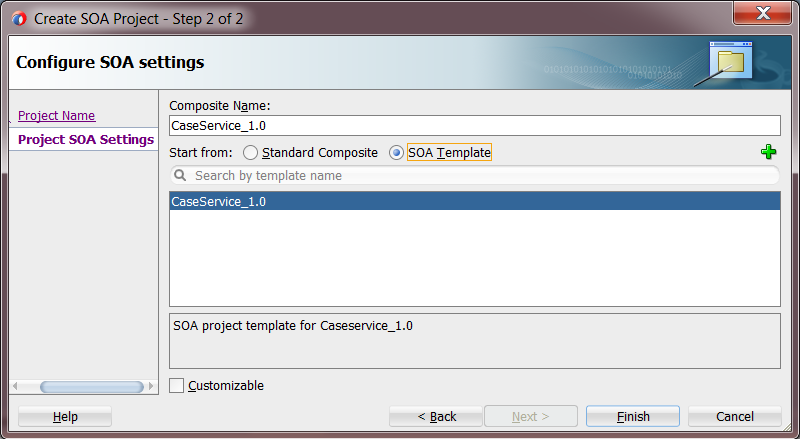
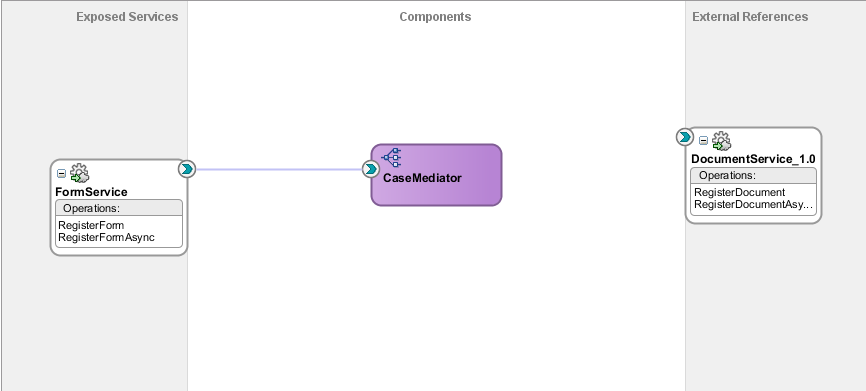
Storyboard X-masterclass 12c

# 1. Kick-start a SOA project using templates

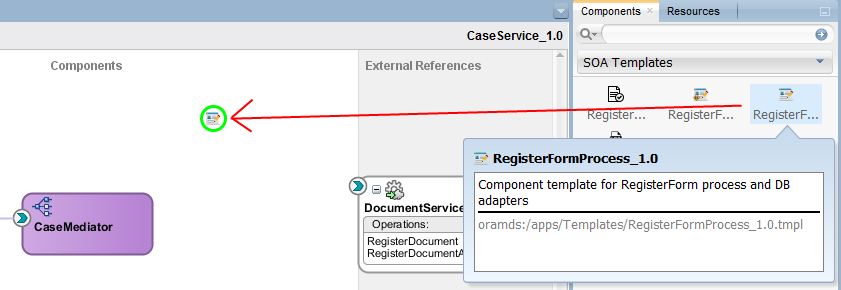
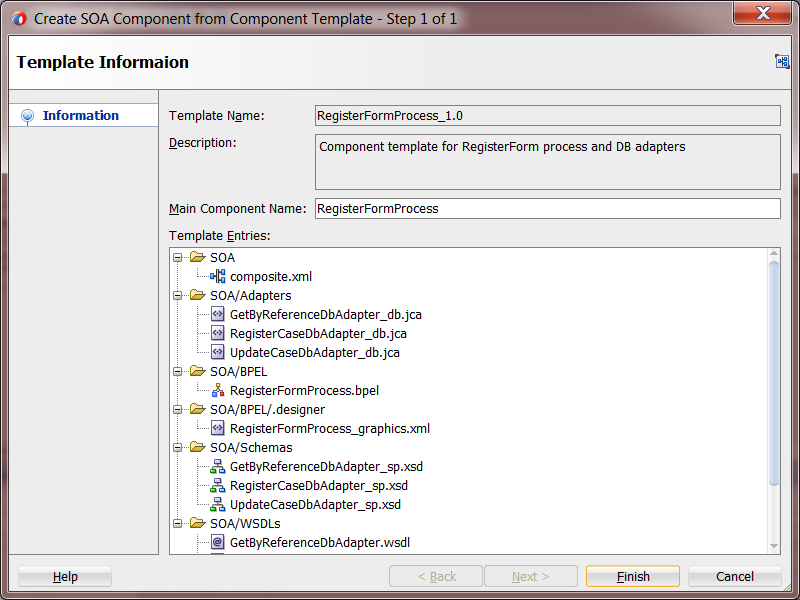
Create the CaseService composite using SOA templates (result is CaseServiceBasedOnTemplates).

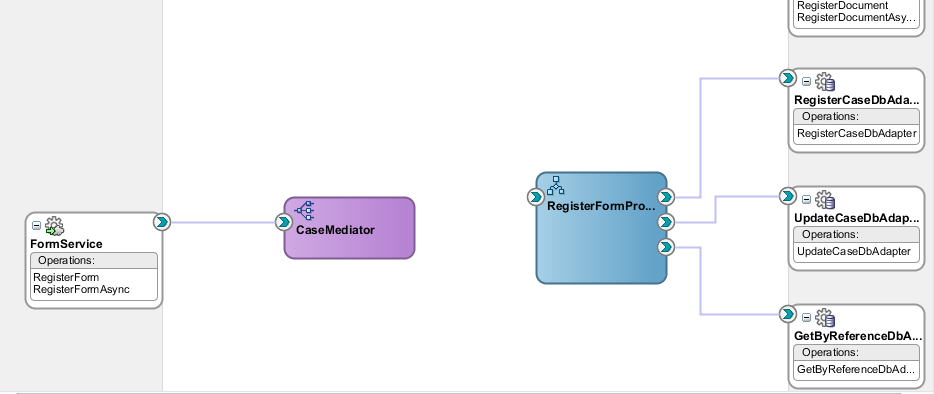
1. Create a new SOA project (CaseService\_1.0) using the corresponding project template. This template includes a SOAP Service Binding with WSDL from the MDS, a Mediator, Schematron validations and SOAP Reference Binding to DocumentService.



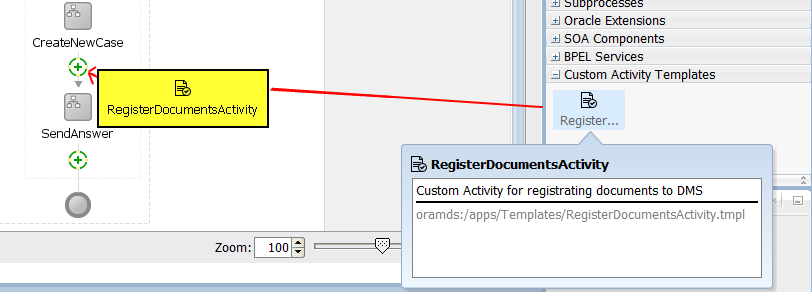


1. Create a new BPEL process (RegisterFormProcess) using the component corresponding template. Templates can found in the Component palette, under SOA templates, and can be dragged and dropped onto the composite editor. This template includes a BPEL process skeleton with Adapters and Fault Handling for the initial registration/creation of new cases.

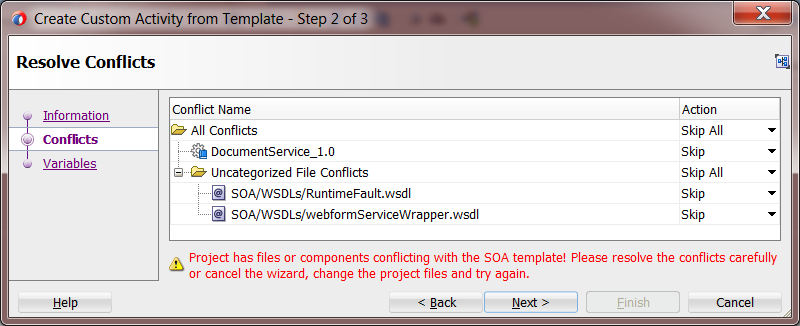
  




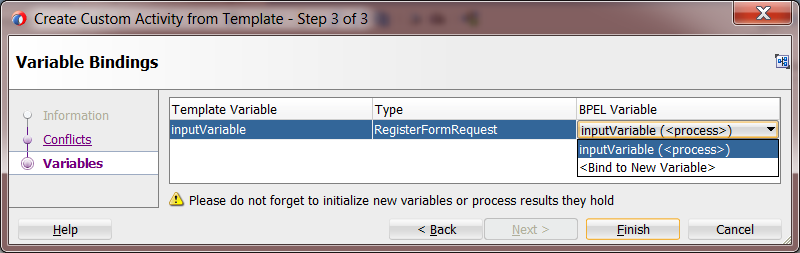
1. Create a new Scope (RegisterDocument) using custom activity template. Templates can found in the Component palette, under Custom Activity Templates, and can be dragged and dropped onto the BPEL editor. This template includes transformation(s), an Invoke of DocumentService and the needed Partner Link.

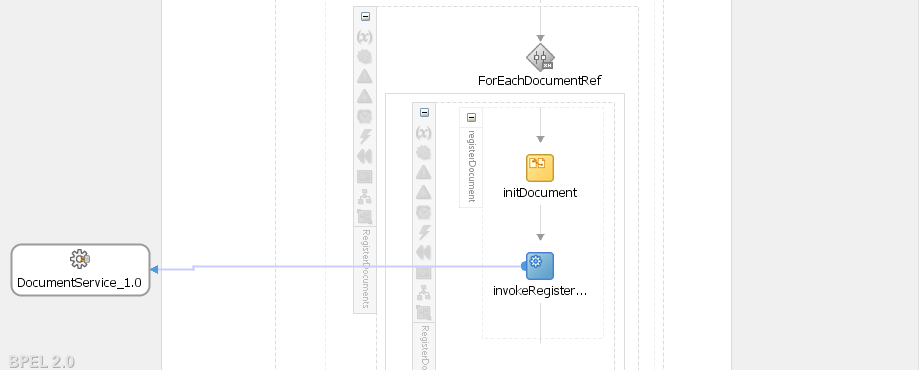


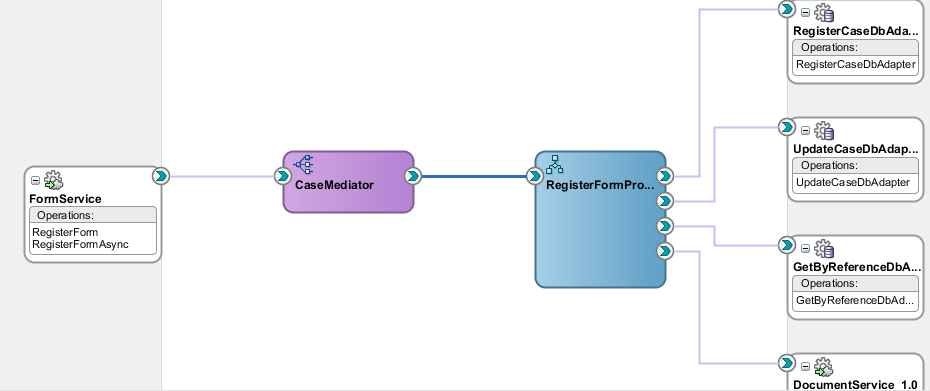
Skip or override conflicts were needed, but for now skip all conflicts.



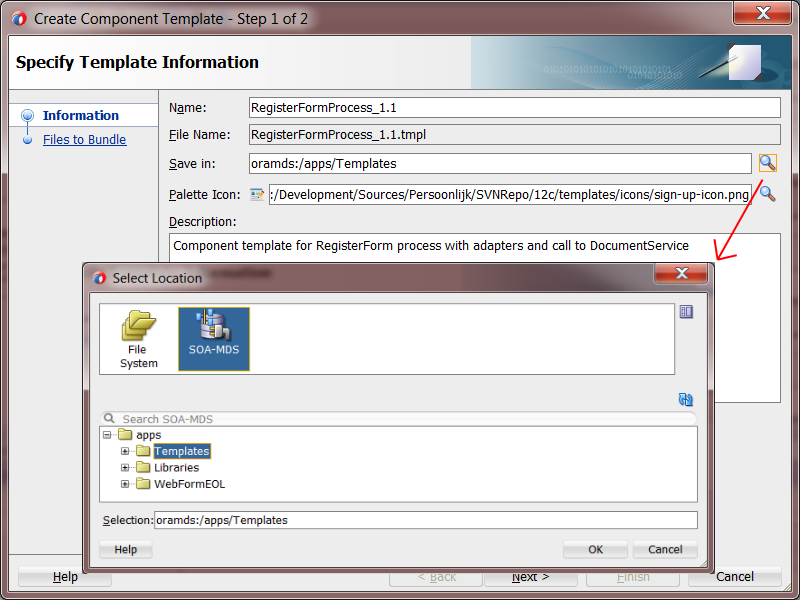
Assign the existing inputVariable to use in the transformations. You can also bind a new Variable, but then you need to initialize it yourself.



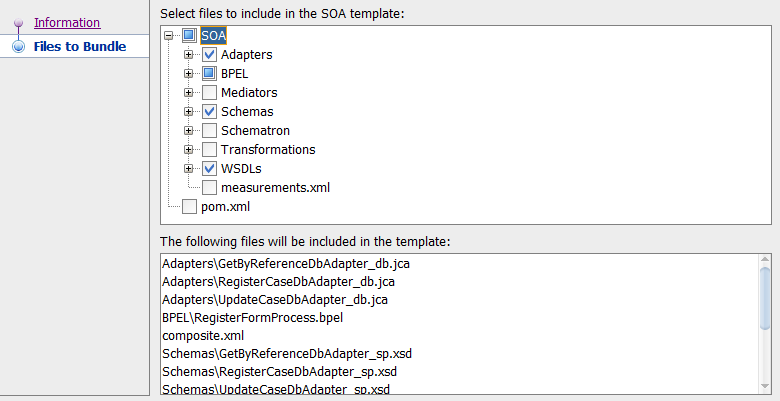




1. Create a new Component template based on the current RegisterForm. Right-click on the BPEL process in the composite overview and choose the Create Component Template option.



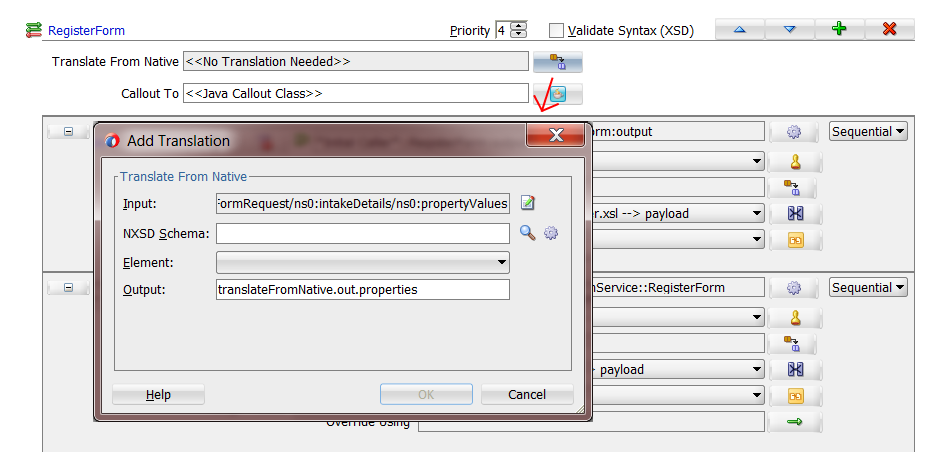
Choose which file to include in template. Depending files can’t be excluded / deselected.



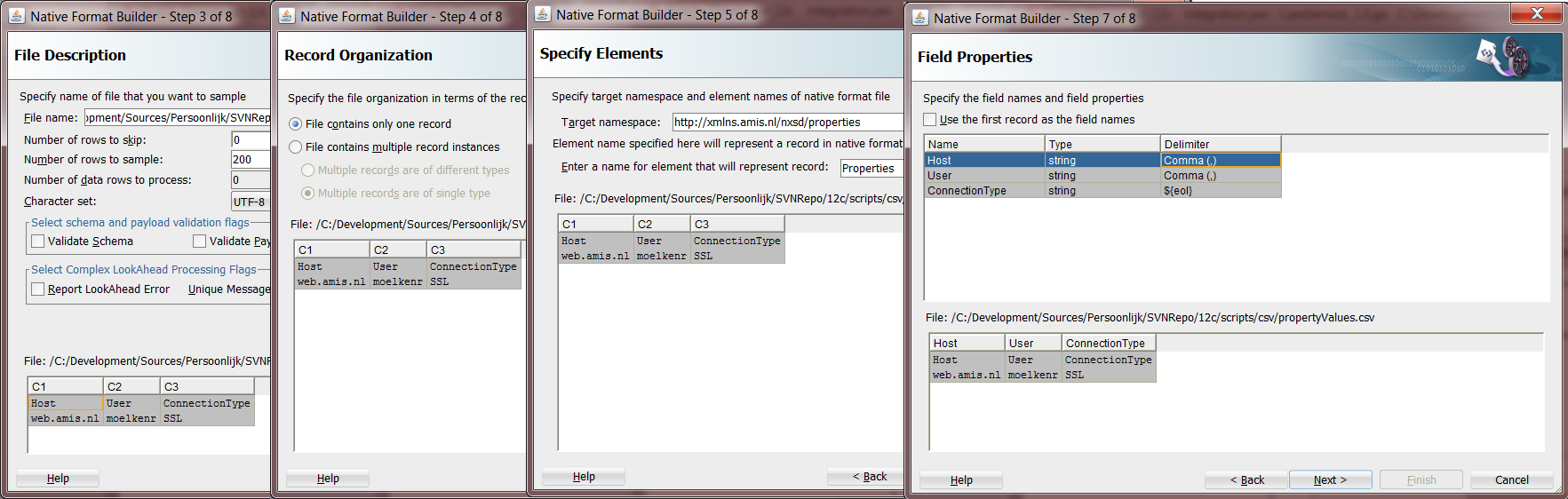
## 1.2. Updated Mediator Component

In the mediator an extra action Translate from Native is added. With this action you can transform a CSV/JSON/String element of the request to XML using a **NXSD transformation**. The transformed output can then be used in the routing rules of the Mediator. This action can be done on **operation** level, but also on a specific **routing rule** for that operation.

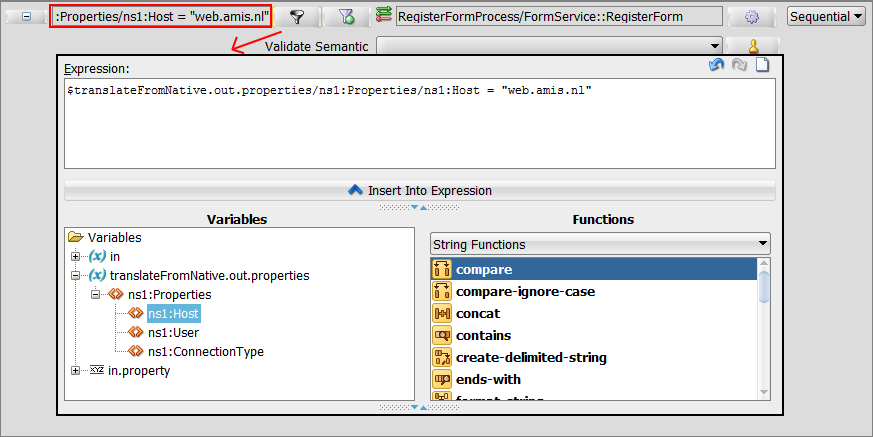
1. In the created project (CaseService\_1.0) open de Mediator (CaseMediator) component. For the RegisterForm operation create a new “Translate to Native” action.
2. Select the propertiyValues element from the input payload:  
   $in.payload/fmr:FormRequest/ns0:intakeDetails/ns0:propertyValues



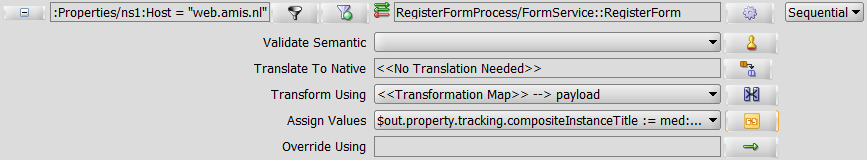
1. Create a new NXSD schema (propertyValues.xsd) based on the propertyValues.csv file (under scripts). De element consist of a **single-row** delimited values string, so it is not needed to skip any header rows.

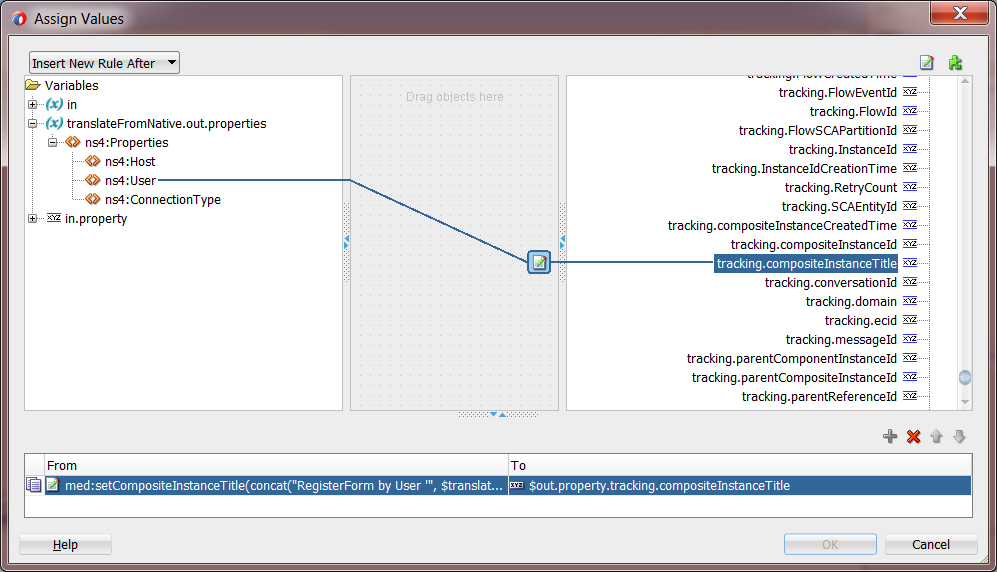


1. The newly created translation and output variable can be used within routing rules, transformations and assigns. Add a filter to the routing rule to filter on the Host element.

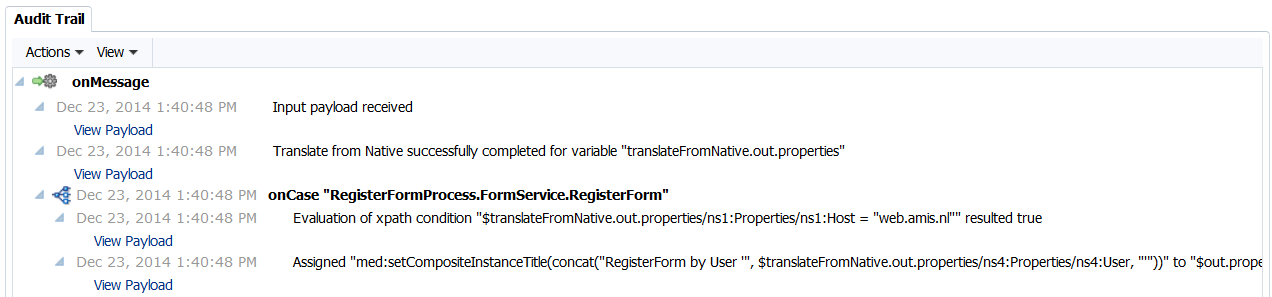


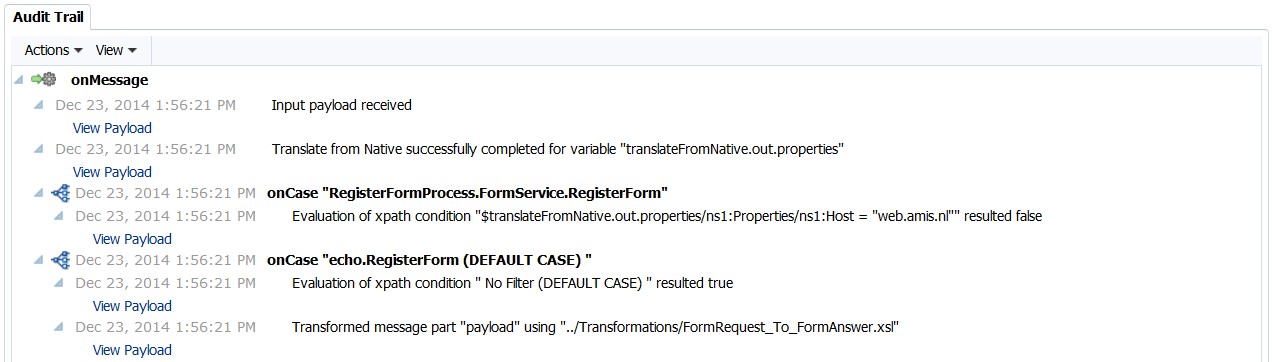
1. The assign of single elements has gotten a new editor much like the editor used in BPEL. Investigate this by adding a assign of the Composite Title.





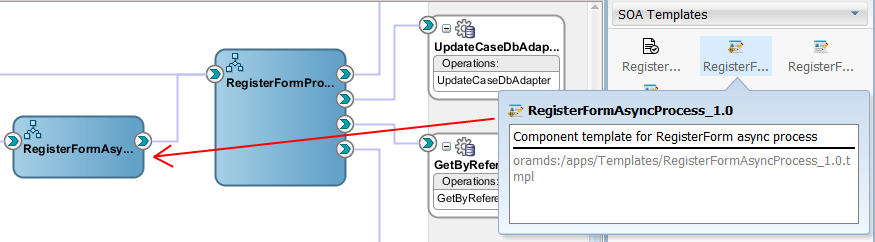
When a new instance is initiated the translation is executed. The values are used on runtime in the filter of the routing rule. If the value is correct the BPEL process is called else a reply is echoed.



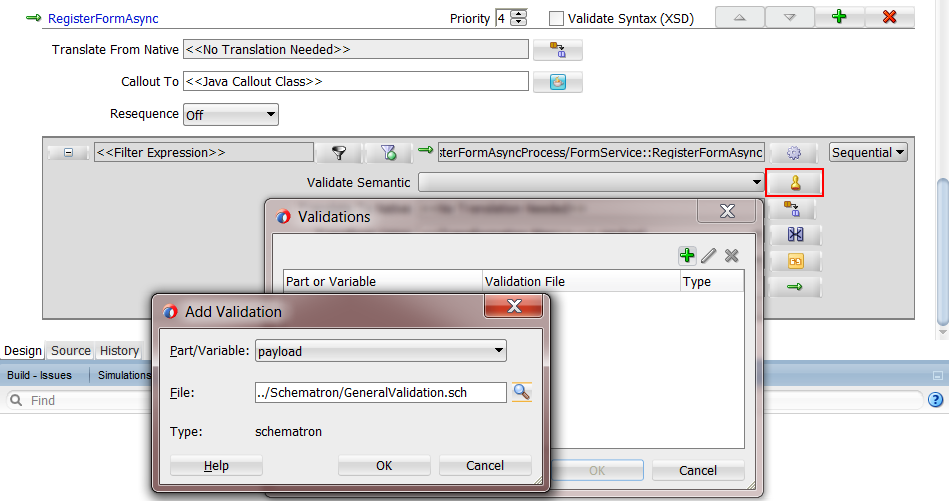


A second new action is Override Using. With this action you can override routing rule options using a DVM or Business Rules. Based on an element value of the request you can change the routing rule option dynamically. This action can only be added to async / one-way operations. With the key value you have the possibility to override the XSLT transformation file,  Filter expression, Execution Type, Syntax Validation (XSD), Semantic Validation (Schematron) file instead of creating a bunch of static routing rule.

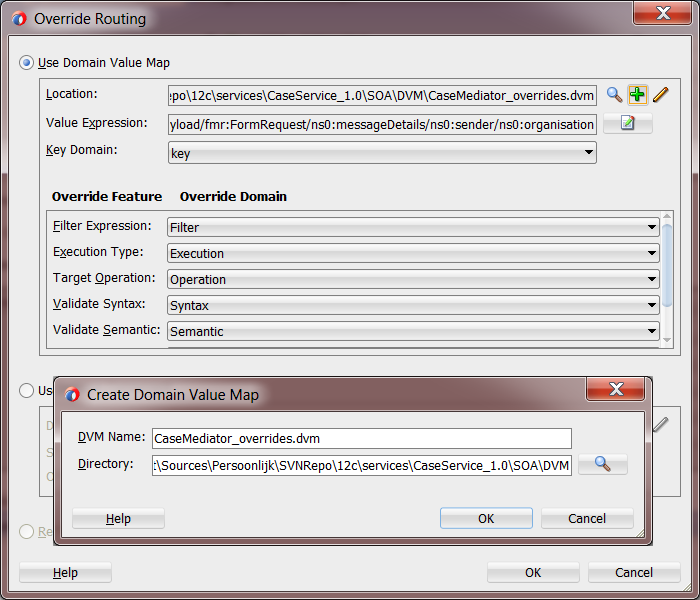
1. Prepare the composite by dragging and dropping the RegisterFormAsyncProcess component template and wire it to the CaseMediator and RegisterFormProcess component.



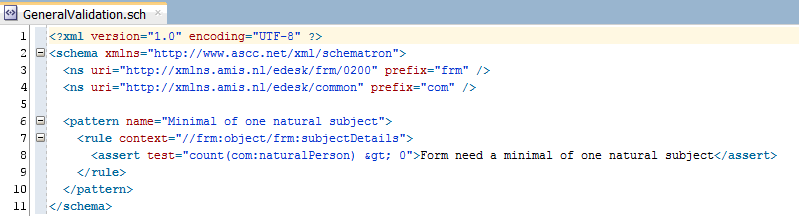
1. Add the GeneralValidation schematron schema to the routing rule. The schematron can be added under the Validate Schematic action.

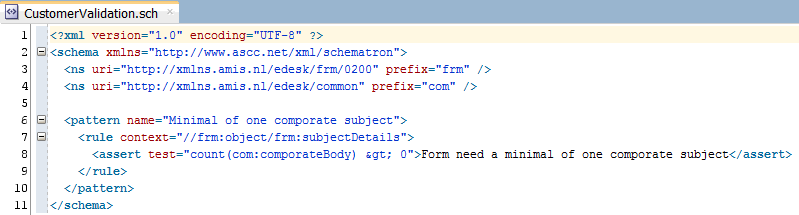


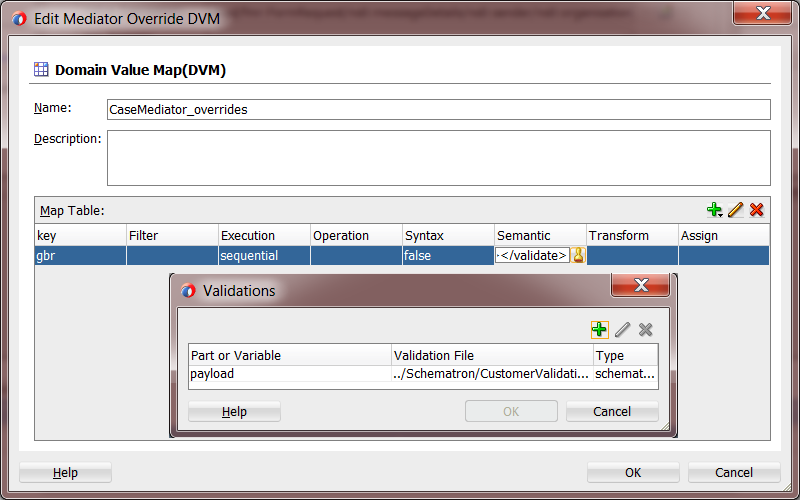
1. Create a new Override Using a DVM file and switch schematron validation for a specific sender/organisation value: xp20:lower-case ($in.payload/fmr:FormRequest/ns0:messageDetails/ns0:sender/ns0:organisation)

`

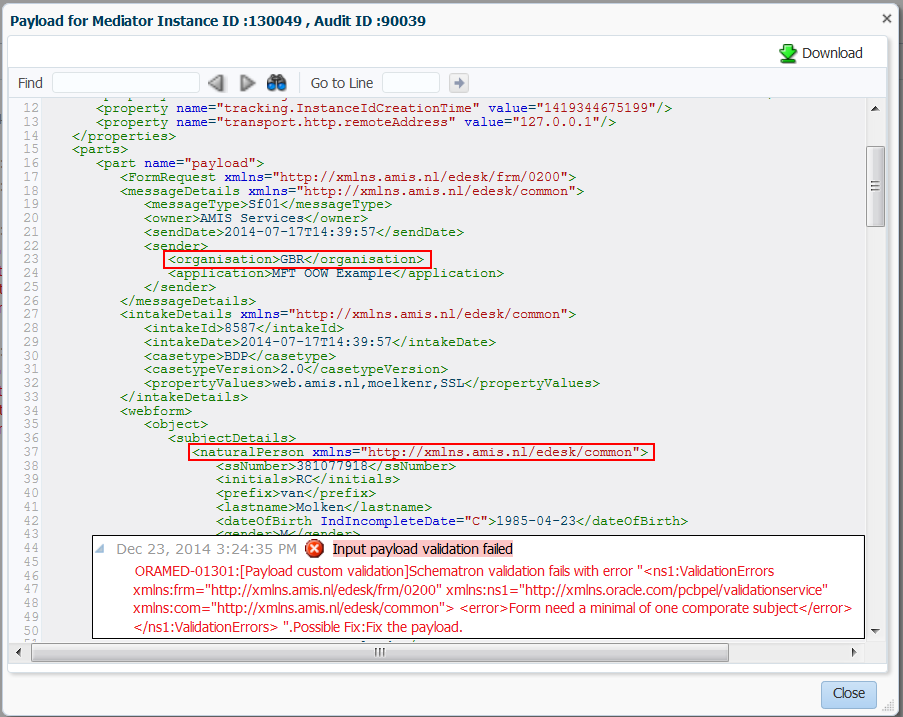
1. When editing the DVM you can change the values of the Mediator actions within a routing rule. For instance changing the schematron schema to use.  
     
   **GeneralValidation.sch** checks if the form has a minimum of one natural subject.  
   **CustomerValidation.sch** checks if the form has a minimum of one company subject.







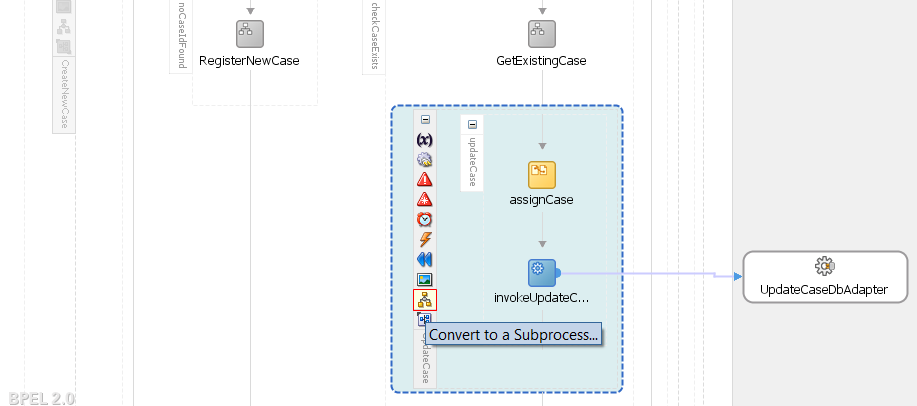
When a instance is initiated for sender AMIS the validation expects one natural subject because the GeneralValidation schema is used, but when the sender is GBR the validation uses a different schema which expects one company subject.



## 1.3. Re-use of BPEL code using Sub-processes

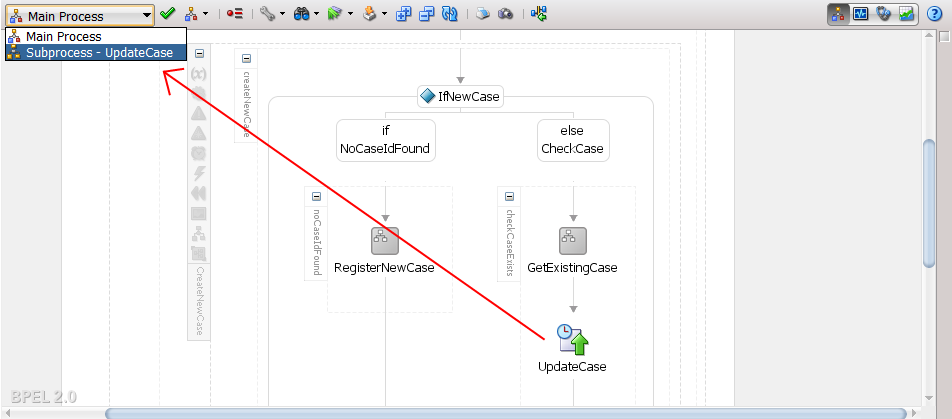
Re-use of code can be managed using BPEL Sub-processes, investigate this new feature by creating a inline sub-process from an existing scope in RegisterFormProcess.

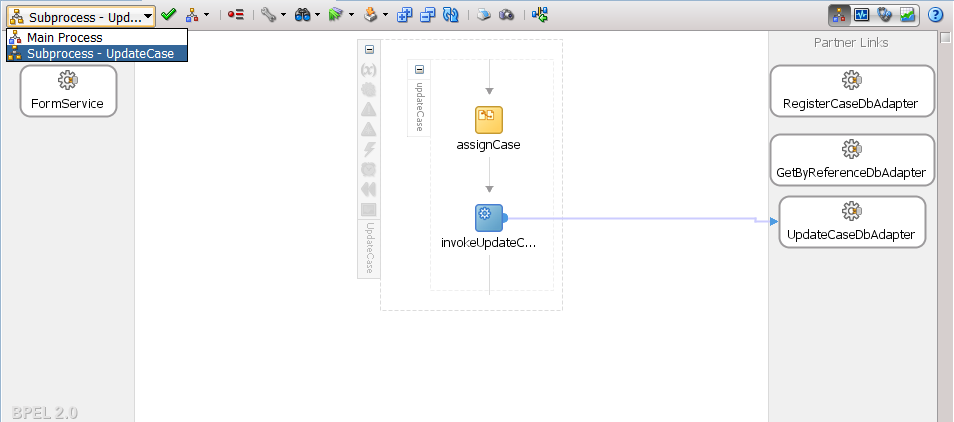
1. Open the RegisterFormProcess BPEL component and search for the UpdateCase scope under the CreateNewCase scope.
2. Right click on exiting scope, or click on the icon to create a (inline) sub-process.



|  |  |
| --- | --- |
|  | You can give it a name, choose to replace it with the call activity and some extra information as the label for the call activity, comment about the sub-process and an image to display for the call activity. |

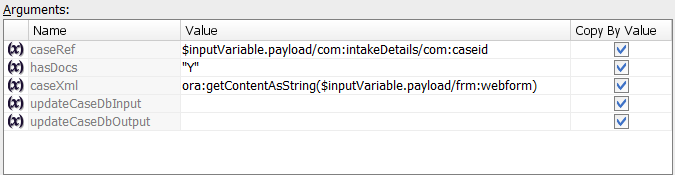
1. What happens with scope if it is converted to a sub-process?





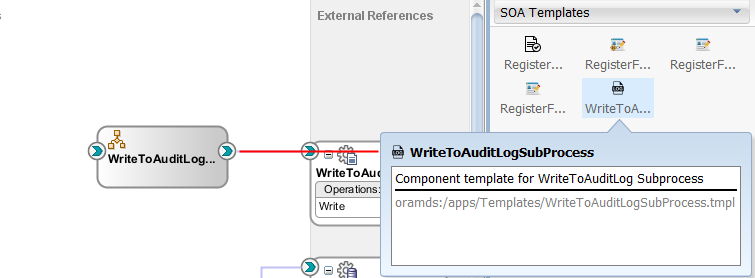
If you initialize variables with a default value, feature when creating variables, you should delete them so they can be overwritten using the call activity. You can re-initialize them without selecting a variable or delete the from tag from the source.

1. Assign values to the arguments of the UpdateCase call activity. To assign an expression first select the Copy By Value checkbox.

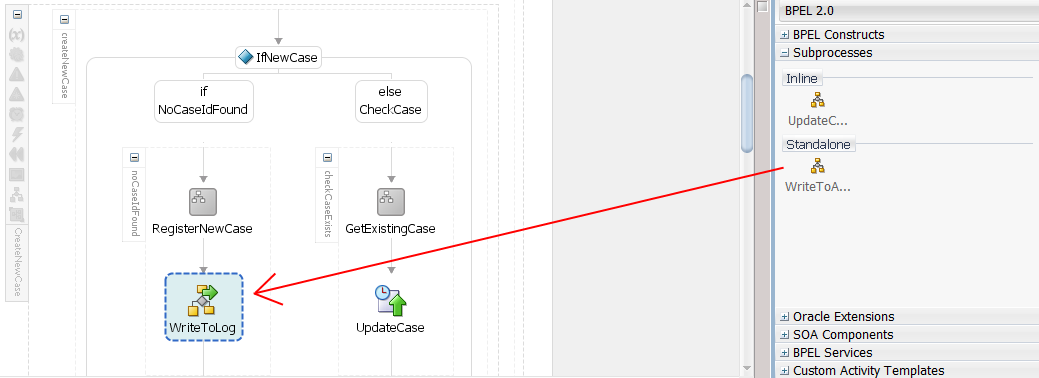


You can also create standalone subprocesses in the composite editor. Create a new Subporcess by dragging and dropping the Subprocess component onto the editor or using a template which includes the Subprocess.

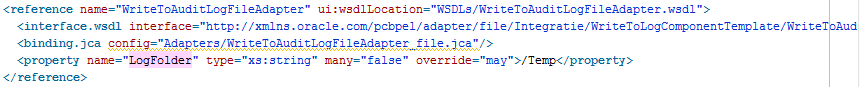
1. Prepare the composite by dragging and dropping the WriteToAuditLogSubprocess component template.



1. Open the RegisterFormProcess component and add a call activity to the standalone Subprocess. You can drag and drop the Subprocess from the Component palette which creates the call activity. Don’t forget to assign the arguments with relevant input.



1. The subprocess uses a File adapter with a logical folder, so don’t forget to set the LogFolder property in the composite.xml.



The call to the Subprocess is included in the audit logging.

