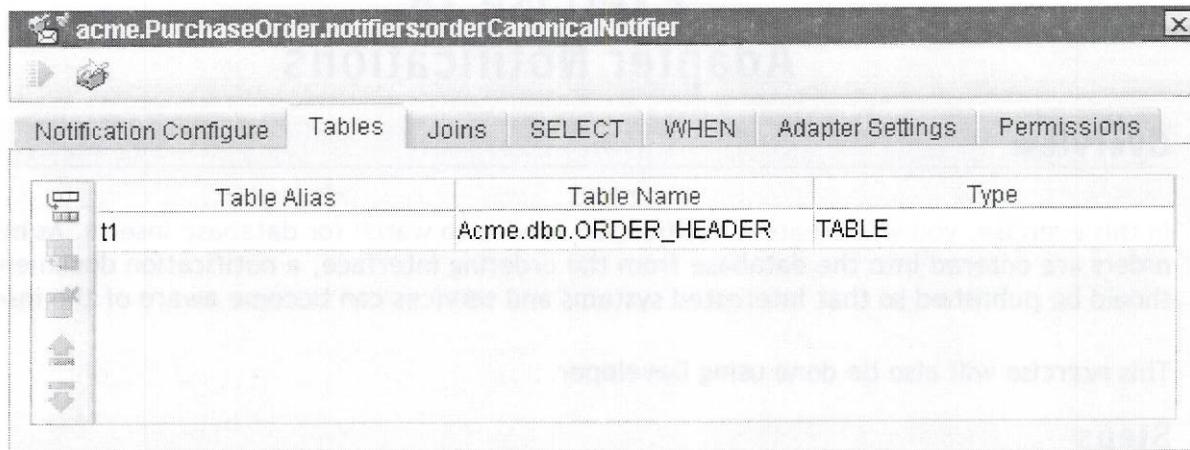
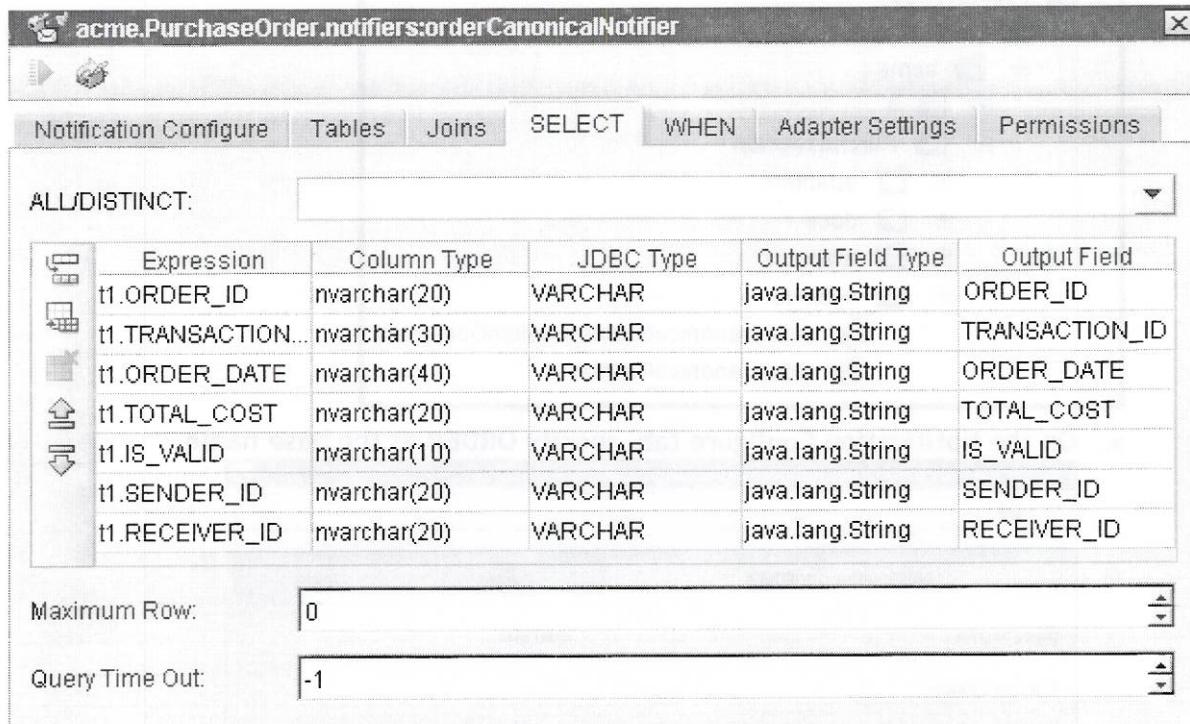


- b. On the Tables tab, select the Acme.dbo.ORDER_HEADER table.



Skip the Joins tab.

- c. On the Select tab, click the “Fill in all rows to the table” button. Before this button becomes enabled, you may have to insert the first row by clicking “Insert Row”.



Skip the When, Adapter Settings, and Permissions tabs and save your service.

2. In the IS Administrator console, select Adapters → JDBC Adapter → Polling Notifications link. You should see your new notification service listed.

The screenshot shows a table titled "JDBC Adapter Polling Notifications". It has columns for "Notification Name" (with a dropdown arrow), "Package Name" (with a dropdown arrow), "State" (with a dropdown arrow), and "Edit Schedule" and "View Schedule" buttons. The entry listed is "acme.PurchaseOrder.notifiers:orderCanonicalNotifier" under the package "Acme". The "State" dropdown is set to "Disabled".

3. Edit your notification schedule by clicking the Edit Schedule icon for your notification service. Specify the following parameters and then select Save Settings:
- Interval = 10
 - Overlap = *unchecked*
 - Immediate = *unchecked*

The screenshot shows a configuration dialog for the notification service. It has fields for "Interval: (seconds)" (set to 10), "Overlap" (unchecked), and "Immediate" (unchecked). Below this is a section titled "Cluster settings" with fields for "Coordination mode" (Standby), "Max process time: (seconds)" (180), and "Max setup time: (seconds)" (180). At the bottom is a "Save Settings" button.

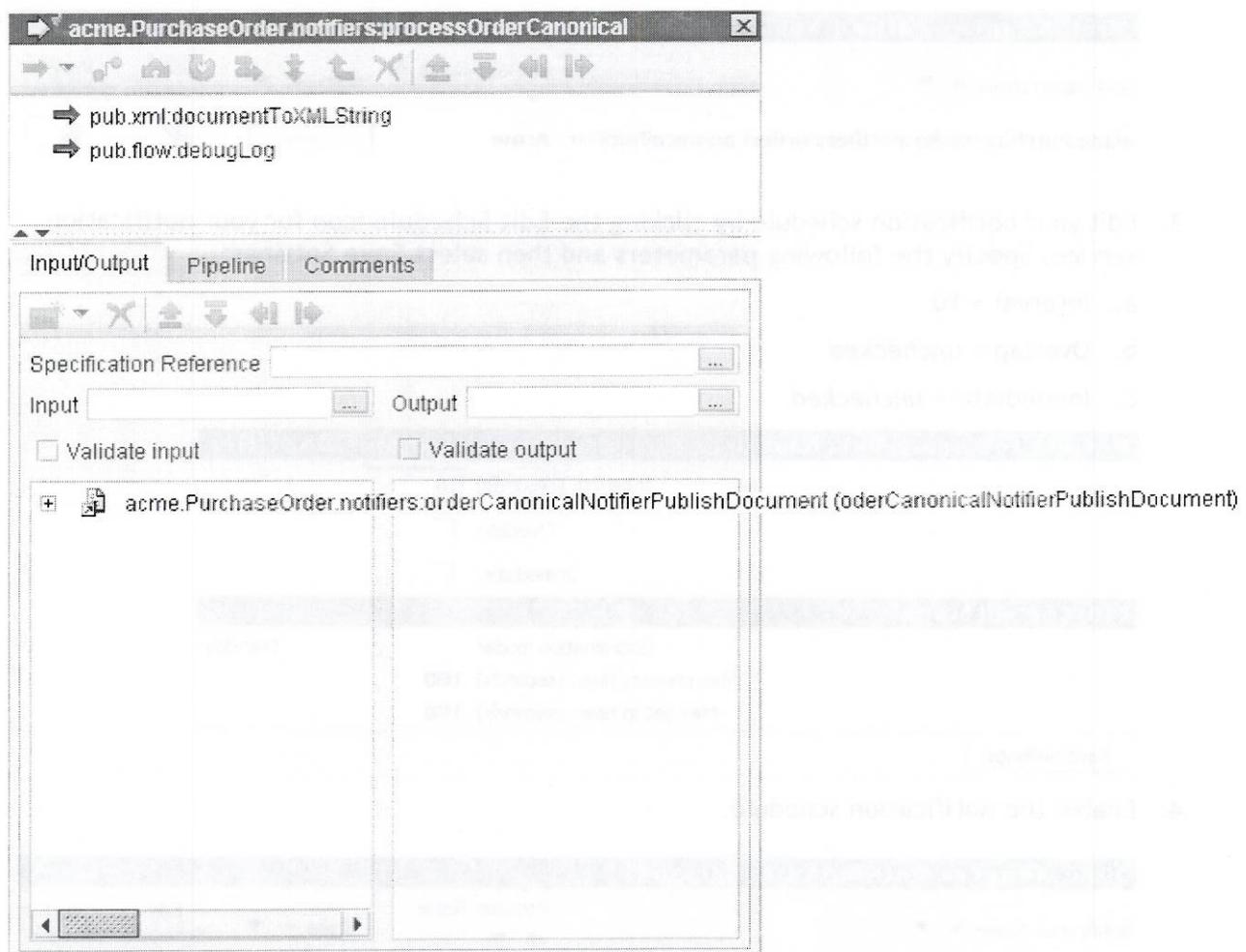
4. Enable the notification schedule.

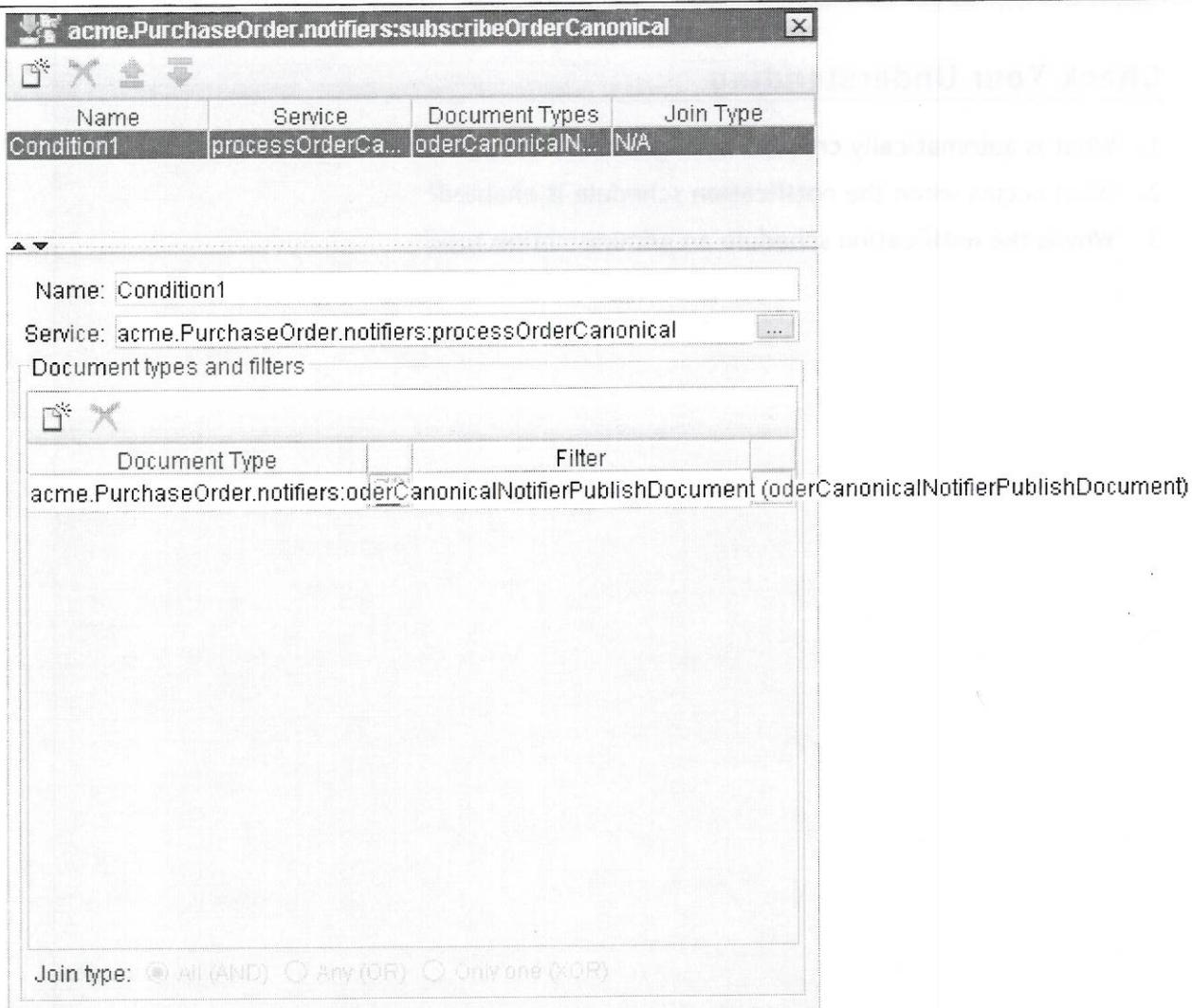
The screenshot shows the same table as before, but the "State" dropdown for the entry "acme.PurchaseOrder.notifiers:orderCanonicalNotifier" is now set to "Enabled".

5. As in “Exercise 16: Broker Pub/Sub” create a handling Service called “acme. Purchase Order.notifiers: processOrderCanonical” and a Trigger called “acme. PurchaseOrder. notifiers:subscribeOrderCanonical”. Equivalent to Exercise 16 the service receives a document reference to the “acme. PurchaseOrder. notifiers: orderCanonicalNotifier PublishDocument” document type. Reminder: Make sure the name of the Document in the handling service input uses the fully qualified name.

Use the `documentToXMLString` to convert the document to an XML representation and the `debugLog` service to display the XML document.

6. The trigger subscribes to the abovementioned notification document and invokes the “processOrderCanonical” service.





7. Test by running the `insertOrderCanonical` service from the previous exercise. You can load the same file as in the previous exercise. This is ...\\IntegrationServer\\packages\\AcmeSupport\\publ\\order_canonical_input.txt. You should see the an XML representation of your orderHeader document within the polling interval (10 Seconds) in the Server Log.

```
Command Prompt - tail -f c:\SoftwareAG\IntegrationServer\logs\server.log
2010-03-18 16:37:28 CET [ISPI.0090.0004C] + + + + -- <?xml version="1.0"?>
<ORDER_ID>123</ORDER_ID>
<TRANSACTION_ID>123</TRANSACTION_ID>
<ORDER_DATE>11/09/05</ORDER_DATE>
<TOTAL_COST>15</TOTAL_COST>
<IS_VALID>true</IS_VALID>
<SENDER_ID>88-888-8888</SENDER_ID>
<RECEIVER_ID>11-111-1111</RECEIVER_ID>
<_env>
  <locale></locale>
  <activation>wm627ca9f20-32a4-11df-be68-a5e85b59f784</activation>
  <businessContext>wm6:27ca9f20-32a4-11df-be68-a5e85b59f784\snnull\snnull:wm627ca9f20-32a4-11df-be
68-a5e85b59f784:null:IS_61</businessContext>
  <uuid>46d3d5f0329811df8395cdac9c998f003_1268926646833</uuid>
  <trackId>46d3d5f0329811df8395cdac9c998f003_1268926646833</trackId>
  <age>0</age>
  <flags>16</flags>
  <enqueueTime>Thu Mar 18 16:37:26 CET 2010</enqueueTime>
  <recvTime>Thu Mar 18 16:37:26 CET 2010</recvTime>
  <pubId>J_jRQtEo1DEurgAVIwwwADxZQs__DefaultClient</pubId>
</_env>
```

Check Your Understanding

1. What is automatically created and updated when you work with your notification service?
2. What occurs when the notification schedule is enabled?
3. Why is the notification schedule an administration task?

Exercise 20:

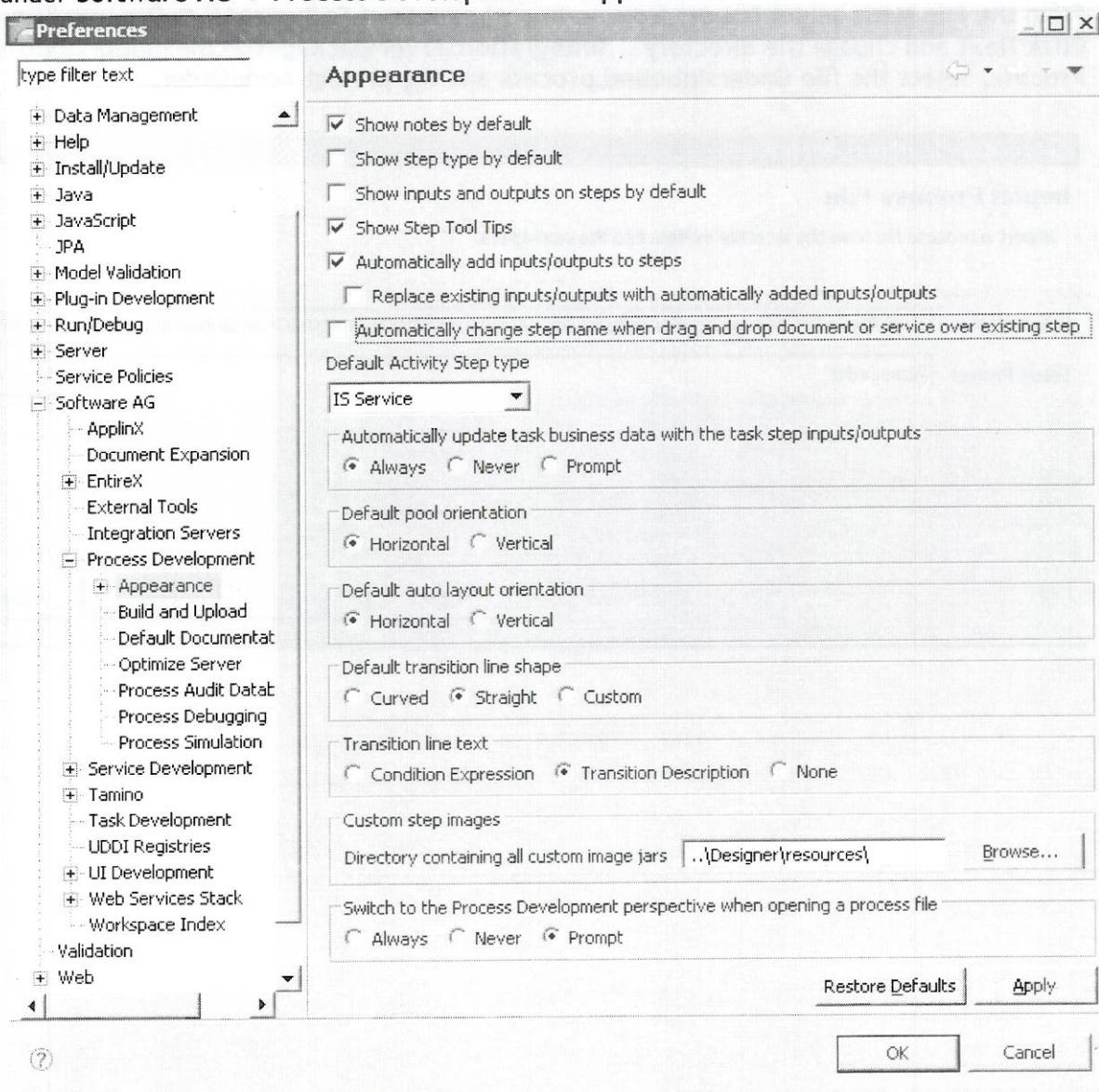
Use Services In a Business Process

Overview

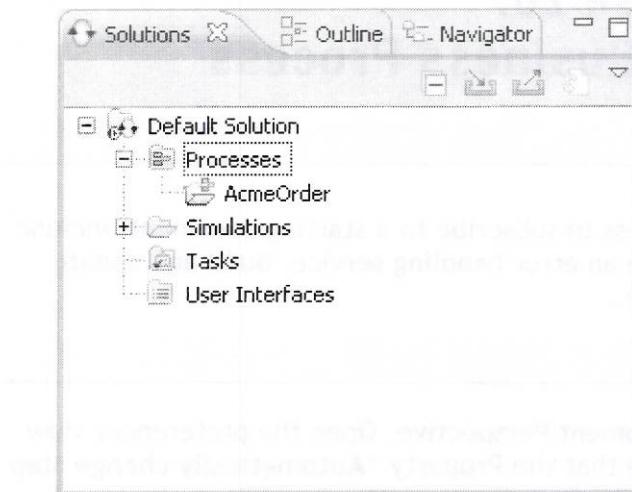
Use Designer to modify an existing business process to subscribe to a starting document, include services from previous exercises, and incorporate an error handling service. Build and update the model, and then test your process in Designer.

Steps

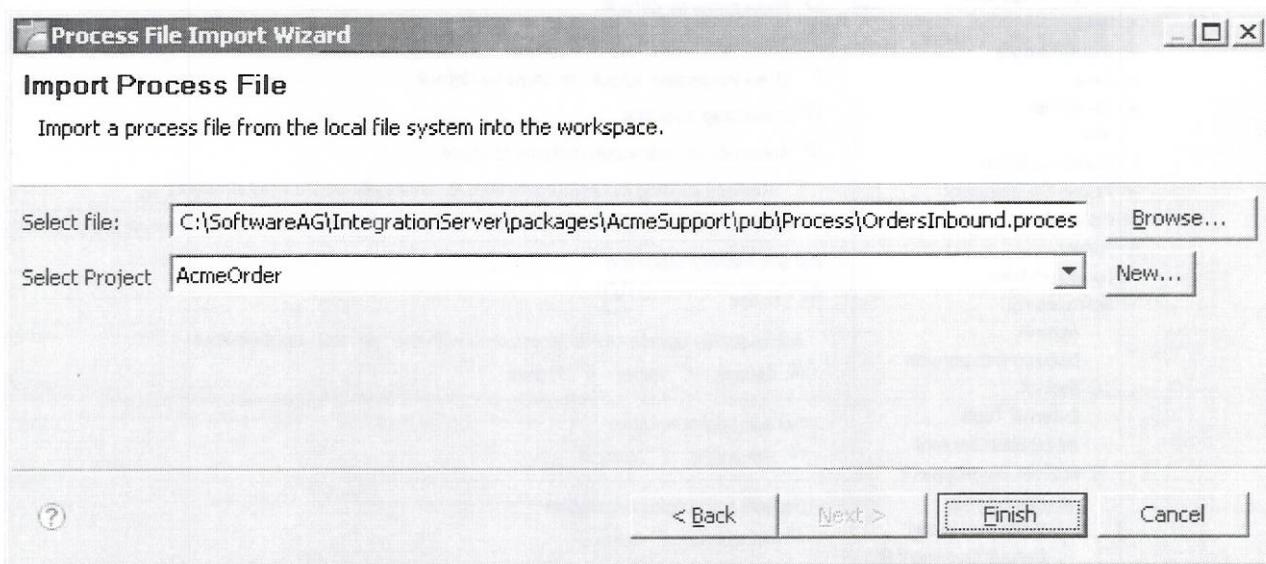
1. Open Designer and select the Process Development Perspective. Open the preferences view under “Window” ➔ “Preferences” and verify that the Property “Automatically change step name when drag and drop document or service over existing step” which can be found under Software AG ➔ Process Development ➔ Appearance is turned off.



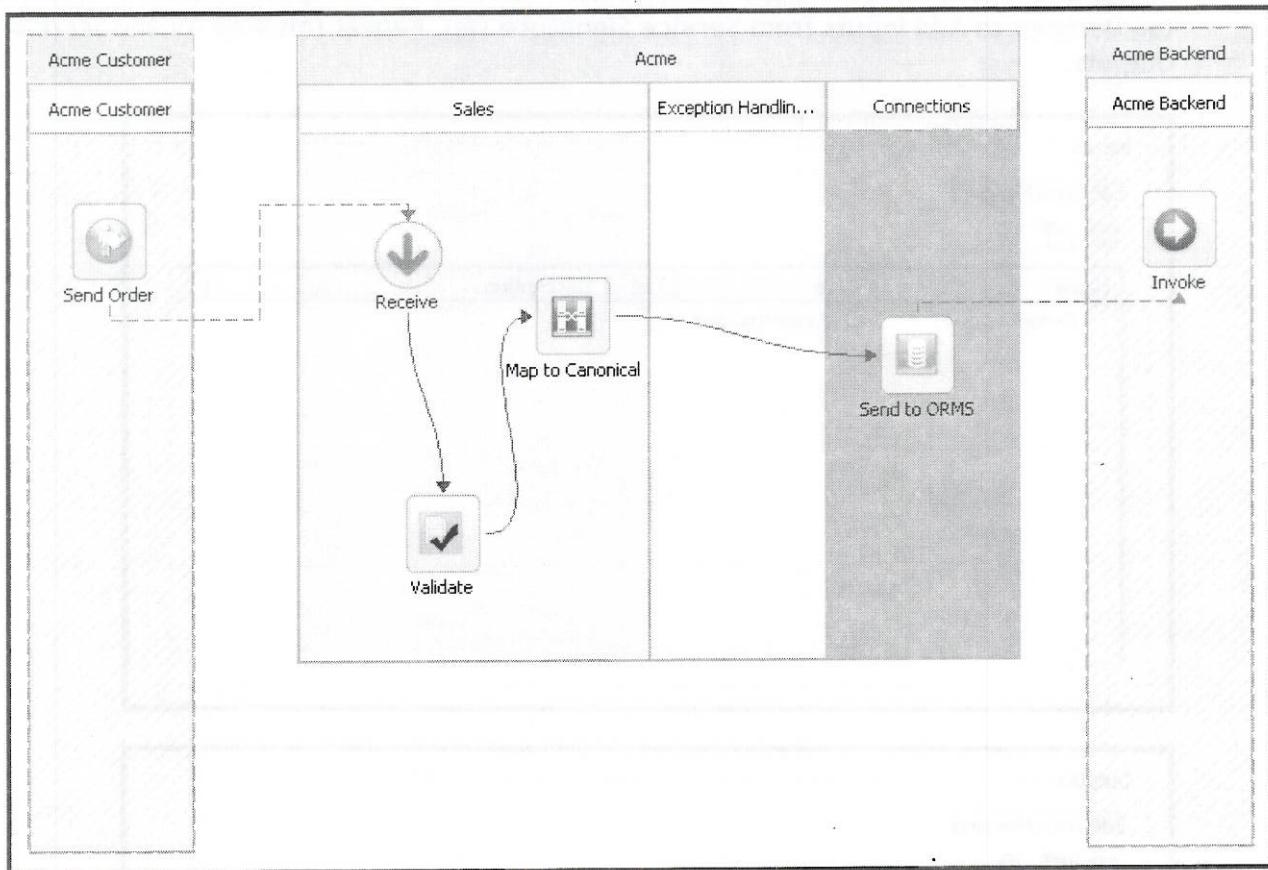
2. Using the File menu **create** a Process Project called AcmeOrder.



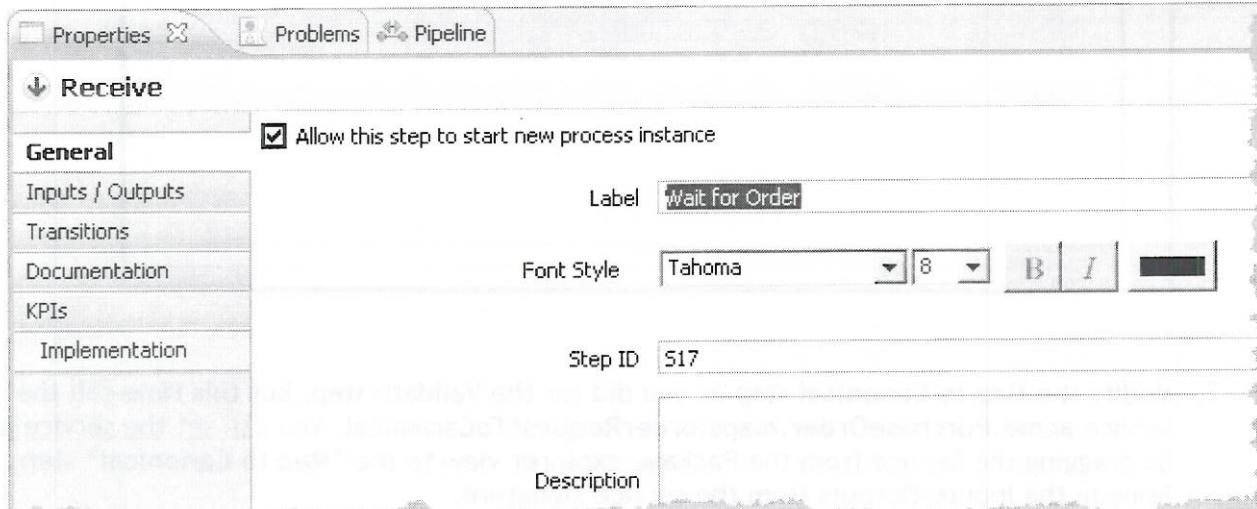
3. From the File Menu select Import from an import source of SoftwareAG ⇒ Process File. Click Next and choose the directory ...\\IntegrationServer\\packages\\AcmeSupport\\pub\\Process. Select the file OrdersInbound.process and the project AcmeOrder.



4. Examine the flow of the process so that you understand what it is intended to do.

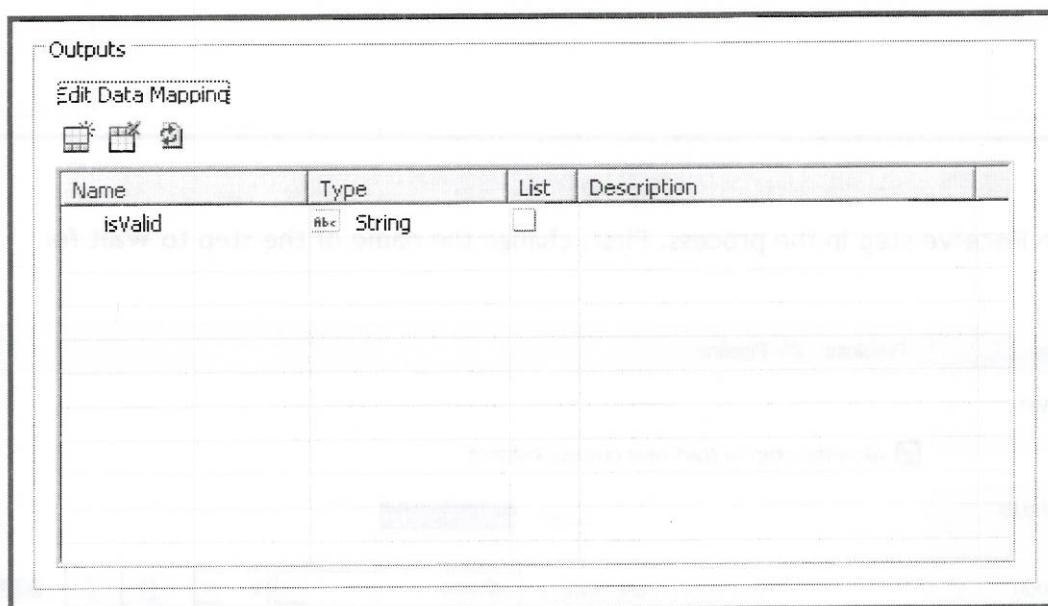
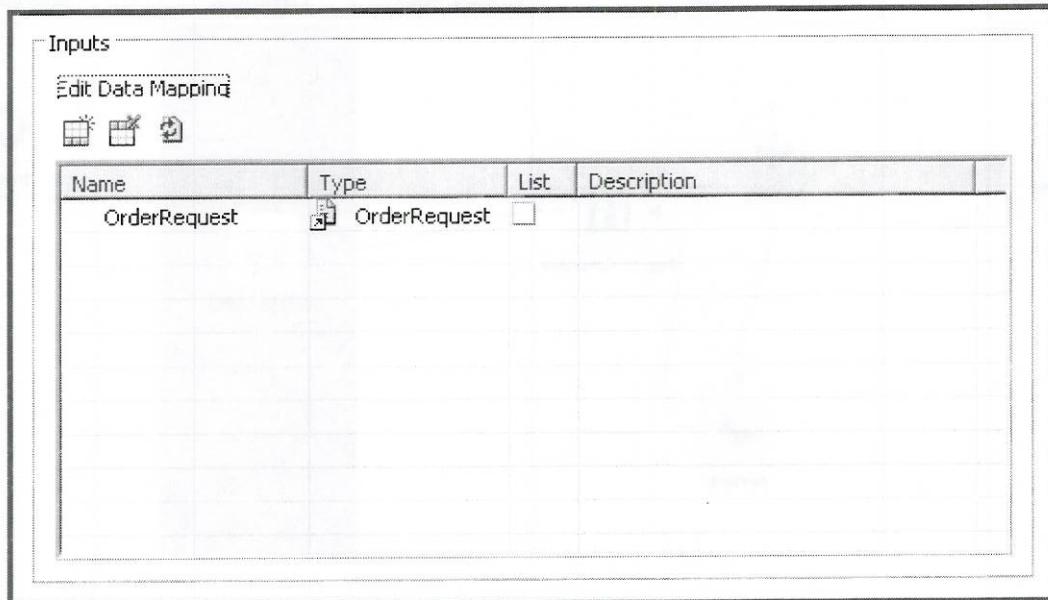


5. Modify the Receive step in the process. First, change the name of the step to Wait for Order.



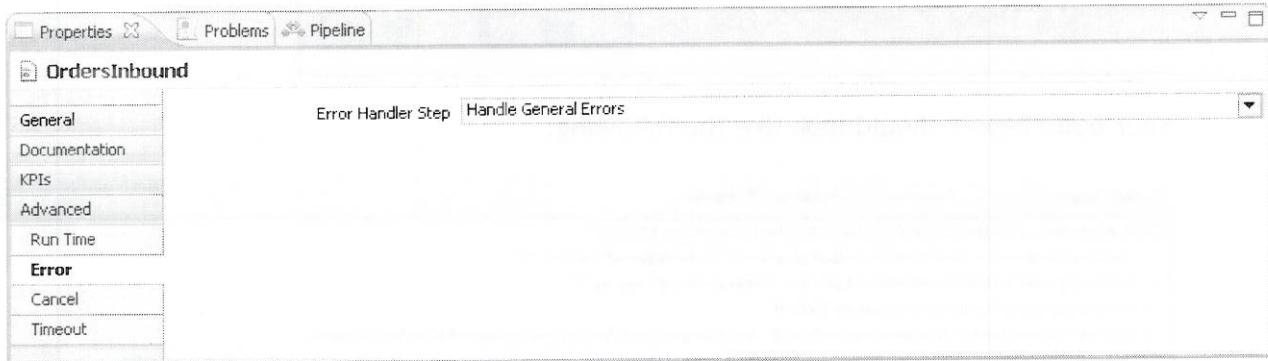
In the **Implementations** tab, set the **Receive Document** field to `acme.PurchaseOrder`.
docs. **request:** `OrderRequest`.

6. Modify the Validate step Implementation properties to call the service `acme.PurchaseOrder.utils:inspectLineItems`. Then modify the Inputs/Outputs property and tell Designer to Add Inputs from Service Signature (2). Repeat this step for the Service Outputs.

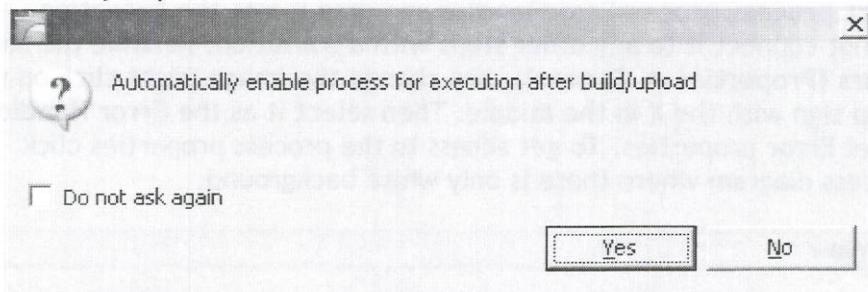


7. Modify the Map to Canonical step as you did for the Validate step, but this time call the service `acme.PurchaseOrder.maps:orderRequestToCanonical`. You can set the service also by dragging the Service from the Package explorer view to the “Map to Canonical” step. Bring in the Inputs/Outputs from the service signature.
8. Modify the Send to ORMS step as you did for the two previous steps, but this time use the service `acme.PurchaseOrder.utils:insertOrderCanonical`. Remember to bring in the Inputs/Outputs from the service signature.

9. Now use the drag and drop development feature to add a error handler step for the process. Locate the Package Navigator view in Designer and expand the AcmeSupport package. Find the service acmeSupport.process:processErrorHandler and drag it into the Exception Handling swimlane. Do not connect it to any other steps with a transition. Rename the step to **Handle General Errors** (**Properties** \Rightarrow **General**), and change the image (right click on the process step) to the stop sign with the X in the middle. Then select it as the **Error Handler Step** on the process level **Error** properties. To get access to the process properties click somewhere in your process diagram where there is only white background.



10. Save the process. Then build and upload the process by clicking on the  Icon. Check for errors in the Build Report, and make corrections as necessary. If asked whether you want to enable your process for execution, choose Yes.



Your build report should look like the following:

```
Build Report Properties Problems Pipeline
Build of process "OrdersInbound" started at Fri Mar 19 16:10:52 CET 2010
  Checking database for prior build and upload of process: "OrdersInbound" version: "1"
  Retrieving previous build information for process: "OrdersInbound" version: "1"
  Testing connection for Integration Server "Default"
  Process "OrdersInbound" has been previously built. Any changes to the flows in this version will be reflected below.
  Deleting flow from the Integration Server where it was last generated. AcmeOrder.OrdersInbound.OrdersInbound_1.Default:Wait_for_Order
  Regenerating flow service for Step "Validate" (ID: 524) Step ID: "524"
  Generating flow services for "Validate" ... Step ID: "524"
  Flow service "AcmeOrder.OrdersInbound.OrdersInbound_1.Default:Validate" generated Step ID: "524" Step Name: "Validate"
  Regenerating flow service for Step "Send to ORMS" (ID: 534) Step ID: "534"
  Generating flow services for "Send to ORMS" ... Step ID: "534"
  Flow service "AcmeOrder.OrdersInbound.OrdersInbound_1.Default:Send_to_ORMS" generated Step ID: "534" Step Name: "Send to ORMS"
  Regenerating flow service for Step "Handle General Errors" (ID: 573) Step ID: "573"
  Generating flow services for "Handle General Errors" ... Step ID: "573"
  Flow service "AcmeOrder.OrdersInbound.OrdersInbound_1.Default:Handle_General_Errors" generated Step ID: "573" Step Name: "Handle General Errors"
  Regenerating flow service for Step "Wait for Order" (ID: 548) Step ID: "548"
  Generating flow services for "Wait for Order" ... Step ID: "548"
  Flow service "AcmeOrder.OrdersInbound.OrdersInbound_1.Default:Wait_for_Order" generated Step ID: "548" Step Name: "Wait for Order"
  Regenerating flow service for Step "Map to Canonical" (ID: 527) Step ID: "527"
  Generating flow services for "Map to Canonical" ... Step ID: "527"
  Flow service "AcmeOrder.OrdersInbound.OrdersInbound_1.Default:Map_to_Canonical" generated Step ID: "527" Step Name: "Map to Canonical"
  Generating triggers for "OrdersInbound" ...
  Generating transition trigger for process "OrdersInbound" Integration Server "Default"
  Generating subscription trigger for process "OrdersInbound" Integration Server "Default"
  Generating process engine fragments for process "OrdersInbound"
  Generating process engine fragment for Integration Server "Default"
  Saving process engine fragment for Integration Server "Default"
  Saving build information for process: "OrdersInbound" version: "1" to database...
  Checking Quality of Service settings for previous generation of process: "OrdersInbound".
Build of process "OrdersInbound" completed successfully at Fri Mar 19 16:12:58 CET 2010
Upload for execution of process "OrdersInbound" completed successfully.
Restoring Quality of Service settings from previous values for process: "OrdersInbound".
```

11. Open the Service Development Perspective, and review the generated package **AcmeOrder** and generated services. Make sure the mapping for **Handle General Errors** is correct and modify if necessary.

Note: you may have to refresh the Package Navigator view to see the generated code. To do this right-click on Default in the Package Navigator and select Refresh (this will reload everything under the Integration Server packages folder)

12. Switch back to the Process Development Perspective and debug the process by clicking the debug button (). When prompted for inputs to the OrderRequest document, load the file ...\\IntegrationServer\\packages\\AcmeSupport\\pub\\publish_order_request_input.txt.

Continue through the process in debug mode, using the Step Into, Step Over, or Step Out buttons in the Trace window.

Check Your Understanding

1. Why do you create swimlanes within a process? What effect do they have on the execution of the process?
2. What occurs throughout the system when you perform a build and upload from Designer?
3. What are the different start mechanisms for a process?

2007. Diese eingeschränkt lizenzierte und verdeckt abgespeckte Software ist nur für den Betrieb von privaten und kleinen Büros sowie für kleinere Betriebsstätten bestimmt und darf nicht kommerziell genutzt werden. Sie ist weiterhin ausdrücklich untersagt, sie zu kopieren, zu verbreiten oder zu verändern.

Die Software ist lizenziert, um auf einer einzigen Station mit einem Benutzer gleichzeitig eine einzige Anwendung auszuführen. Es darf jedoch nicht mehr als eine Anwendung auf einer Station ausgeführt werden.

Die Software ist lizenziert, um auf einer einzigen Station mit einem Benutzer gleichzeitig eine einzige Anwendung auszuführen. Es darf jedoch nicht mehr als eine Anwendung auf einer Station ausgeführt werden.

Die Software ist lizenziert, um auf einer einzigen Station mit einem Benutzer gleichzeitig eine einzige Anwendung auszuführen. Es darf jedoch nicht mehr als eine Anwendung auf einer Station ausgeführt werden.

Die Software ist lizenziert, um auf einer einzigen Station mit einem Benutzer gleichzeitig eine einzige Anwendung auszuführen. Es darf jedoch nicht mehr als eine Anwendung auf einer Station ausgeführt werden.

Die Software ist lizenziert, um auf einer einzigen Station mit einem Benutzer gleichzeitig eine einzige Anwendung auszuführen. Es darf jedoch nicht mehr als eine Anwendung auf einer Station ausgeführt werden.

Die Software ist lizenziert, um auf einer einzigen Station mit einem Benutzer gleichzeitig eine einzige Anwendung auszuführen. Es darf jedoch nicht mehr als eine Anwendung auf einer Station ausgeführt werden.

This page intentionally left blank

Check Your Understanding: Answers to the Questions

Exercise 1: Start the Integration Server, Broker, and MWS

1. What is the URL to access the Integration Server? <http://localhost:5555/>
2. What is the URL for MWS? <http://localhost:8585/>
3. Why is the Broker Monitor set to Automatic start, but not the Broker Server? The Broker Monitor is responsible for starting and monitoring the Broker Server. If the Broker Server terminates for whatever reason, broker monitor tries to restart it again.

Exercise 2: Packages and Folders

1. If you place the folders in the wrong parent folder, how could you correct it? You can drag and drop folders with the mouse.
2. Why is a consistent folder structure in all packages important? This makes it easier for new people on a project to understand the inner workings of a package.

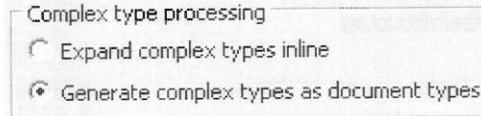
Exercise 3: Create a Service

1. Why is the order of the services important? The order of service invocation determines the runtime behaviour of your own service.
2. How many inputs can the pub.string:toUpperCase service accept? It accepts 4 Inputs. All inputs marked with a small square like language, country and variant are optional and have a default value that usually does “the right thing”™. For the default values being used and the possible values that can be set, consult the services documentation in ... _documentation\Developer\8-0-SP1_Integration_Server_Built-In_Services_Reference.pdf
3. Where would the server log appear if the server is not running through the Command Prompt? Either in the WEBMDB database or in the server.log file, depending on how logging is set up.

Exercise 4: Document Types

1. Why are additional documents created in the acme.PurchaseOrder.docs.request folder when the OrderRequest schema is imported? Because you explicitly said so when importing the

schema by leaving the checkbox



on its default value

“Generate ... as document types”. The generated Document type docType_PartnerRoleDescription is reused in several places inside the OrderRequest document, making maintenance of it much easier.

2. What is the benefit of using a schema for import over using a DTD? Both DTD and XML schemas allow the same functionality. So there is not much of a difference. But DTD's are written in their own language, while schemas are written using XML syntax and can be processed with the standard XML toolset. Furthermore DTD's are more oriented towards text based documents (books) while schemas are oriented towards data based documents (orders, invoices).

3. What are the two ways to indent a document type element under another? You can use drag and drop with the mouse or the arrow buttonw in the top toolbar

Exercise 5: Flow Services - BRANCH

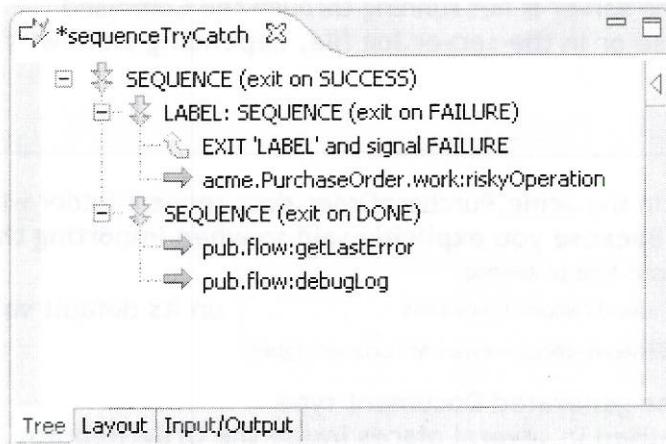
4. When are regular expressions useful in branch? When you want to perform an operation on a set of similar input values.
5. Can you combine a switch variable with Evaluate labels=True? No, this wouldn't make sense.
6. What are the special test values that can be used as labels in a branch statement? Those values are “\$default”, which is used when none of the other input values matches and “\$null”, which is used when the switch variable is not specified. The usage of “\$null” makes no sense when the “Evaluate labels” option is set to true.

Exercise 6: Building Flow Services - LOOP

1. What would happen if the MAP step is not indented under the Loop? There MAP step would be executed only once. The value of \$iteration outside the loop is not defined.
2. How many employees could you have added? Does Loop have a limit? There is no limit set by the LOOP statement itself. It can handle arbitrary large arrays.
3. Why do you want to use document references rather than creating the document in the service input? Sending in the document as argument to the service allows for more flexibility when using the service.

Exercise 7: Building Flow Services – SEQUENCE

1. Rather than using a service you know will fail, how can you throw an Exception in Flow? You can use an EXIT ‘LABEL’ statement with signal set to FAILURE like in the screenshots below:



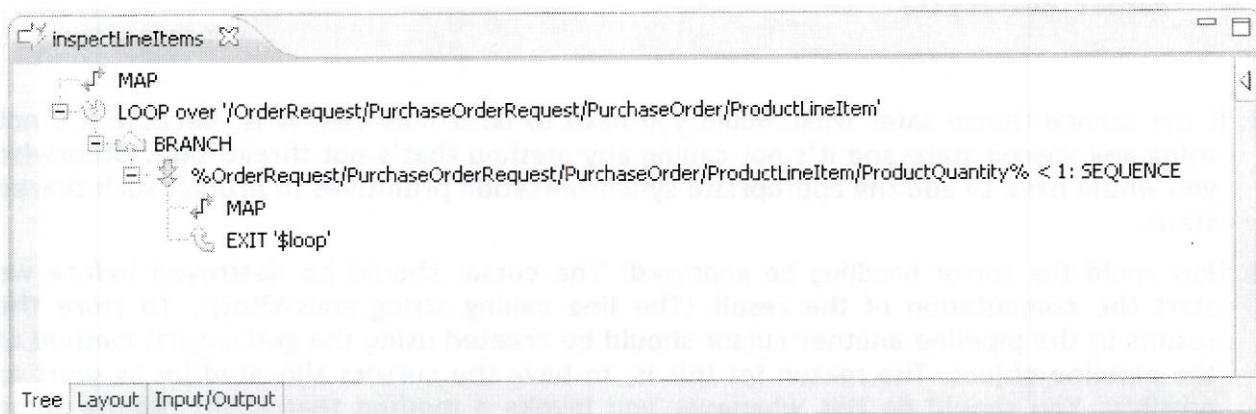
Property	Value
General	
Comments	
Label	LABEL
Exit from	FAILURE
Signal	
Failure message	

2. What happens if the riskyOperation service works (doesn't fail) ? Only the 1st and the 2nd SEQUENCE statements get executed. The content of the file is read into memory and is returned to the caller.

Exercise 8: Validation Service

1. Why did we set isValid to true at the very beginning? isValid is an Output variable. As such, it should always been initialized.
2. Is there another way we could have validated this particular value with writing Flow or Java? We could have written a Java service, but dealing with such nested structures in Java produces deeply nested code that is hard to maintain. When using Flow, we could have used other loop (eg REPEAT) or mapping (eg using Indexes) functionality.

Extra credit solution:



Exercise 9: Mapping Service

1. How is a transformer different from a normal service? It requires explicit assignment of its mappings. It does not do implicit mapping. Transformers do not receive a full copy of the pipeline and all transformers in a MAP step can execute in parallel.
2. What if the transformer you want to use is not in the transformer drop-down list? You can use any service as transformer by selecting the browse (Browse...) button at the bottom of the transformer dropdown.
3. Why did we need to LOOP over ProductLineItems? Why not just map from ProductLineItems to Items? Because we needed to apply transformers to some of the source values. The structure of the two documents is different as well.

Exercise 10: Create a Java Service

- What exactly is each line of the Java code doing in the endsWith service? See the inline comments below:

```

// get a cursor to access the pipeline

IDataCursor cursor = pipeline.getCursor();

// retrieve the two input values from the pipeline. NOTE: there is
// no test that these values are actually present!

String string = IDataUtil.getString(cursor, "string");
String suffix = IDataUtil.getString(cursor, "suffix");

// compute the value to be returned

String value = string.endsWith(suffix) ? "true" : "false";

// store the return value in the pipeline

IDataUtil.put(cursor, "value", value);

// destroy (release) the cursor, as we do not need it any longer.

cursor.destroy();

```

- Is the service thread safe? What would you have to do if not? Yes, it is. Because it's not using any shared state and it's not calling any method that's not thread safe. Otherwise you would have to add the appropriate synchronization primitives to protect such shared state.
- How could the cursor handling be improved? The cursor should be destroyed before we start the computation of the result (The line calling `string.endsWith()`). To store the results in the pipeline another cursor should be created using the `getCursor()` method of the pipeline object. The reason for this is, to have the cursors allocated for as short as possible. You should do this whenever you invoke a method that might require some time to compute its result. In the present example the overhead caused by cursor management outweighs the benefits of a shorter cursor lifetime.

Exercise 11: Monitoring Services

- Why is it necessary to create remote server aliases? By design, MWS will communicate only with one instance of Integration server. When resubmitting a service invocation, MWS tells this Integration server where it wants the service instance to be scheduled for execution. To do so, MWS is sending the name of the remote server alias that should be used to resolve the final execution server.
- Under what circumstances would it be acceptable to resubmit a service? Why? Those circumstances depend only on the service execution to be resubmitted. If a failed service had already executed half of its statements, then those statements may have caused some state changes where it may not be viable to do those changes again

(Imagine a service giving a 3% raise to all employees, that failed after processing the first 100 employees).

Exercise 12: Invoking Services

1. Why and when would you use an HTTP URL alias for your services? You can use an HTTP URL alias if the name of the service, to be called by your clients might change from time to time. It also allows invoking services using a much shorter URL.
2. How do the services find their input data? They all depend on the presence of the node object in the input pipeline. This is a parsed representation of the XML document that was sent to integration server.
3. How do the services return their result? They return their result by XML encoding the content of the output pipeline.

Exercise 13: Create a Flat File Schema

1. Why can't flat files be imported like XML documents? Because a flat file contains no metadata like field names. Also it would be pure guesswork to find the correct delimiter characters.
2. What is the meaning of Nth field? Nth field is the name of an extractor, that returns a part of the data stored in a record, which is delimited by special delimiter characters.

Exercise 14: Create a Flat File Dictionary

1. What is the difference between a dictionary and a schema? A schema describes the records that are contained in a single flat file. A dictionary serves as a repository of record and composite definitions, which can be used across multiple schemas.
2. Why should you create the IS document type when the schema is complete? At this time all information is available to create the IS document type.

Exercise 15: Web Service Descriptors and Custom Faults

1. When would you create a Provider WSD when a Consumer WSD? You create a provider when you (Integration Server) provides a WEB service. You create a consumer WSD when you want to consume external WEB services.
2. How and when are WSC's created? They are implicitly created when you create a WEB service consumer or provider.
3. Can you have more than one custom SOAP Fault Document? Yes. All you have to do is the addition of more error document types to the Response/Fault document list of the required operations.

Exercise 16: Broker Pub/Sub

1. What happens when a document is made publishable? The document is modified to contain a new document reference at the top level called _env. This envelope document contains data that is used internally by the broker to process the document. The second thing that happens is the publication of the new document type to the broker.

