - Borrower
 - LoanData

- Contains
- Normalize Space
- String
- Starts With
- String Length
- Substring
- Substring Before
- Substring After
- Translate
- Concat
- String Literal**

Input All

If

Boolean Condition

If1 [If] - Navigator

BPEL Logical View

- variables
 - CreditCheckDe
 - CreditCheckDe
 - LoanApproveD
- Correlation Sets
- Sequence
 - Receive1
 - Assign1
 - LoanAppro
 - CreditCheck
 - If1

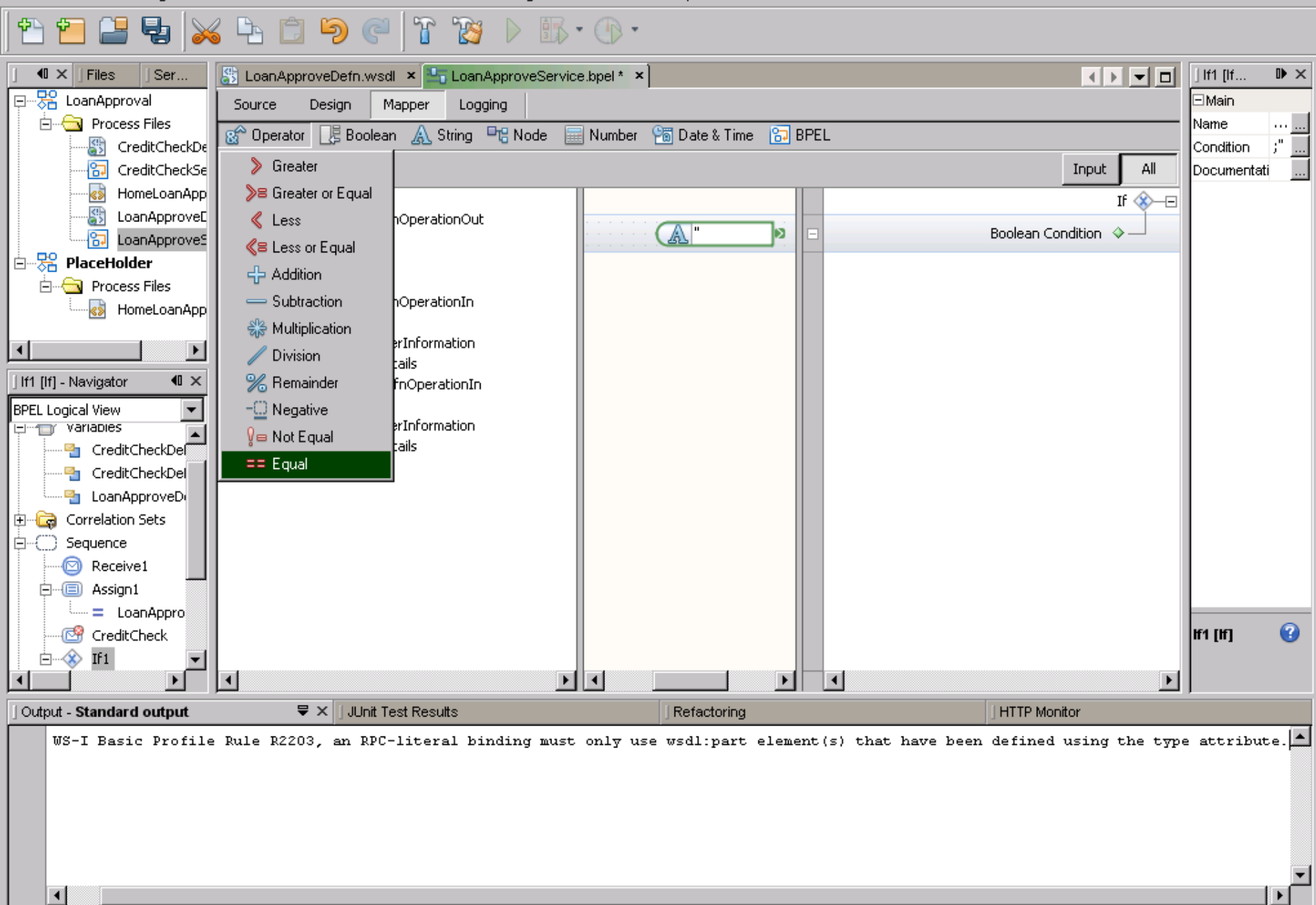
If1 [If...]

- Main
- Name
- Condition
- Documentati

If1 [If] ?

Output - Standard output | JUnit Test Results | Refactoring | HTTP Monitor

WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.



LoanApproval

- Process Files
 - CreditCheckDe
 - CreditCheckSe
 - HomeLoanApp
 - LoanApproveD
 - LoanApproveS
- Placeholder
 - Process Files
 - HomeLoanApp

Files | Ser...

LoanApproveDefn.wsdl * | LoanApproveService.bpel * x

Source Design Mapper Logging

Operator Boolean String Node Number Date & Time BPEL

- Greater
- Greater or Equal
- Less
- Less or Equal
- Addition
- Subtraction
- Multiplication
- Division
- Remainder
- Negative
- Not Equal
- Equal**

Input All

If

Boolean Condition

If1 [If] - Navigator

BPEL Logical View

- variables
 - CreditCheckDe
 - CreditCheckDe
 - LoanApproveD
- Correlation Sets
- Sequence
 - Receive1
 - Assign1
 - LoanAppro
 - CreditCheck
 - If1

Output - Standard output | JUnit Test Results | Refactoring | HTTP Monitor

WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.



Files | Ser...

- LoanApproval
 - Process Files
 - CreditCheckDefn
 - CreditCheckSe
 - HomeLoanApp
 - LoanApproveD
 - LoanApproveS
 - Placeholder
 - Process Files
 - HomeLoanApp

If1 [If] - Navigator

BPEL Logical View

- variables
 - CreditCheckDefn
 - CreditCheckDefn
 - LoanApproveD
- Correlation Sets
- Sequence
 - Receive1
 - Assign1
 - LoanAppro
 - CreditCheck
 - If1

LoanApproveDefn.wsdl * x | LoanApproveService.bpel * x

Source Design Mapper Logging

Operator Boolean String Node Number Date & Time BPEL

All Output

Input All

Variables

- CreditCheckDefnOperationOut
 - part1
 - Status
 - SSN
- CreditCheckDefnOperationIn
 - part1
 - BorrowerInformation
 - LoanDetails
- LoanApproveDefnOperationIn
 - part1
 - BorrowerInformation
 - LoanDetails

Diagram showing a variable assignment and a Boolean Condition node.

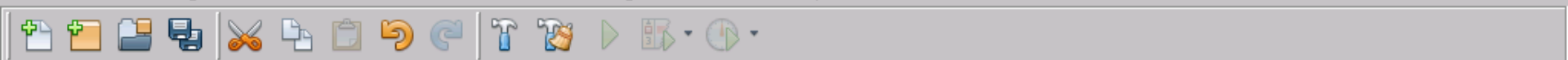
If1 [If] ...

- Main
- Name ...
- Condition ...
- Documentati ...

If1 [If] ?

Output - Standard output | JUnit Test Results | Refactoring | HTTP Monitor

WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.



Files | Ser...

- LoanApproval
 - Process Files
 - CreditCheckDefn
 - CreditCheckSe
 - HomeLoanApp
 - LoanApproveD
 - LoanApproveS
 - Placeholder
 - Process Files
 - HomeLoanApp

If1 [If] - Navigator

BPEL Logical View

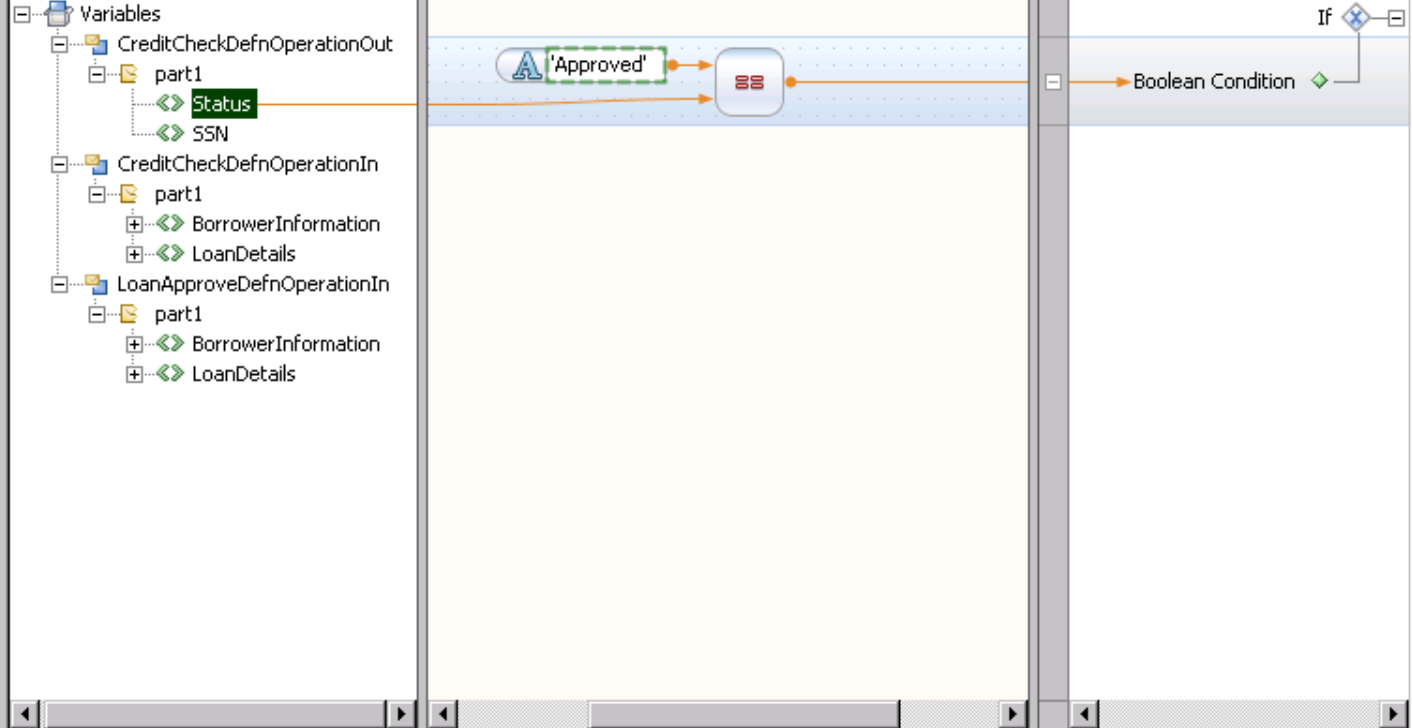
- variables
 - CreditCheckDefn
 - CreditCheckDel
 - LoanApproveD
- Correlation Sets
- Sequence
 - Receive1
 - Assign1
 - LoanAppro
 - CreditCheck
 - If1

LoanApproveDefn.wsdl * x | LoanApproveService.bpel * x

Source Design Mapper Logging

Operator Boolean String Node Number Date & Time BPEL

All Output Input All



If1 [If] ...

Main

Name ...

Condition ...

Documentati ...

If1 [If] ?

Output - Standard output | JUnit Test Results | Refactoring | HTTP Monitor

```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```

Phase suivante : Ecriture de la réponse dans un fichier

The screenshot displays the NetBeans IDE interface for editing a BPEL process. The main workspace shows a BPEL process diagram in the Design view. The process flow is as follows:

- Process Start
- Receive1 (Message In)
- Assign1 (Assign)
- CreditCheck (Message In)
- If1 (Decision)
- Invoke1 (Message Out)
- Invoke2 (Message Out)
- Process End

A context menu is open over the 'CreditCheck' activity, with the 'New' submenu expanded. The 'WSDL Document...' option is selected. The 'CreditCheck' activity is highlighted in blue, and a blue arrow points from the 'WSDL Document...' option to it. The 'CreditCheck' activity is also highlighted in blue, and a blue arrow points from the 'WSDL Document...' option to it. The 'CreditCheck' activity is also highlighted in blue, and a blue arrow points from the 'WSDL Document...' option to it.

Output - Standard output

JUnit Test Results

Refactoring

HTTP Monitor

```
WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attribute.
```

New WSDL Document [X]

Steps

1. Choose File Type
- 2. Name and Location**
3. Abstract Configuration
4. Concrete Configuration

Name and Location

File Name:

Project:

Folder:

Created File:

Target Namespace:

Import XML Schema File(s)

XML Schema(s):

< Back Next > Finish Cancel Help

One way

New WSDL Document

Steps

1. Choose File Type
2. Name and Location
- 3. Abstract Configuration**
4. Concrete Configuration

Abstract Configuration

Port Type Name:

Operation Name:

Operation Type:

Input:

Message Part Name	Element Or Type
part1	xsd:string

Generate partnerlinktype automatically.

New WSDL Document

Steps

1. Choose File Type
2. Name and Location
3. Abstract Configuration
4. **Concrete Configuration**

Concrete Configuration

Binding Name:

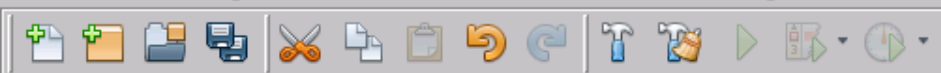
Binding Type:

Binding Subtype: File Transport

Service Name:

Port Name:

< Back Next > Finish Cancel Help



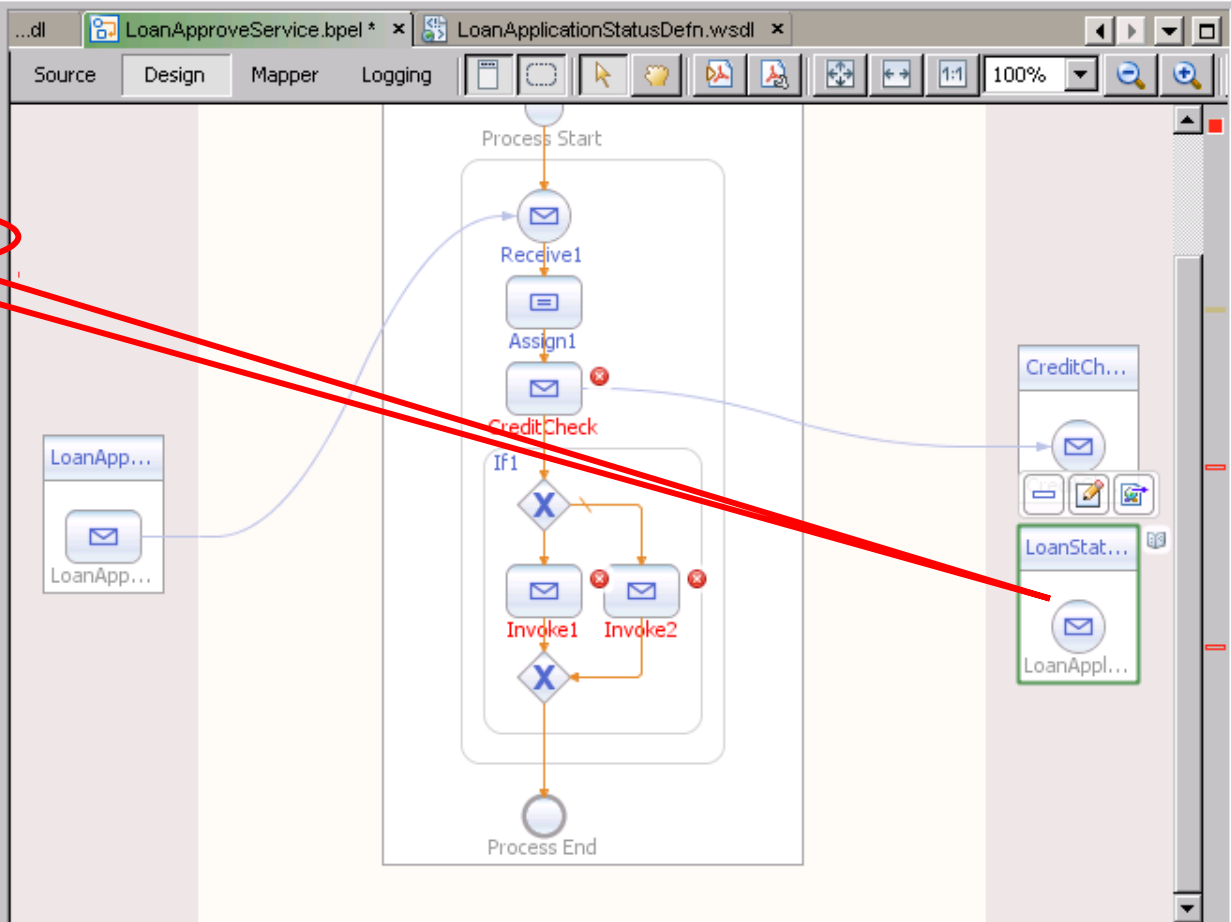
Project Explorer showing file structure:

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl**
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - Placeholder
 - Process Files
 - HomeLoanApplication.xsd

PartnerLink1 [Partner Link] - Navigator

BPEL Logical View

- sequence
 - Receive1
 - Assign1
 - LoanApproveDefnOperationIn=
 - CreditCheck
 - If1
 - If1
 - Else
 - Invoke2
 - CreditCheckServicePartner
 - LoanStatusPartner



Palette

- Web Se...
 - Invoke
 - Receive
 - Reply
 - Partner Linl
- Basic A...
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensat
 - Compensat
- Structu...
 - If
 - While

Per WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attrib...



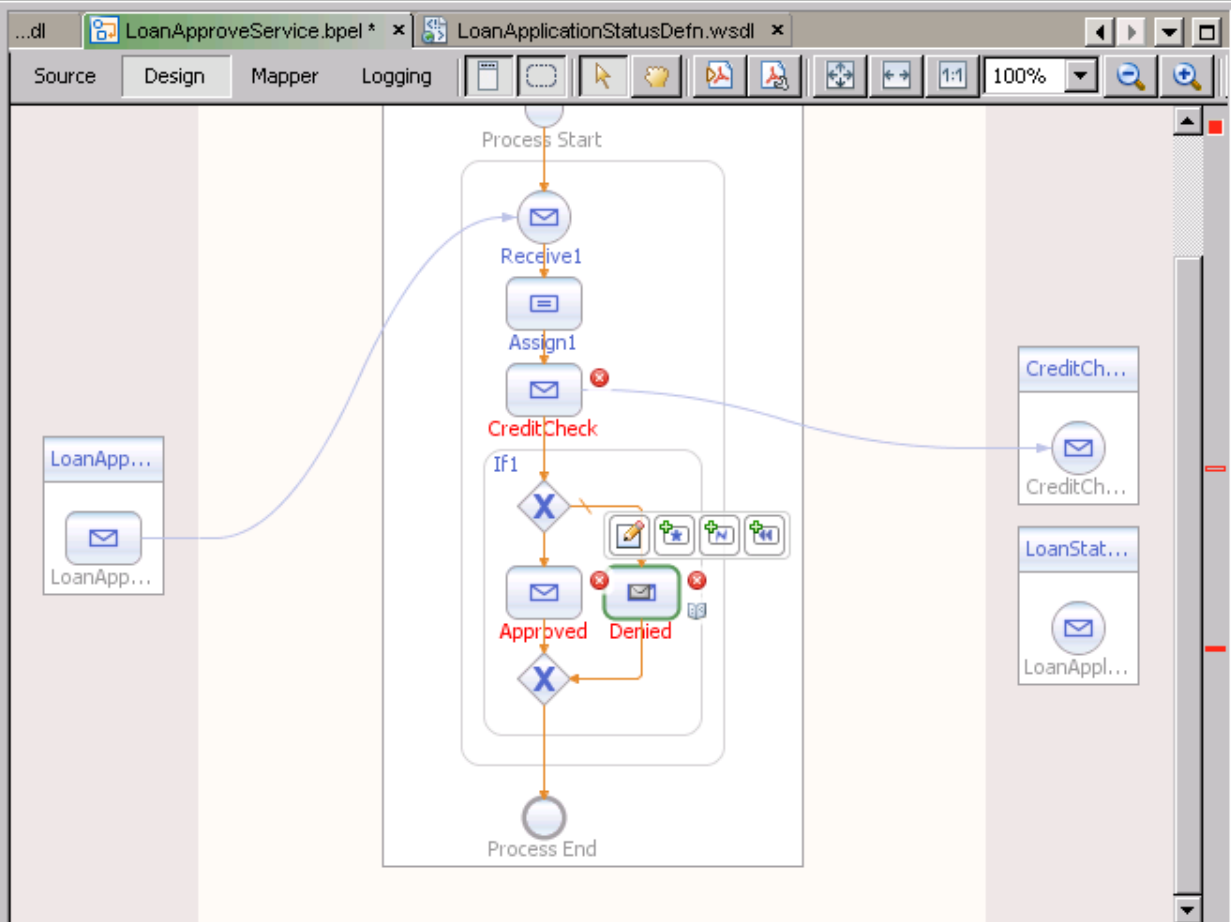
Project Explorer showing the file structure:

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- Placeholder
 - Process Files
 - HomeLoanApplication.xsd

Invoke2 [Invoke] - Navigator

BPEL Logical View

- sequence
 - Receive1
 - Assign1
 - LoanApproveDefnOperationIn=
 - CreditCheck
 - If1
 - If1
 - Approved
 - Else
 - Denied
 - CreditCheckServicePartner



Output - Standard output

```

    Per WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attrib
    
```


Approved [Invoke] - Property Editor

Main | Correlations

Name:

Partner Link:

Operation:

Input Variable:

Output Variable:

Approved [Invoke] - Navigator

BPEL Logical View

- Correlation Sets
- Sequence
 - Receive1
 - Assign1
 - LoanApproveDefnOperationIn=Loan
 - CreditCheck
 - If1
 - If1
 - Approved
 - Else
 - Denied
 - CreditCheckServicePartner
 - LoanStatusPartner

Output - Standard output

JUnit Test Results | Refactoring | HTTP Monitor

Per WS-I Basic Profile Rule R2203, an RPC-literal binding must only use wsdl:part element(s) that have been defined using the type attrib

The screenshot displays the NetBeans IDE 6.1 interface for the LoanApproval project. The main editor shows a BPEL diagram for LoanApproveService, with a 'Process Start' node and a 'Denied [Invoke]' activity. The 'Denied [Invoke] - Property Editor' dialog is open, showing the following configuration:

- Name: Denied
- Partner Link: LoanStatusPartner
- Operation: LoanApplicationStatusDefnOperation
- Input Variable: (empty)
- Output Variable: (empty)

The 'Input Variable Chooser' dialog is also open, showing the variable 'LoanApplicationStatusDefnOperationIn' selected from the 'LoanApproveService' namespace.

The left sidebar shows the project tree and a BPEL Logical View. The right sidebar shows a palette of BPEL activities. The bottom status bar shows 'Output - Standard output' with a message: 'Per WS-I Basic Profile Rule R2203, an RPC-literal bin'.



Projects | Files | Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- Placeholder
 - Process Files
 - HomeLoanApplication.xsd

Assign2 [Assign] - Navigator

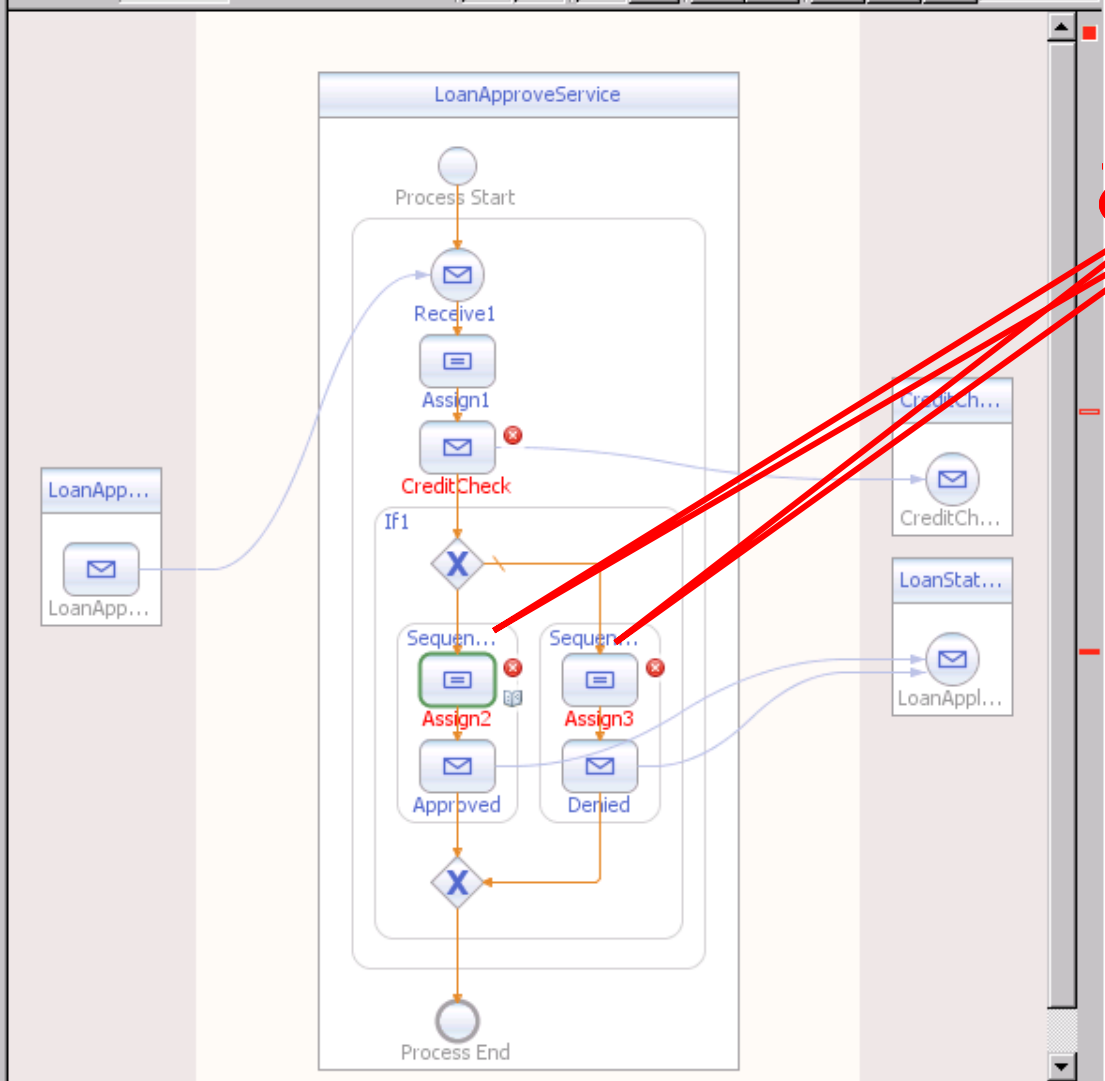
BPEL Logical View

- Receive1
 - Assign1
 - LoanApproveDefnOperationIn=LoanApprove
 - CreditCheck
 - If1
 - If1
 - Sequence1
 - Assign2
 - Approved
 - Else
 - Sequence2
 - Assign3
 - Denied
- CreditCheckServicePartner
- LoanStatusPartner

LoanApproveService.bpel * | LoanApplicationStatusDefn.wsdl

Source | Design | Mapper | Logging

1:1 | 100%



Palette

- Web Se...
- Invoke
- Receive
- Reply
- Partner Linl
- basic A...
- Assign**
- Empty
- Wait
- Throw
- ReThrow
- Exit
- Compensat
- Compensat
- Structu...
- If
- While
- RepeatUnti
- ForEach
- Pirk

Assign... | Main | Name | Assignment: 0 | Documentati

Assign2 [... ?]



LoanApproveDefn.wsdl x LoanApproveService.bpel * x LoanApplicationStatusDefn.wsdl x

Source Design Mapper Logging

Operator Boolean String Node Number Date & Time BPEL

All Output

- Contains
- Normalize Space
- String
- Starts With
- String Length
- Substring
- Substring Before
- Substring After
- Translate
- Concat**
- String Literal

Input All

Variables Partner Links

Assign2 [...

Output - Standard output x JUnit Test Results Refactoring HTTP Monitor



LoanApproveDefn.wsdl | LoanApproveService.bpel | LoanApplicationStatusDefn.wsdl

Source | Design | Mapper | Logging

Operator | Boolean | String | Node | Number | Date & Time | BPEL

All | Output

Variables

- LoanApplicationStatusDefnOperationIn
- part1
- CreditCheckDefnOperationOut
- CreditCheckDefnOperationIn
- LoanApproveDefnOperationIn
- part1
 - BorrowerInformation
 - Name
 - SSN
 - Address
 - LoanDetails
 - LoanAmount
 - HomePrice
 - HomeAddress
- Partner Links

Concat

- String
- String
- String
- String

LoanApplicationStatusDefnOperationIn

- part1

CreditCheckDefnOperationOut

- part1
 - Status
 - SSN

CreditCheckDefnOperationIn

- part1
 - BorrowerInformation
 - Name
 - SSN
 - Address
 - Addr1
 - CityState
 - Zip
 - Telephone
 - LoanDetails
 - LoanAmount
 - HomePrice

Assign...

Main

Name ...

Assignment: 1

Documentati ...

Assign2 [...]



LoanApproveDefn.wsdl x LoanApproveService.bpel * x LoanApplicationStatusDefn.wsdl x

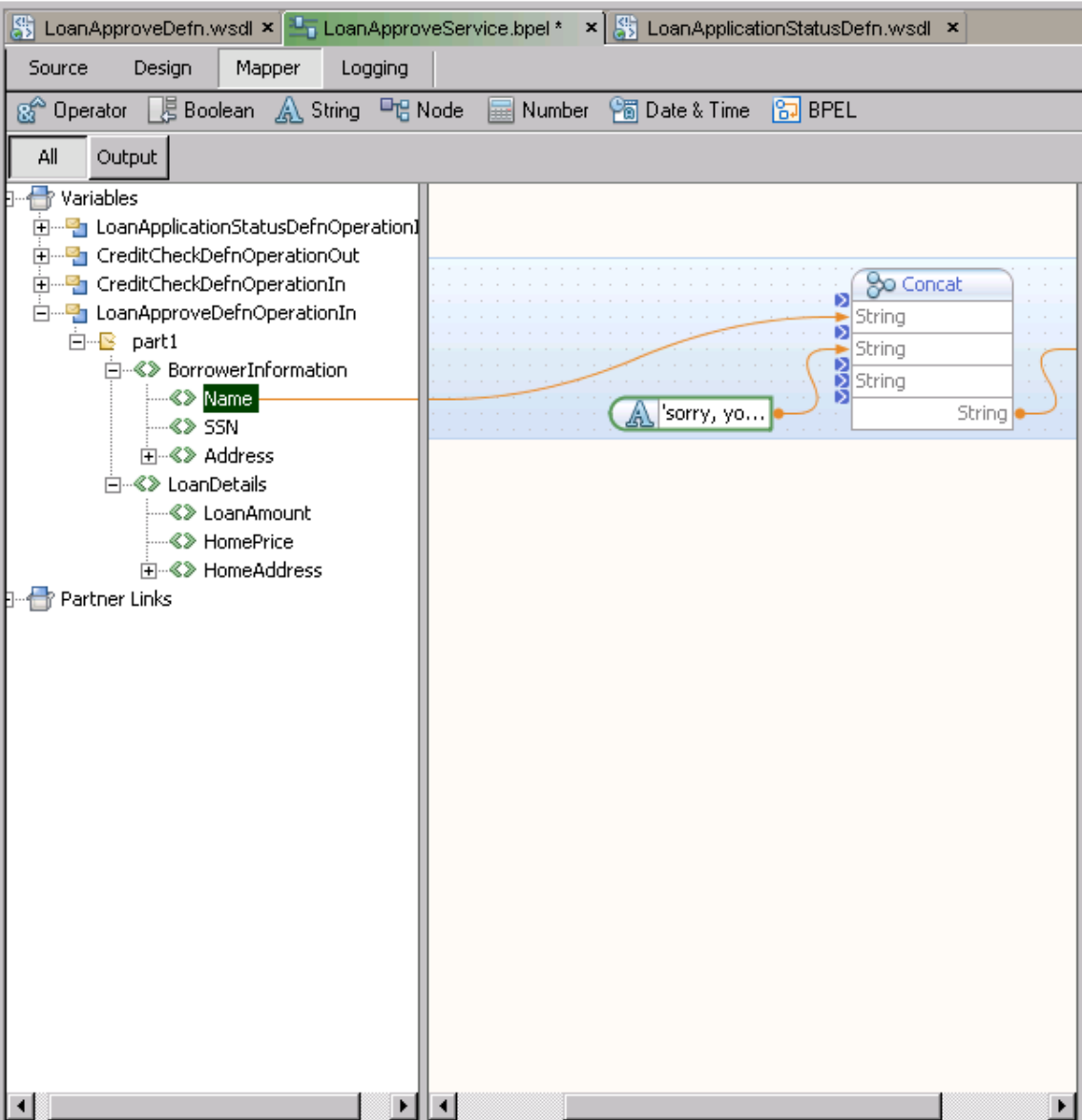
Source Design Mapper Logging

Operator Boolean String Node Number Date & Time BPEL

All Output

Variables

- LoanApplicationStatusDefnOperationIn
- CreditCheckDefnOperationOut
- CreditCheckDefnOperationIn
- LoanApproveDefnOperationIn
 - part1
 - BorrowerInformation
 - Name
 - SSN
 - Address
 - LoanDetails
 - LoanAmount
 - HomePrice
 - HomeAddress
- Partner Links



Assign3

Main

Name ...

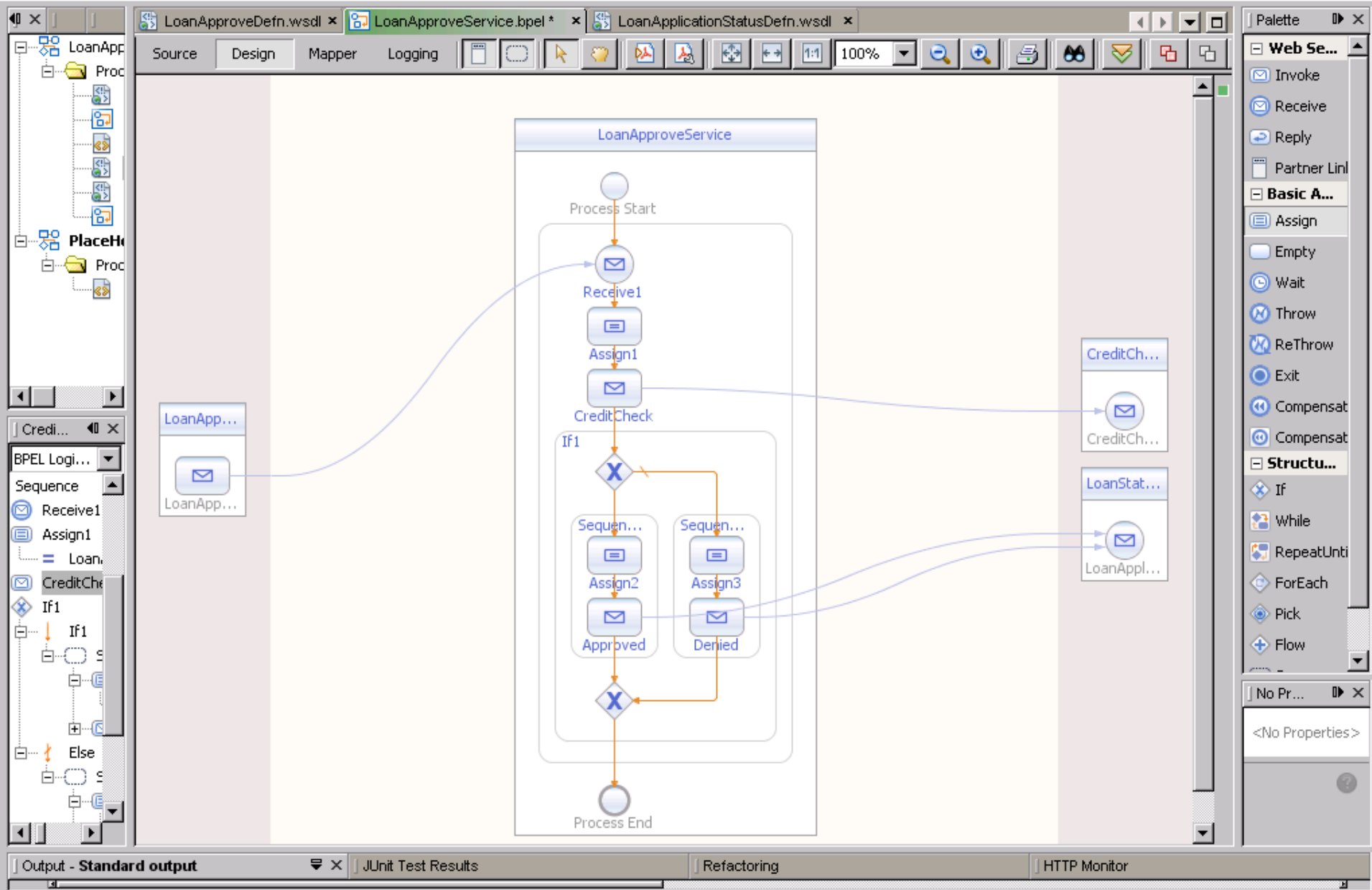
Assignment: 1

Documentati ...

Variables

- LoanApplicationStatusDefnOperationIn
- LoanApproveDefnOperationIn
- LoanApplicationStatusDefnOperationOut
- CreditCheckDefnOperationIn
- LoanApproveDefnOperationIn
- Partner Links
 - CreditCheckServicePartner
 - LoanStatusPartner

Processus achevé



source



LoanApproveDefn.wsdl x LoanApproveService.bpel * x LoanApplicationStatusDefn.wsdl x

Source Design Mapper Logging

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <process
3      name="LoanApproveService"
4      targetNamespace="http://enterprise.netbeans.org/bpel/LoanApproval/LoanApproveService"
5      xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
6      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
7      xmlns:tns="http://enterprise.netbeans.org/bpel/LoanApproval/LoanApproveService" xmlns:sxt="http://www.
8  <import namespace="http://j2ee.netbeans.org/wsdl/LoanApproveDefn" location="LoanApproveDefn.wsdl" impor
9  <import namespace="http://j2ee.netbeans.org/wsdl/CreditCheckDefn.wsdl" location="CreditCheckDefn.wsdl"
10 <import namespace="http://j2ee.netbeans.org/wsdl/LoanApplicationStatusDefn" location="LoanApplicationSt
11 <partnerLinks>
12     <partnerLink name="CreditCheckServicePartner" xmlns:tns="http://j2ee.netbeans.org/wsdl/CreditCheckDe
13     <partnerLink name="LoanStatusPartner" xmlns:tns="http://j2ee.netbeans.org/wsdl/LoanApplicationStatus
14     <partnerLink name="LoanApprovePartner" xmlns:tns="http://j2ee.netbeans.org/wsdl/LoanApproveDefn" par
15 </partnerLinks>
16 <variables>
17     <variable name="LoanApplicationStatusDefnOperationIn" xmlns:tns="http://j2ee.netbeans.org/wsdl/LoanA
18     <variable name="CreditCheckDefnOperationOut" xmlns:tns="http://j2ee.netbeans.org/wsdl/CreditCheckDef
19     <variable name="CreditCheckDefnOperationIn" xmlns:tns="http://j2ee.netbeans.org/wsdl/CreditCheckDefn
20     <variable name="LoanApproveDefnOperationIn" xmlns:tns="http://j2ee.netbeans.org/wsdl/LoanApproveDefn
21 </variables>
22 <sequence>
23     <receive name="Receive1" createInstance="yes" partnerLink="LoanApprovePartner" operation="LoanApprov
24     <assign name="Assign1">
25         <copy>
26             <from variable="LoanApproveDefnOperationIn" part="part1"></from>
27             <to variable="LoanApproveDefnOperationIn" part="part1"/>
28         </copy>
29     </assign>
30     <invoke name="CreditCheck" partnerLink="CreditCheckServicePartner" operation="CreditCheckDefn.wsdlOp

```

4:8 INS

Source (suite et fin)

```
27         <to variable="LoanApproveDefnOperationIn" part="part1"/>
28     </copy>
29 </assign>
30 <invoke name="CreditCheck" partnerLink="CreditCheckServicePartner" operation="CreditCheckDefn.wsdlOp
31 <if name="If1">
32     <condition>'Approved' = $CreditCheckDefnOperationOut.part1/Status</condition>
33     <sequence name="Sequence1">
34         <assign name="Assign2">
35             <copy>
36                 <from>concat('your loan has been approved', $LoanApproveDefnOperationIn.part1/BorrowerIn
37                     <to variable="LoanApplicationStatusDefnOperationIn" part="part1"/>
38             </copy>
39         </assign>
40         <invoke name="Approved" partnerLink="LoanStatusPartner" operation="LoanApplicationStatusDefnOp
41     </sequence>
42 </else>
43     <sequence name="Sequence2">
44         <assign name="Assign3">
45             <copy>
46                 <from>concat($LoanApproveDefnOperationIn.part1/BorrowerInformation/Name, 'sorry, your
47                     <to variable="LoanApplicationStatusDefnOperationIn" part="part1"/>
48             </copy>
49         </assign>
50         <invoke name="Denied" partnerLink="LoanStatusPartner" operation="LoanApplicationStatusDefnO
51     </sequence>
52 </else>
53 </if>
54 </sequence>
55 </process>
56
```

4:8 INS

Output - Standard output | JUnit Test Results | Refactoring | HTTP Monitor



Projects | Files | Services

LoanApproval

- Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- LoanApprovalServiceAssembly**
 - Service Assembly
 - Process Files
 - JBI Modules
 - Test
- Placeholder
 - Process Files
 - HomeLoanApplication.xsd

Service Assembly - Navigator

Logical View

- LoanApproval
- Connections
 - CreditCheckDefn.wsdlPortTypeBindingPort <-> Cr

...dl LoanApproveService.bpel x LoanApplicationStatusDefn.wsdl x Input.xml x



```
1 <soapenv:Envelope xsi:schemaLocation="http://schemas.xmlsoap.org/soap/en
2 <soapenv:Body>
3 <hom:LoanInformation>
4 <BorrowerInformation>
5 <Name>Toto</Name>
6 <SSN>366677</SSN>
7 <Address>
8 <Addr1?string?</Addr1>
9 <CityState?string?</CityState>
10 <Zip?string?</Zip>
11 <Telephone?string?</Telephone>
12 </Address>
13 </BorrowerInformation>
14 <LoanDetails>
15 <LoanAmount?string?</LoanAmount>
16 <HomePrice?string?</HomePrice>
17 <HomeAddress>
18 <Addr1?string?</Addr1>
19 <CityState?string?</CityState>
20 <Zip?string?</Zip>
21 <Telephone?string?</Telephone>
```

1:1 INS

Output | JUnit Test Results

Refactoring

HTTP Monitor

Standard output x XML check x

```
XML validation started.
0 Error(s), 0 Warning(s).
XML validation finished.
```



Projects Files Services

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel

- LoanApprovalServiceAssembly
 - Service Assembly
 - Process Files
 - JBIs Modules
 - Test
 - Placeholder
 - Process Files
 - HomeLoanAp

- Add JBI Module...
- New
- Build
- Clean and Build
- Clean
- Deploy**
- Undeploy
- Test Alt+F6
- Debug (BPEL)
- Set as Main Project
- Open Required Projects
- Close
- Rename...
- Move...
- Copy...
- Delete Supprimer
- Find... Ctrl+F
- Reverse Engineer...
- Versioning
- Local History
- Properties

LoanApproveService.bpel LoanApplicationStatusDefn.wsdl Input.xml

```

1 <soapenv:Envelope xsi:schemaLocation="http://schemas.xmlsoap.org/soap/en
2   <soapenv:Body>
3     <hom:LoanInformation>
4       <BorrowerInformation>
5         <Name>Toto</Name>
6         <SSN>366677</SSN>
7         <Address>
            <Addr1?string?</Addr1>
            <CityState?string?</CityState>
            <Zip?string?</Zip>
            <Telephone?string?</Telephone>
          </Address>
        </BorrowerInformation>
        <LoanDetails>
          <LoanAmount?string?</LoanAmount>
          <HomePrice?string?</HomePrice>
          <HomeAddress>
            <Addr1?string?</Addr1>
            <CityState?string?</CityState>
            <Zip?string?</Zip>
            <Telephone?string?</Telephone>
          </HomeAddress>
        </LoanDetails>
      </hom:LoanInformation>
    </soapenv:Body>
  </soapenv:Envelope>

```

Service Assembly - Navigator

Local View

- Connections
 - LoanApproveDefn
 - LoanApplicationSt

Standard output x

```

file=C:\
[start-service-a
Starting a s
host=loc:
port=484:
name=Loa
run:
BUILD SUCCESSFUL

```

Refactoring HTTP Monitor

process x GlassFish V2 x build.xml (run) x

approvalServiceAssembly/dist/LoanApprovalServiceAssembly.zip

Nouveau cas de test

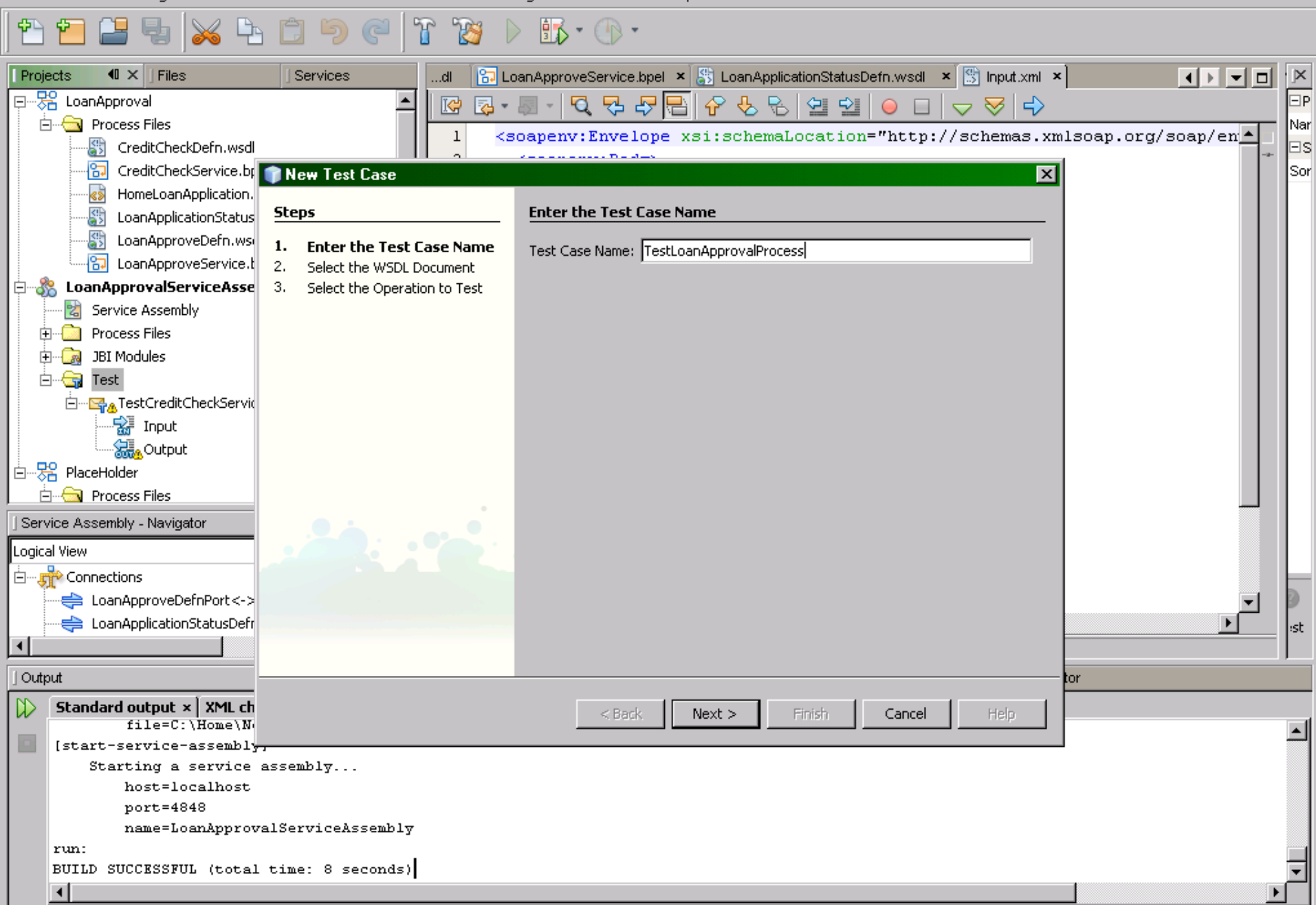
The screenshot displays the NetBeans IDE interface for a service assembly project named "LoanApprovalServiceAssembly".

- Project Explorer:** Shows the project structure with folders for "Process Files", "LoanApprovalServiceAssembly", "Service Assembly", "Process Files", "JBI Modules", "Test", "Placeholder", and "Process Files". A context menu is open over the "Test" folder, with options "New Test Case" (highlighted) and "Delete All Results".
- Service Assembly - Navigator:** Shows the logical view of the service assembly, including connections between "LoanApproveDefnPort" and "LoanApproveDefnPort", and "LoanApplicationStatusDefnPort" and "LoanApplicationStatusDefnPortTypeRole_partner".
- Main Editor:** Displays the XML code for "Input.xml". The code is as follows:

```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
2   <soapenv:Body>
3     <hom:LoanInformation>
4       <BorrowerInformation>
5         <Name>Toto</Name>
6         <SSN>366677</SSN>
7         <Address>
8           <Addr1?string?/>
9           <CityState?string?/>
10          <Zip?string?/>
11          <Telephone?string?/>
12        </Address>
13      </BorrowerInformation>
14      <LoanDetails>
15        <LoanAmount?string?/>
16        <HomePrice?string?/>
17        <HomeAddress>
18          <Addr1?string?/>
19          <CityState?string?/>
20          <Zip?string?/>
21          <Telephone?string?/>

```
- Output Window:** Shows the output of the build process, including the file path and the successful completion of the service assembly build.

```
file=C:\Home\NetBeansProjects\LoanApprovalServiceAssembly\dist\LoanApprovalServiceAssembly.zip
[start-service-assembly]
Starting a service assembly...
host=localhost
port=4848
name=LoanApprovalServiceAssembly
run:
BUILD SUCCESSFUL (total time: 8 seconds)
```



The screenshot shows the NetBeans IDE interface with a 'New Test Case' dialog box open. The dialog box has a green title bar and a close button. It is divided into two main sections: 'Steps' and 'Enter the Test Case Name'.

Steps:

1. Enter the Test Case Name
2. Select the WSDL Document
3. Select the Operation to Test

Enter the Test Case Name:

Test Case Name:

At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

The background shows the IDE's project tree on the left, with 'LoanApprovalServiceAssembly' selected. The main editor area displays a SOAP envelope XML snippet:

```
1 <soapenv:Envelope xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/...>
```

The 'Standard output' window at the bottom left shows the following text:

```
file=C:\Home\N...
[start-service-assembly]
Starting a service assembly...
host=localhost
port=4848
name=LoanApprovalServiceAssembly
run:
BUILD SUCCESSFUL (total time: 8 seconds)
```



Projects | Files | Services | ...dl | LoanApproveService.bpel | LoanApplicationStatusDefn.wsdl | Input.xml

LoanApproval

- Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bp
 - HomeLoanApplication.
 - LoanApplicationStatus
 - LoanApproveDefn.wsdl
 - LoanApproveService.t
- LoanApprovalServiceAsse
 - Service Assembly
 - Process Files
 - JBI Modules
 - Test
 - TestCreditCheckServic
 - Input
 - Output
 - Placeholder
 - Process Files

Service Assembly - Navigator

Logical View

- Connections
 - LoanApproveDefnPort <->
 - LoanApplicationStatusDefn

Output

Standard output x XML ch
file=C:\Home\N
[start-service-assembly
Starting a service assembly...
host=localhost
port=4848
name=LoanApprovalServiceAssembly
run:
BUILD SUCCESSFUL (total time: 8 seconds)

New Test Case

Steps

1. Enter the Test Case Name
2. **Select the WSDL Document**
3. Select the Operation to Test

Select the WSDL Document

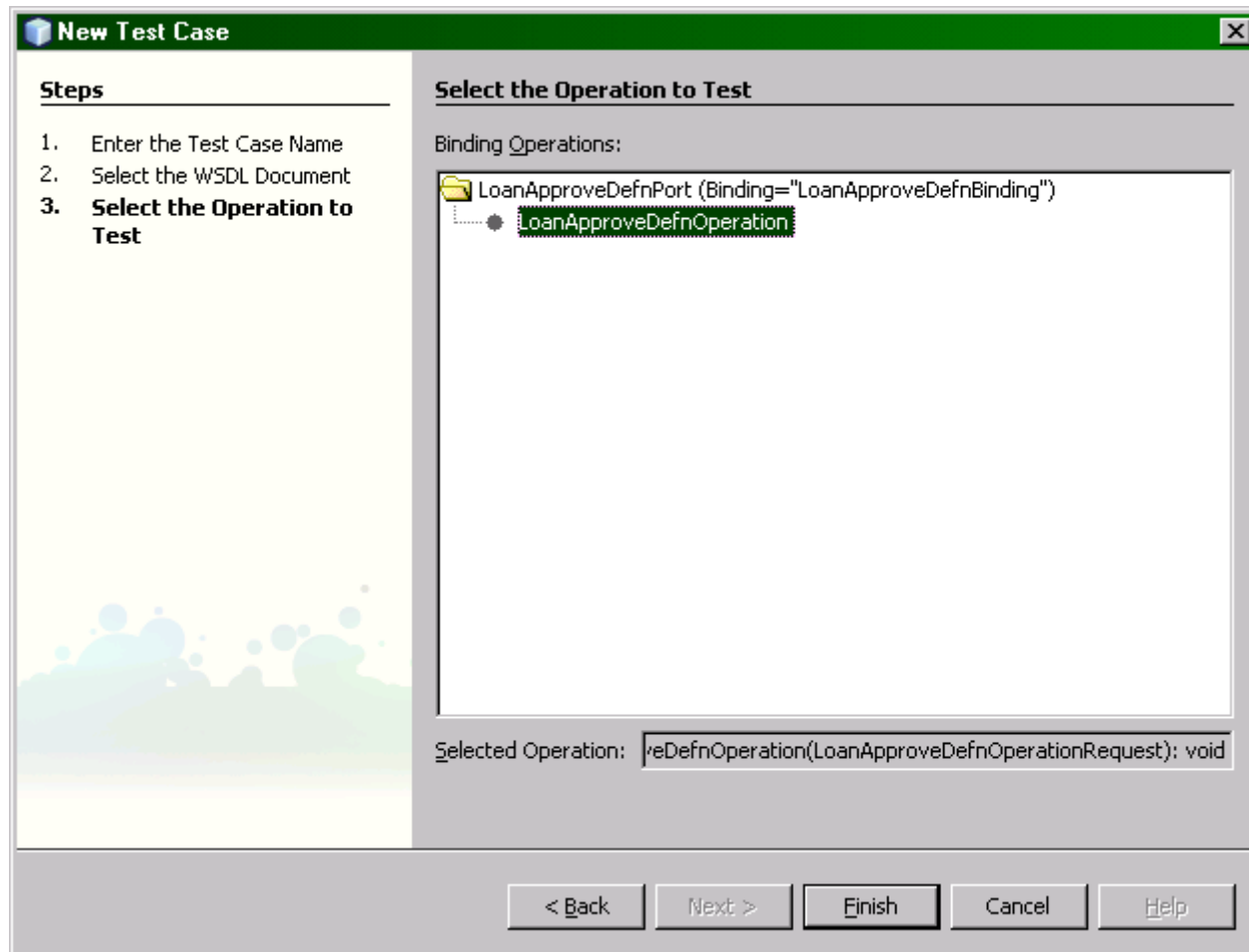
WSDL Documents:

- LoanApprovalServiceAssembly - Process Files
- LoanApproval - Source Packages
 - CreditCheckDefn.wsdl
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl**

Selected WSDL: jme\NetBeansProjects\LoanApproval\src\LoanApproveDefn.wsdl

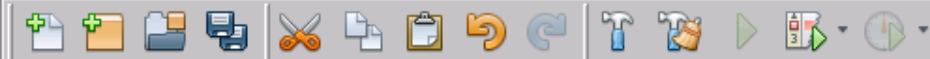
< Back Next > Finish Cancel Help

1 <soapenv:Envelope xsi:schemaLocation="http://schemas.xmlsoap.org/soap/en



Entrée du cas de test

```
LoanApproveDefn.wsdl x LoanApproveService.bpel x LoanApplicationStatusDefn.wsdl x Input.xml x
1 <soapenv:Envelope xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/ http://schemas.xmlsoap.org/soap/envelope/">
2   <soapenv:Body>
3     <hom:LoanInformation>
4       <BorrowerInformation>
5         <Name>?string?</Name>
6         <SSN>?string?</SSN>
7         <Address>
8           <Addr1>?string?</Addr1>
9           <CityState>?string?</CityState>
10          <Zip>?string?</Zip>
11          <Telephone>?string?</Telephone>
12        </Address>
13      </BorrowerInformation>
14      <LoanDetails>
15        <LoanAmount>?string?</LoanAmount>
16        <HomePrice>?string?</HomePrice>
17        <HomeAddress>
18          <Addr1>?string?</Addr1>
19          <CityState>?string?</CityState>
20          <Zip>?string?</Zip>
21          <Telephone>?string?</Telephone>
22        </HomeAddress>
23      </LoanDetails>
24    </hom:LoanInformation>
25  </soapenv:Body>
26 </soapenv:Envelope>
```

LoanApproveDefn.wsdl x LoanApproveService.bpel x LoanApplicationStatusDefn.wsdl x Input.xml * x

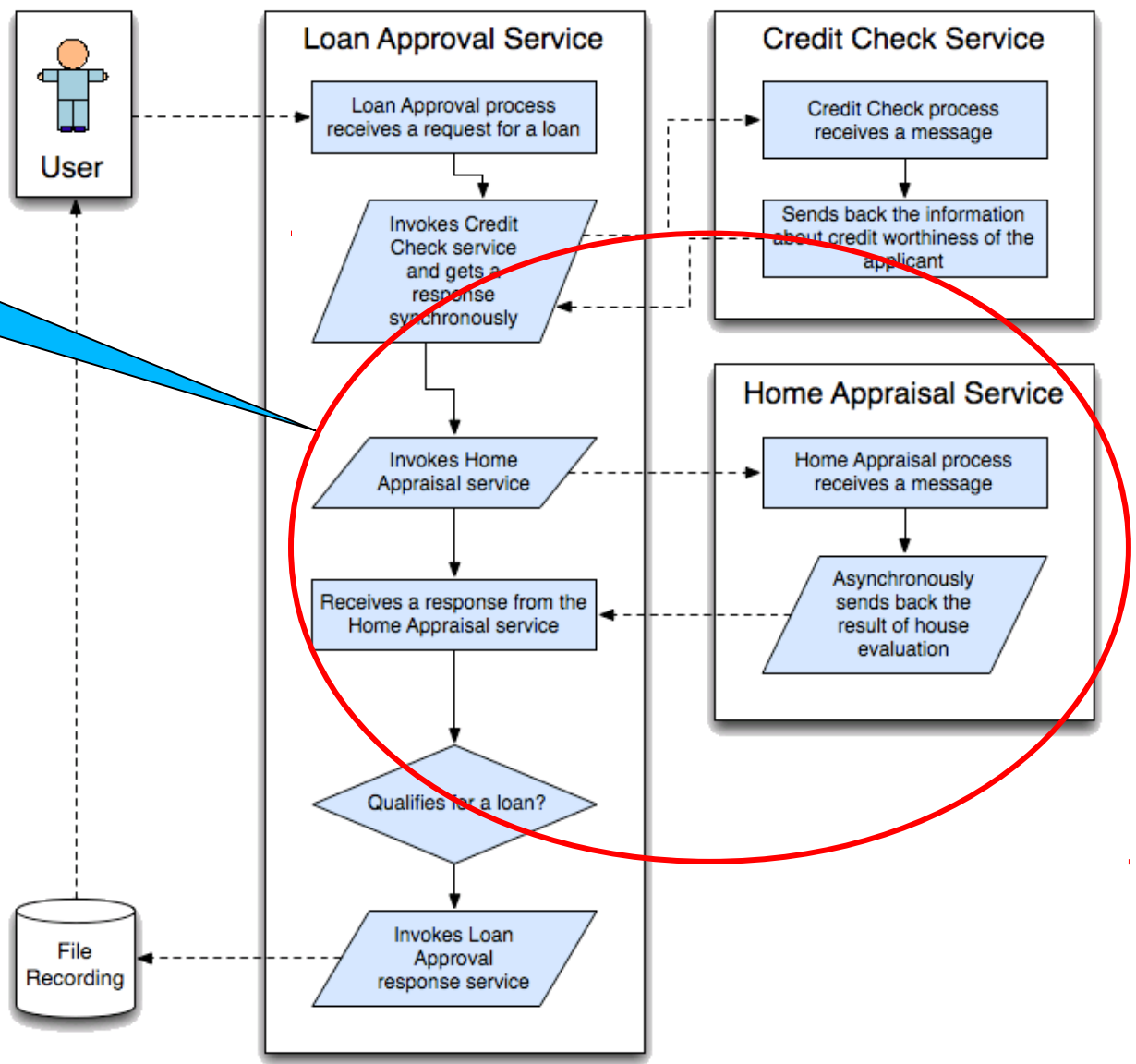


```
1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://schemas.xmlsoap.org/soap/envelope/">
2   <soapenv:Body>
3     <hom:LoanInformation>
4       <BorrowerInformation>
5         <Name>Paul Durand</Name>
6         <SSN>666677723</SSN>
7       <Address>
8         <Addr1?string?/></Addr1>
9         <CityState?string?/></CityState>
10        <Zip?string?/></Zip>
11        <Telephone?string?/></Telephone>
12      </Address>
13    </BorrowerInformation>
14    <LoanDetails>
15      <LoanAmount?string?/></LoanAmount>
16      <HomePrice?string?/></HomePrice>
17      <HomeAddress>
18        <Addr1?string?/></Addr1>
19        <CityState?string?/></CityState>
20        <Zip?string?/></Zip>
21        <Telephone?string?/></Telephone>
22      </HomeAddress>
23    </LoanDetails>
24  </hom:LoanInformation>
25 </soapenv:Body>
26 </soapenv:Envelope>
```

5:1 INS

Soit le processus suivant:

Suite du sujet





LoanApproval

- Process
 - New
 - WSDL Document...
 - BPEL Process...
 - XML Schema...
 - Other...
 - Local History
 - Find... Ctrl+F
 - Paste Ctrl+V
 - Tools
- Cred
- Cred
- Hom
- Loar
- Loar
- Loar

LoanAppro...
Placeholder

NetBeans

Welcome to NetBeans IDE 6.1 My NetBeans

Get Started

- Quick Start Tutorial
- What's New
- Take a Tour

Try a Sample Project

- Java
- Web Services
- Enterprise
- UML
- Identity Blueprints
- C/C++
- Web
- RESTful Web Services
- Mobility
- SOA
- Ruby
- NetBeans Modules

Extend Your IDE

Add more tools into NetBeans IDE delivered by NetBeans team for Desktop, Web, Enterprise, Mobile Java development and Ruby, C/C++, SOA, UML development. Plus many third party plugins for enhanced productivity.

Learn More About

- General Java Programming
- Web Apps
- Ruby and Rails
- SOA
- Swing GUIs
- Java EE
- Mobile Apps
- UML Modeling

New WSDL Document [X]

Steps

1. Choose File Type
- 2. Name and Location**
3. Abstract Configuration
4. Concrete Configuration

Name and Location

File Name:

Project:

Folder:

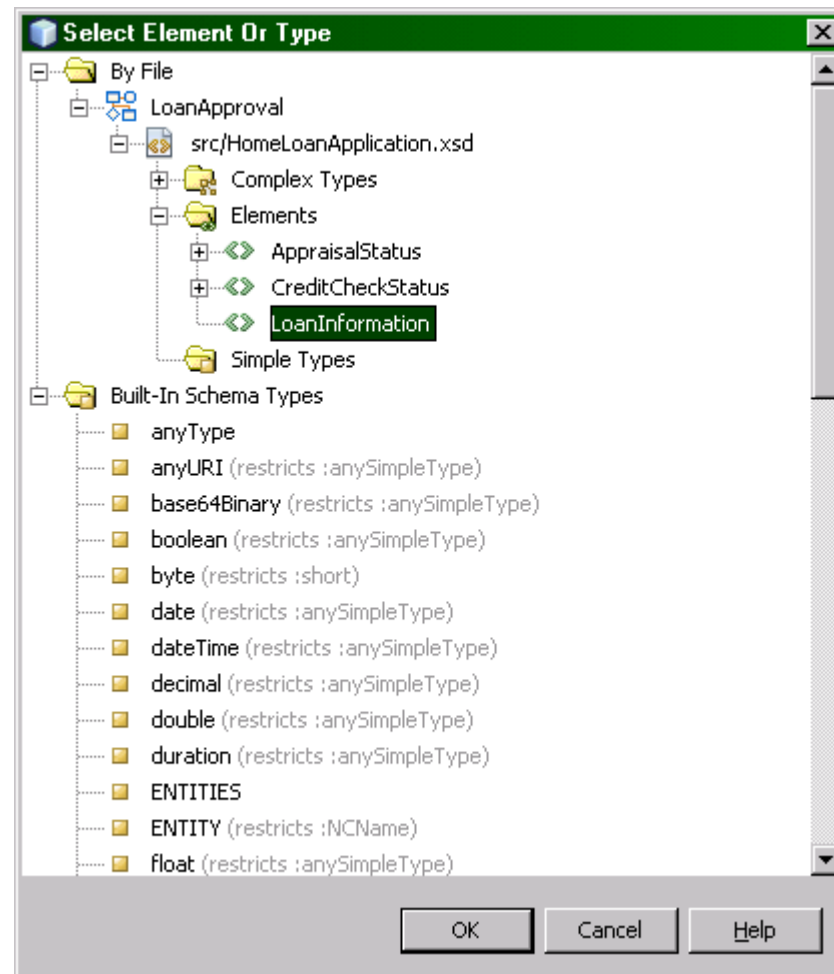
Created File:

Target Namespace:

Import XML Schema File(s)

XML Schema(s):

< Back Next > Finish Cancel Help



New WSDL Document [X]

Steps

1. Choose File Type
2. Name and Location
- 3. Abstract Configuration**
4. Concrete Configuration

Abstract Configuration

Port Type Name: HomeAppraisalDefnPortType

Operation Name: HomeAppraisalDefnOperation

Operation Type: One-Way Operation

Input:

Message Part Name	Element Or Type
part1	ns:LoanInformation

Generate partnerlinktype automatically.

< Back Next > Finish Cancel Help



LoanApproval

- Process
 - New
 - WSDL Document...
 - BPEL Process...
 - XML Schema...
 - Other...
 - Local History
 - Find... Ctrl+F
 - Paste Ctrl+V
 - Tools
- Cred
- Cred
- Hom
- Loar
- Loar
- Loar

LoanAppro...
Placeholder

NetBeans

Welcome to NetBeans IDE 6.1 My NetBeans

Get Started

- Quick Start Tutorial
- What's New
- Take a Tour

Try a Sample Project

- Java
- Web Services
- Enterprise
- UML
- Identity Blueprints
- C/C++
- Web
- RESTful Web Services
- Mobility
- SOA
- Ruby
- NetBeans Modules

Extend Your IDE

Add more tools into NetBeans IDE delivered by NetBeans team for Desktop, Web, Enterprise, Mobile Java development and Ruby, C/C++, SOA, UML development. Plus many third party plugins for enhanced productivity.

Learn More About

- General Java Programming
- Web Apps
- Ruby and Rails
- SOA
- Swing GUIs
- Java EE
- Mobile Apps
- UML Modeling

New WSDL Document

Steps

1. Choose File Type
- 2. Name and Location**
3. Abstract Configuration
4. Concrete Configuration

Name and Location

File Name:

Project:

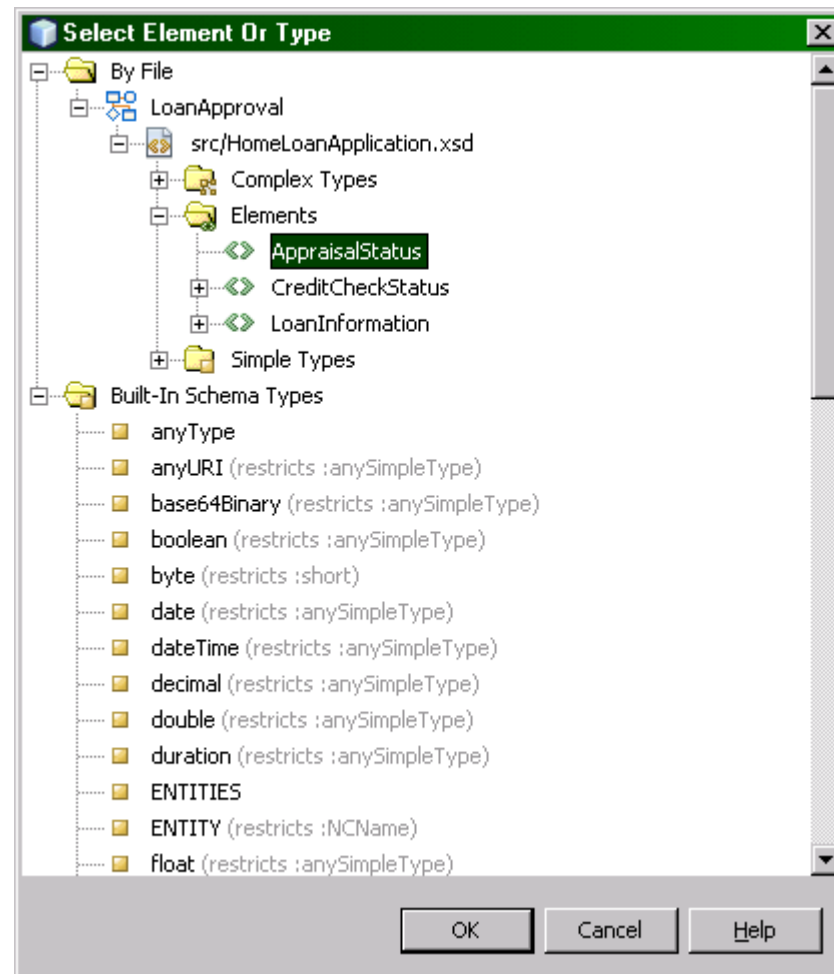
Folder:

Created File:

Target Namespace:

Import XML Schema File(s)

XML Schema(s):



New WSDL Document

Steps

1. Choose File Type
2. Name and Location
- 3. Abstract Configuration**
4. Concrete Configuration

Abstract Configuration

Port Type Name: HomeAppraisalCallbackDefnPortType

Operation Name: HomeAppraisalCallbackDefnOperation

Operation Type: One-Way Operation

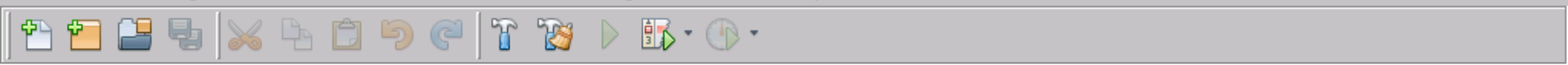
Input:

Message Part Name	Element Or Type
part1	ns:AppraisalStatus

Add Remove

Generate partnerlinktype automatically.

< Back Next > Finish Cancel Help



Project Explorer showing a tree view of the LoanApproval project. The tree includes folders for Process Files, LoanApproval, and LoanApprovalServiceAssembly. Under LoanApproval, there are files like LoanApproveService.bpel and LoanApprovalService.wsdl. Under LoanApprovalServiceAssembly, there is a Placeholder.

Context menu for the LoanApproval folder. The menu items are: New (with a sub-menu), Local History, Find... (Ctrl+F), Paste (Ctrl+V), and Tools.

WSDL Editor showing the structure of HomeAppraisalCallbackDefn.wsdl. The editor displays a tree view of the WSDL document, including elements like HomeAppraisalCallbackDefnOperationRequest, Port Types, HomeAppraisalCallbackDefnPortType, Bindings, Services, Extensibility Elements, and HomeAppraisalCallbackDefn.

Output window showing JUnit Test Results, Refactoring, and HTTP Monitor. The window is currently empty.

New BPEL Process

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name:

Project:

Folder:

Created File:

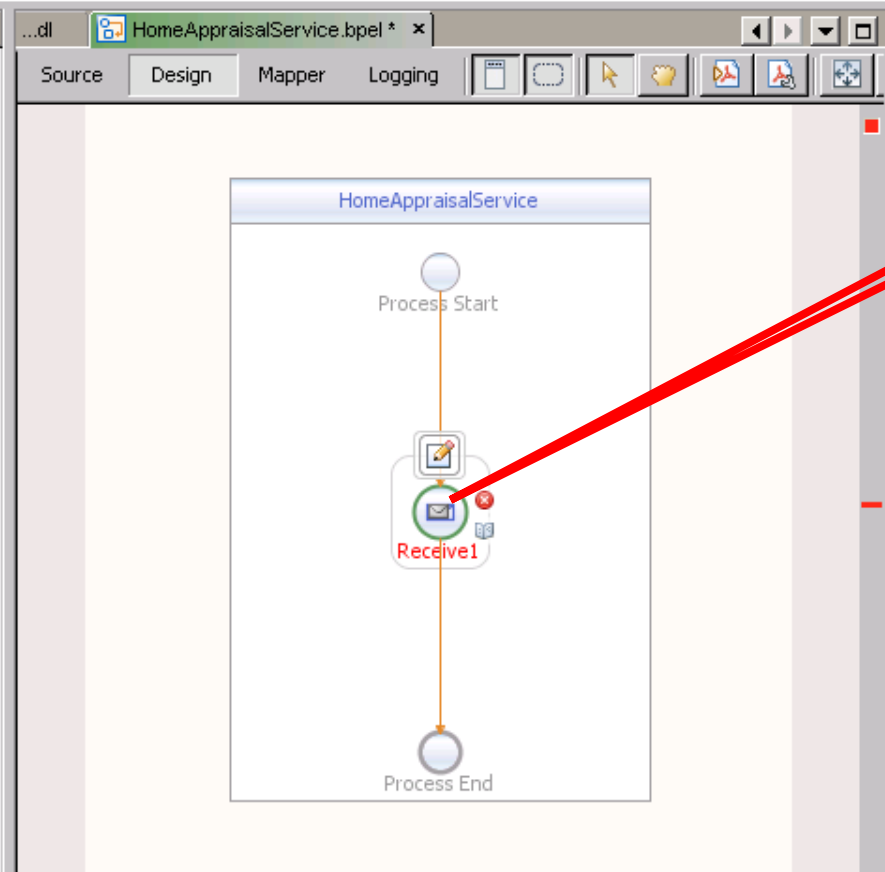
Target Namespace:

< Back Next > Finish Cancel Help



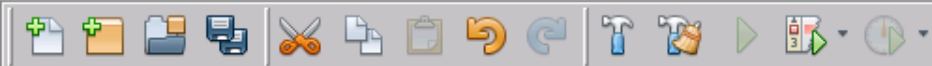
Files Services Receiv...

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - LoanApprovalServiceAssembly
 - Placeholder



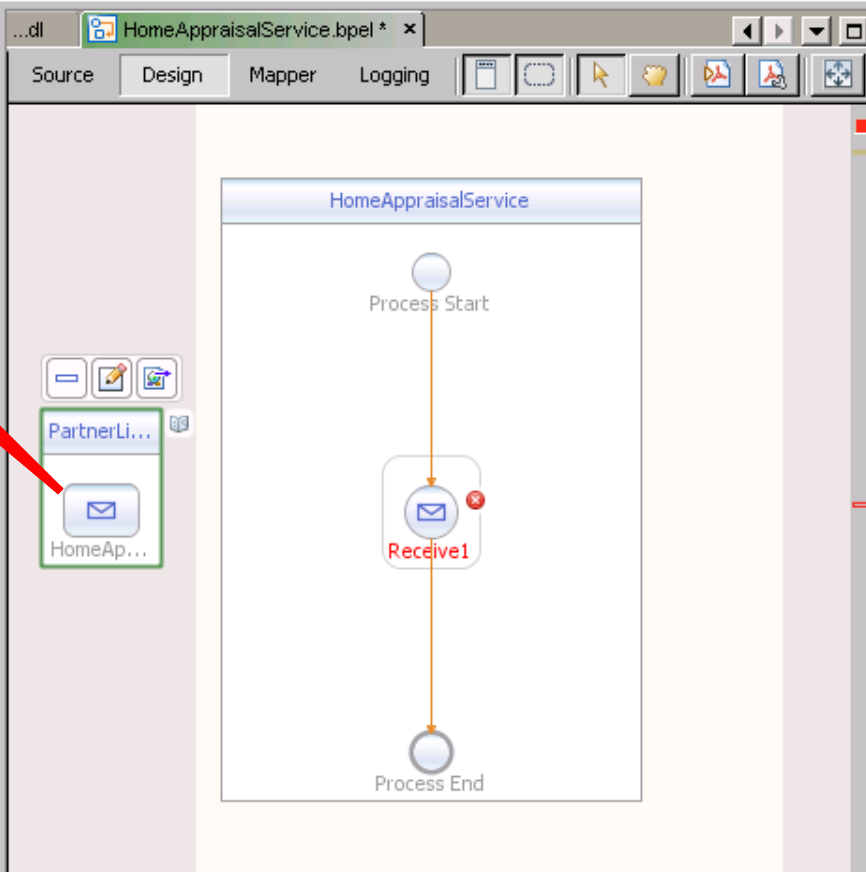
Palette

- Web Service
 - Invoke
 - Reply
 - Partner Link
 - Receive
- Basic Activities
 - Assign
 - Wait
 - ReThrow
 - Compensate
 - Empty
 - Throw
 - Exit
 - CompensateScope
- Structured Activities
 - If
 - ForEach
 - Sequence
 - While
 - Pick
 - Scope
 - RepeatUntil
 - Flow



Files Services Partner...

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl**
 - HomeAppraisalService.bpel
 - HomeLoanApplication.wsdl
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - LoanApprovalServiceAssembly
 - Placeholder



Palette

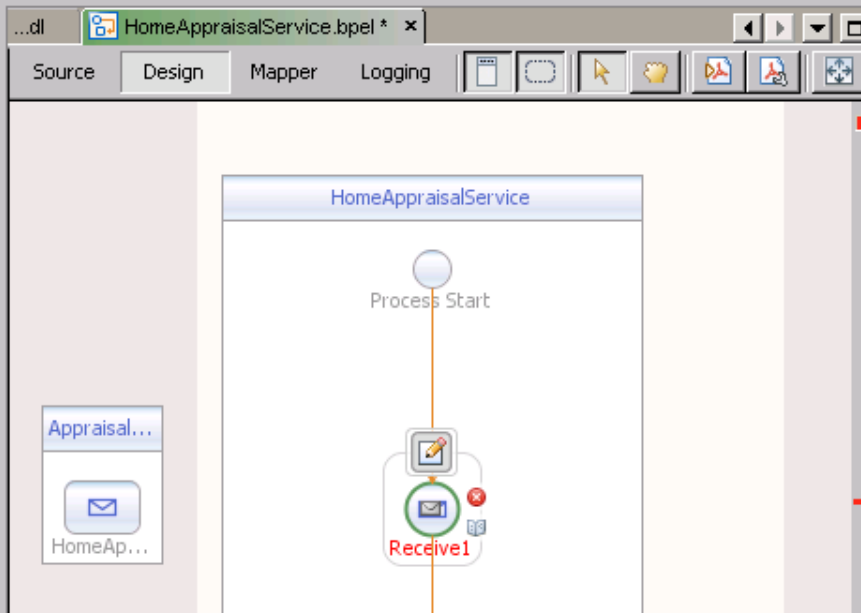
- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities
 - If
 - While
 - RepeatUntil
 - ForEach
 - Pick
 - Flow
 - Sequence
 - Scope



Files Services Receiv...

LoanApproval

- Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- LoanApprovalServiceAssembly
- Placeholder



Palette

Web Service

- Invoke
- Receive
- Reply
- Partner Link

Basic Activities

- Assign
- Empty
- Wait
- Throw
- ReThrow
- Exit
- Compensate
- CompensateScope

Structured Activities

- If
- While
- RepeatUntil
- ForEach
- Pick
- Flow
- Sequence
- Scope

Receive1 [Receive] - Property Editor

Main Correlations

Name:

Partner Link:

Operation:

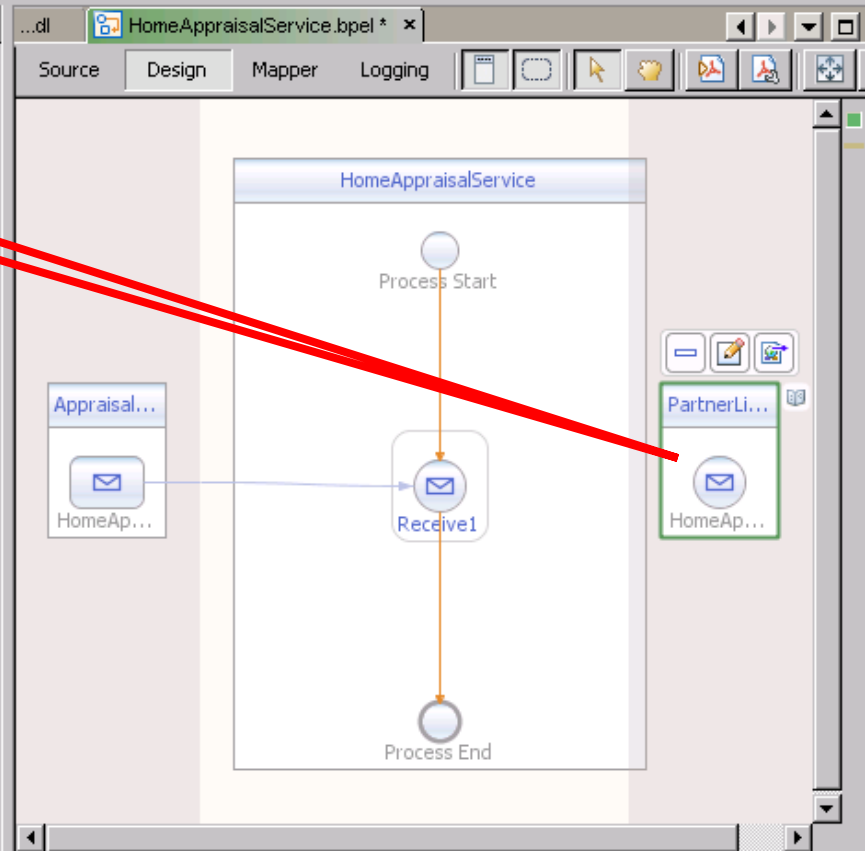
Input Variable:

Create Instance



Files Services Partner...

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl**
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - LoanApprovalServiceAssembly
 - Placeholder



Palette

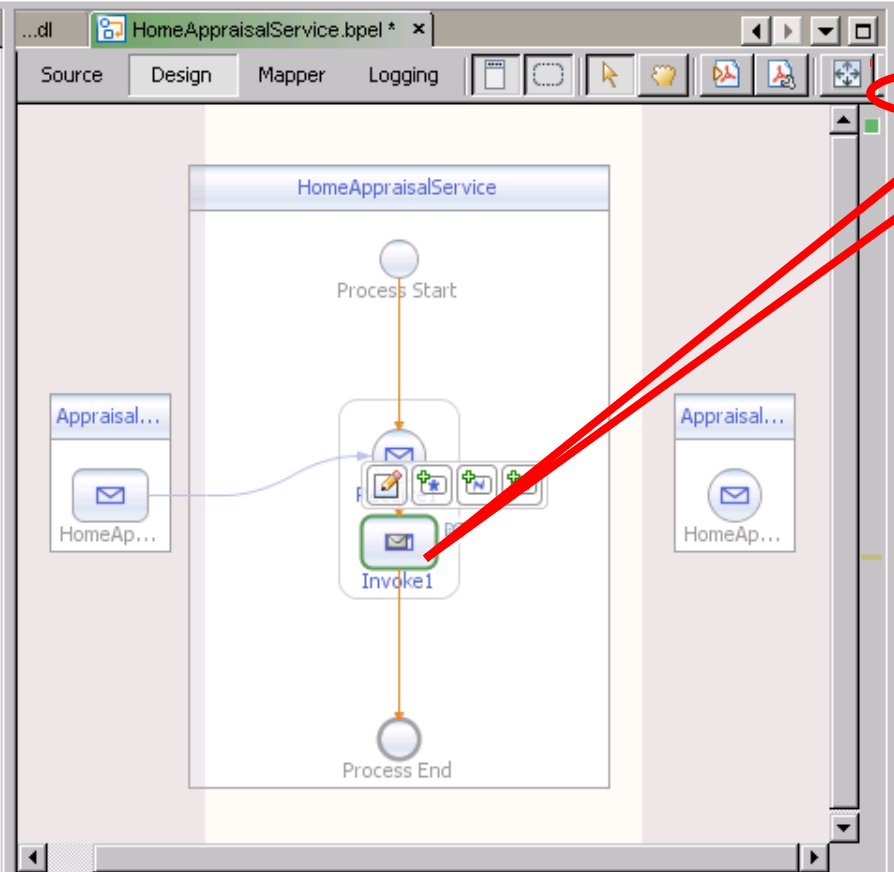
- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities
 - If
 - While
 - RepeatUntil
 - ForEach
 - Pick
 - Flow
 - Sequence
 - Scope

Output JUnit Test Results Refactoring HTTP Monitor



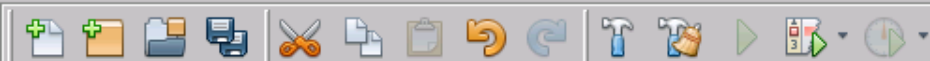
Files Services Invoke...

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - LoanApprovalServiceAssembly
 - Placeholder



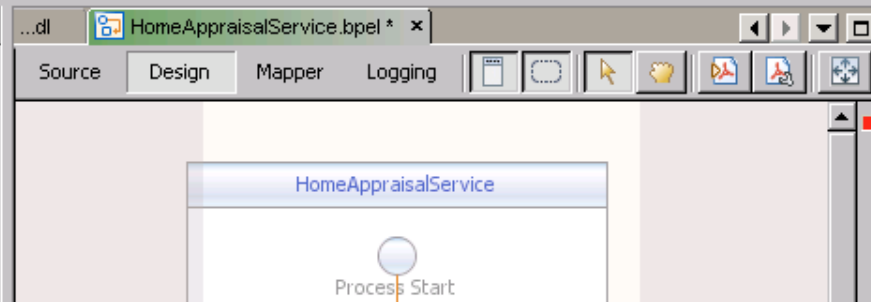
Palette

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities
 - If
 - While
 - RepeatUntil
 - ForEach
 - Pick
 - Flow
 - Sequence
 - Scope



Files Services Invoke...

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - LoanApprovalServiceAssembly
 - Placeholder



Palette

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities
 - If
 - While
 - RepeatUntil
 - ForEach
 - Pick
 - Flow
 - Sequence
 - Scope

Properties

Invoke1 [Invoke] - Property Editor

Main Correlations

Name: Invoke1

Partner Link: AppraisalCallbackPartner

Operation: HomeAppraisalCallbackDefnOperation

Input Variable: Create ... Browse ...

Output Variable: Create ... Browse ...

Ok Cancel Help

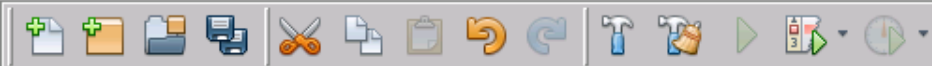
New Input Variable

Name: HomeAppraisalCallbackDefnOperationIn

Type: jckDefn:HomeAppraisalCallbackDefnOperationRequest

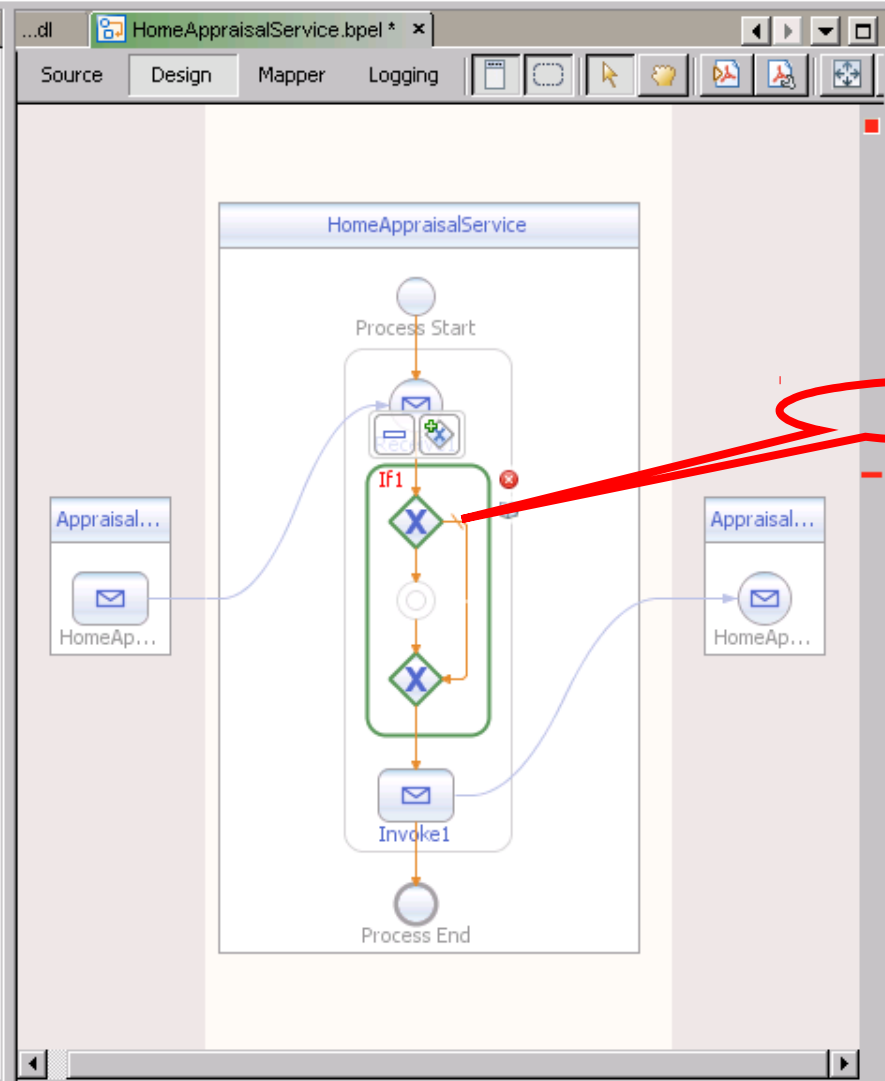
Scope: HomeAppraisalService

Ok Cancel



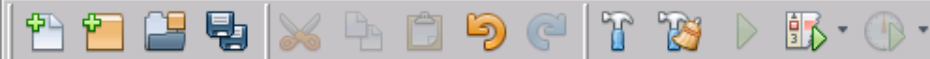
Files Services | If1 [If]...

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - LoanApprovalServiceAssembly
 - Placeholder

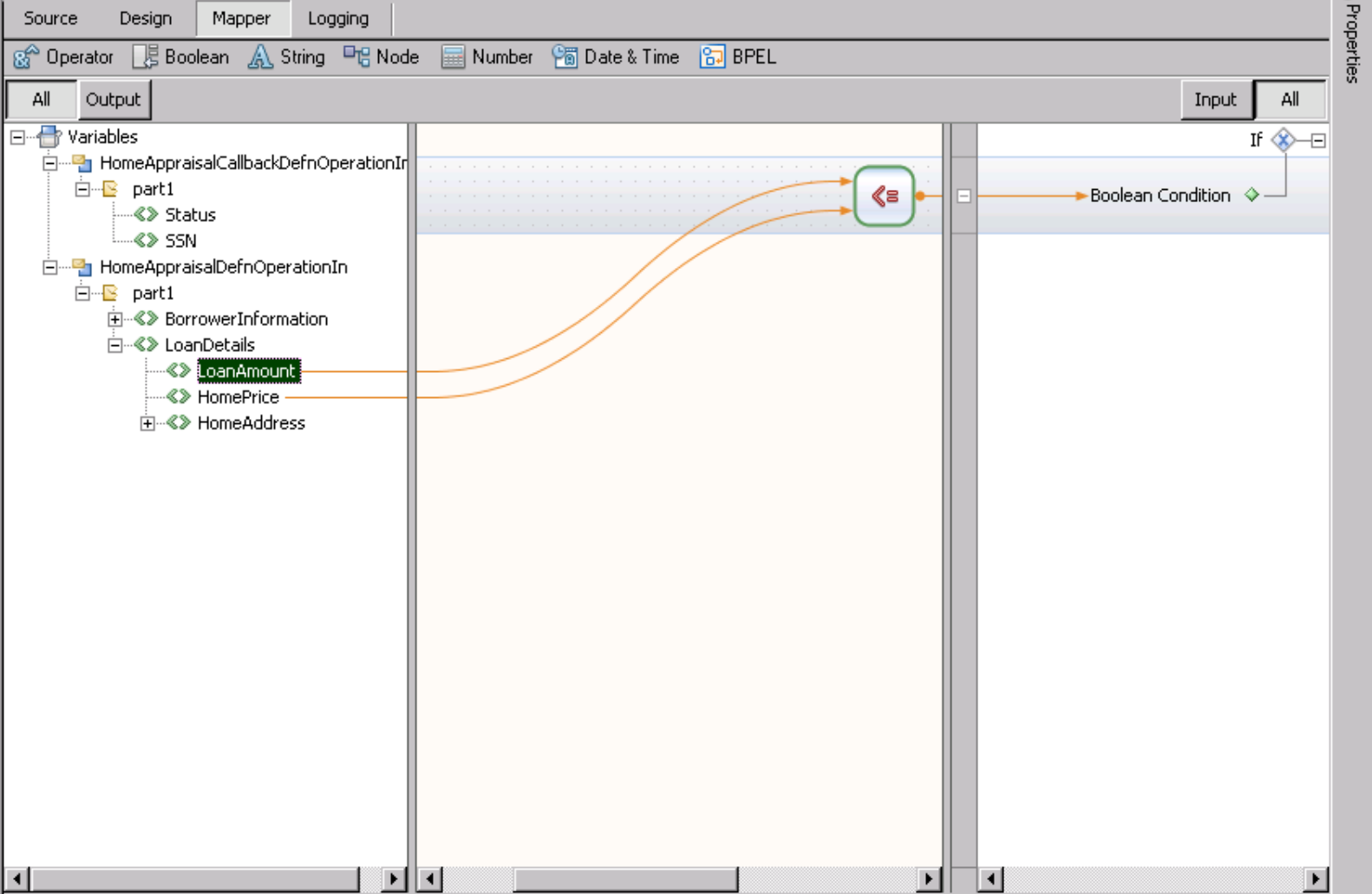


Palette

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities
 - If
 - While
 - RepeatUntil
 - ForEach
 - Pick
 - Flow
 - Sequence
 - Scope



- LoanApproval
 - Process Files
 - CreditCheckDefn
 - CreditCheckServ
 - HomeAppraisalCa
 - HomeAppraisalDe
 - HomeAppraisalSe
 - HomeLoanApplica
 - LoanApplicationS
 - LoanApproveDef
 - LoanApproveSer
 - LoanApprovalService
 - Placeholder

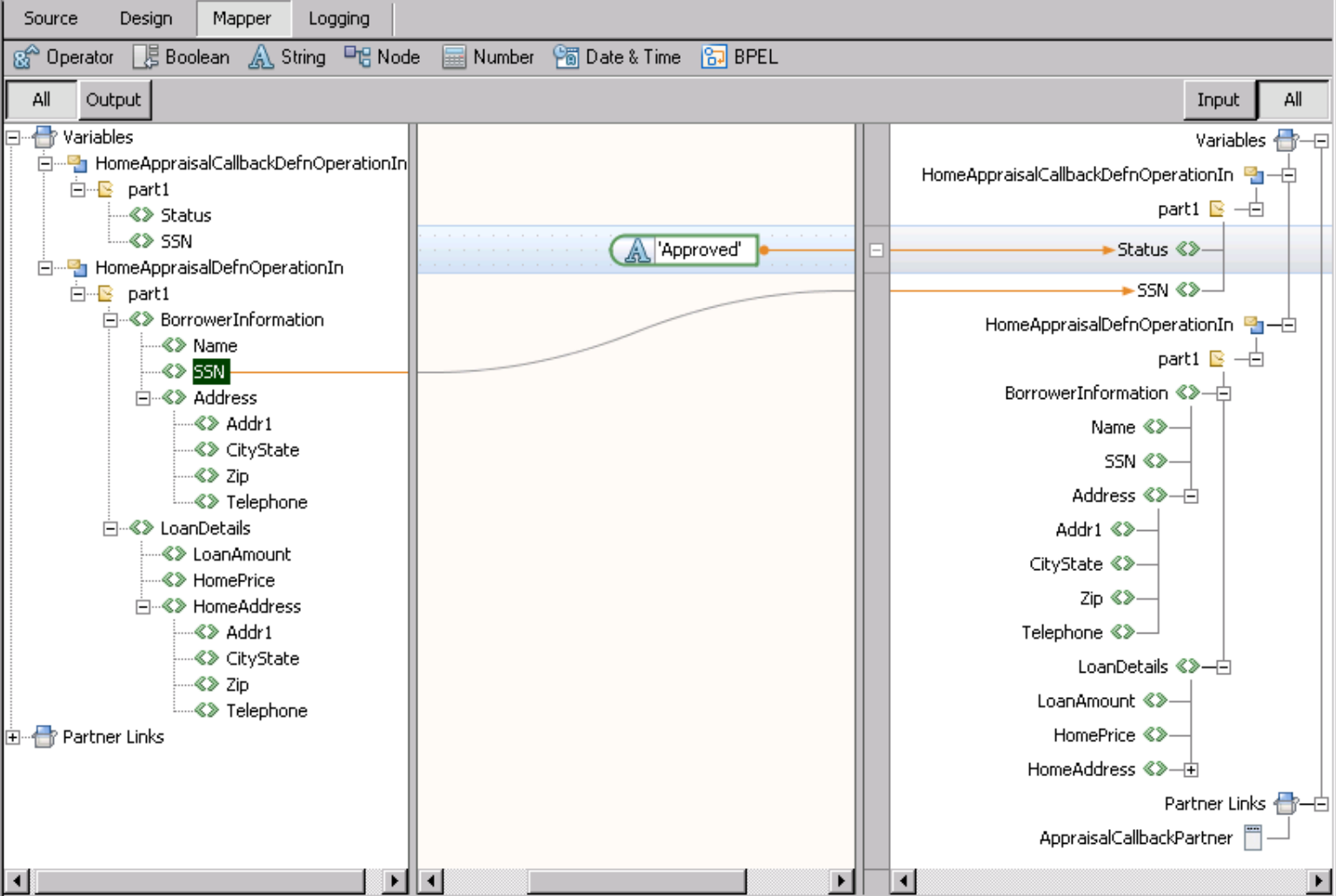




- LoanApproval
 - Process Files
 - CreditCheckDefn
 - CreditCheckServ
 - HomeAppraisalCa
 - HomeAppraisalDe
 - HomeAppraisalSe
 - HomeLoanApplica
 - LoanApplicationS
 - LoanApproveDef
 - LoanApproveSer
 - LoanApprovalService
 - Placeholder

Source Design Mapper Logging

Operator Boolean String Node Number Date & Time BPEL



Output JUnit Test Results Refactoring HTTP Monitor

Properties





Project Explorer showing a tree structure:

- LoanApproval
 - Process Files
 - CreditCheckDefn
 - CreditCheckServ
 - HomeAppraisalCa
 - HomeAppraisalDe
 - HomeAppraisalSe
 - HomeLoanApplica
 - LoanApplicationS
 - LoanApproveDef
 - LoanApproveSer
 - LoanApprovalService
 - Placeholder

Source Editor showing BPEL code with a variable assignment:

```

    <assign>
      <in>
        <input/>
      </in>
      <out>
        <output/>
      </out>
      <copy from='HomeAppraisalDefnOperationIn.part1.SSN' to='HomeAppraisalCallbackDefnOperationIn.part1.SSN'/>
    </assign>
  
```

The diagram shows the flow of data between variables. A variable named 'SSN' is being copied from the 'part1' of 'HomeAppraisalDefnOperationIn' to the 'part1' of 'HomeAppraisalCallbackDefnOperationIn'. The 'Sorry, D...' message is associated with this assignment.

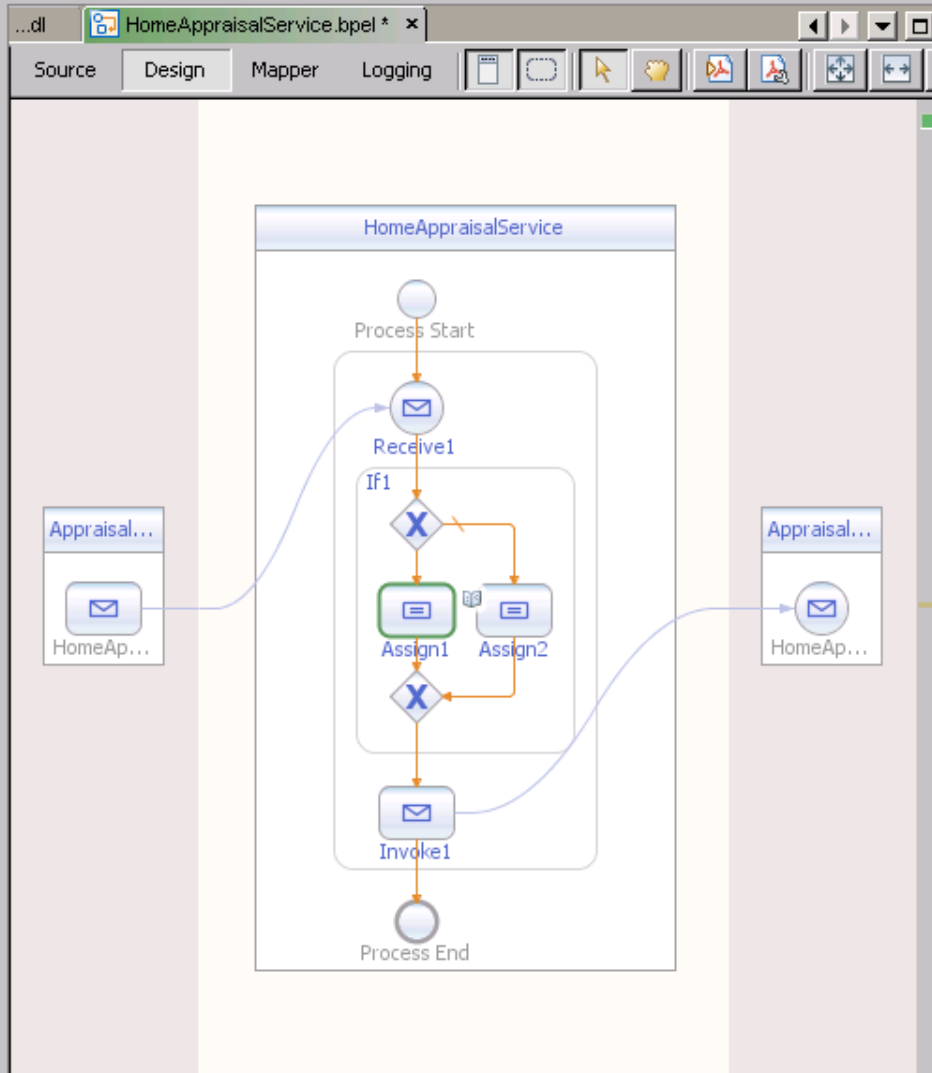
Properties window showing the structure of the variables:

- HomeAppraisalCallbackDefnOperationIn
 - part1
 - Status
 - SSN
- HomeAppraisalDefnOperationIn
 - part1
 - BorrowerInformation
 - Name
 - SSN
 - Address
 - Addr1
 - CityState
 - Zip
 - Telephone
 - LoanDetails
 - LoanAmount
 - HomePrice
 - HomeAddress
 - Addr1
 - CityState
 - Zip
 - Telephone



Files | Serv... | Assi...

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.v
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.w
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- LoanApprovalServiceAssembly
 - Placeholder



Palette

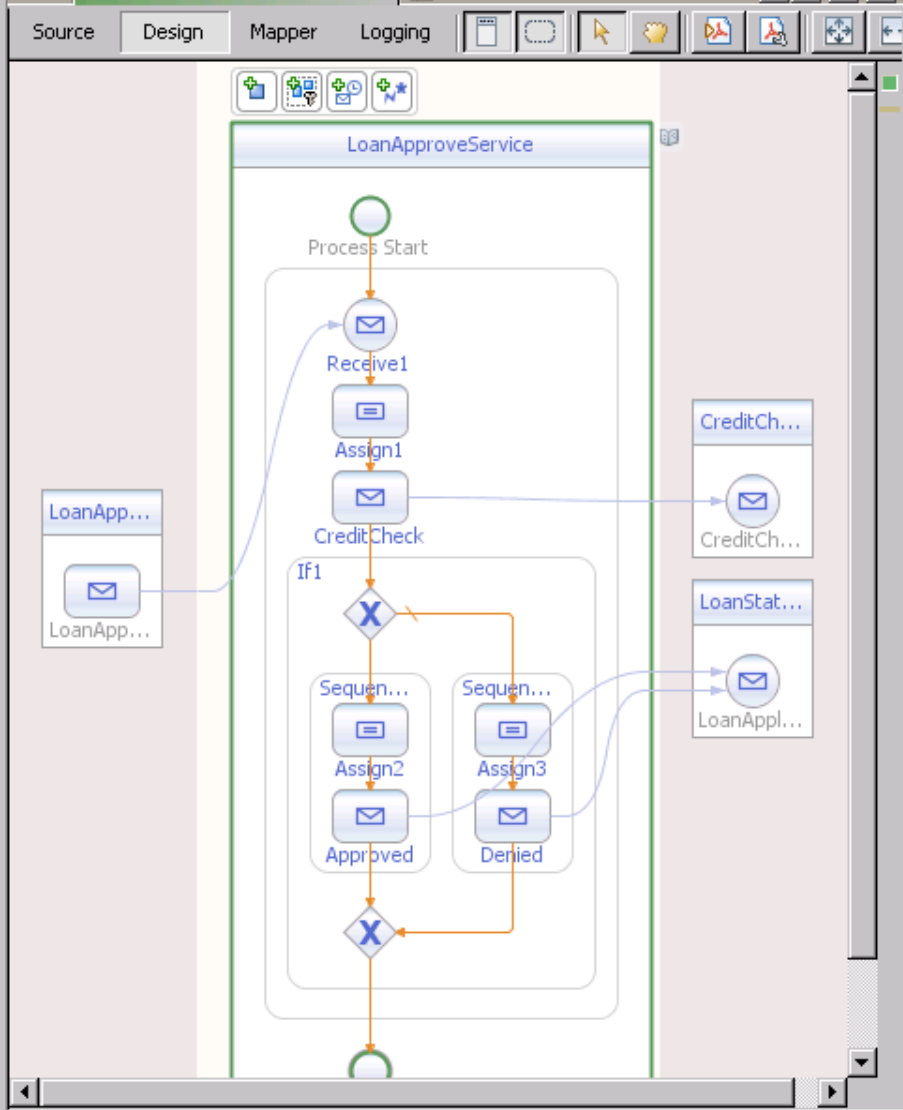
- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities
 - If
 - While
 - RepeatUntil
 - ForEach
 - Pick
 - Flow
 - Sequence
 - Scope



Files | Serv... | Loan...

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - LoanApprovalServiceAssembly
 - Placeholder

Source | Design | Mapper | Logging



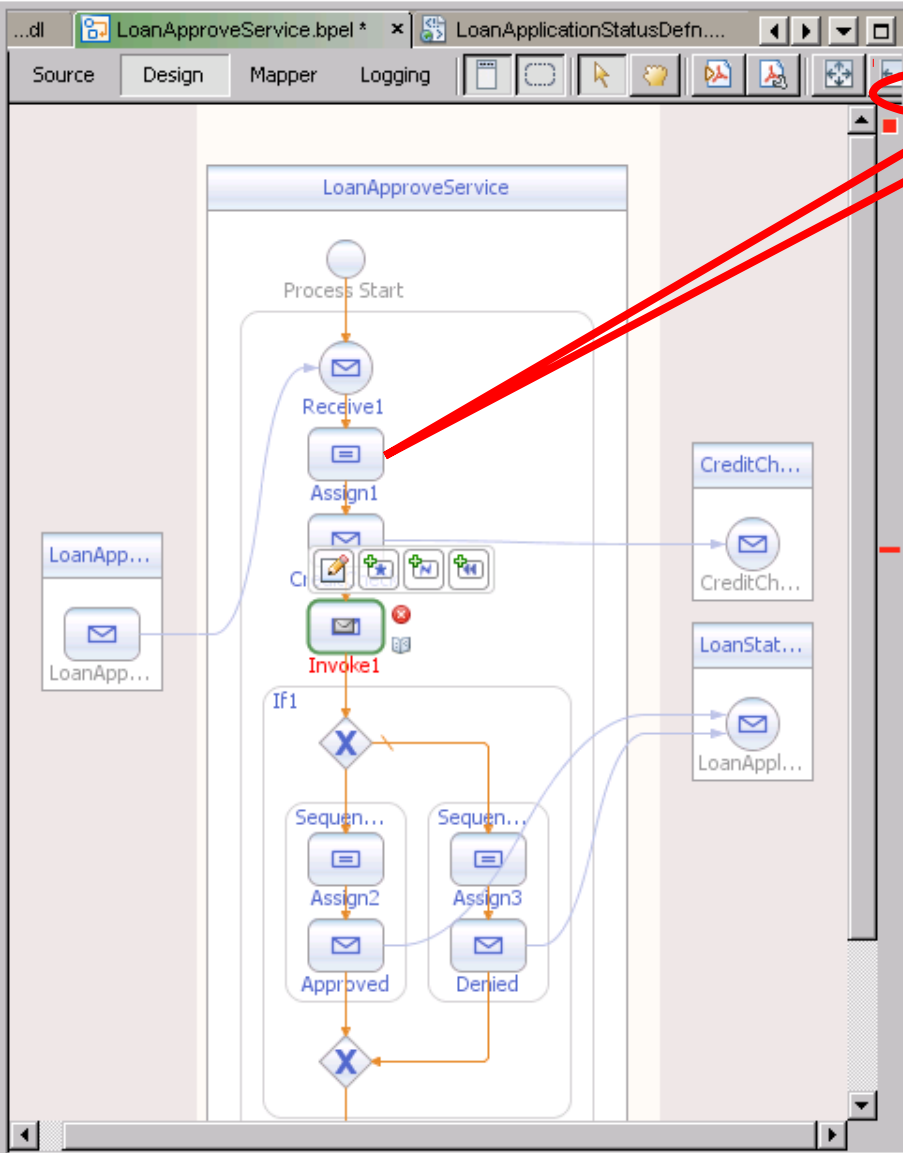
Palette

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities
 - If
 - While
 - RepeatUntil
 - ForEach
 - Pick
 - Flow
 - Sequence
 - Scope



Files | Serv... | Invoke...

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - LoanApprovalServiceAssembly
 - Placeholder



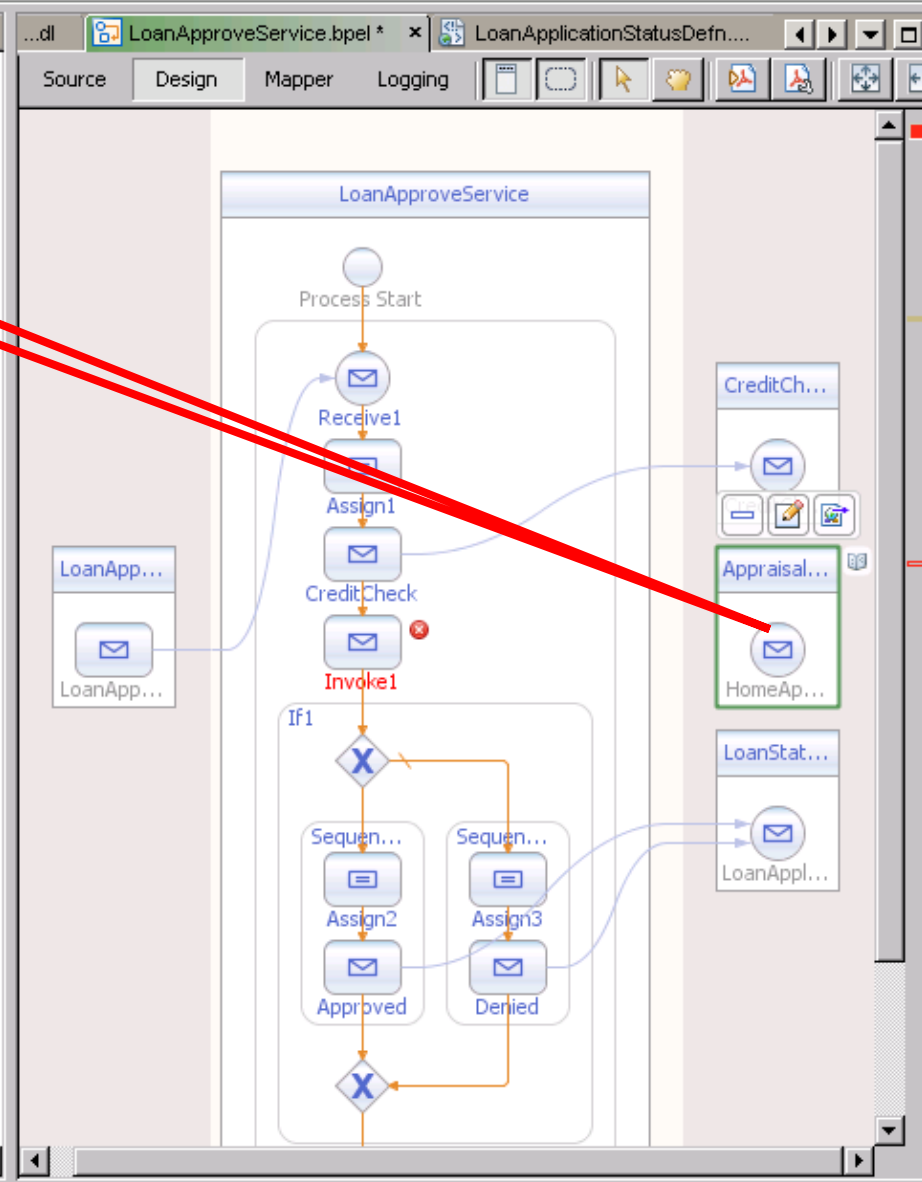
Palette

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Wait
 - ReThrow
 - Compensate
 - Empty
 - Throw
 - Exit
 - CompensateScope
- Structured Activities
 - If
 - ForEach
 - Sequence
 - While
 - Pick
 - Scope
 - RepeatUntil
 - Flow



Files | Serv... | Partne... | ...dl | LoanApproveService.bpel * | LoanApplicationStatusDefn....

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl**
 - HomeAppraisalService.bpel
 - HomeLoanApplication.wsdl
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- LoanApprovalServiceAssembly
- Placeholder



Palette

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities
 - If
 - While
 - RepeatUntil
 - ForEach
 - Pick
 - Flow
 - Sequence
 - Scope

AppraisalProvider [Partner Link] - Property Editor

Name:

WSDL File:

Use Existing Partner Link Type

Partner Link Type:

My Role:

Partner Role:

Use a Newly Created Partner Link Type

Create in File:

Partner Link Type Name:

Process will implement (My Role)

Role Name:

Port Type:

Partner service will implement (Partner Role)

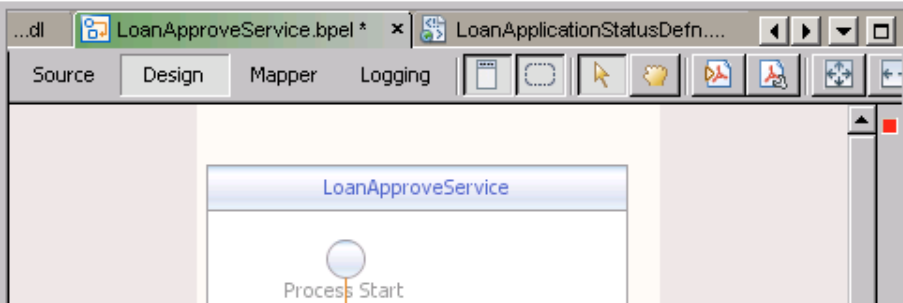
Role Name:

Port Type:



Files | Serv... | Invoke...

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
- LoanApprovalServiceAssembly
- Placeholder



Invoke1 [Invoke] - Property Editor

Main | Correlations

Name:

Partner Link:

Operation:

Input Variable:

Output Variable:

New Input Variable

Name:

Type:

Scope:

Palette

Web Service

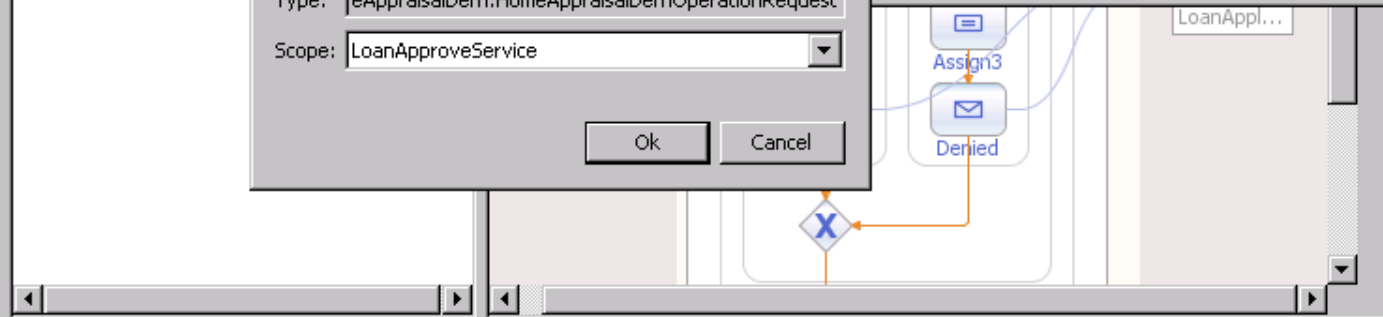
- Invoke
- Receive
- Reply
- Partner Link

Basic Activities

- Assign
- Empty
- Wait
- Throw
- ReThrow
- Exit
- Compensate
- CompensateScope

Structured Activities

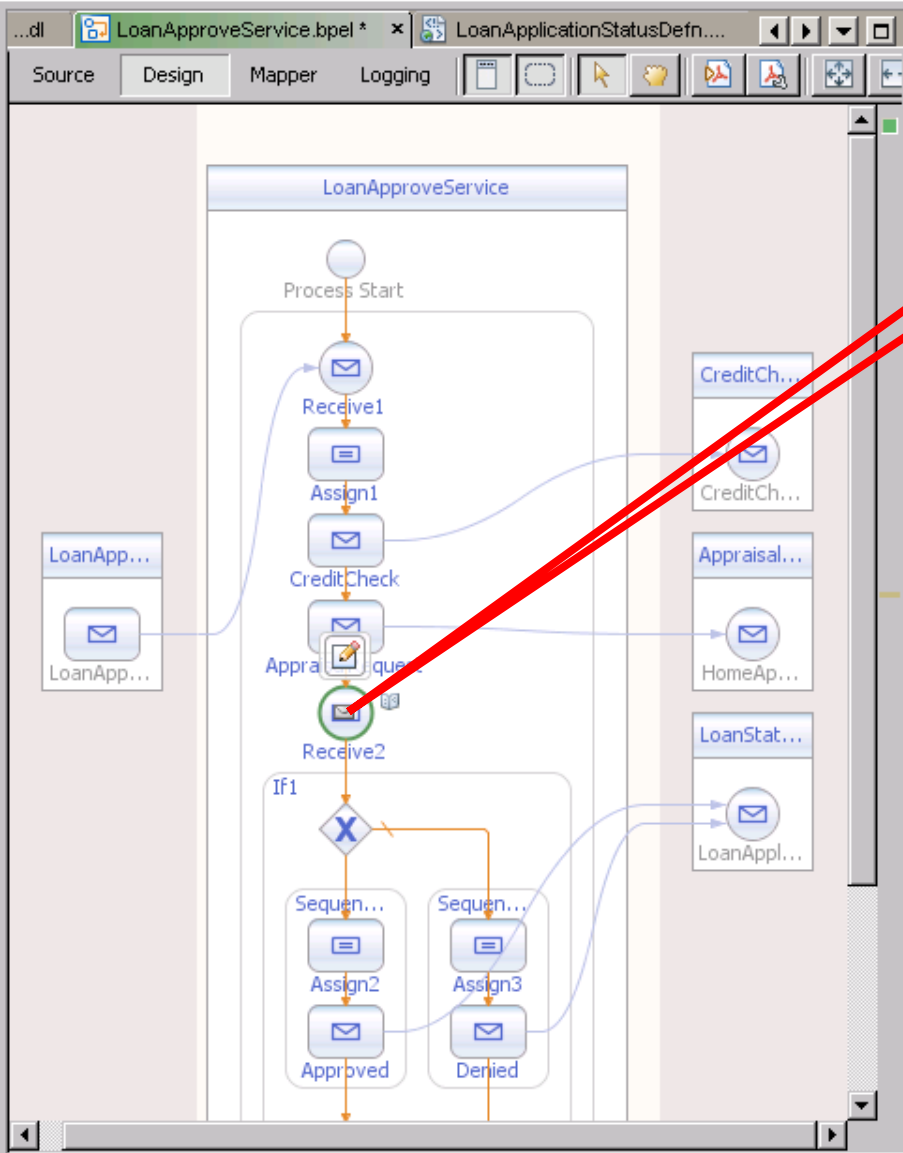
- If
- While
- RepeatUntil
- ForEach
- Pick
- Flow
- Sequence
- Scope





Files | Serv... | Recei...

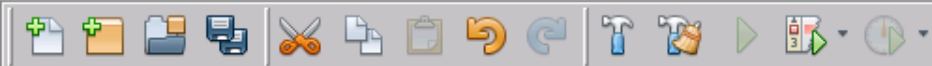
- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - LoanApprovalServiceAssembly
 - Placeholder



Palette

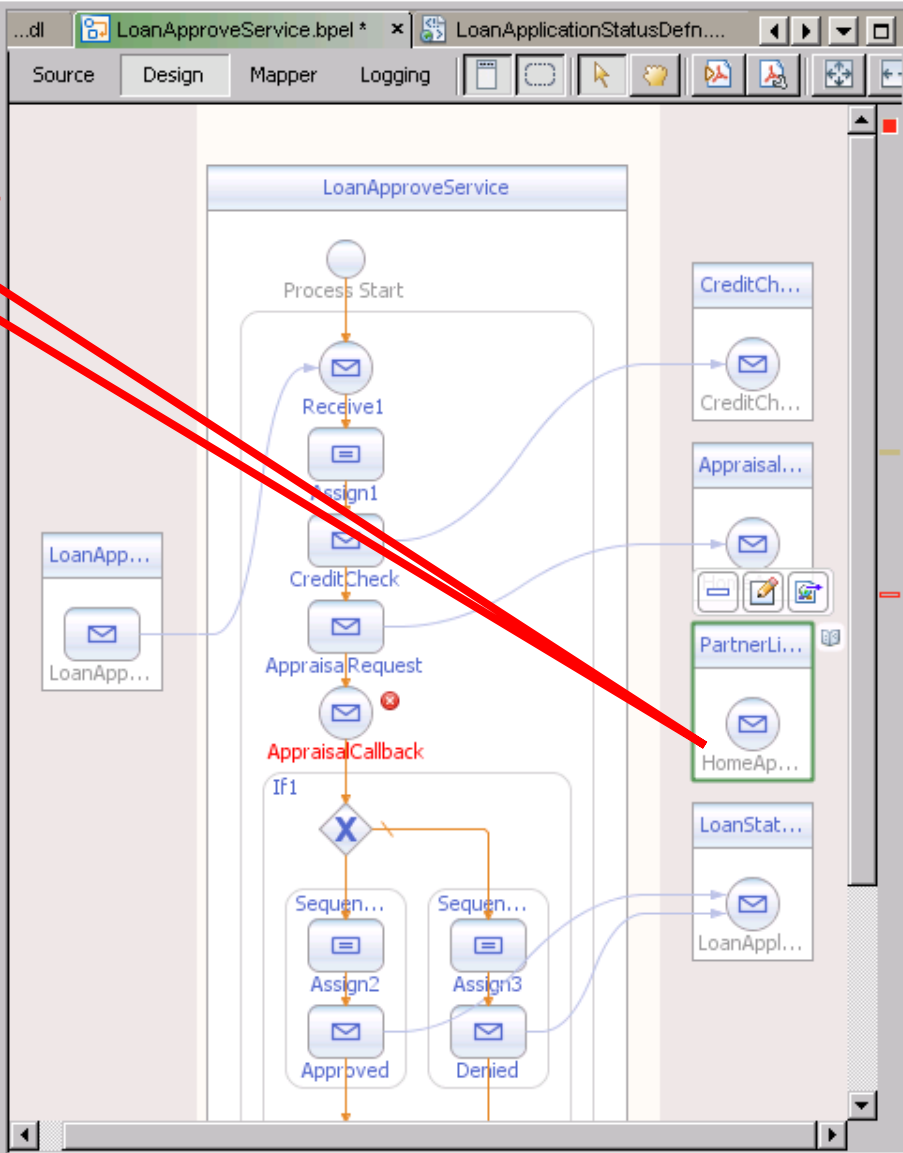
- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Wait
 - ReThrow
 - Compensate
 - Empty
 - Throw
 - Exit
 - CompensateScope
- Structured Activities
 - If
 - ForEach
 - Sequence
 - While
 - Pick
 - Scope
 - RepeatUntil
 - Flow

Container for a set of activities



Files | Serv... | Partne...

- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl**
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - LoanApprovalServiceAssembly
 - Placeholder



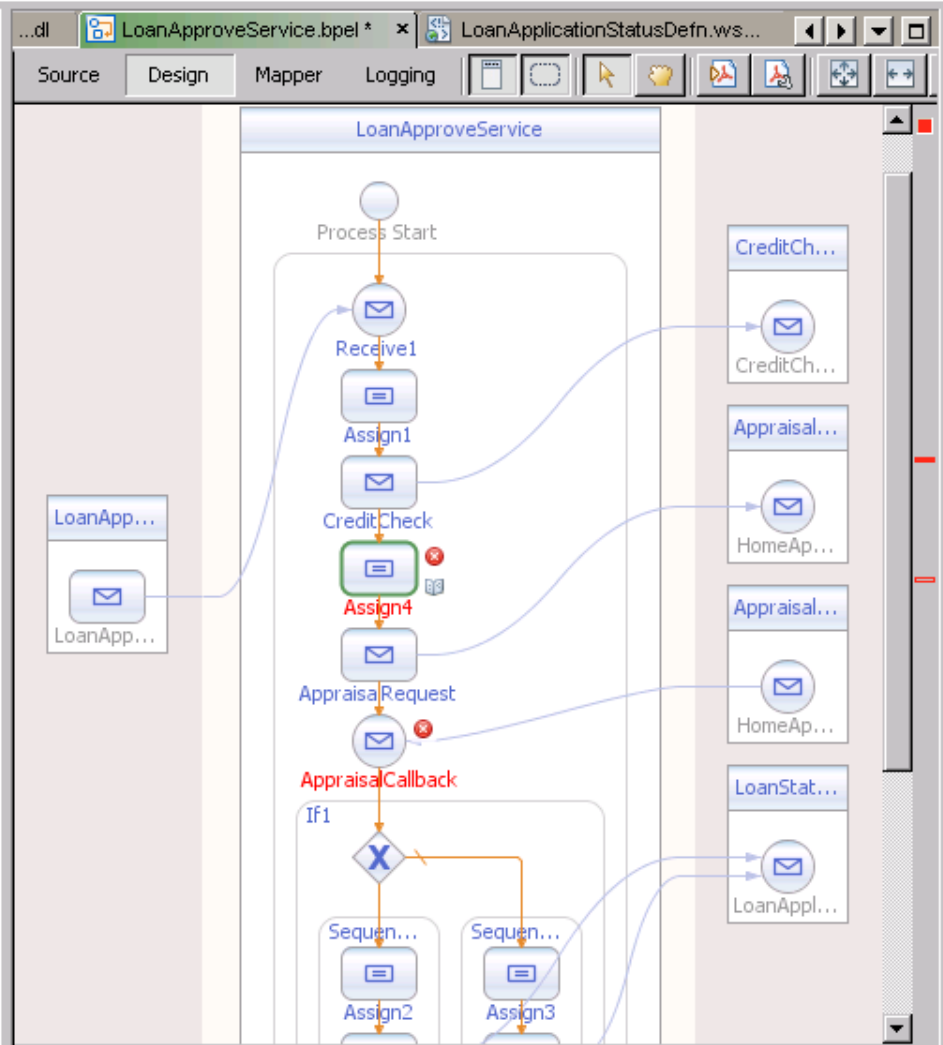
Palette

- Web Service**
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities**
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities**
 - If
 - While
 - RepeatUntil
 - ForEach
 - Pick
 - Flow
 - Sequence
 - Scope



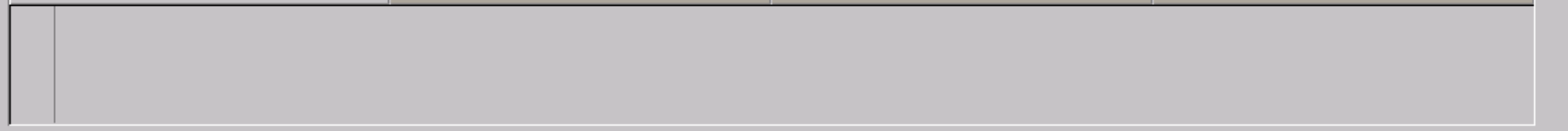
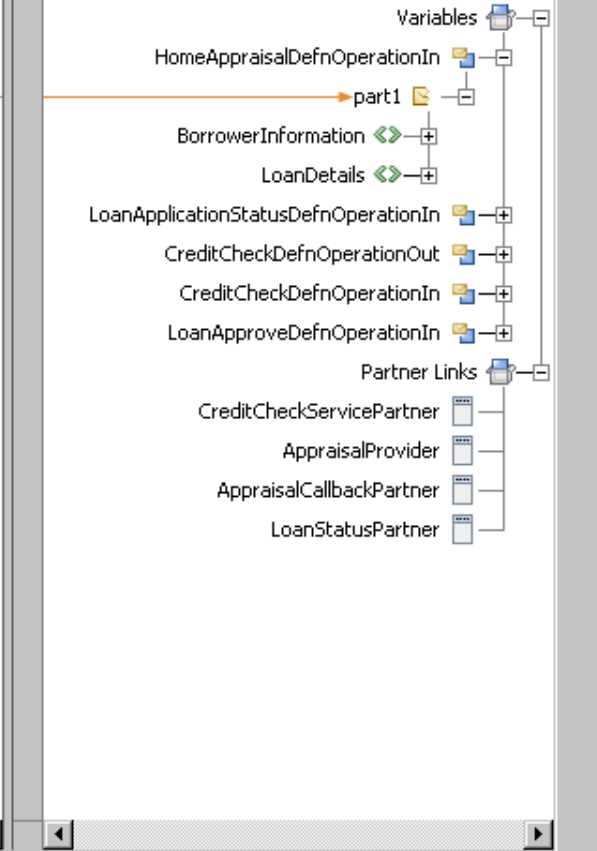
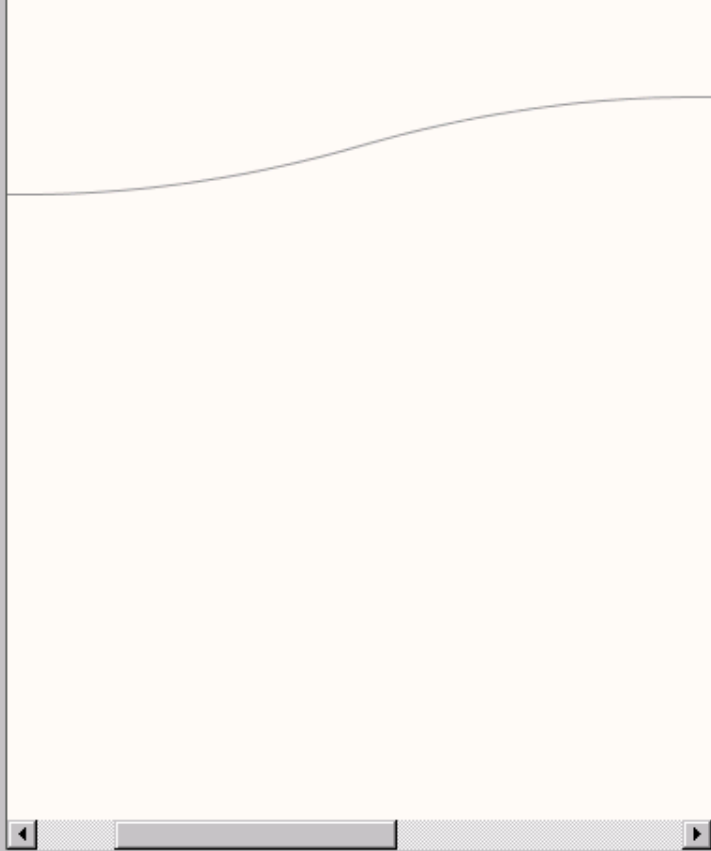
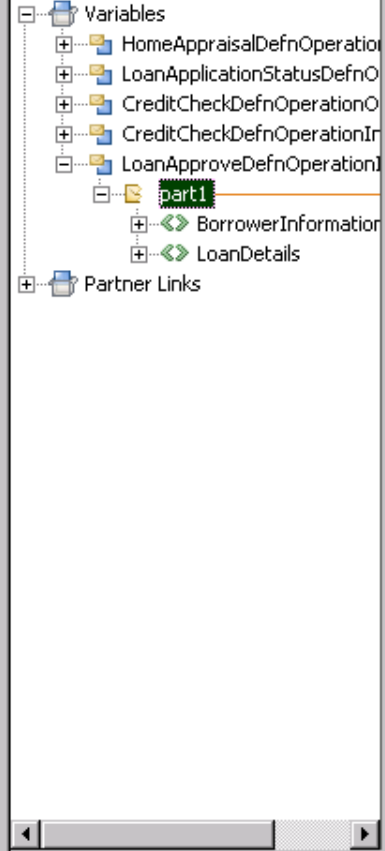
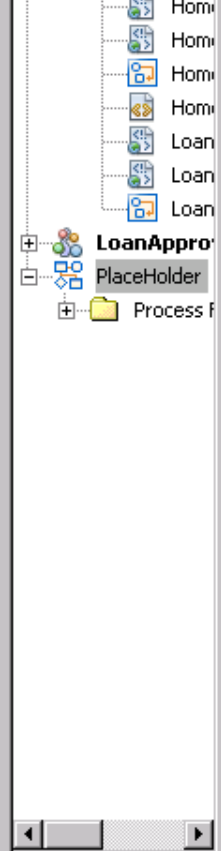
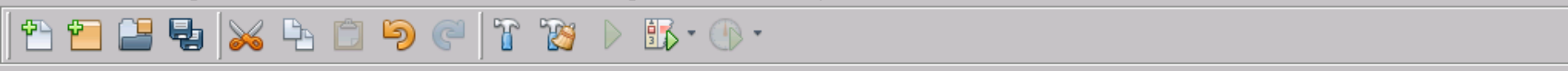
Files | Serv... | Assig...

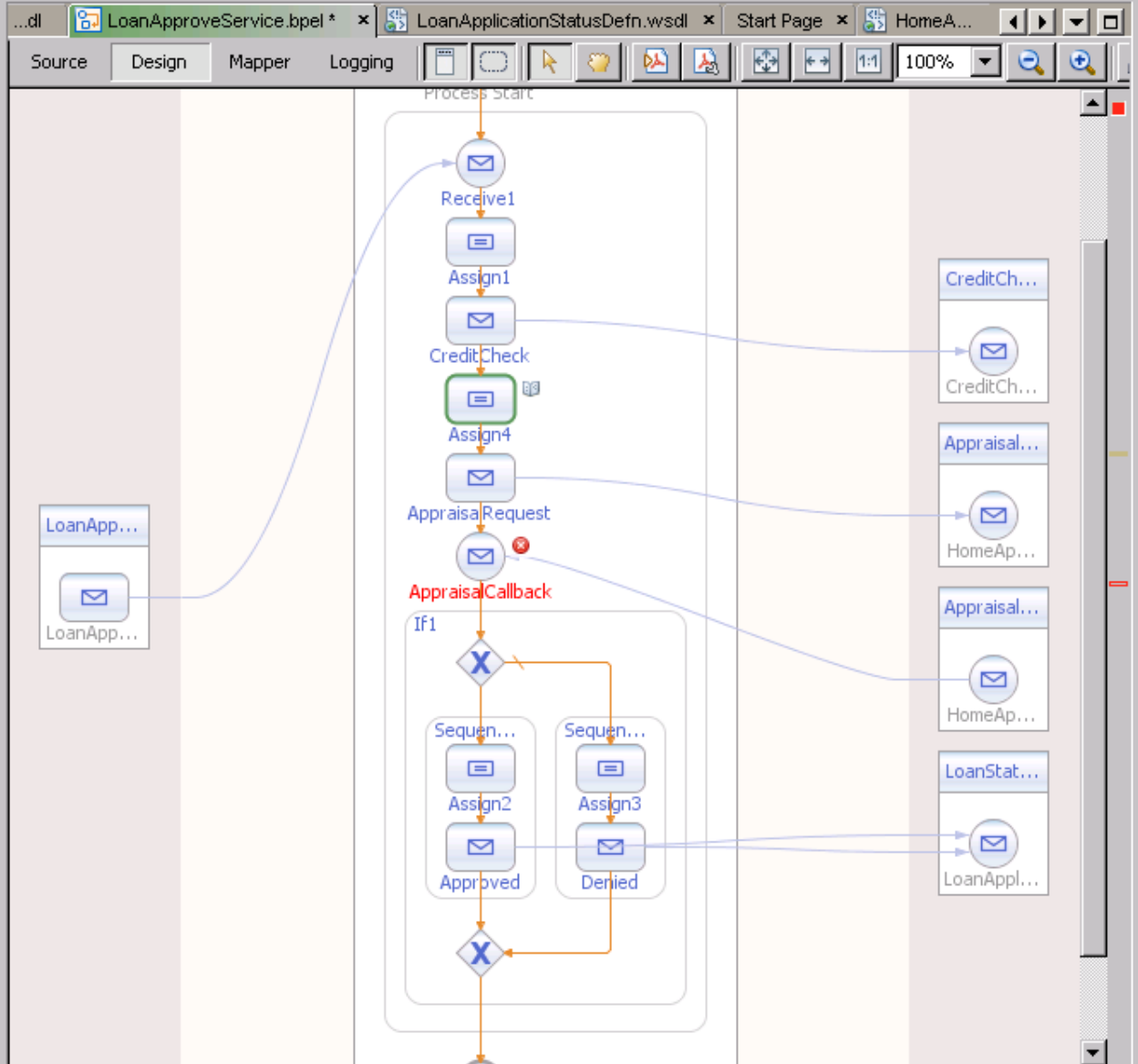
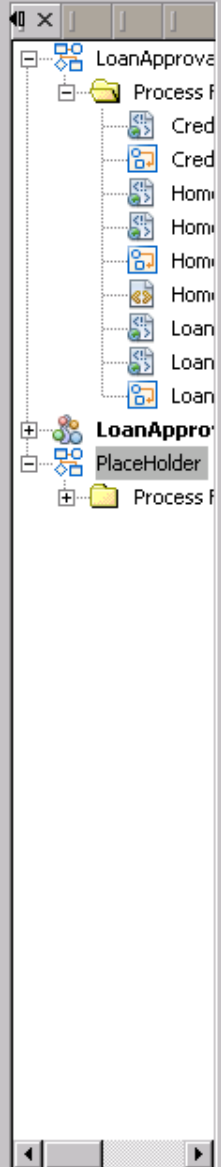
- LoanApproval
 - Process Files
 - CreditCheckDefn.wsdl
 - CreditCheckService.bpel
 - HomeAppraisalCallbackDefn.wsdl
 - HomeAppraisalDefn.wsdl
 - HomeAppraisalService.bpel
 - HomeLoanApplication.xsd
 - LoanApplicationStatusDefn.wsdl
 - LoanApproveDefn.wsdl
 - LoanApproveService.bpel
 - LoanApprovalServiceAssembly**
 - Placeholder
 - Process Files



Palette

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities
 - If
 - RepeatUntil
 - Pick
 - Sequence
 - While
 - ForEach
 - Flow
 - Scope





Palette

- Web Service
 - Invoke
 - Receive
 - Reply
 - Partner Link
- Basic Activities
 - Assign
 - Empty
 - Wait
 - Throw
 - ReThrow
 - Exit
 - Compensate
 - CompensateScope
- Structured Activities
 - If
 - While
 - RepeatUntil
 - ForEach
 - Pick
 - Flow
 - Sequence
 - Scope

Nécessité de corrélér les numéros SSN

The screenshot displays the NetBeans IDE interface with a BPMN diagram for a loan approval process. The diagram is titled "Process Start" and shows a sequence of activities: "Receive1", "Assign1", "CreditCheck", "Assign4", and "AppraisalRequest". The "AppraisalRequest" activity is highlighted with a red border and has a context menu open over it. The context menu options are: "Scroll To Partner Link", "Go To", "Wrap With", "Edit", "Define Correlation ..." (highlighted in green), "Move Up", "Move Down", "Toggle Breakpoint Ctrl+F8", "Delete Supprimer", and "Properties".

The diagram also shows several service call activities (represented by envelope icons) connected to the main process flow: "CreditCh...", "Appraisal...", "Appraisal...", "LoanStat...", and "LoanApp...". The "LoanApp..." activity is also visible in the left palette. The right side of the IDE shows the "Palette" with categories: "Web Service" (Invoke, Reply, Receive, Partner Link), "Basic Activities" (Assign, Empty, Wait, Throw, ReThrow, Exit, Compensate, CompensateScope), and "Structured Activities" (If, While, RepeatUntil, ForEach, Pick, Flow, Sequence, Scope).

The bottom of the IDE shows the "Output" window with tabs for "JUnit Test Results", "Refactoring", and "HTTP Monitor".

Et ainsi de suite....

Fin du module