SWIFT MT/FIN Encoder

Version 1.0

SWIFT MT/FIN Encoder is used to decode SWIFT MT/FIN data into XML format, or encode XML data into SWIFT MT/FIN format. It is built up on **SWIFT SDK** which provides the basic parsing. **SWIFT MT/FIN Encoder** also provides flexible extensions for schema validation, cross-fields semantic validation and text validation, through XSLT, Schematron, JavaScript, JRuby, java, etc, or some combinations of them, you can choose depending on your own preference and expertise.

SWIFT MT/FIN Encoder seamlessly integrates with the powerful Open Source platform **OpenESB** and **GlassFishESB** (JBI based). It can be used in different ways, e.g. in interative **SWIFT Encoder Tester**, hook-up with dozens of **Open JBI components** (e.g. ftp, file, database, smtp, imap, pop3, http, soap, rest, jms, ldap, mq, msmq, snmp, scheduler etc), in BPEL engine, POJO engine.

SWIFT MT/FIN Encoder can also run outside **OpenESB** / **GlassFishESB**, e.g. in Java EE, Java SE.

Potentially, **SWIFT MT/FIN Encoder** can also be used in SeeBeyond/SUN/Oracle **Java CAPS** platform (the prior generations of **OpenESB**) in certain ways. Actually **SWIFT SAI** is a product built up on **Java CAPS**, combining with **SWIFT SDK**.

This PDF document (http://swiftencoder.appspot.com/SwiftEncoder.pdf) is a step-by-step tutorial to **SWIFT MT/FIN Encoder**.

There are also several screen casts showing the steps:

1. Basics http://swiftencoder.appspot.com/SwiftEncoderBasics.html
2. In BPEL http://swiftencoder.appspot.com/SwiftEncoderInBPEL.html
3. In POJO http://swiftencoder.appspot.com/SwiftEncoderInPOJO.html
4. In Java SE http://swiftencoder.appspot.com/SwiftEncoderInJavaSE.html
5. In Java EE http://swiftencoder.appspot.com/SwiftEncoderInJavaEE.html

There is also a page (http://swiftencoder.appspot.com/MTXML_Tester.jsp) which allows you to perform online decode/encode conversions between MT/FIN and XML.

Table of Contents

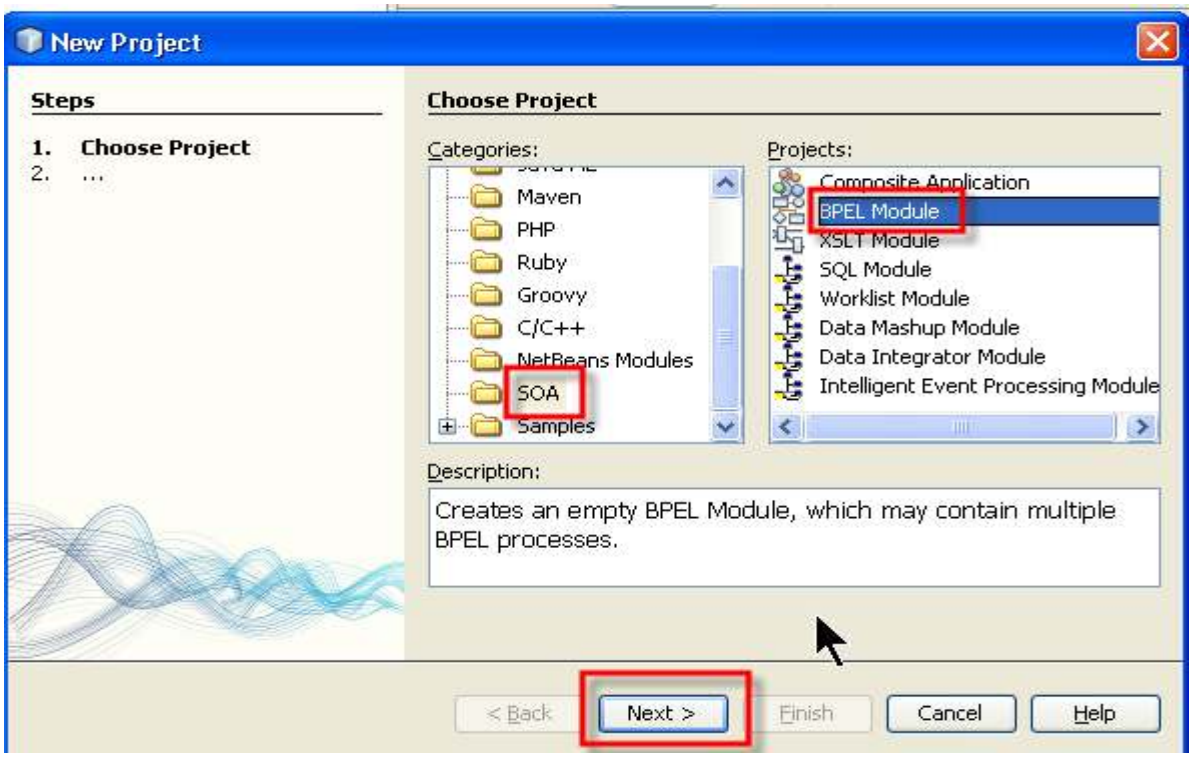
1. Basics of SWIFT MT/FIN Encoder

We'll go through basic functions of SWIFT Encoder, also practice SWIFT Encoder Tester. We'll show the extensions for cross-fields semantic validation and text validation with examples of XSLT, Schematron, java.

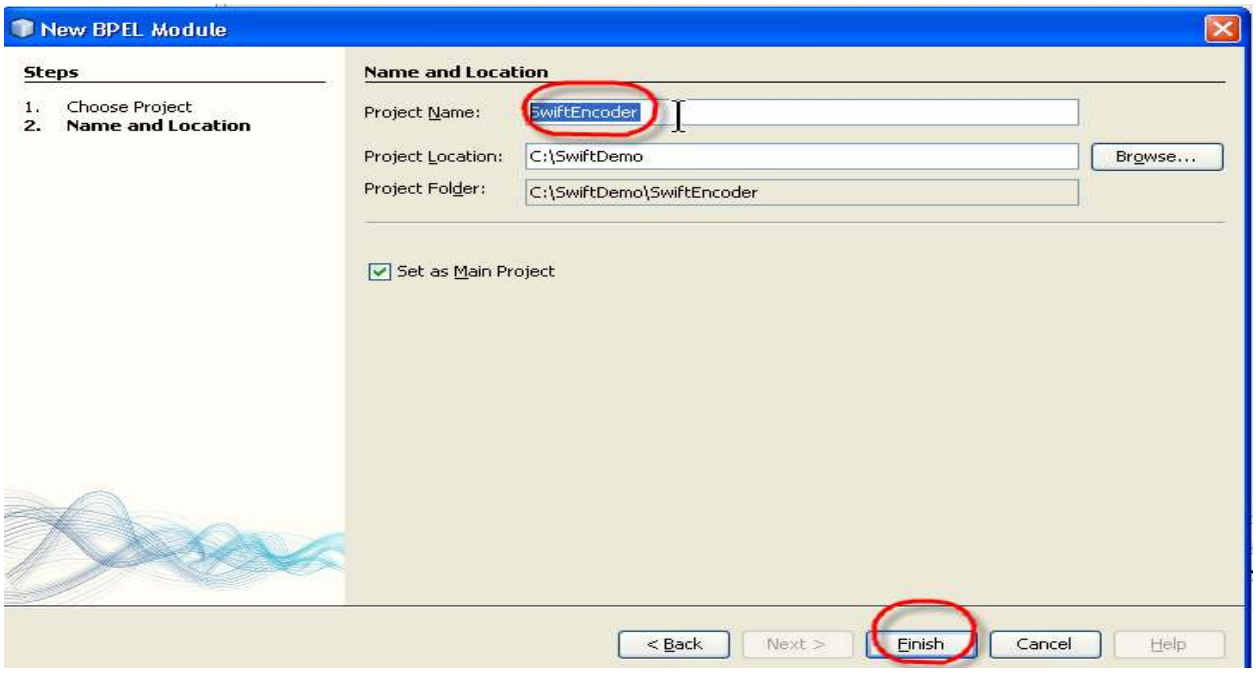
Here is a screen cast:

http://swiftencoder.appspot.com/SwiftEncoderBasics.html

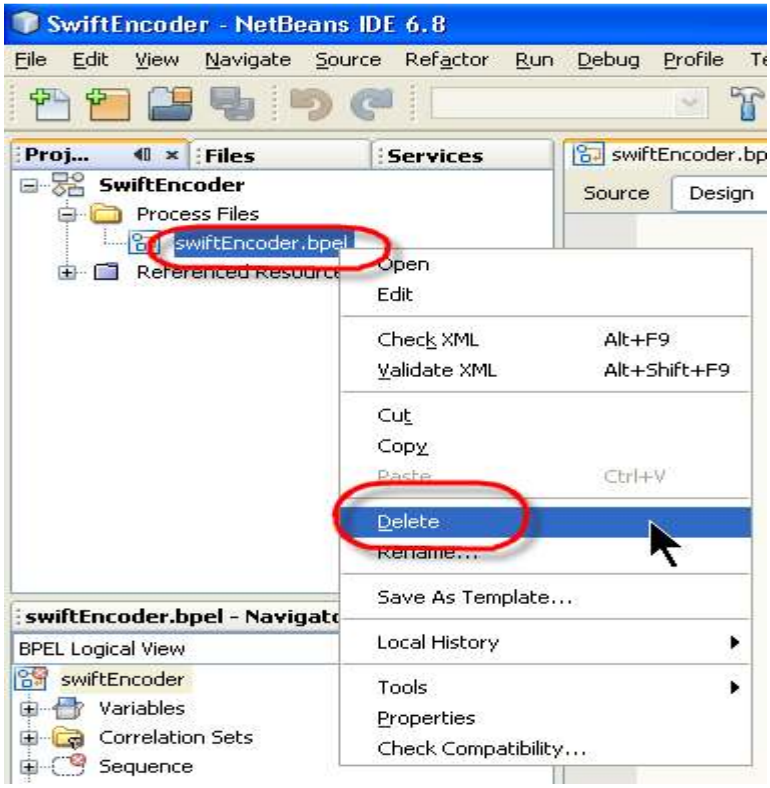
Create a project SOA/BPEL Module.



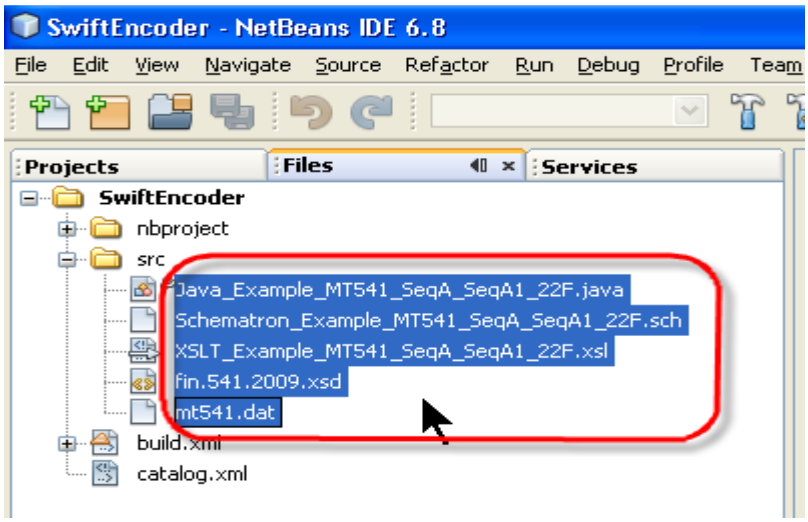
Name it as "SwiftEncoder"



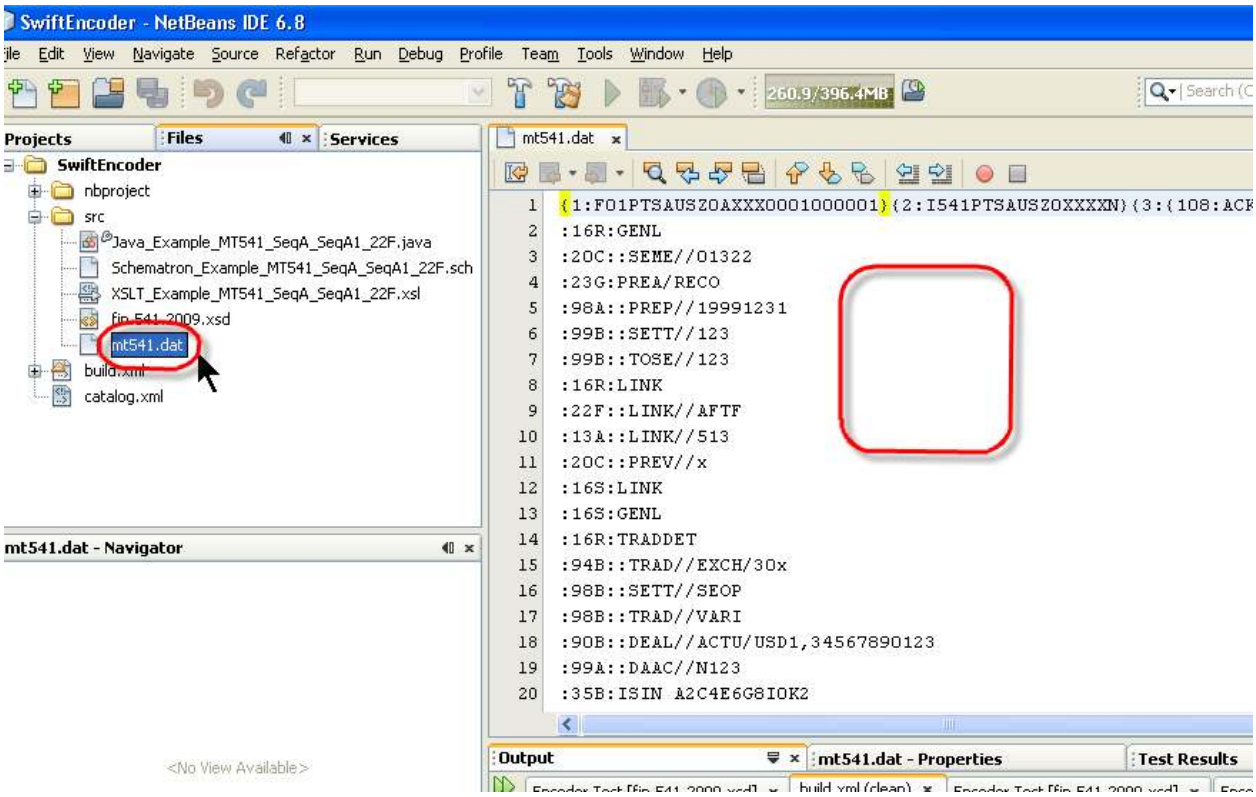
Delete the .bpel file because we don't need it.



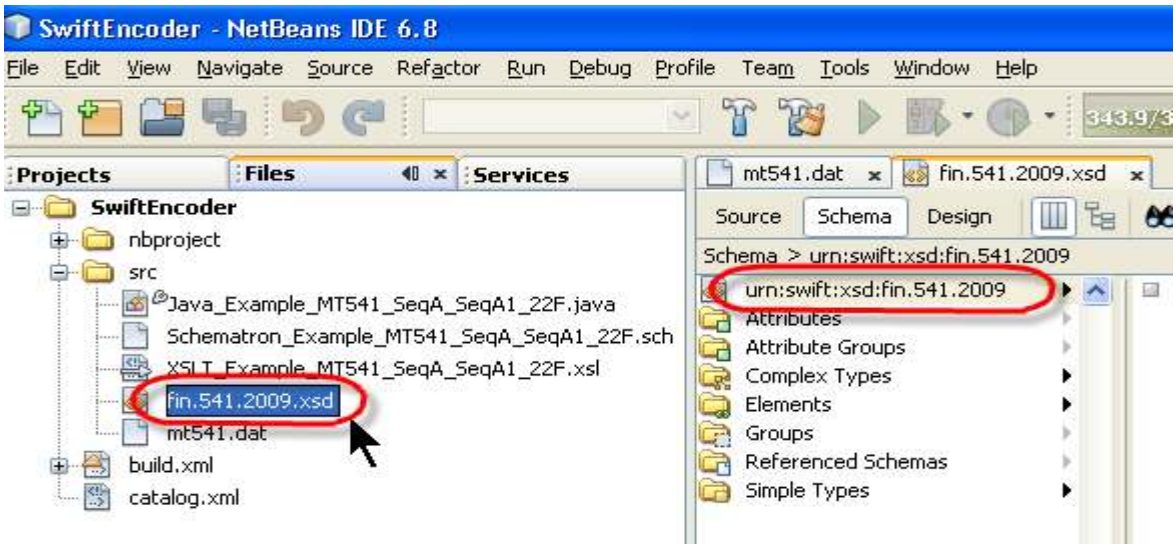
Copy some resource files into project, they will be used later.



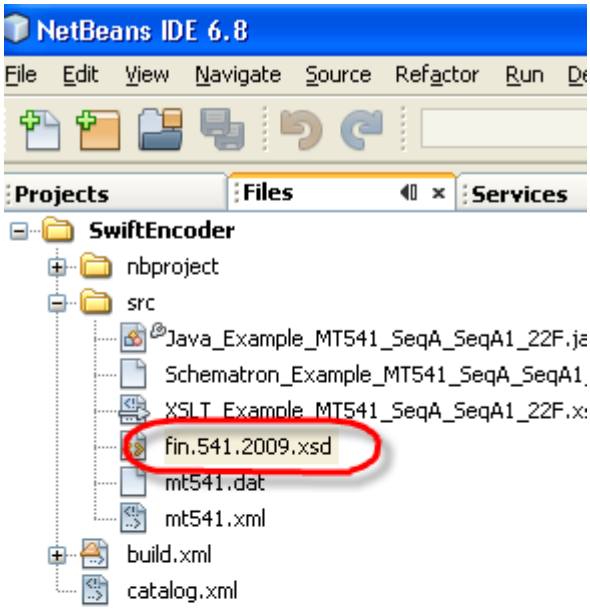
Here is an MT test file, mt541.dat.

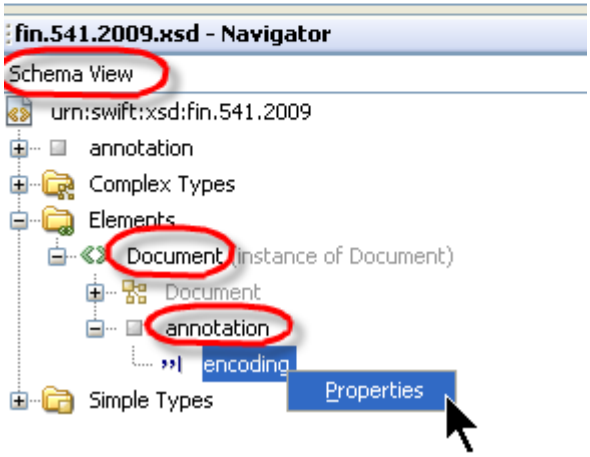


Here is a schema for fin.541.2009.xsd.

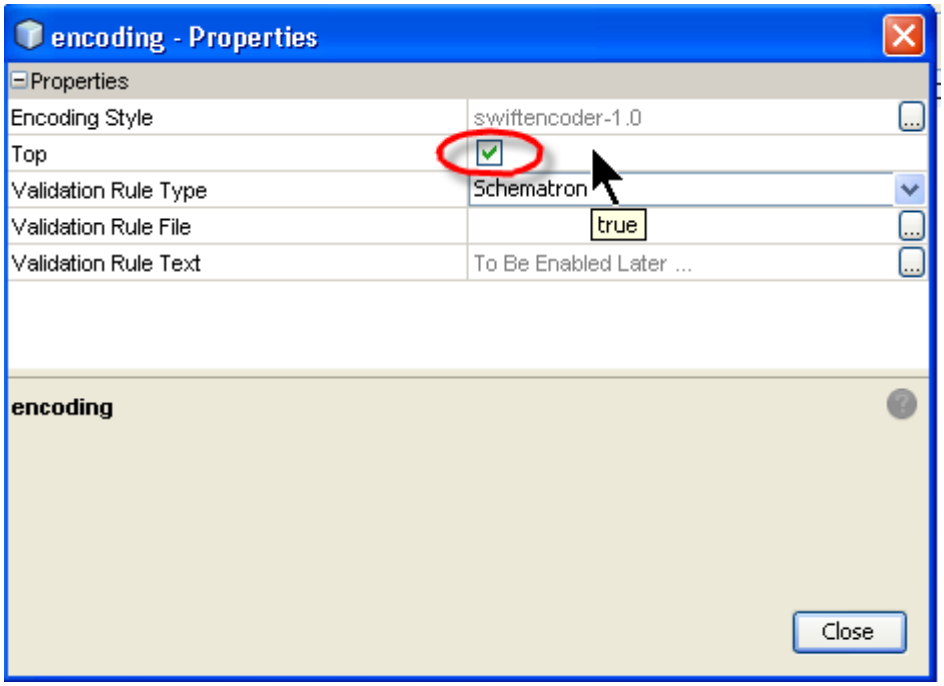


Select schema file. In Schema View, navigate to Document->annotation->encoding. Right-click on “encoding”, select “Properties”.


NetBeans IDE 6.8
File Edit View Navigate Source Refactor Run
Projects
Files
Services
SwiftEncoder
nbproject
src
Java_Example_MT541_SeqA_SeqA1_22F.ja
Schematron_Example_MT541_SeqA_SeqA1
XSLT_Example_MT541_SeqA_SeqA1_22F.x
fin.541.2009.xsd
mt541.dat
mt541.xml
build.xml
catalog.xml

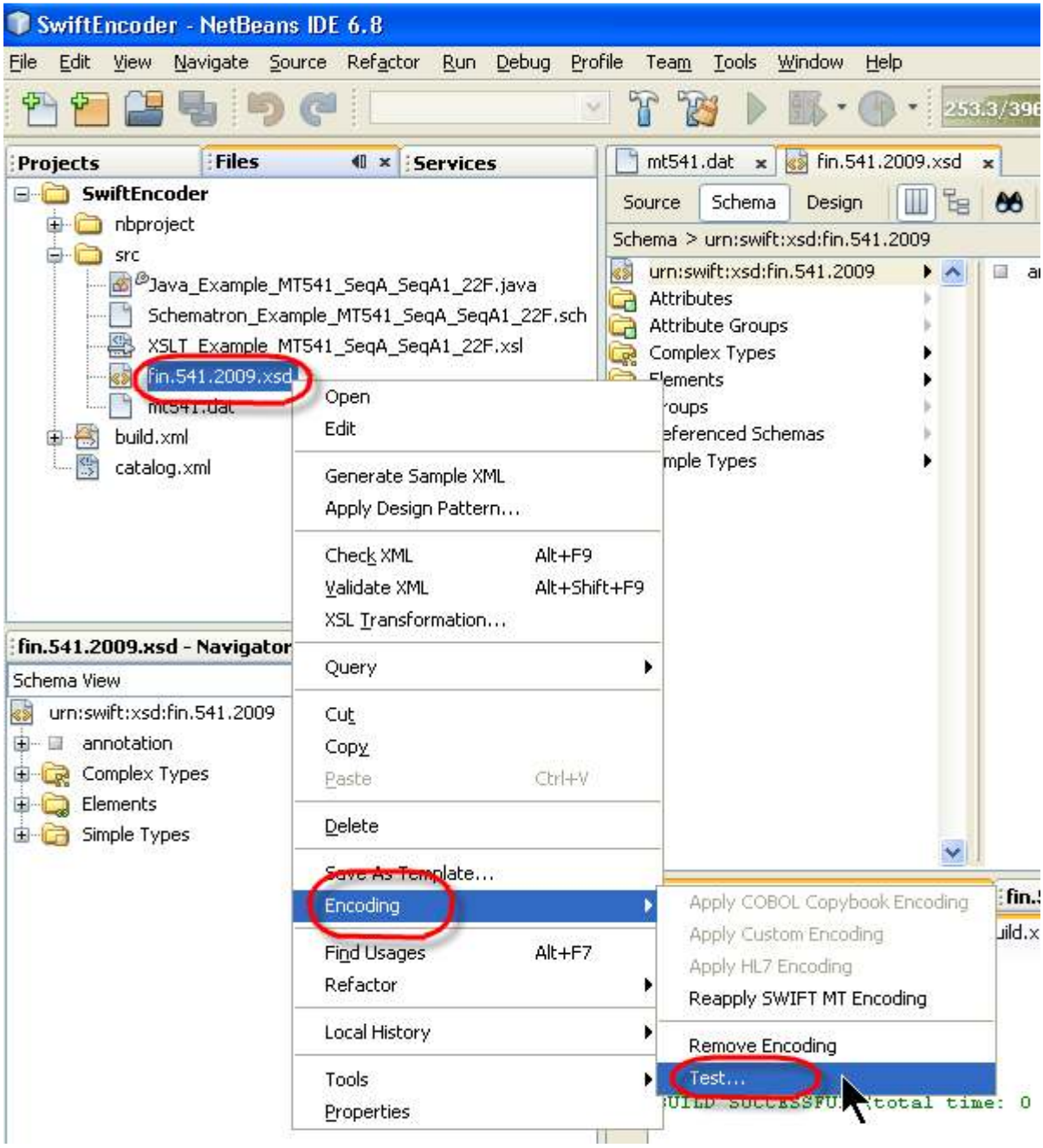

fin.541.2009.xsd - Navigator
Schema View
urn:swift:xsd:fin.541.2009
annotation
Complex Types
Elements
Document (instance of Document)
Document
annotation
encoding
Properties
Simple Types

Check “Top” check box.

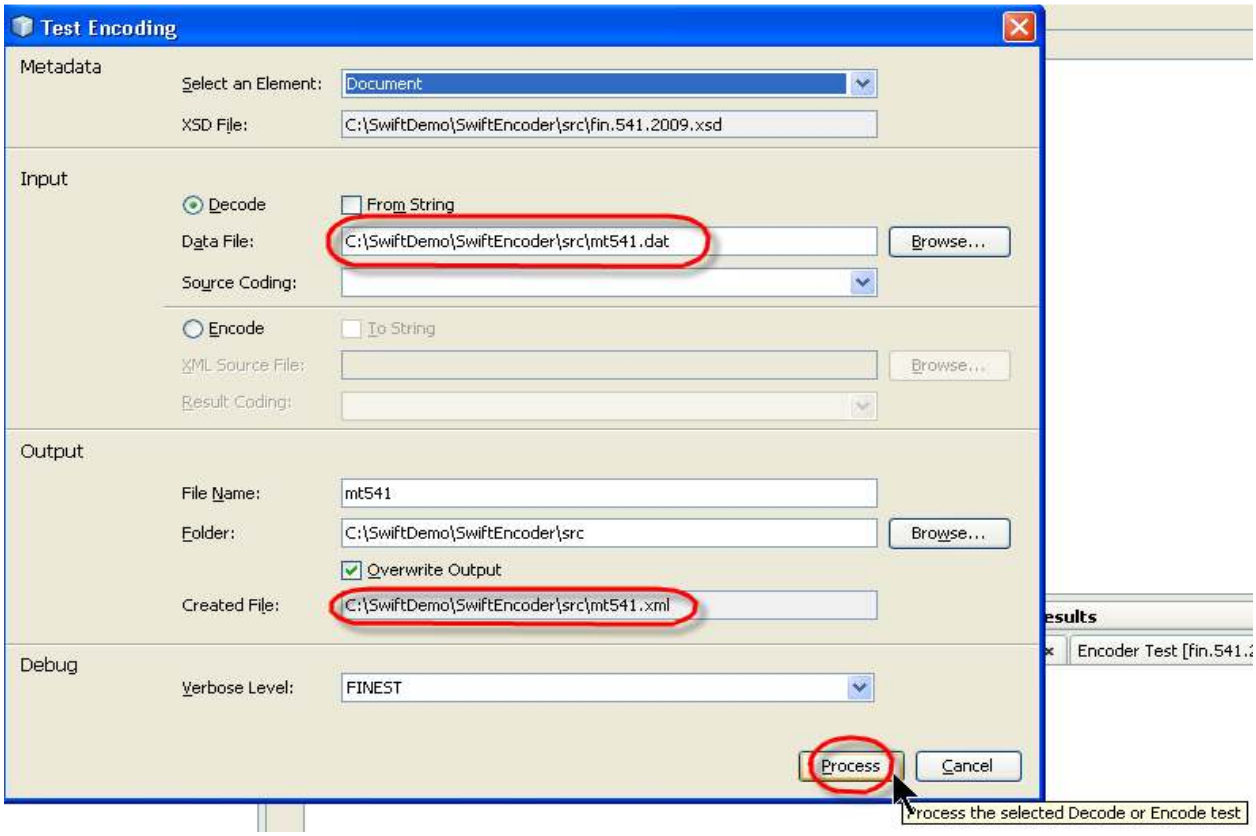


Save all.

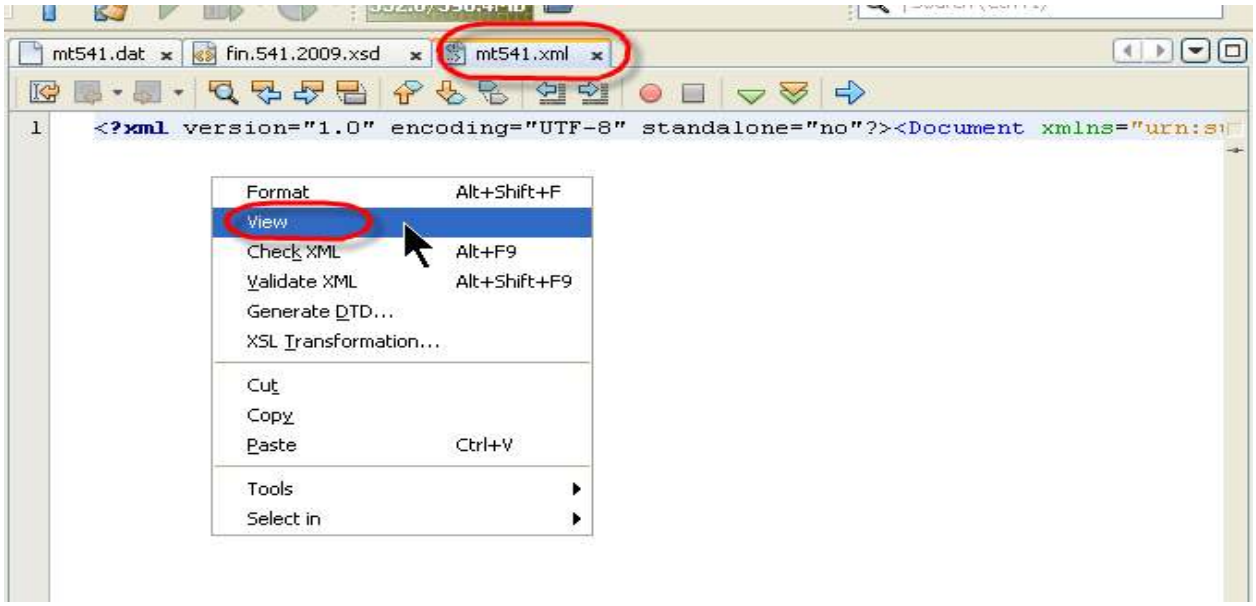
Right-click on schema “fin.541.2009.xsd”, choose “Encoding->Test”.



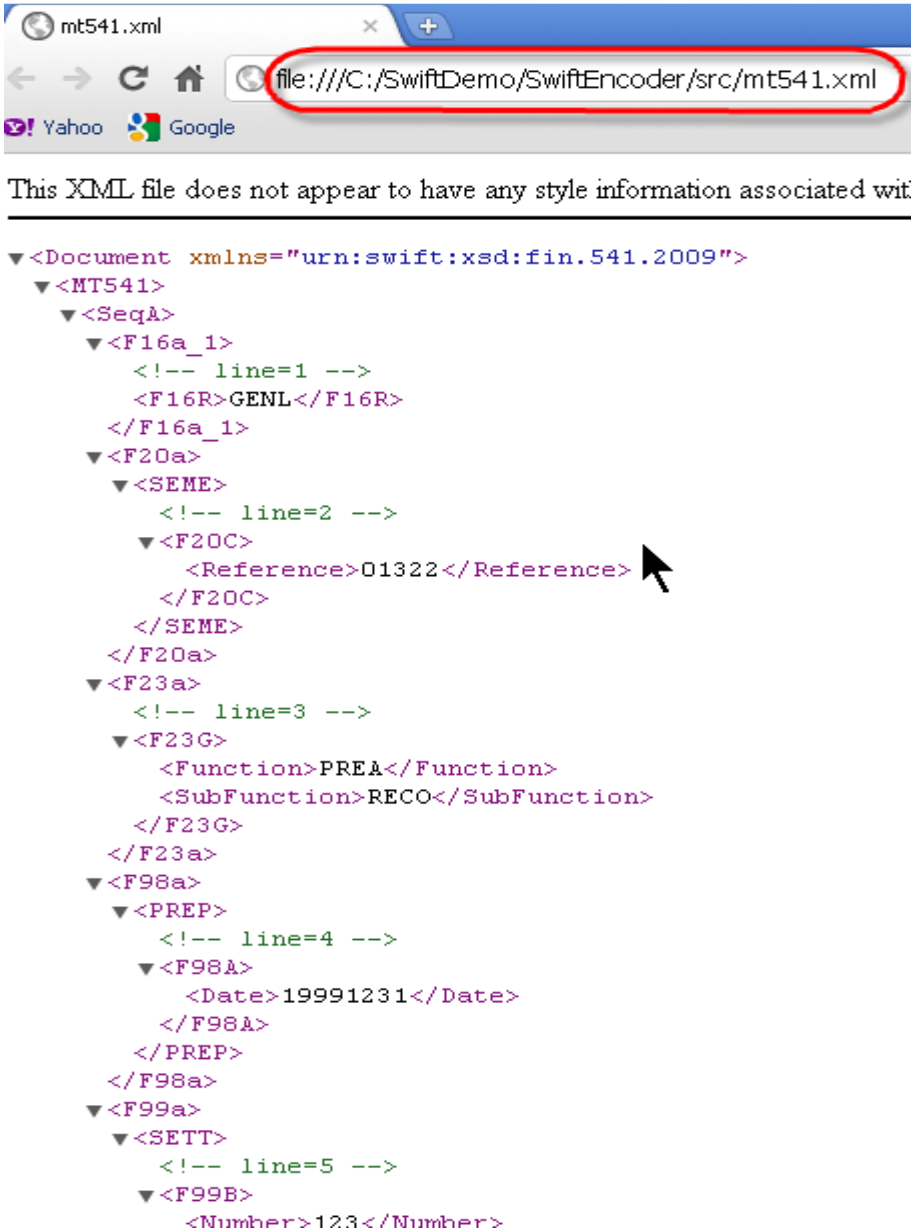
Test “Decode”. Select the input MT data file, mt541.dat. Specify the output XML file, mt541.xml. Click “Process” button.



Output mt541.xml is generated. Right-click, select “View”.

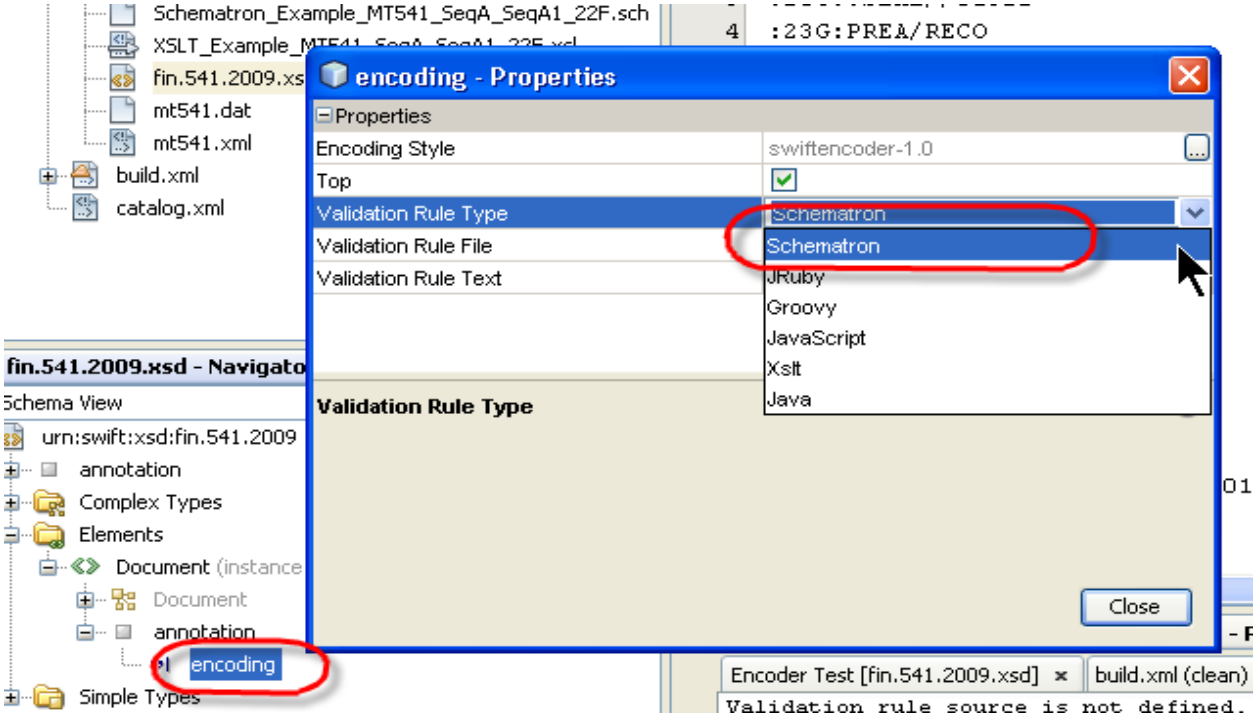


You can see view the XML file mt541.xml, e.g. in a web-browser.

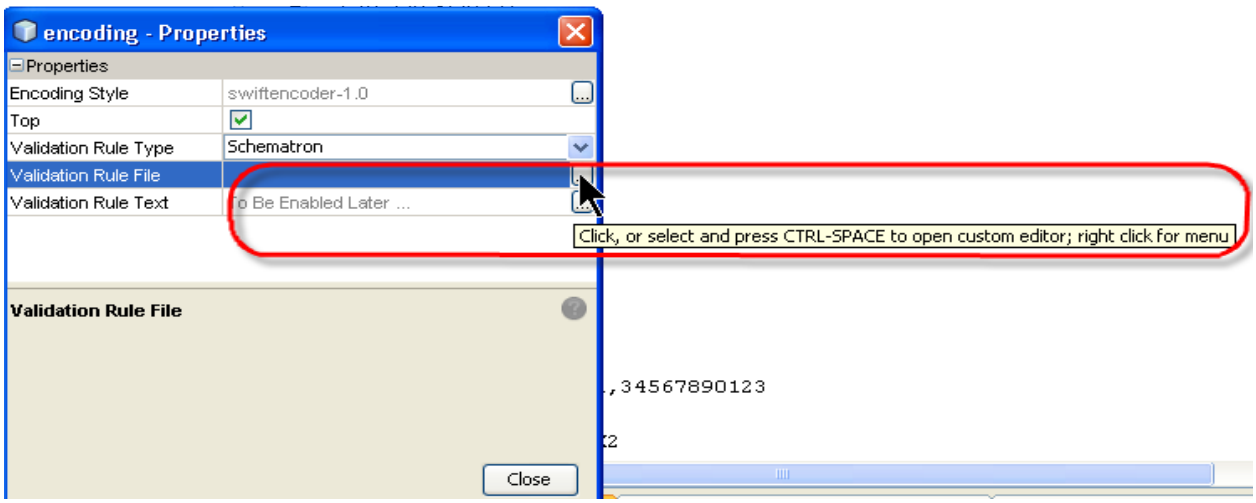


Note: This sample MT data actually contains a semantic validation error, it is caught by default parsing/validation engine. It will be covered below.

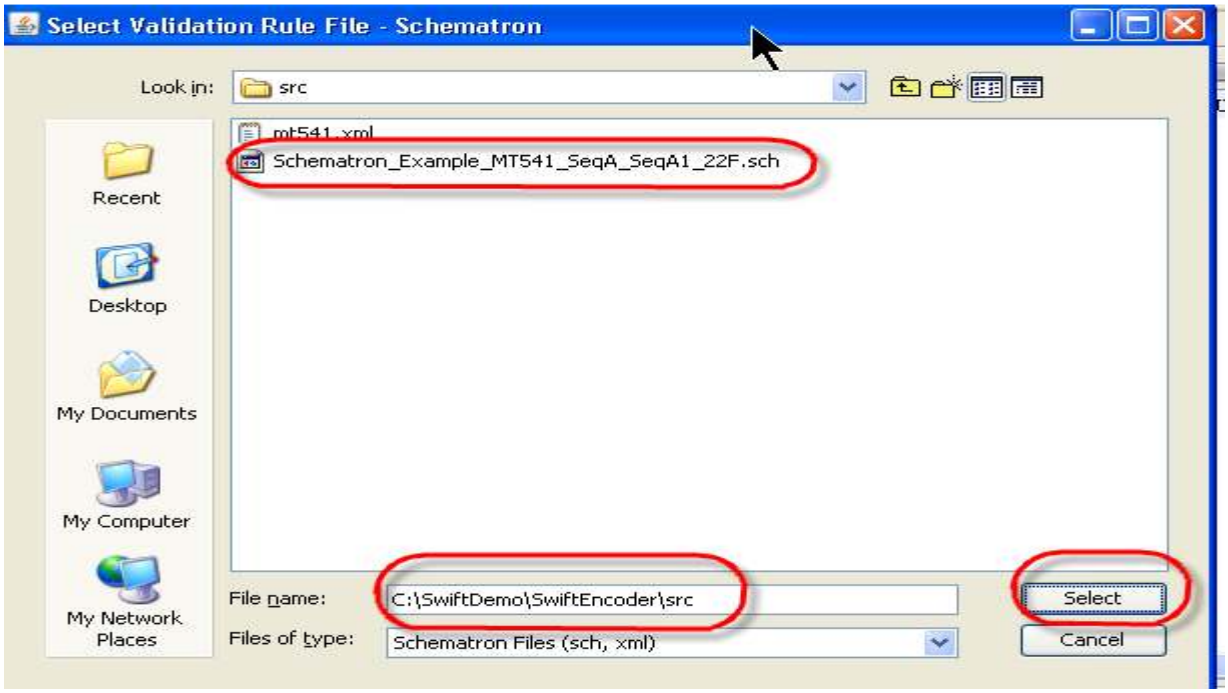
Some validation rule types are plugged. Schematron, XSLT, JavaScript, Java, etc … I'll give some examples below.



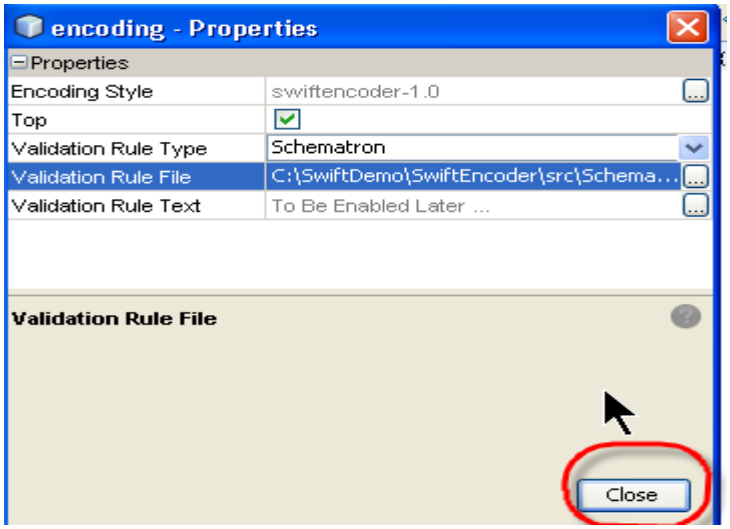
You provide the validation rules via files. For example, select "Schematron", then choose file …



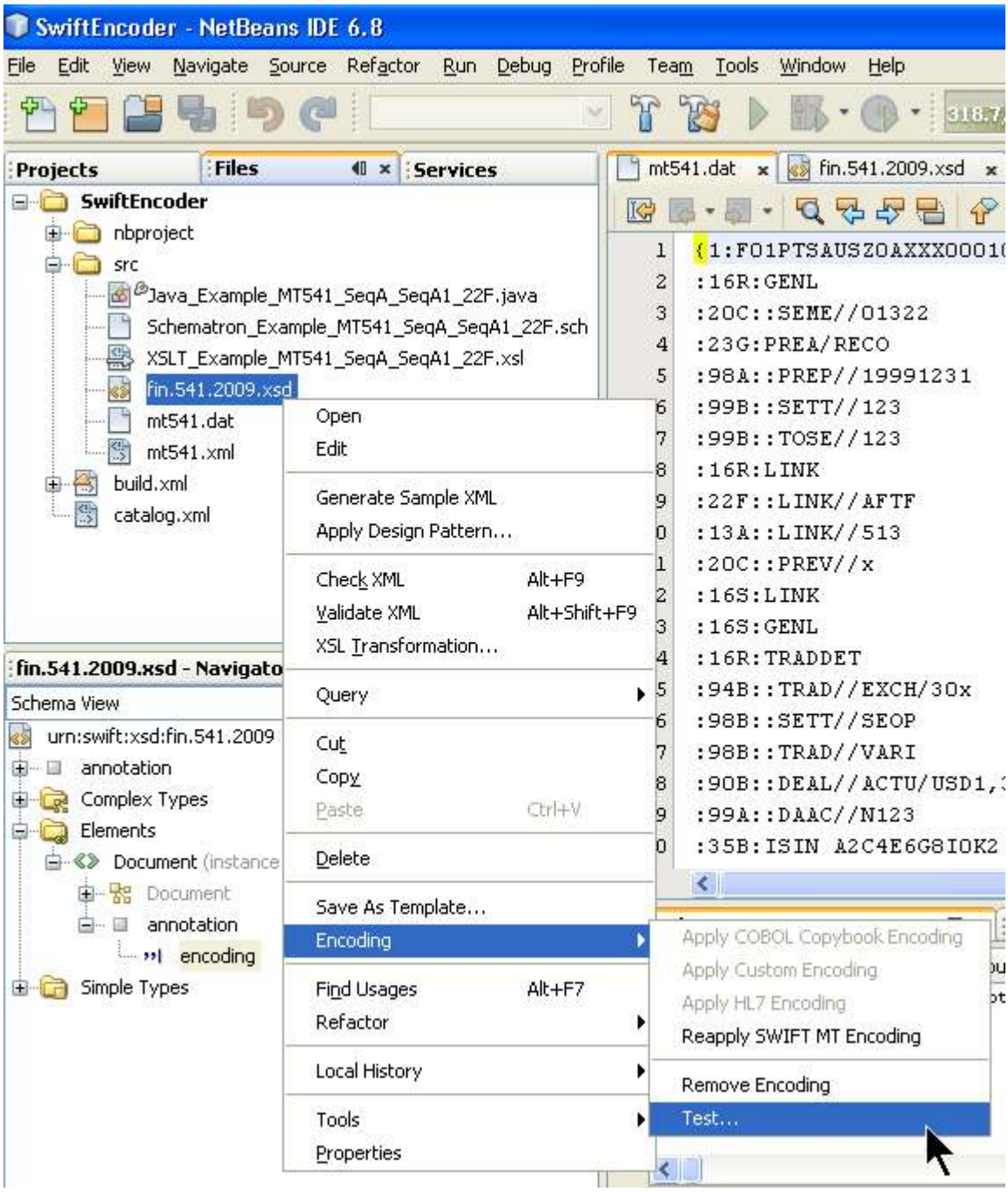
Select the sample Schematron file,
"Schematron_Example_MT51_SeqA_SeqA1_22F.sch".



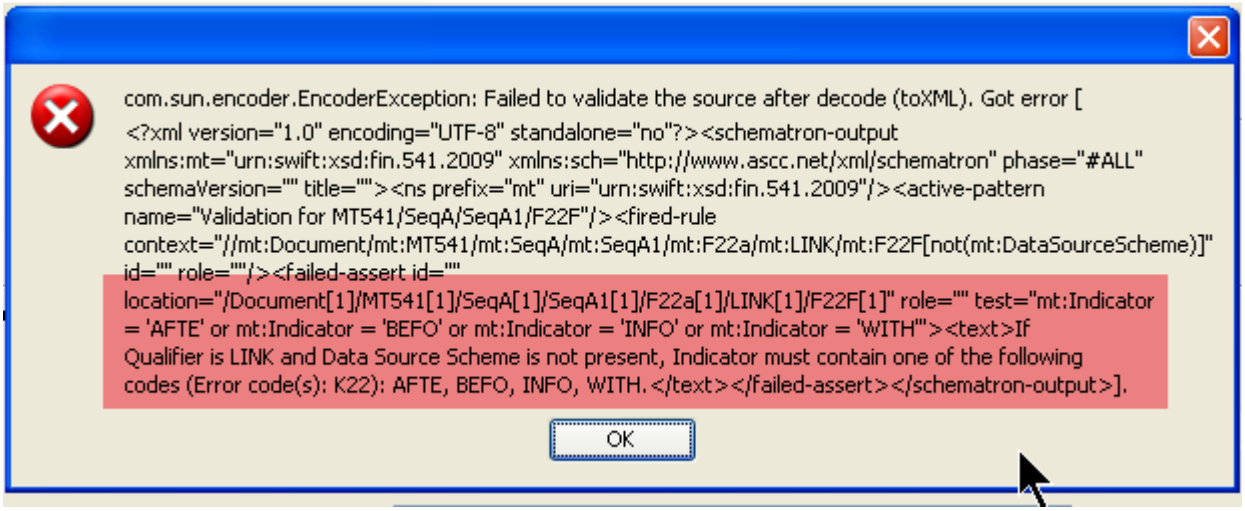
Finish the file selection.



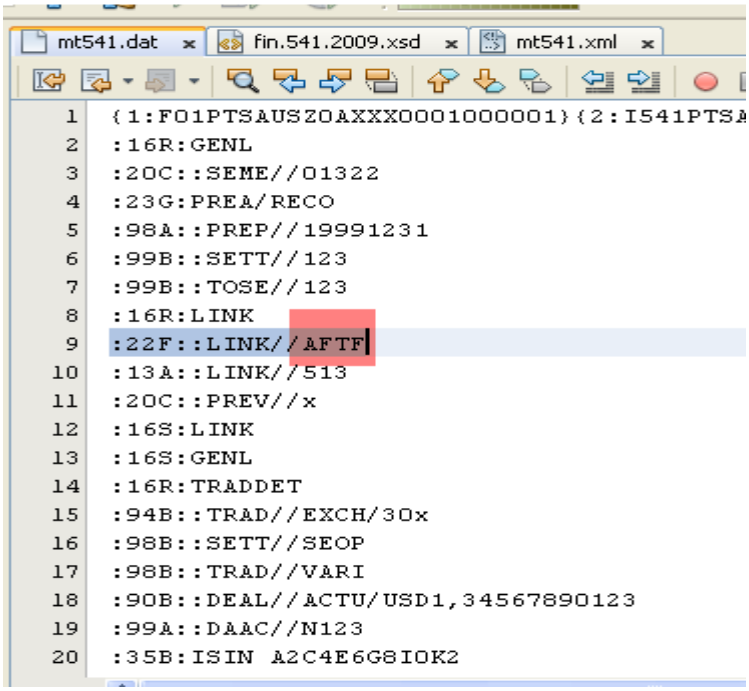
Save all and Test again …



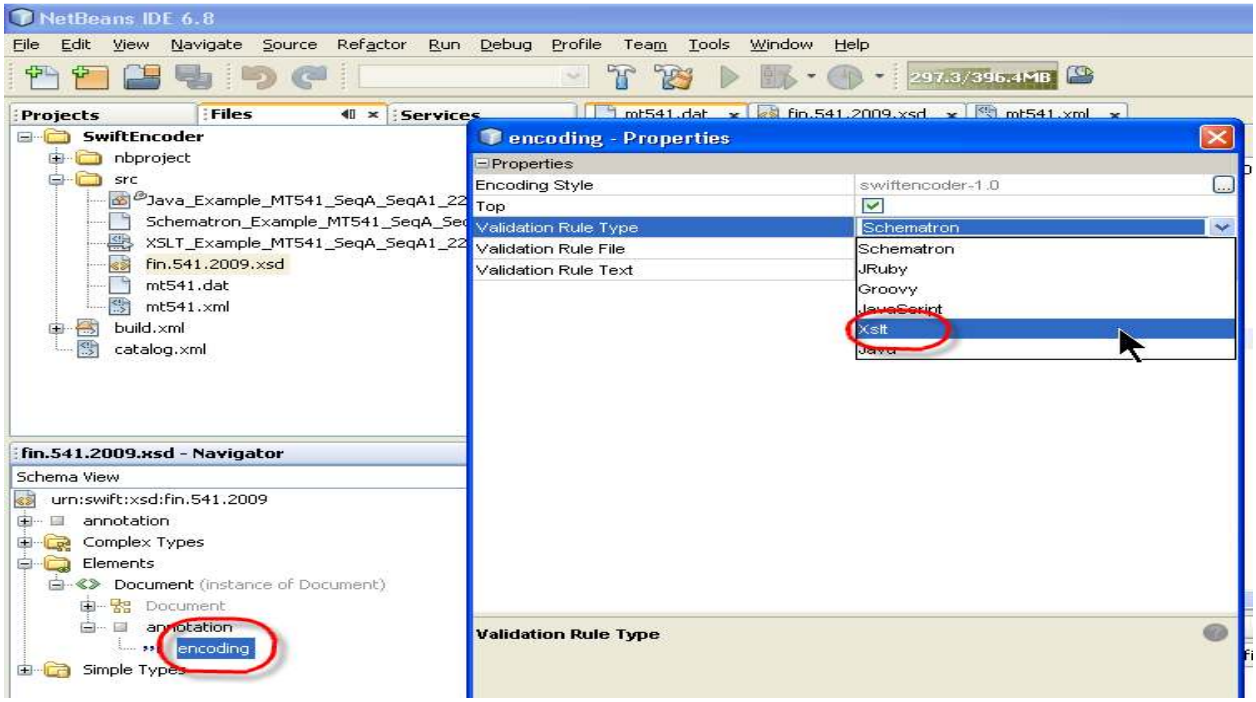
You see an error pop-up window. Schematron rules are taking effect. The data contains a semantic error, as described by the error message.



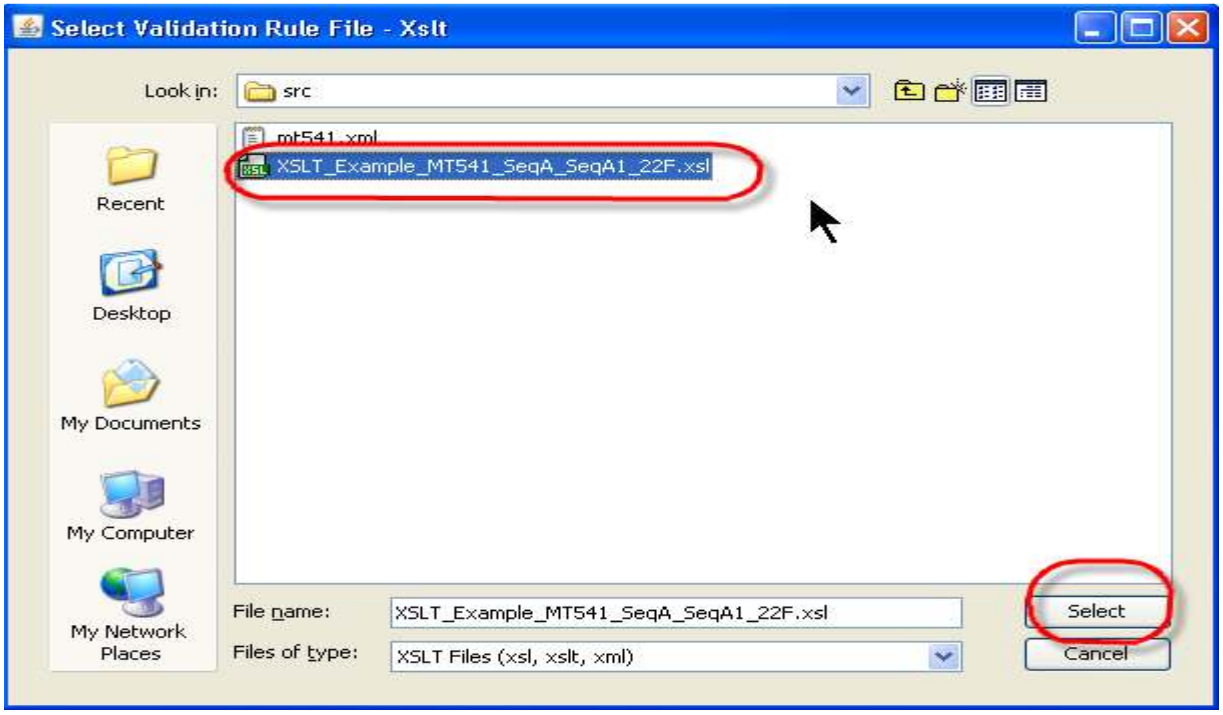
Checking back the MT data, locating the error point, "AFTF" is invalid.



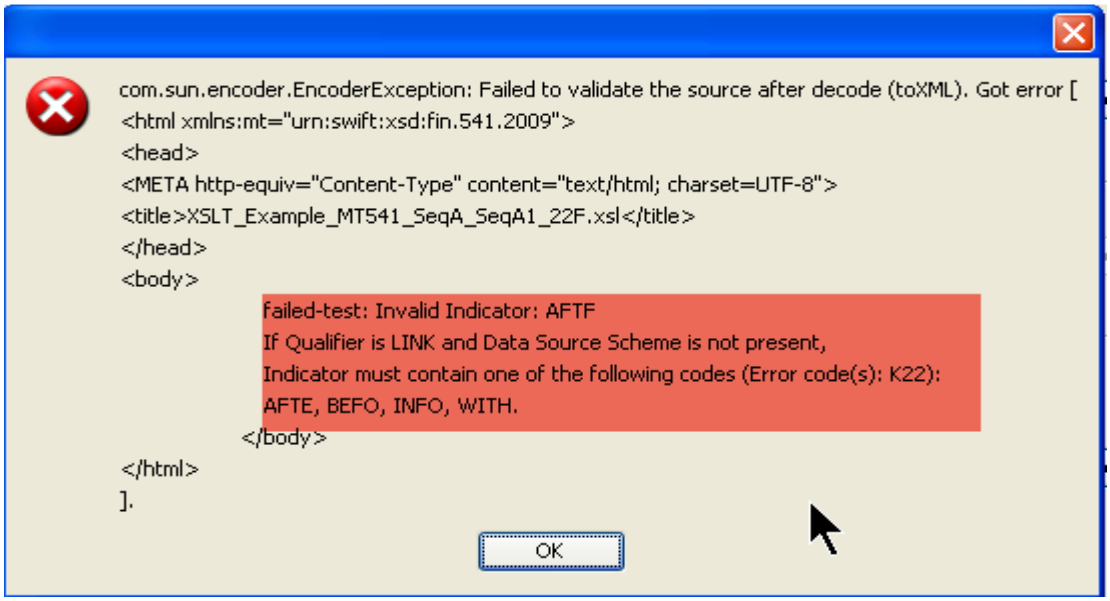
Try to use "XSLT" to validate it …



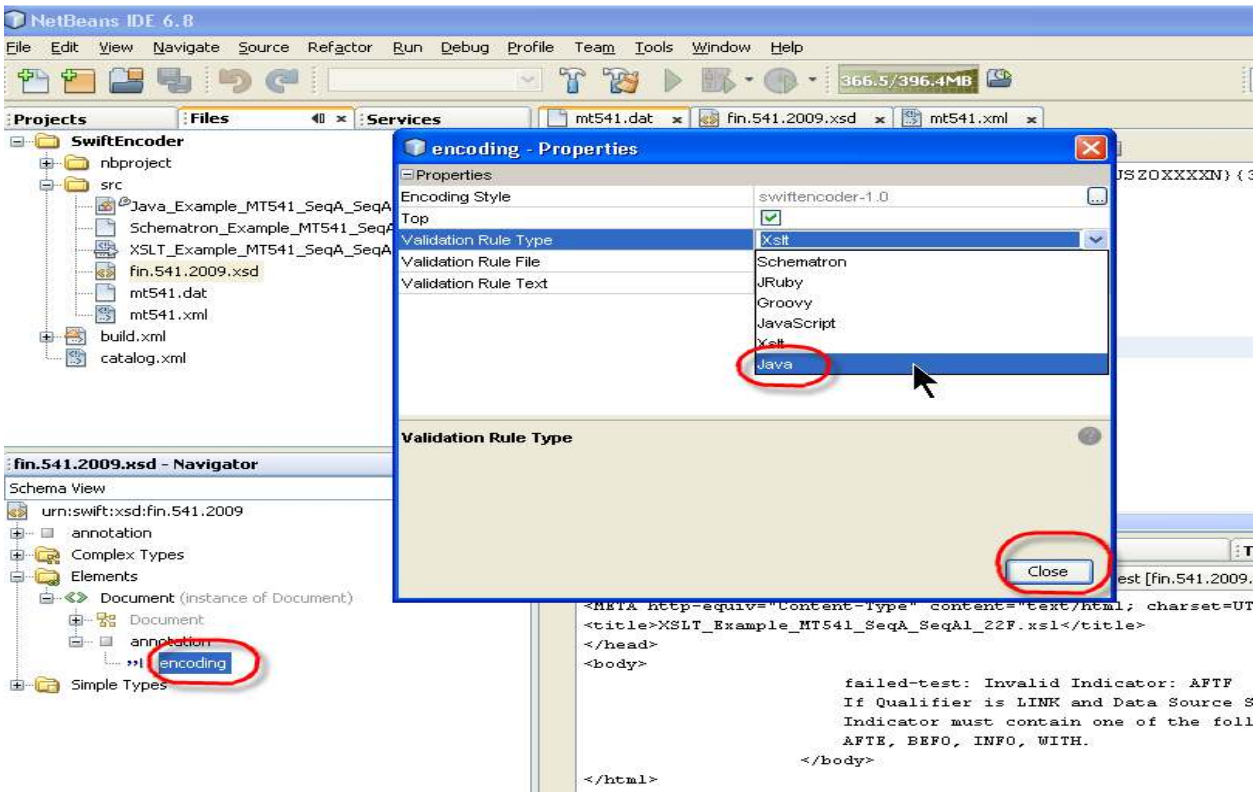
Select the provided XSLT sample code, …, save, test.



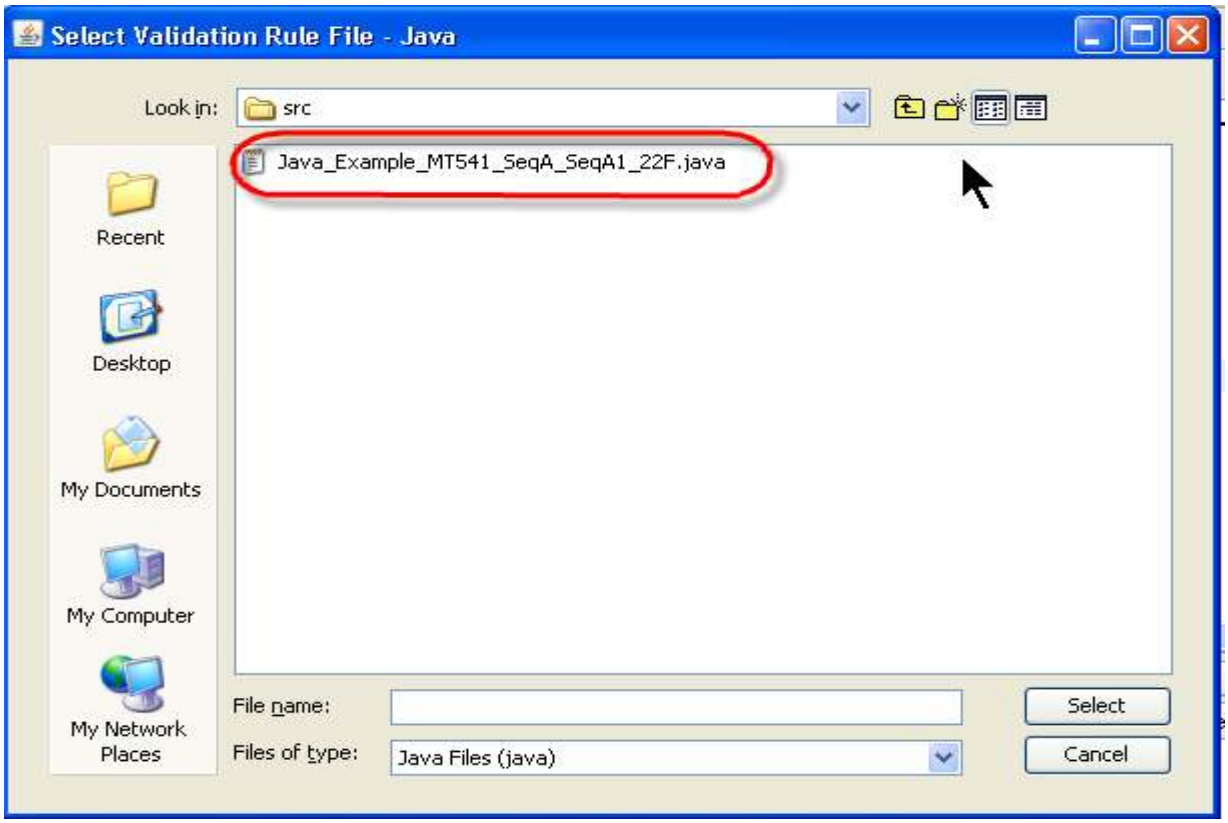
Good. You get an error pop-up window, XSLT rules are taking effect. You also can see the error description.



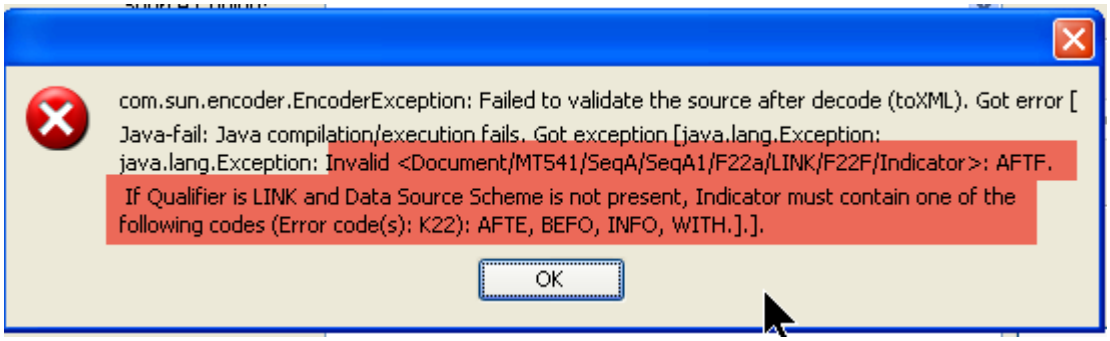
Let's try "Java" …



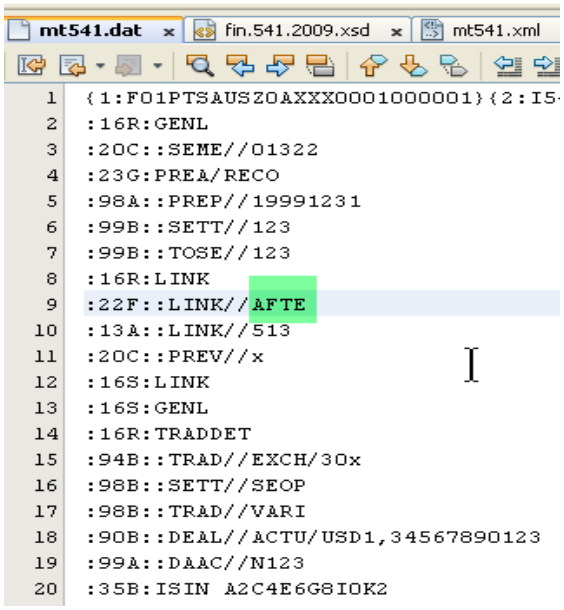
Select the provided Java sample code, …, save, test.



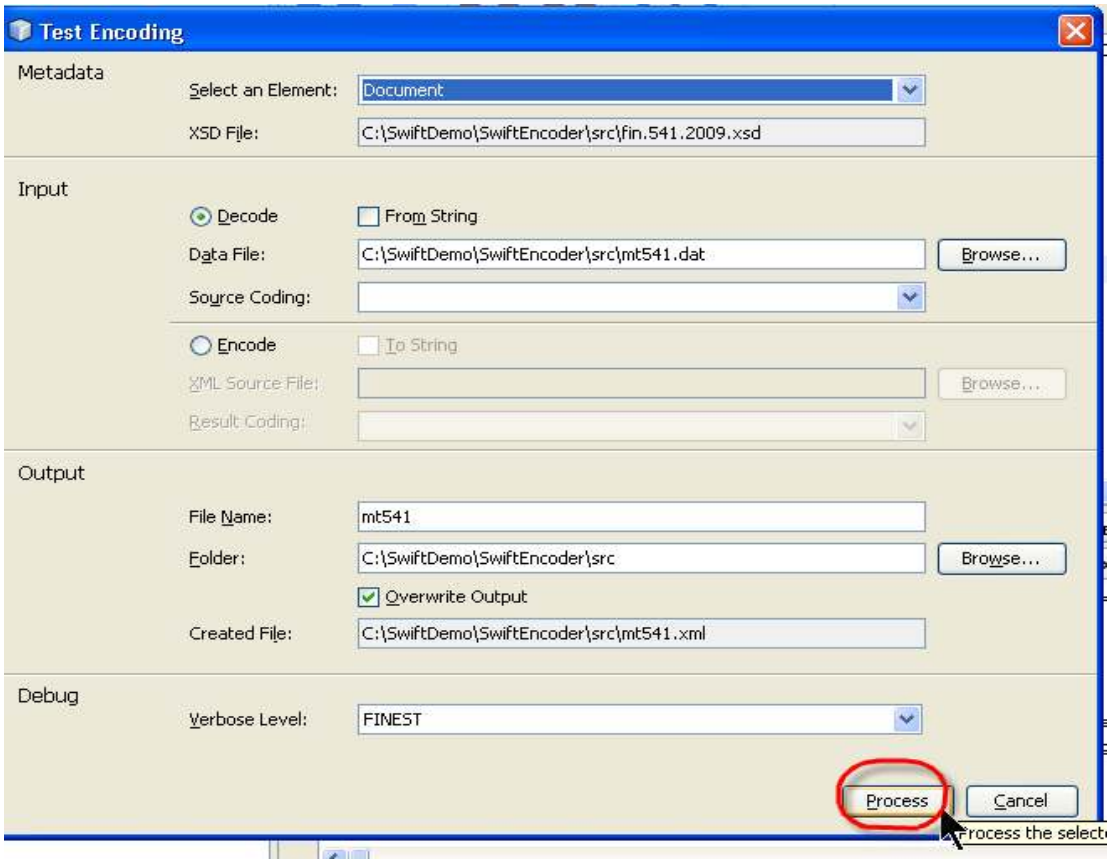
You'll see, Java also catches the data error.



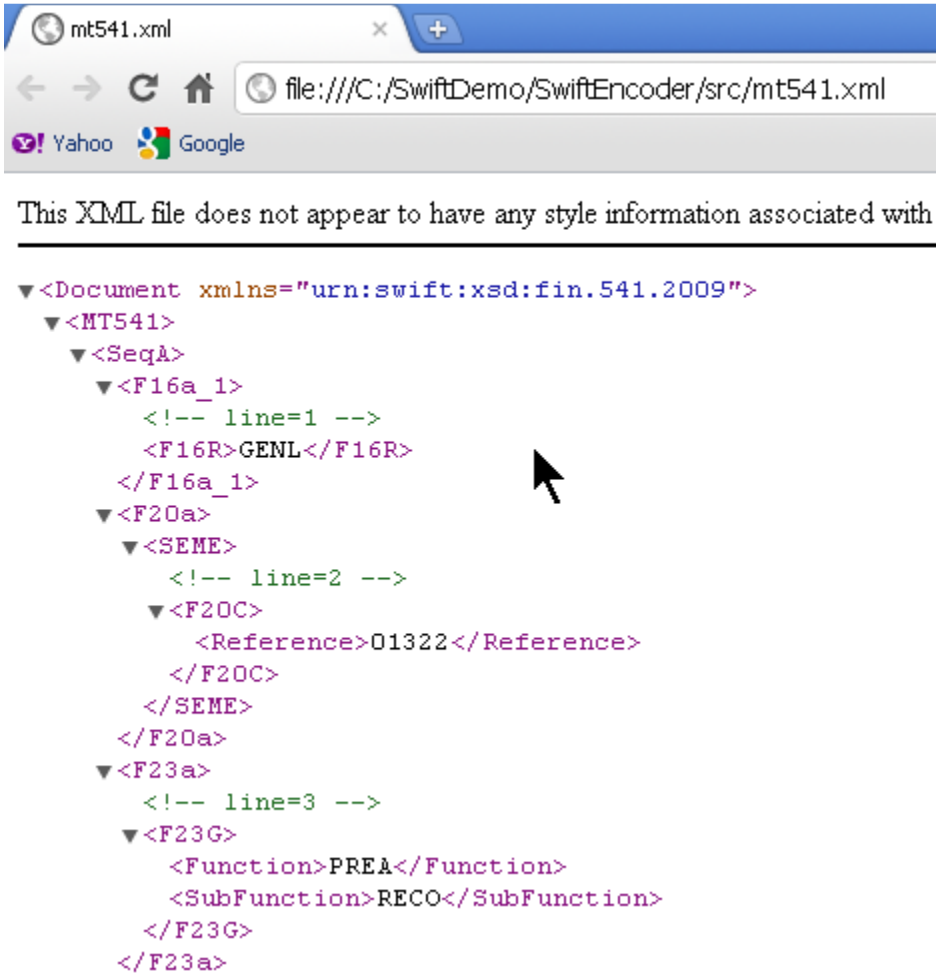
Let's correct the MT data error by changing "AFTF" to "AFTE".



Save the data. Test again …



Ok. Now error goes away. XML output will be generated.



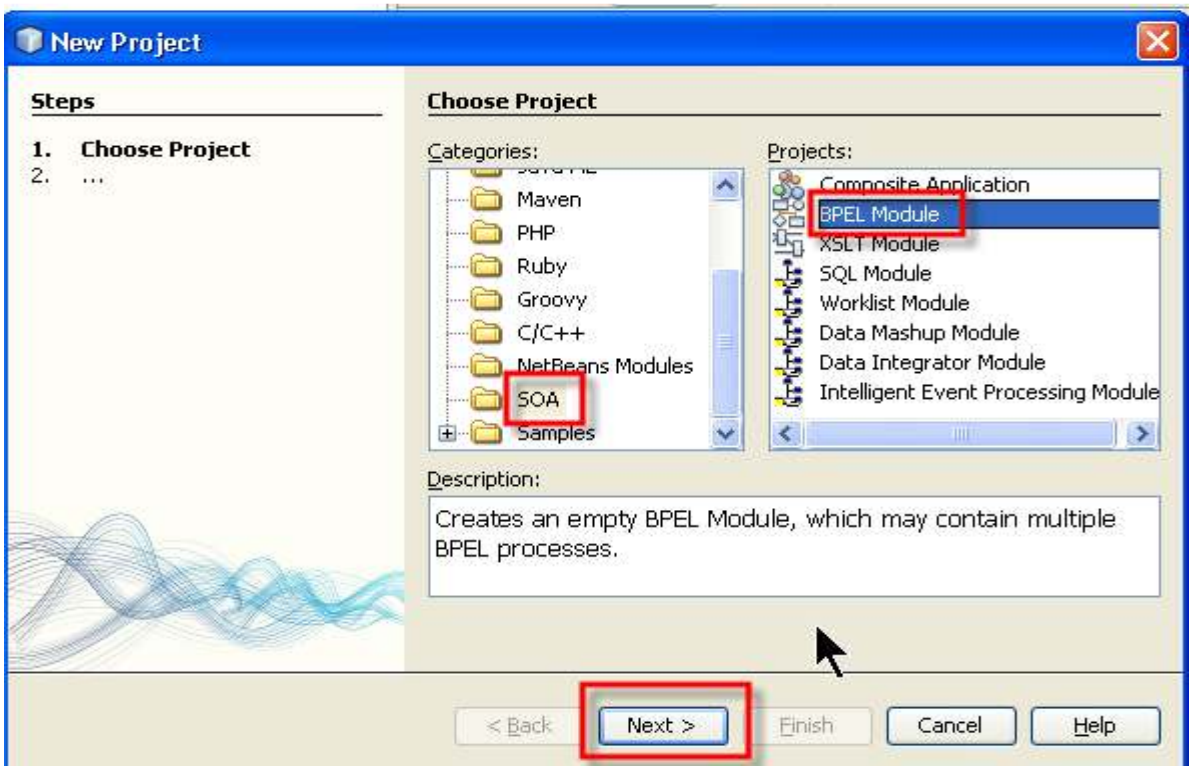
2. Using SWIFT Encoder in BPEL

Demonstrate how to use SWIFT Encoder in BPEL SE (Service Engine) by use of a simple example.

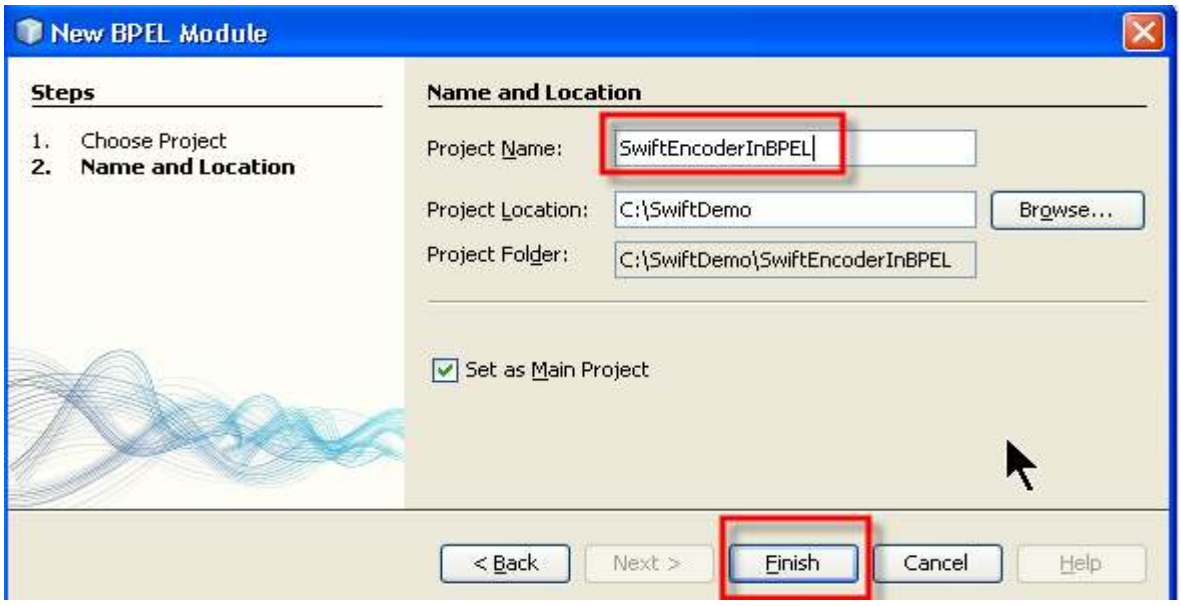
Here is a screen cast:

http://swiftencoder.appspot.com/SwiftEncoderInBPEL.html

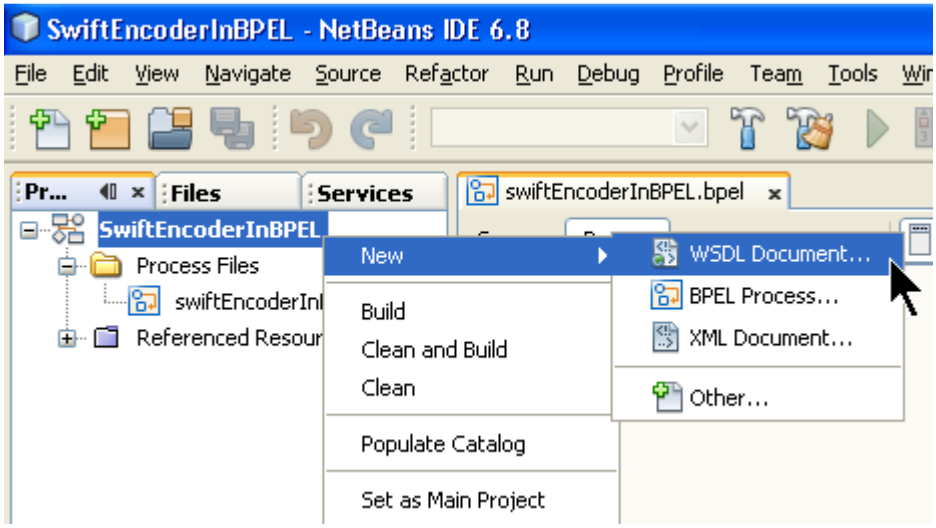
Create an BPEL project.



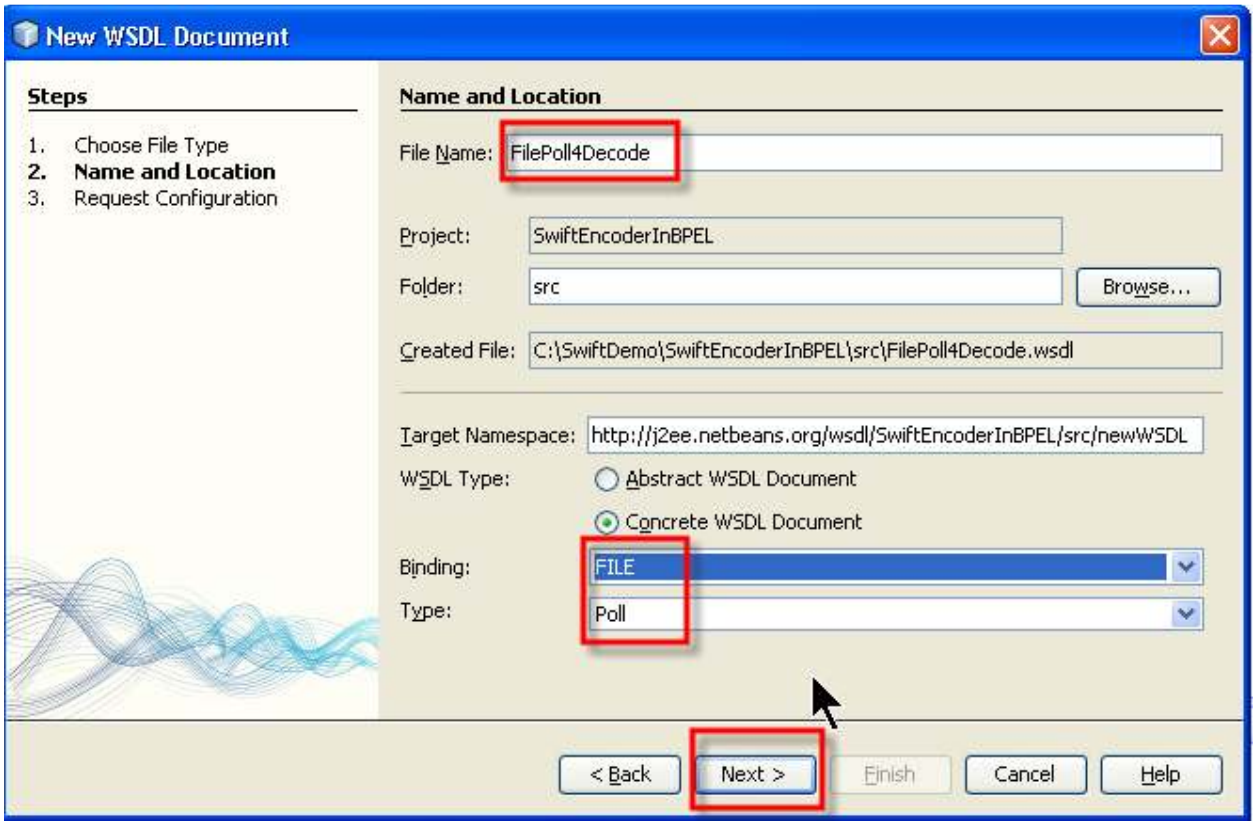
Name it as "SwiftEncoderInBPEL"



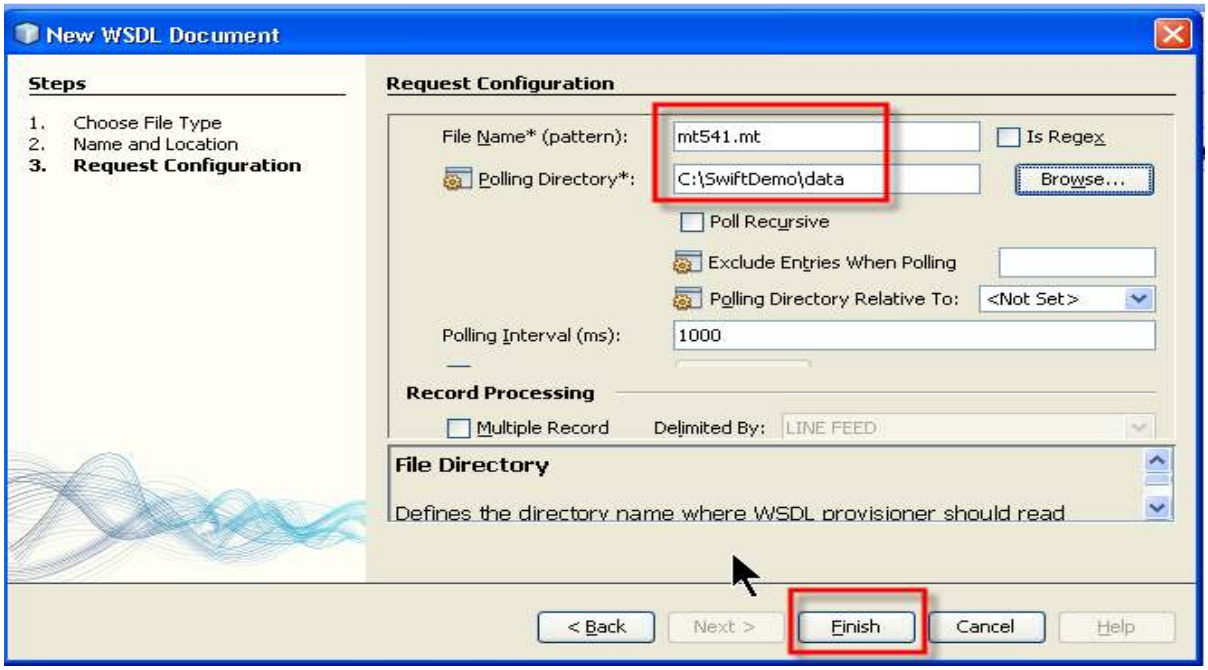
Create an WSDL …



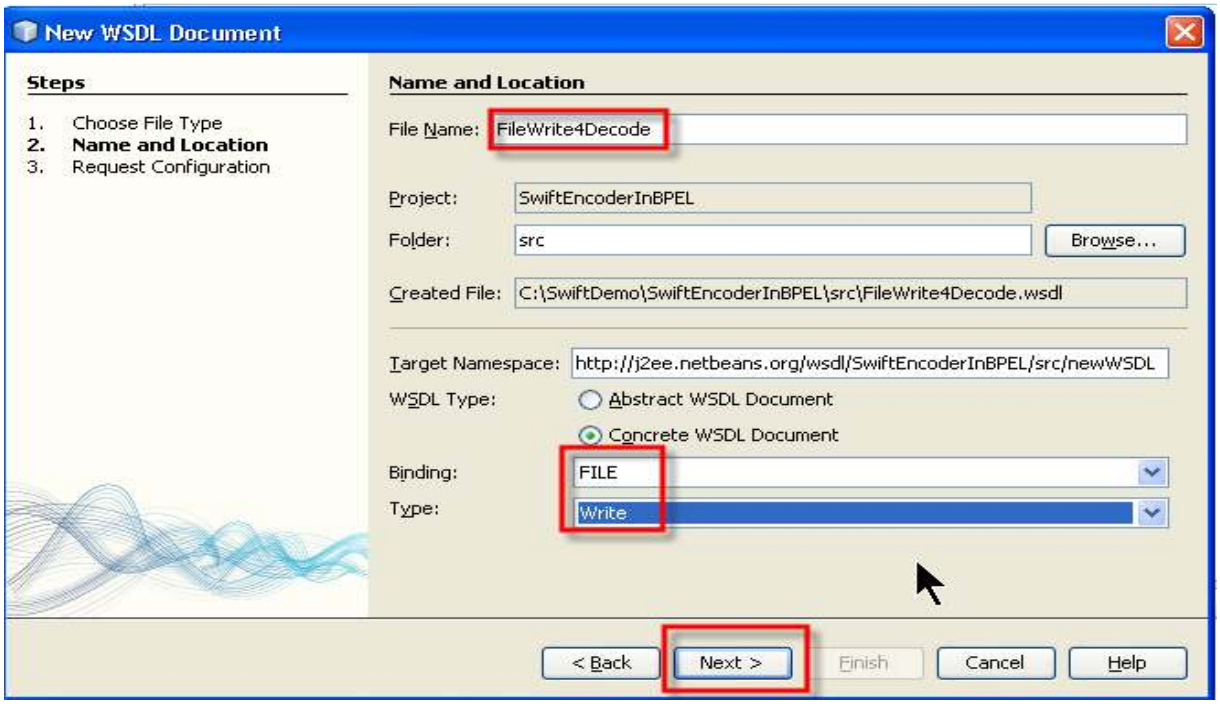
Name it as "FilePoll4Decode", select "FILE", "Poll" …



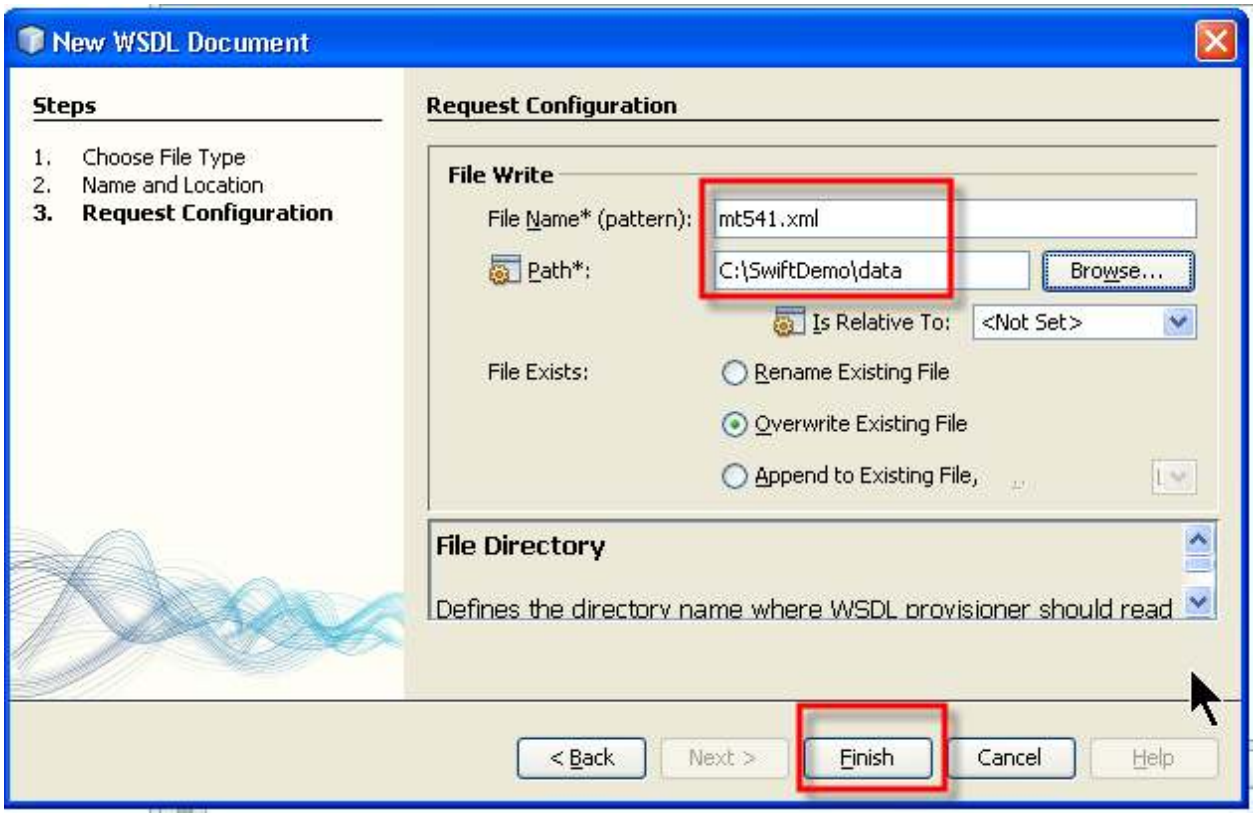
Specify MT file name (mt541.mt) and directory.



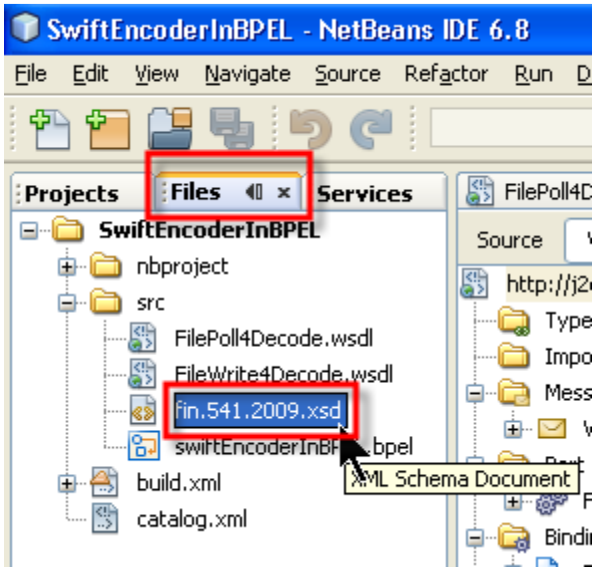
Create another WSDL for "FILE" … "Write", name it as "FileWrite4Decode".



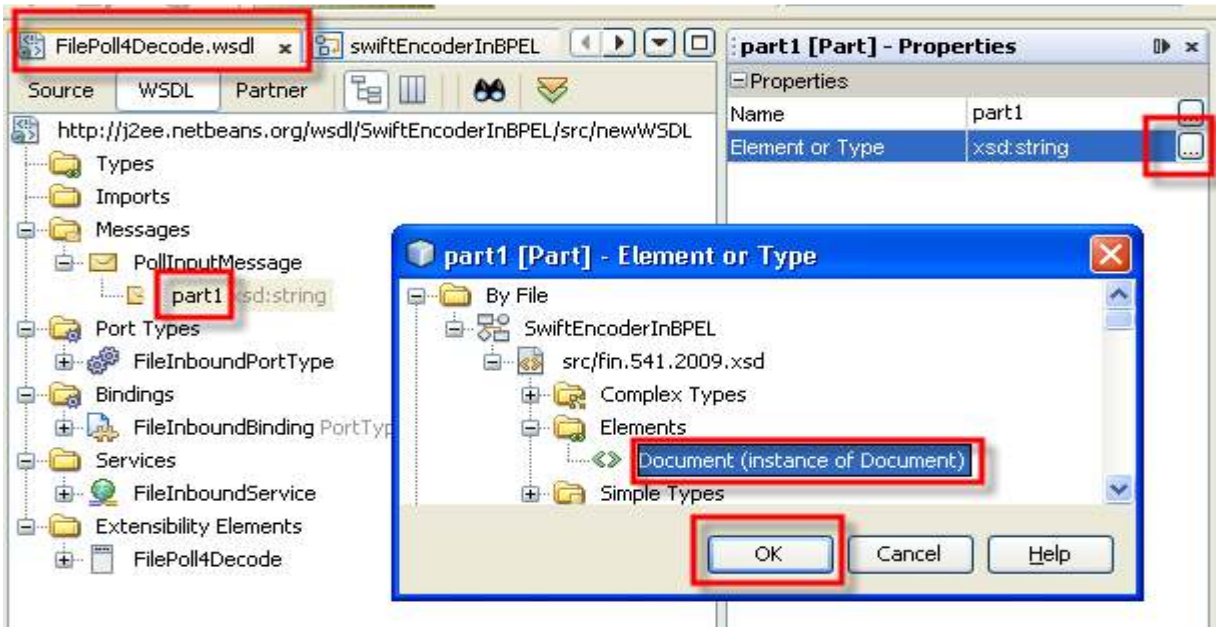
Specify XML file name (mt541.xml) and directory.



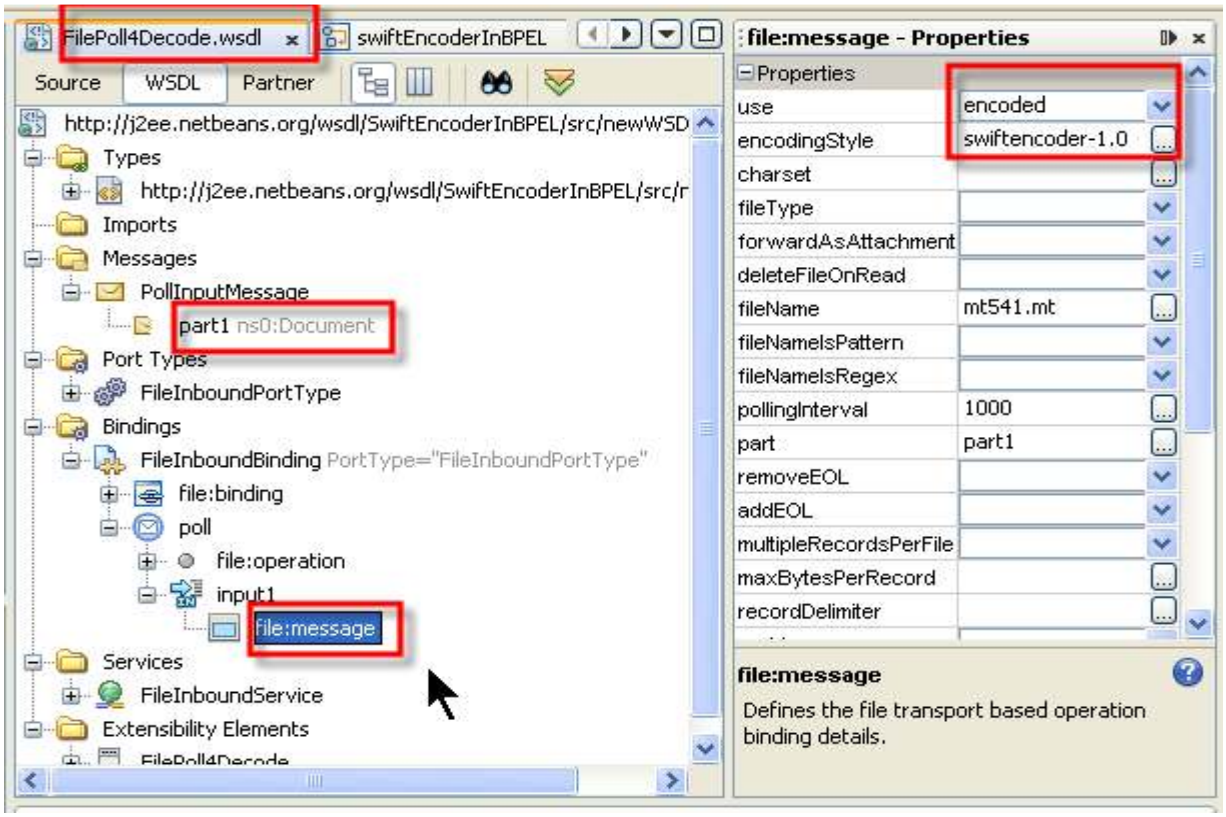
Copy the xsd schema "fin.541.2009.xsd" into project's src folder.



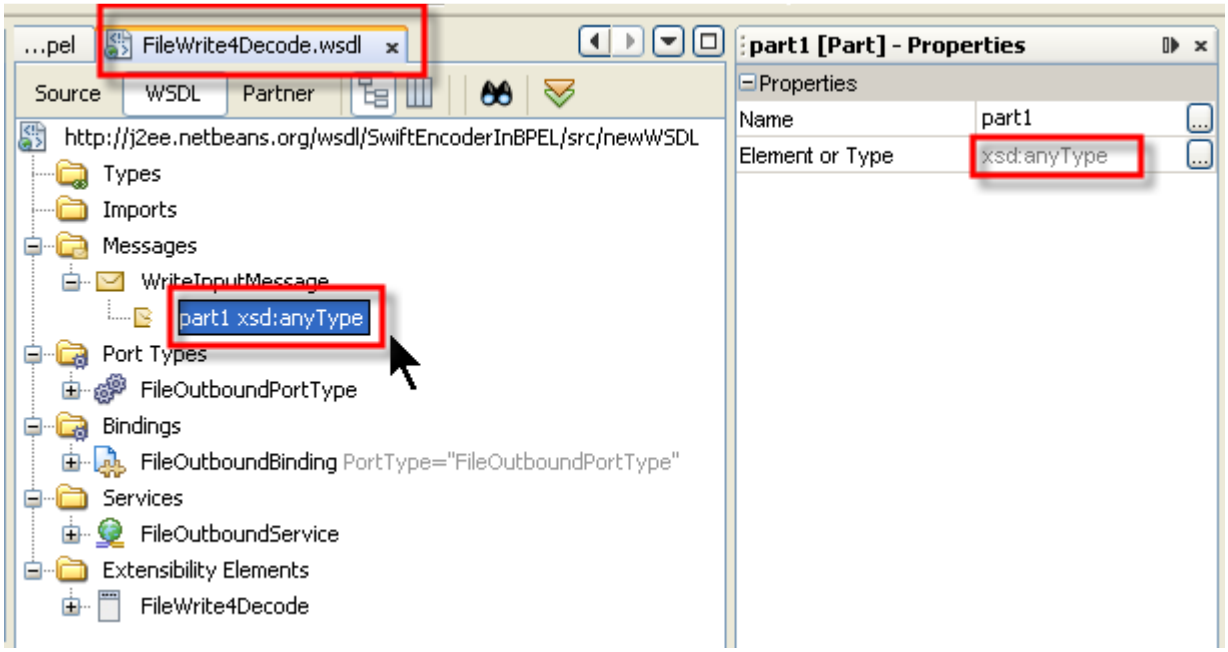
Configure WSDL "FilePoll4Decode" … for part1, change "Element or type" to element "Document" in "fin.541.2009.xsd"



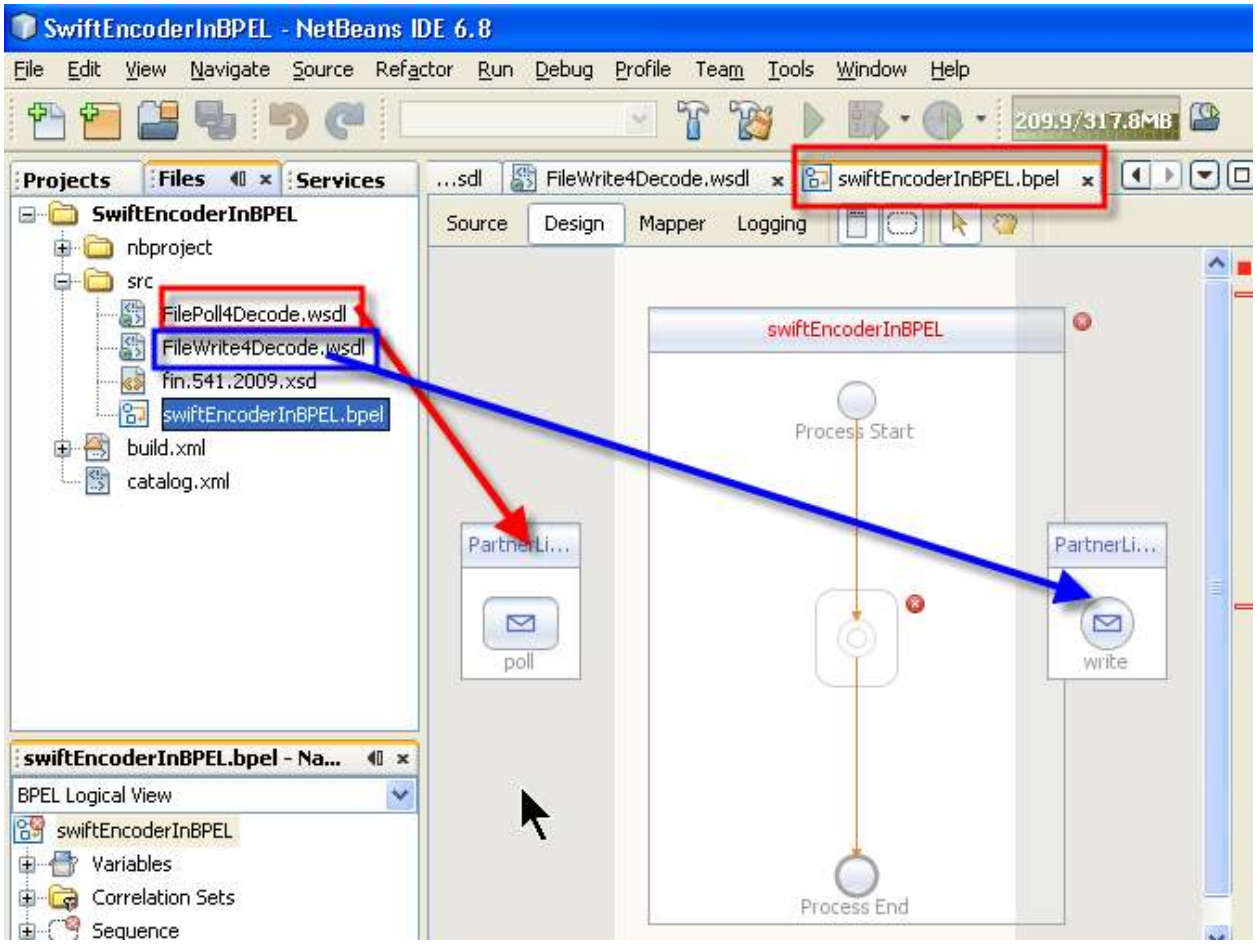
Configure use "Encoded" and encodingStyle "swiftencoder-1.0".



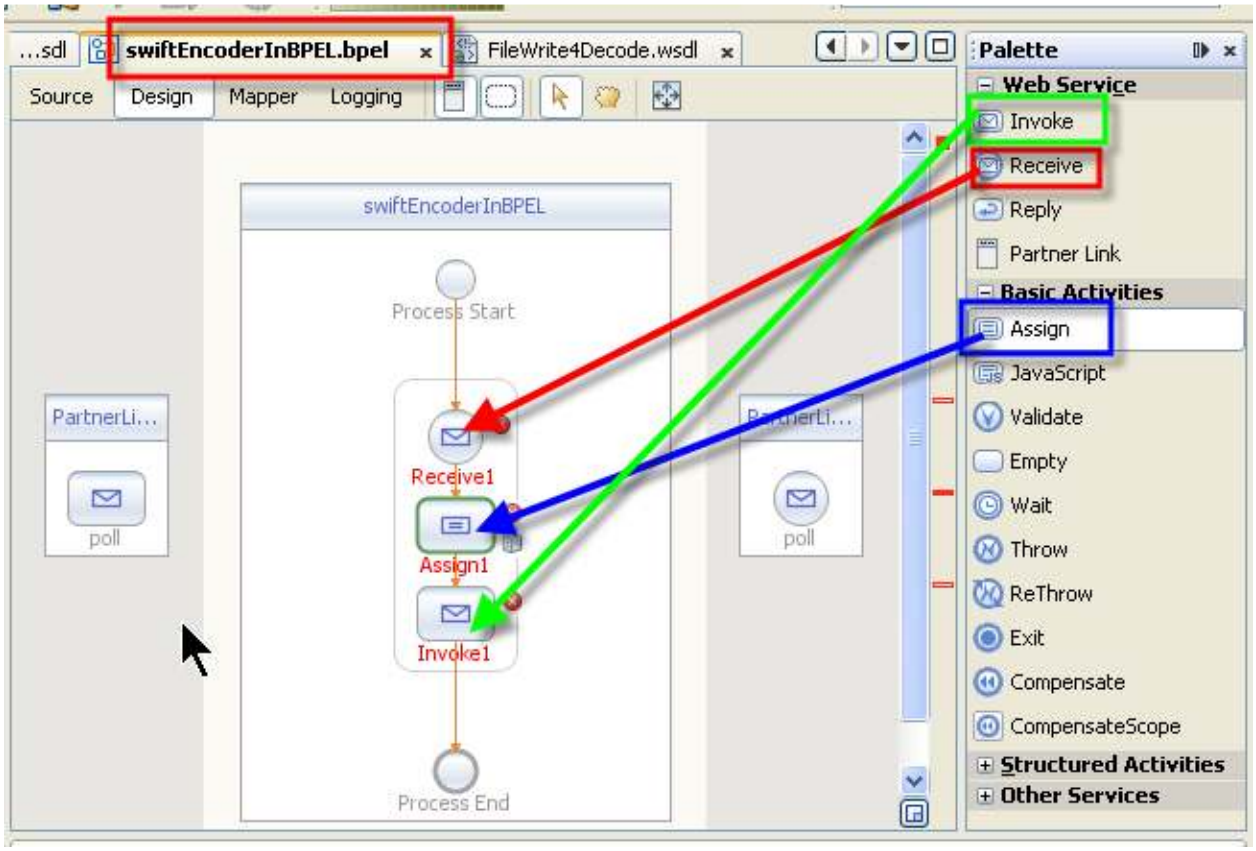
For WSDL “FileWrite4Decode”, change part1 type to “xsd:anyType”.



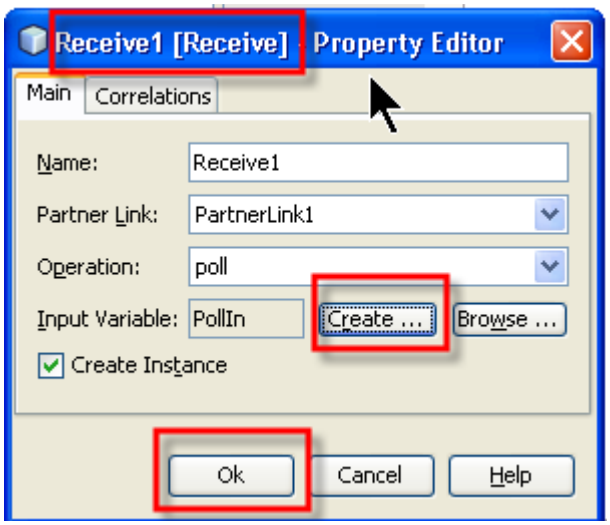
In BPEL editor, drag-n-drop two WSDL files into the panel as indicated.



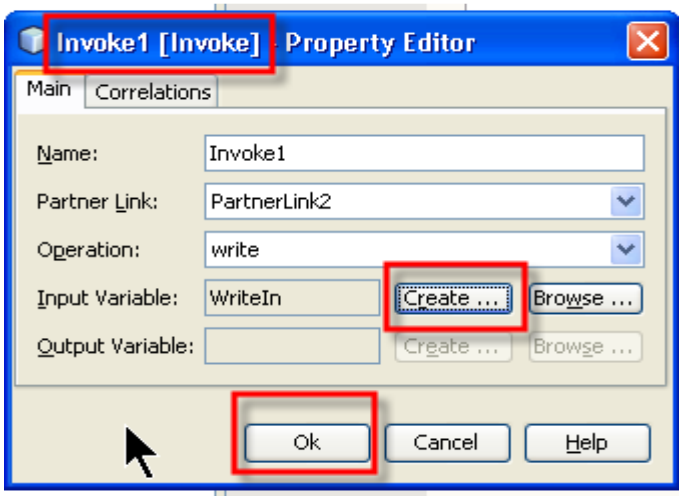
Orchestrate BPEL process: Receive➔Assign➔Invoke … by drag-n-drop as arrows indicated.



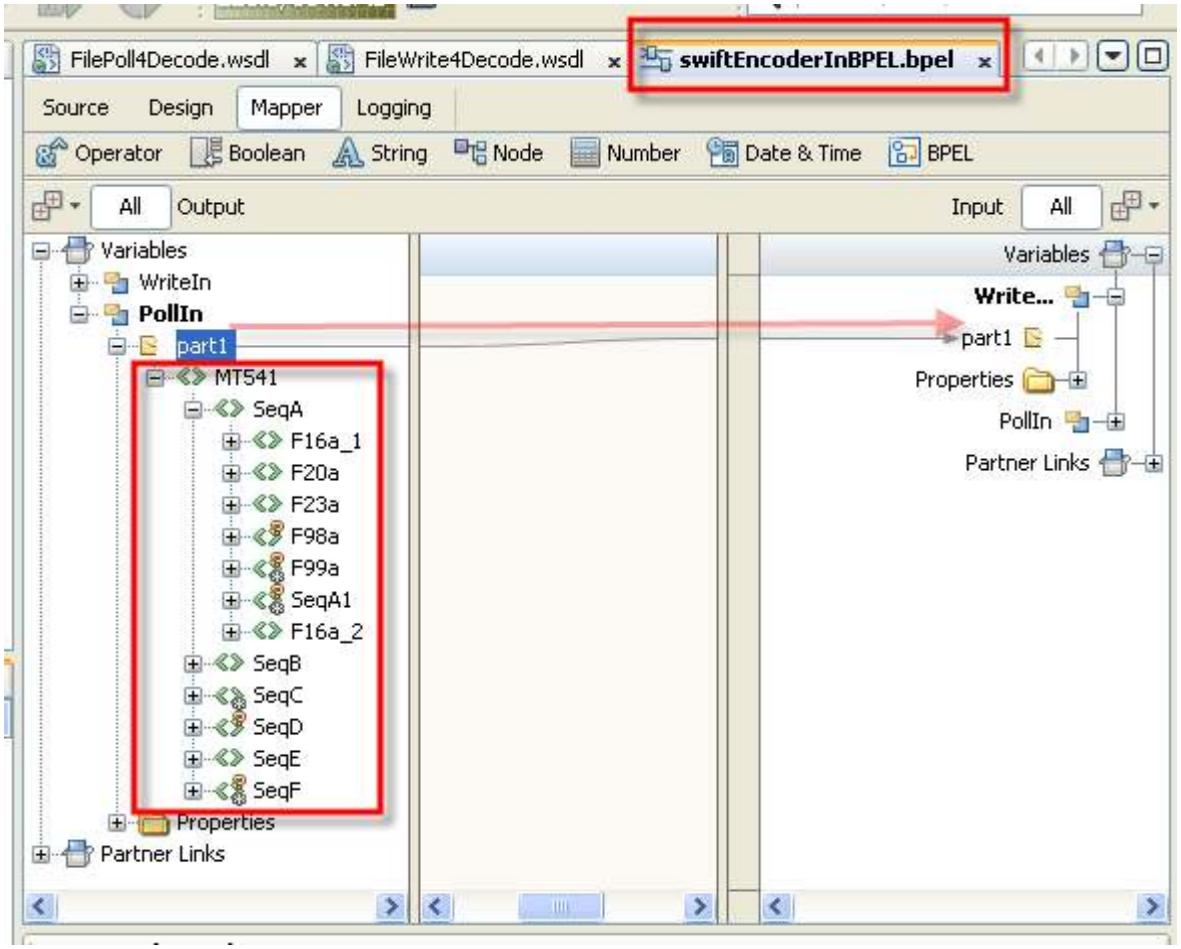
Configure "Receive1"



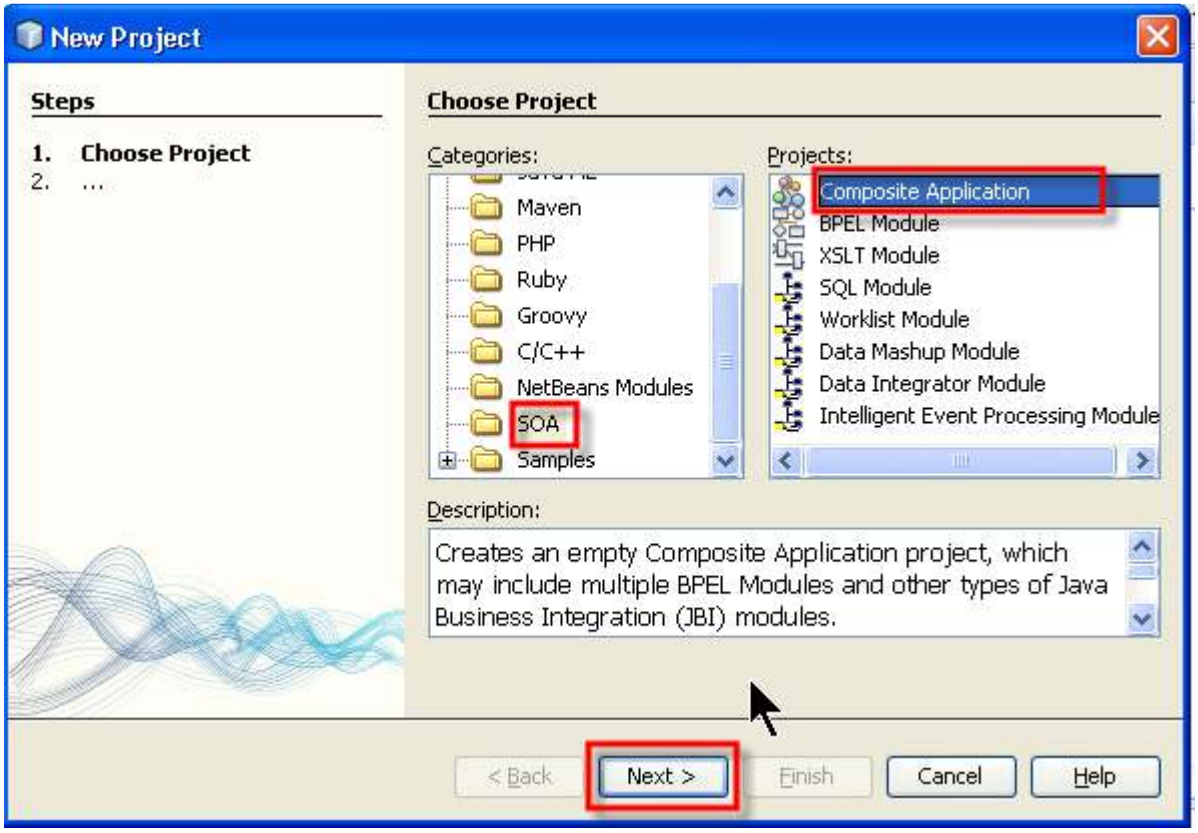
Configure “Invoke1”



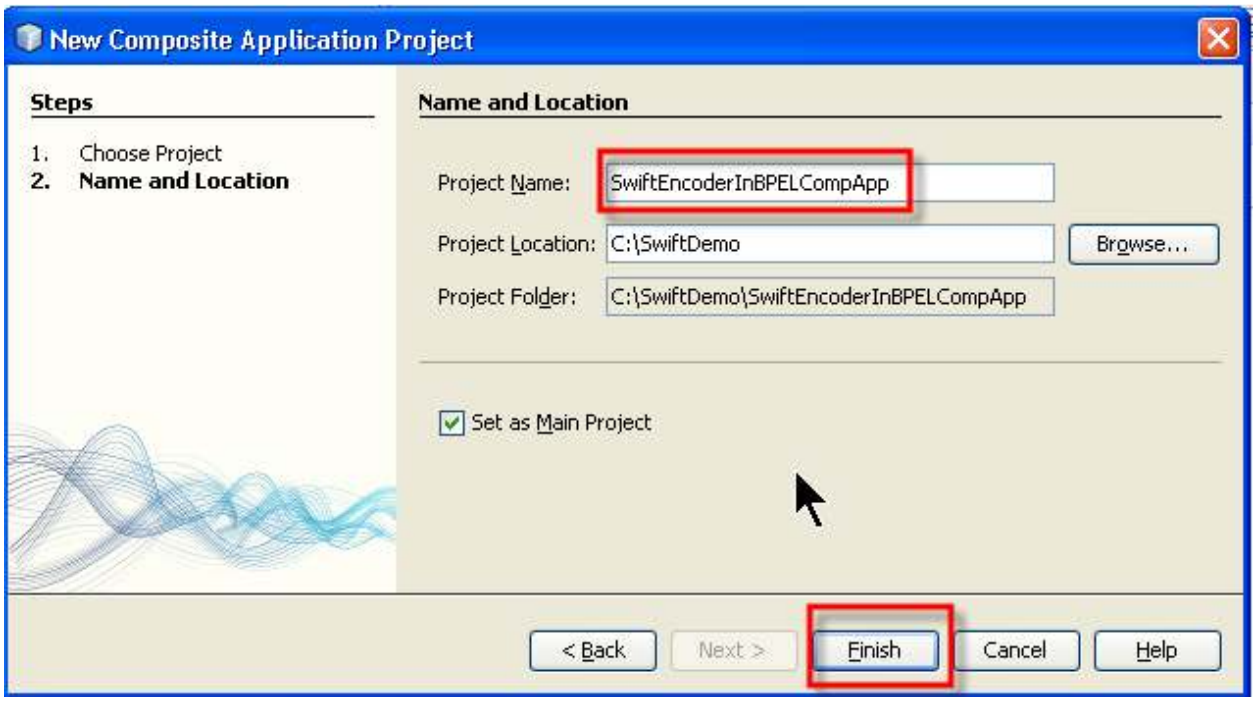
Open “Assign1” map editor. Map PollIn/part1➔Write/part1 as indicated. Note you see the MT541 message structure on left hand side.



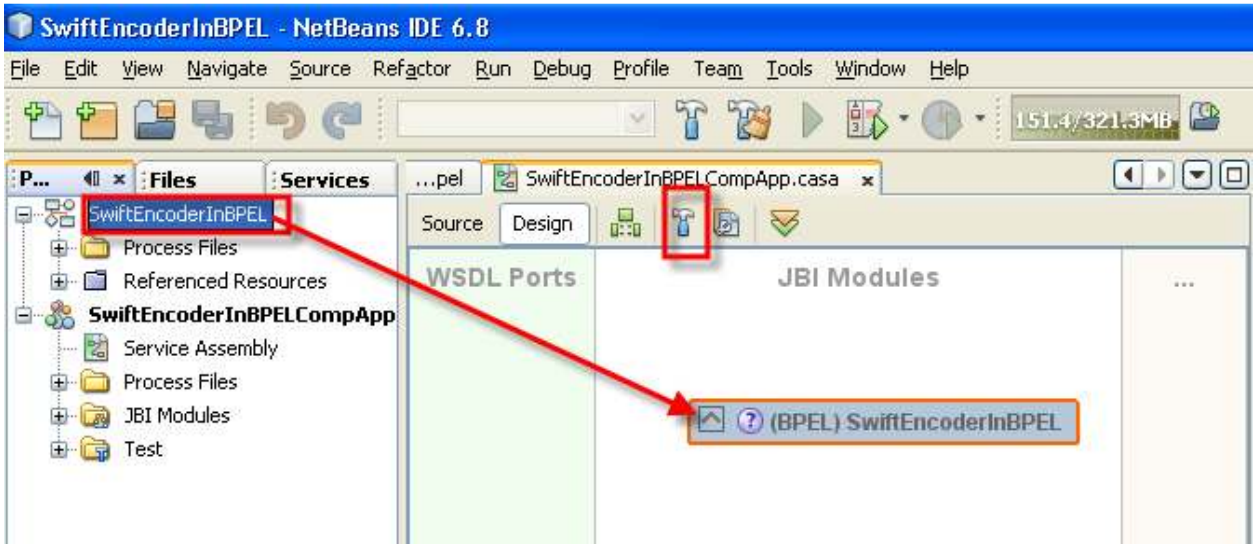
Create a project SOA➔Composite Application



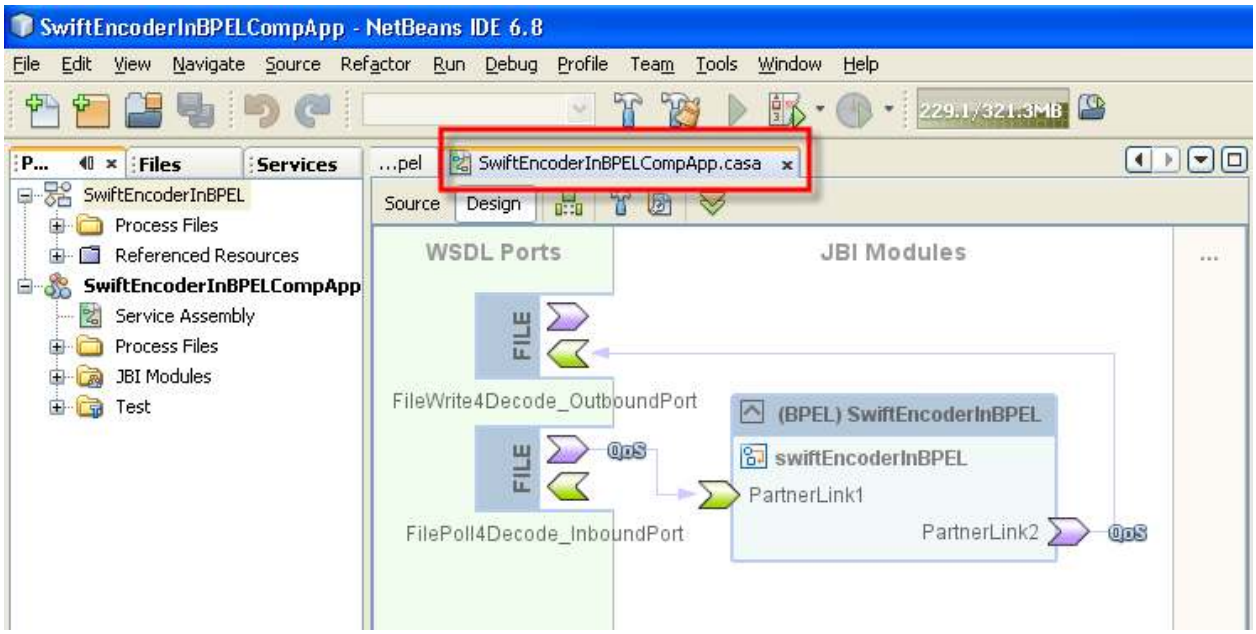
Give name “SwiftEncoderInBPELCompApp”.



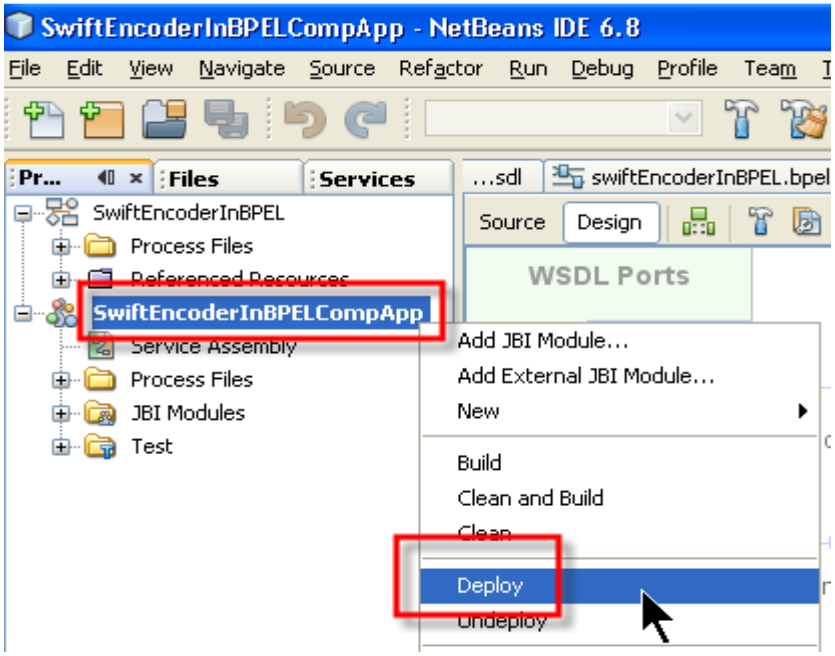
Drag-n-drop project "SwiftEncodeInBPEL" into casa panel. Click build icon.



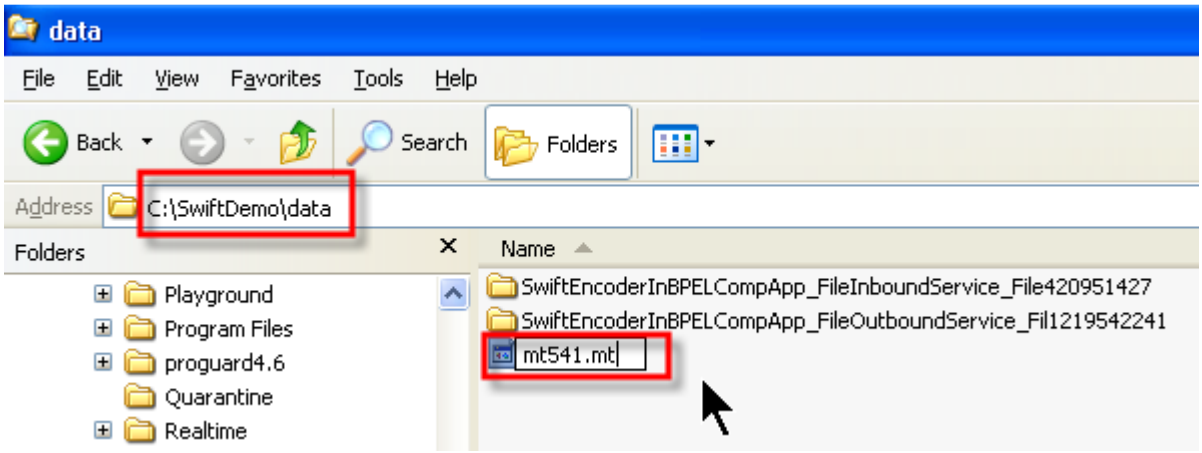
Panel will be refreshed as below.



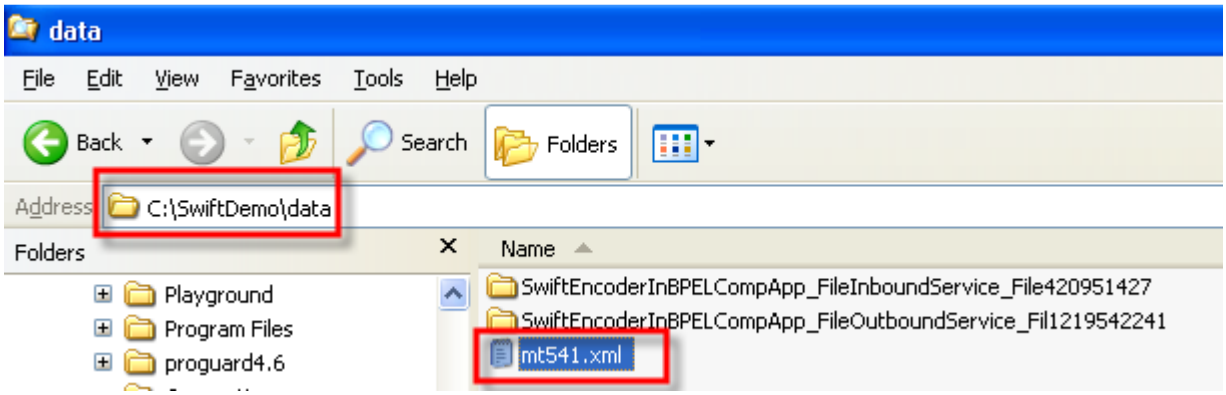
Deploy the composite app "SwiftEncoderInBPELCompApp".



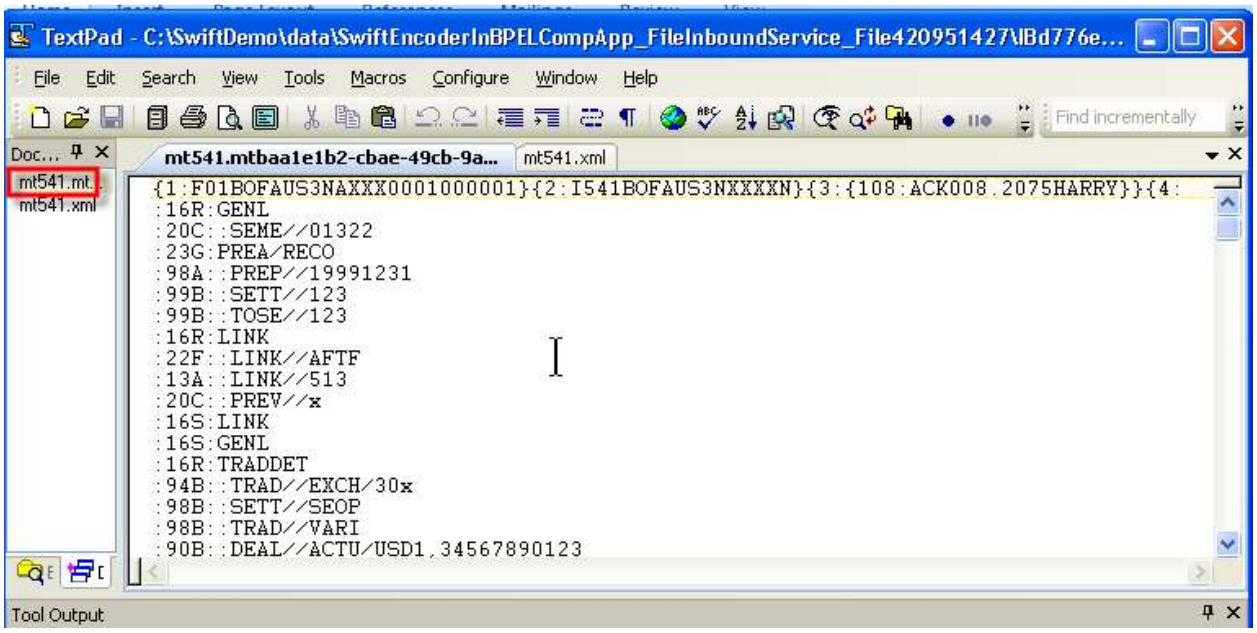
Feed in the input MT file as configured "C:\SwiftDemo\data\mt541.mt".



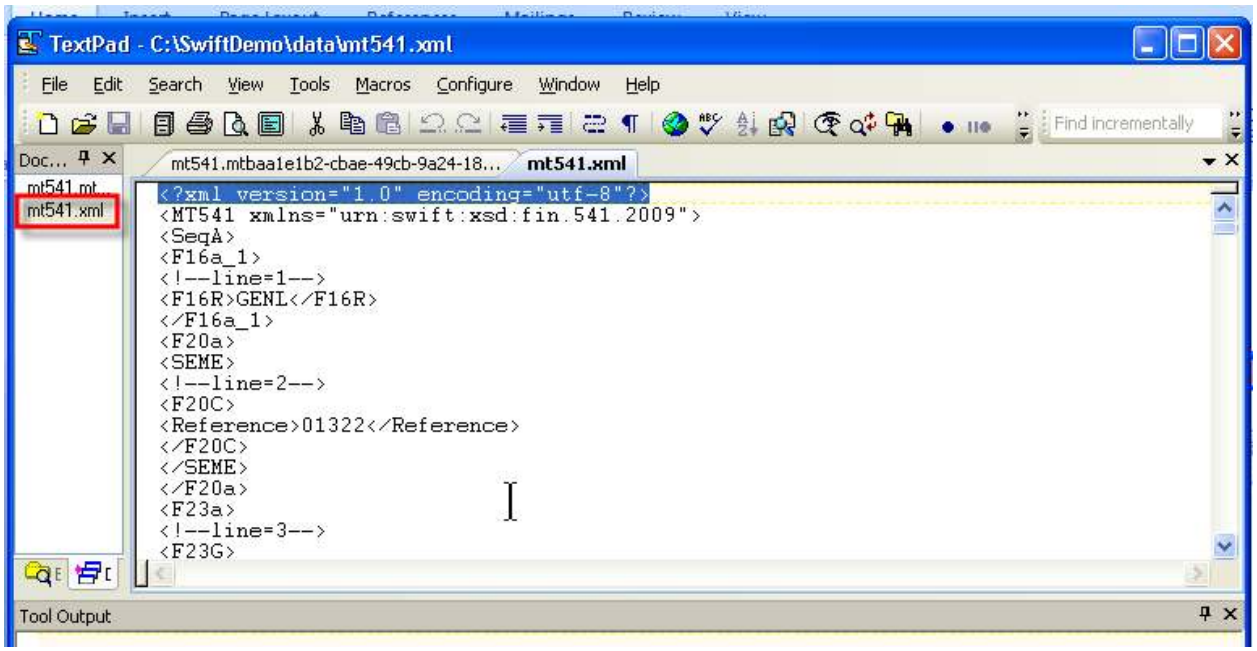
The input file disappears and output XML "mt541.xml" appears.



Take a look at the input MT file “mt541.mt”.



Also take a look the output XML file “mt541.xml”.



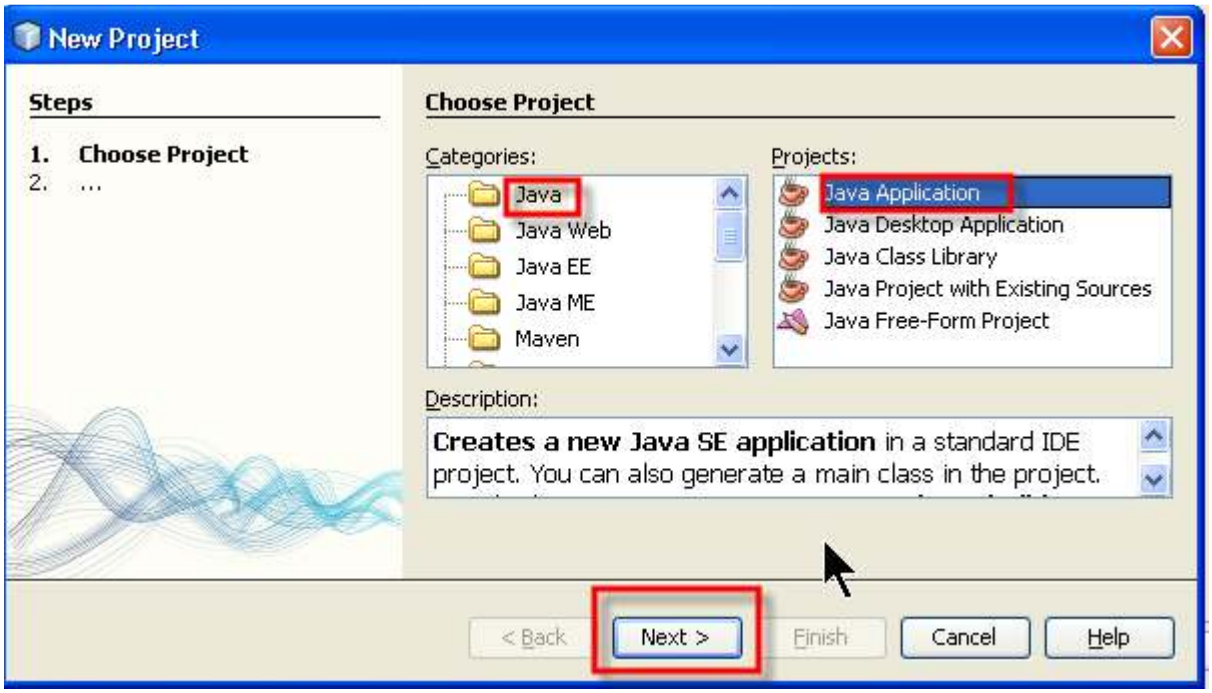
3. Using SWIFT Encoder in POJO

Demonstrate how to use SWIFT Encoder in POJO SE (Service Engine) by use of a simple example.

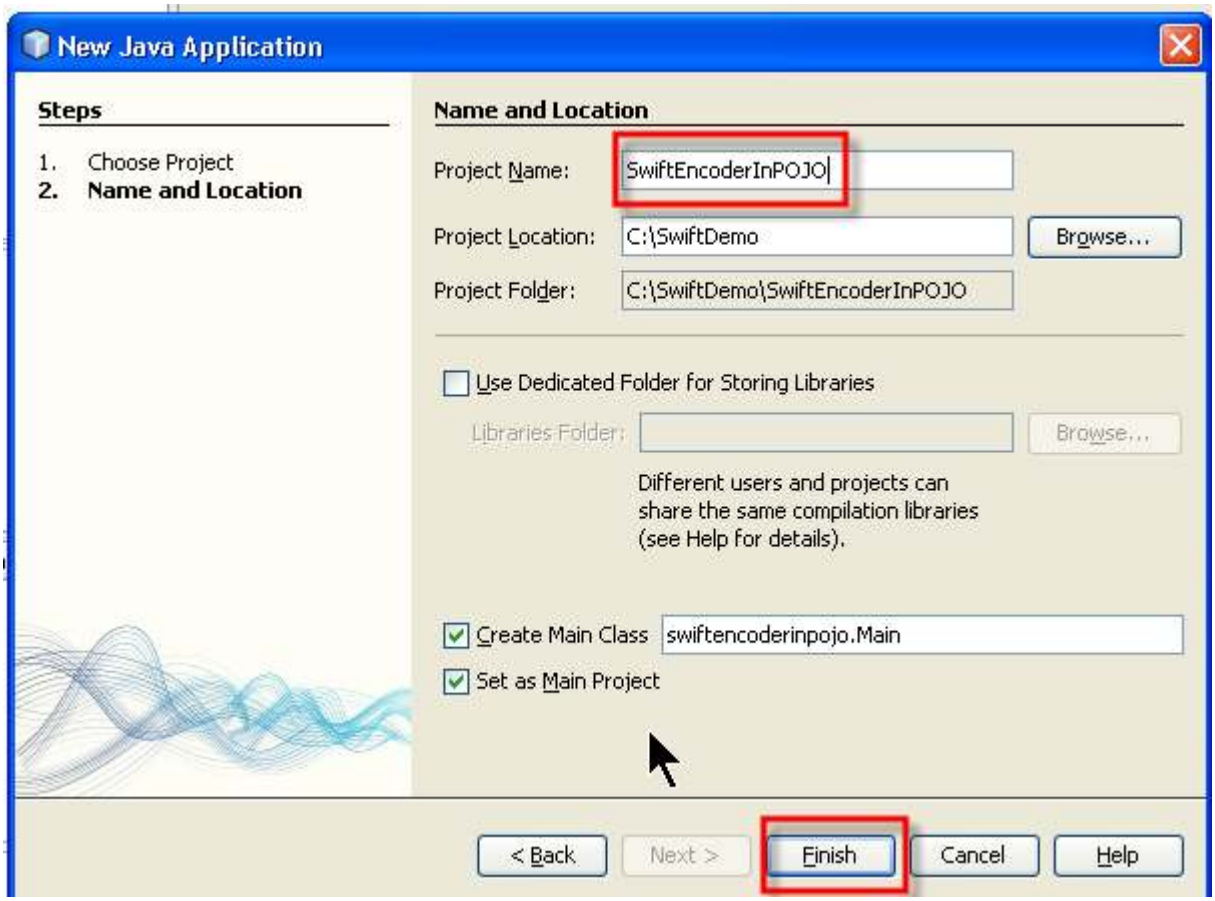
Here is a screen cast:

http://swiftencoder.appspot.com/SwiftEncoderInPOJO.html

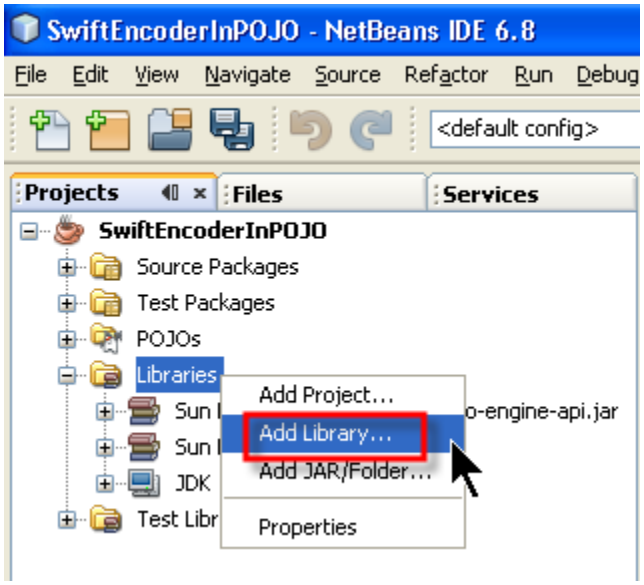
Create a project Java➔Java Application



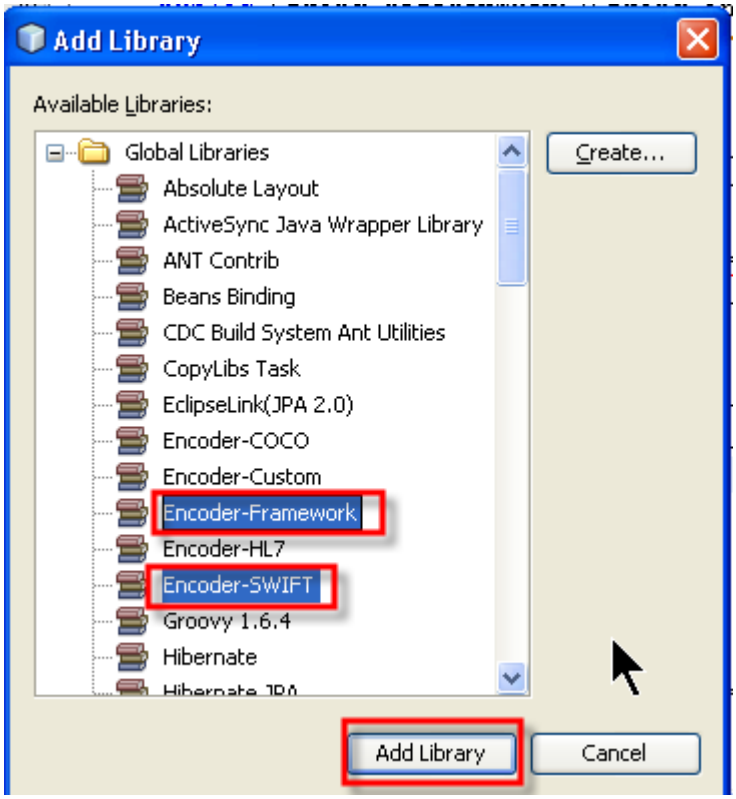
Give name "SwiftEncoderInPOJO"



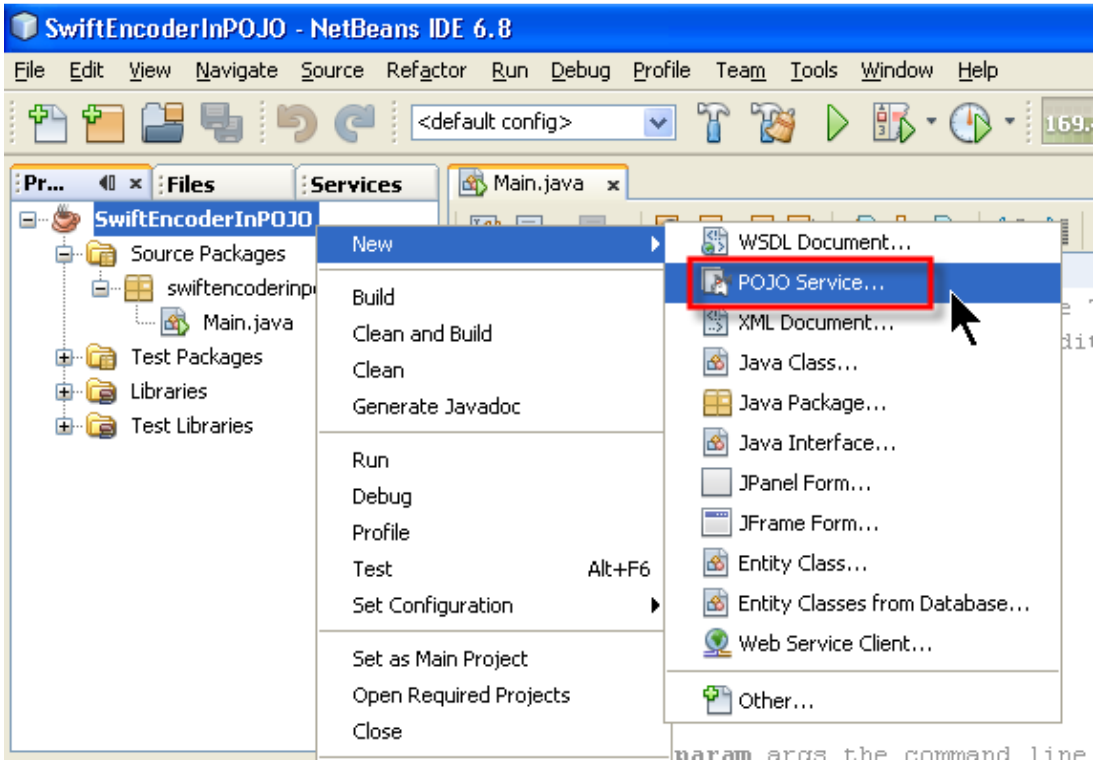
Add Library by right-click on project node “Libraries”.



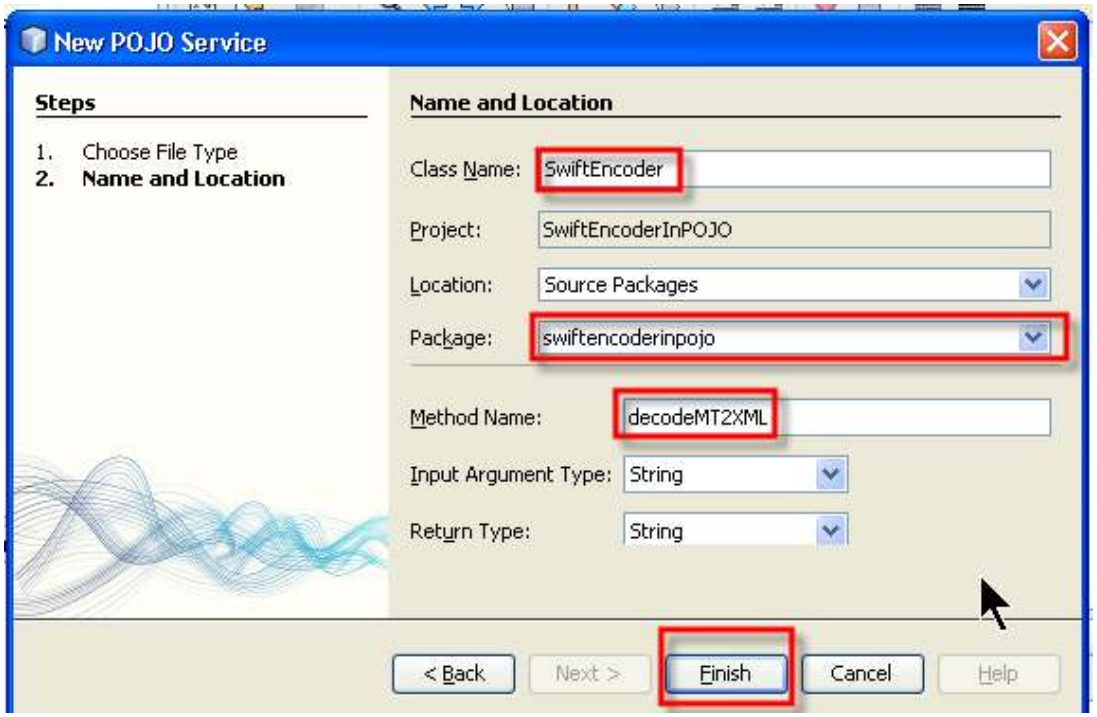
Choose libraries “Encoder-Framework” and “Encoder-SWIFT”.



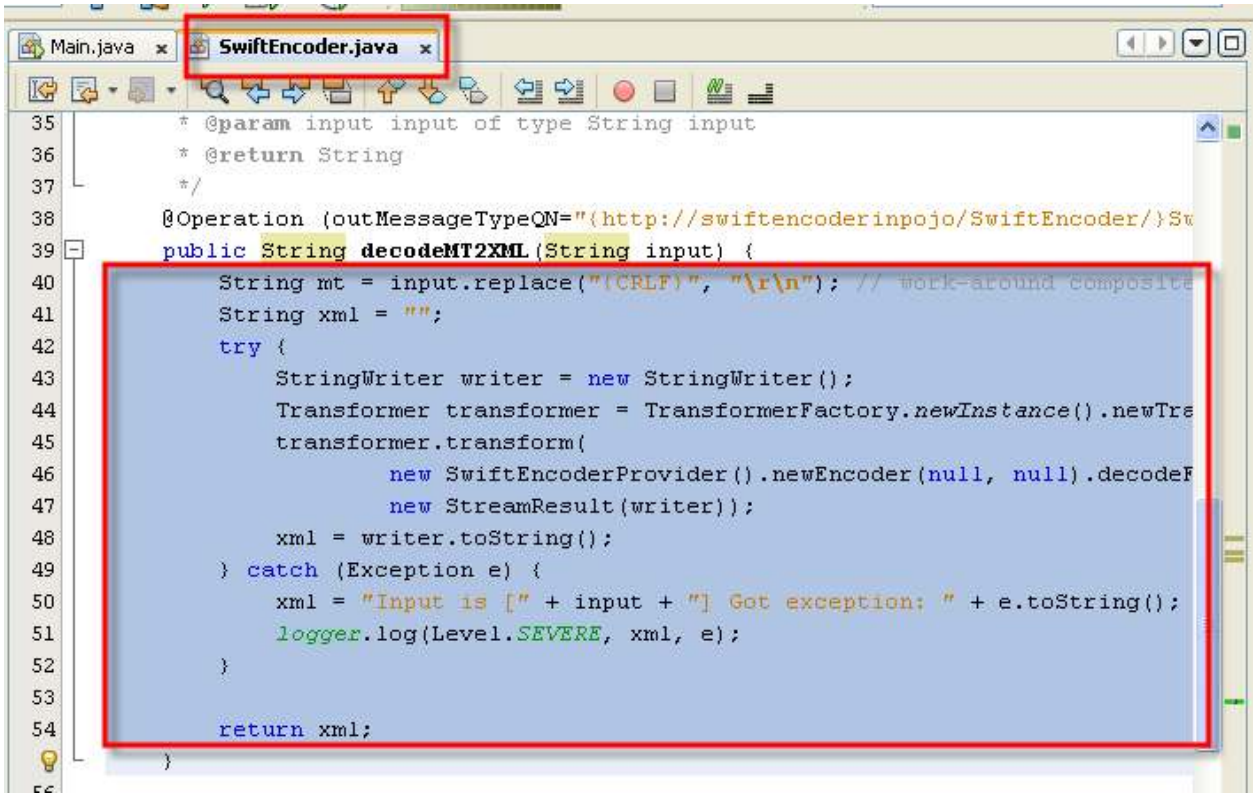
Right-click “SwiftEncoderInPOJO”, choose “New”, “POJO Service …”.



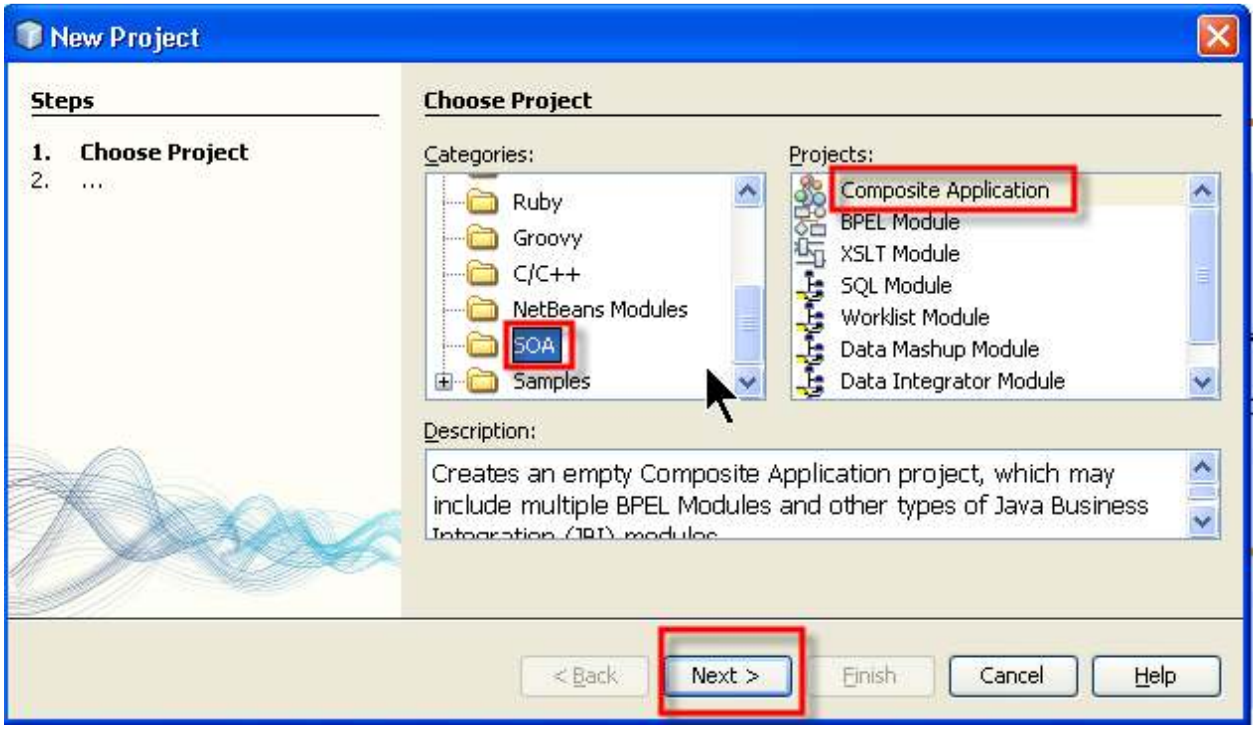
Give name “SwiftEncoder”, package “swiftencoderpojo”, method name “decodeMT2XML” …



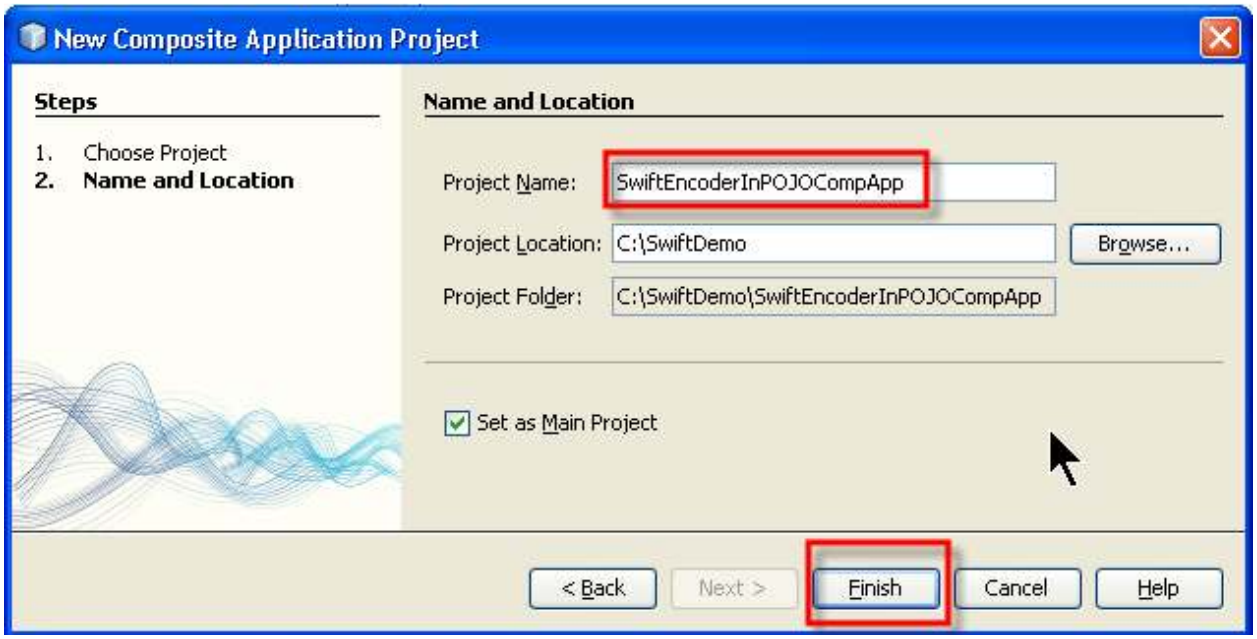
Write the code for method “decodeMT2XML” …



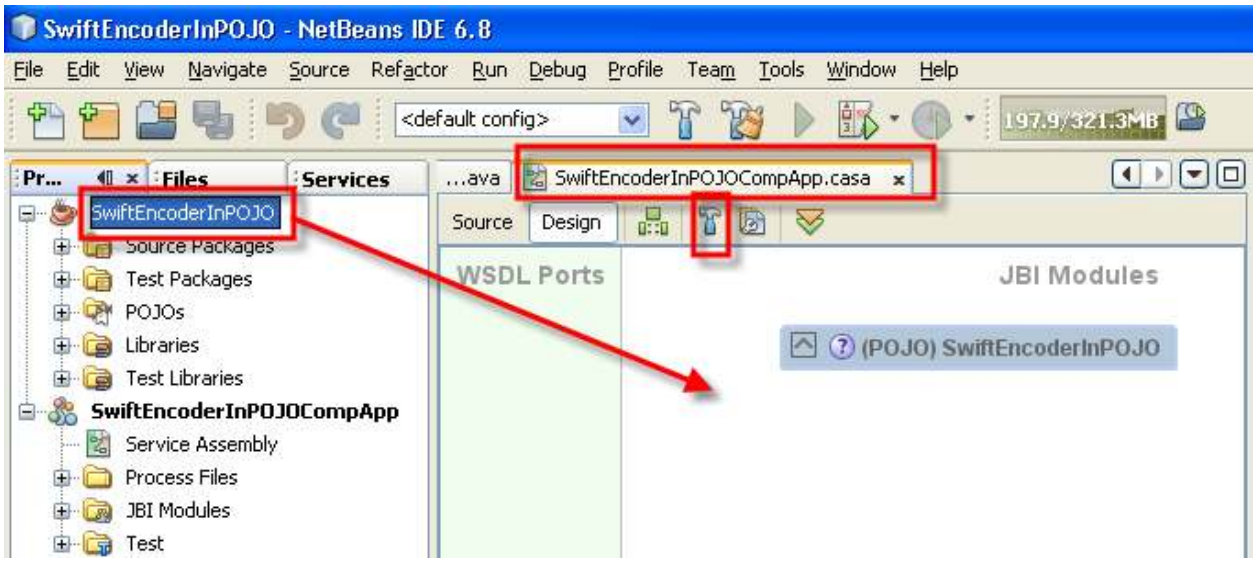
Create a project SOA➔Composite Application



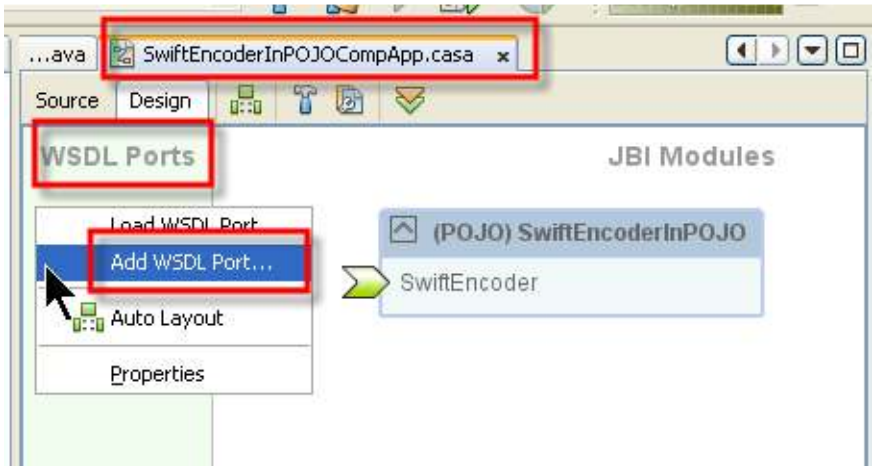
Give project name “SwiftEncoderInPOJOCompApp”.



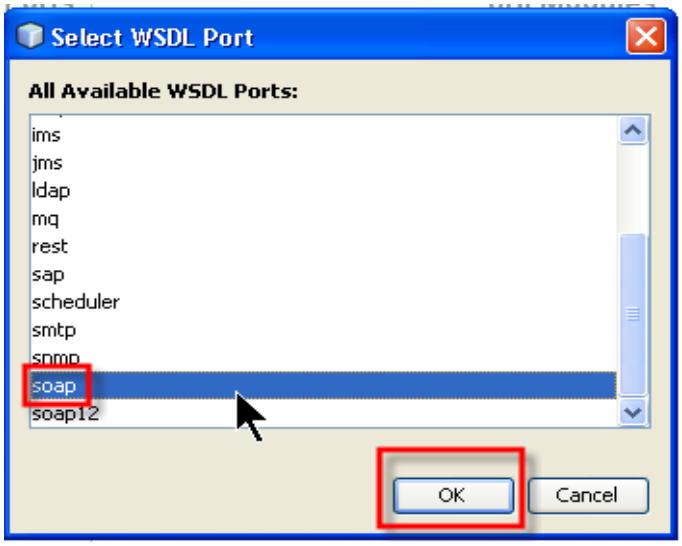
Drag-n-drop “SwiftEncoderPOJO” into casa panel. Then click “Build” button.



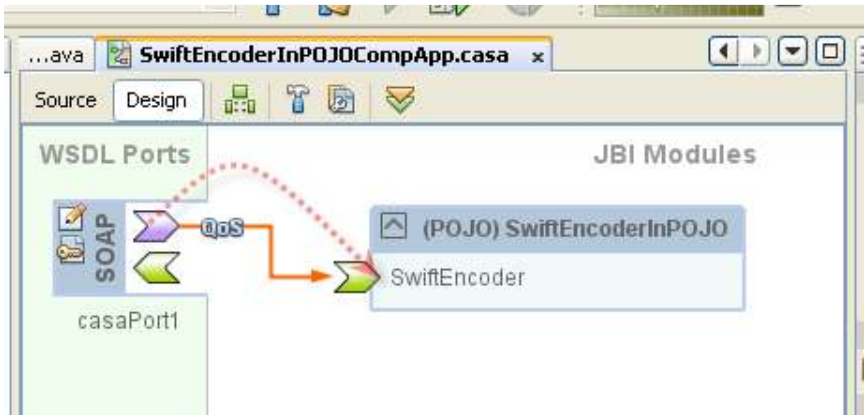
Right-click on the left hand side of casa panel, select "Add WSDL Port" from the pop-up menu.



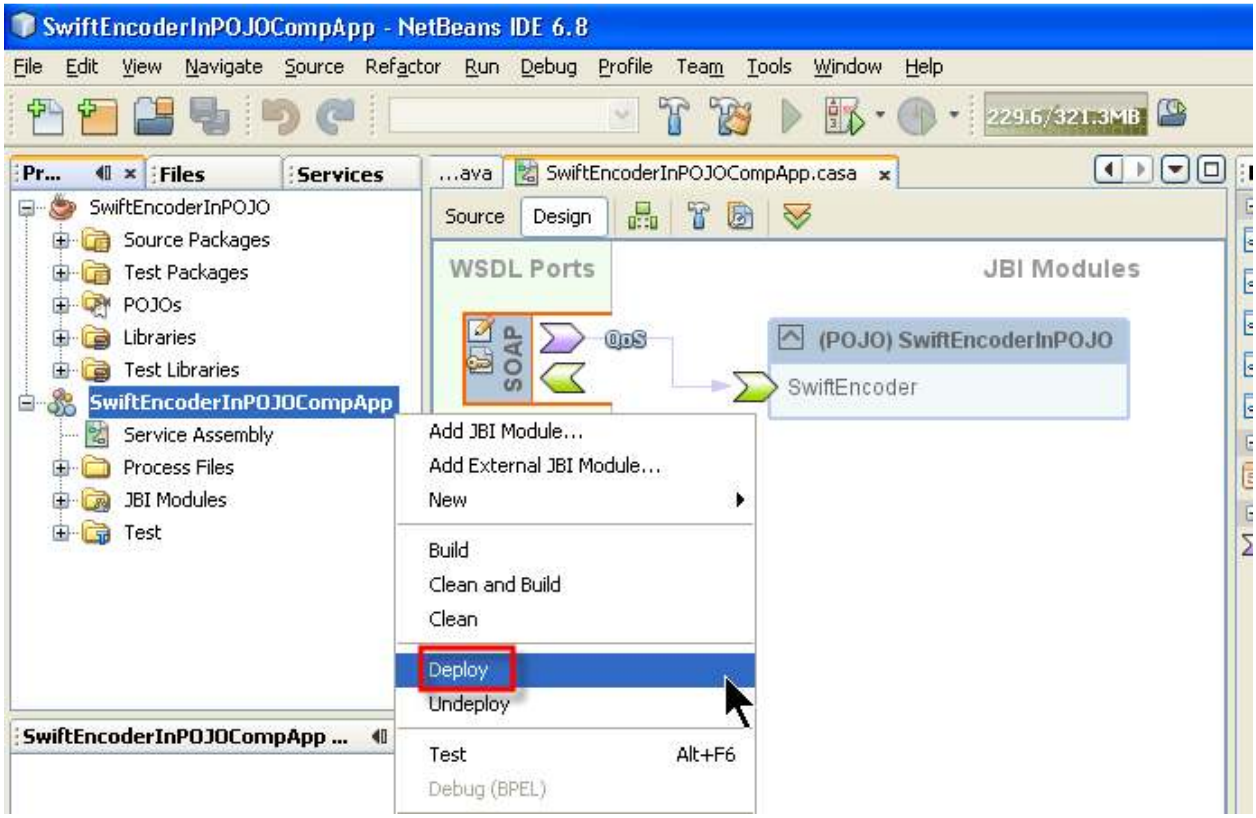
Choose soap WSDL port.



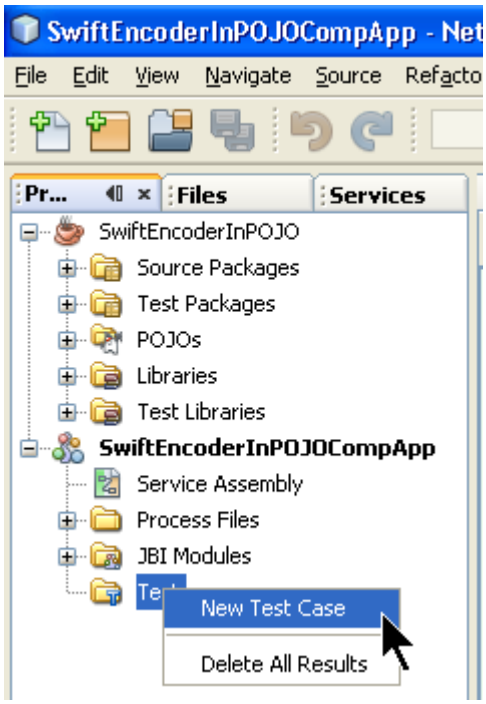
Connect soap port to POJO in casa panel.



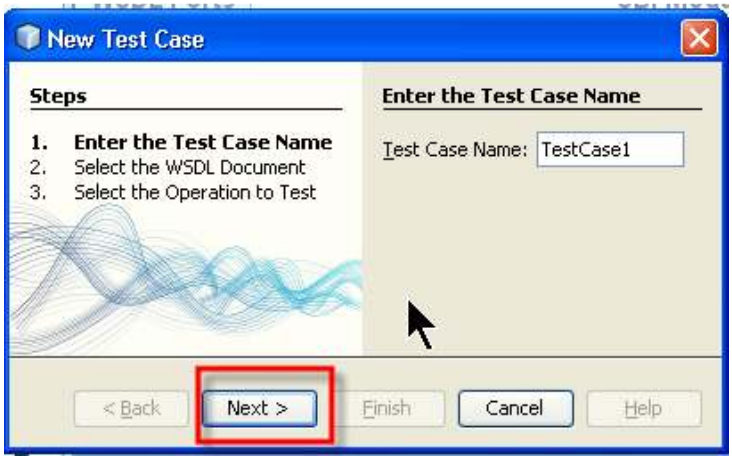
Save all. Deploy composite app “SwiftEncoderInPOJOCompApp”.



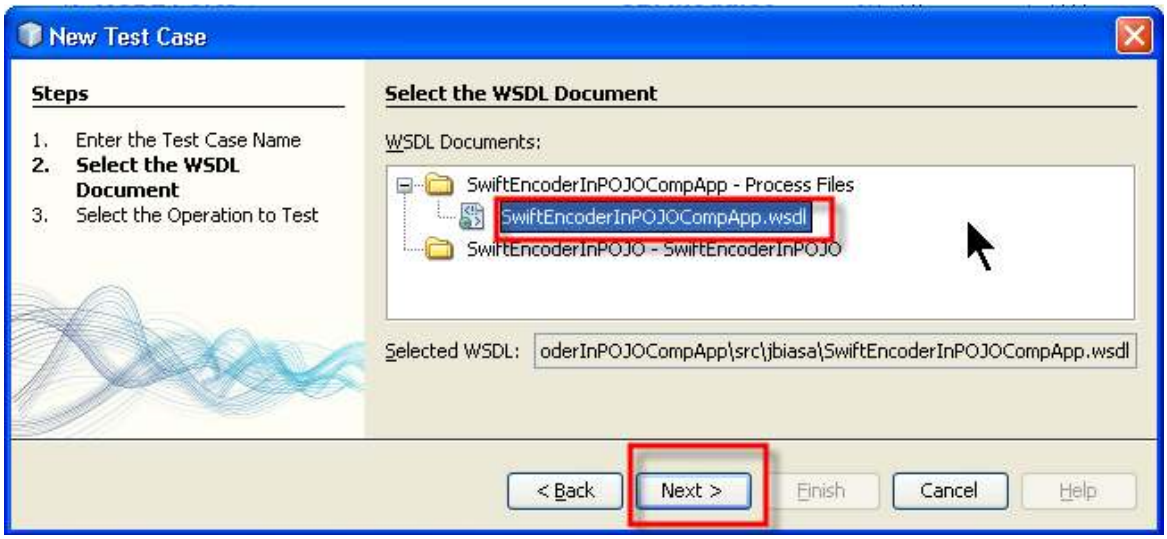
Create Test case in composite app “SwiftEncoderInPOJOCompApp”



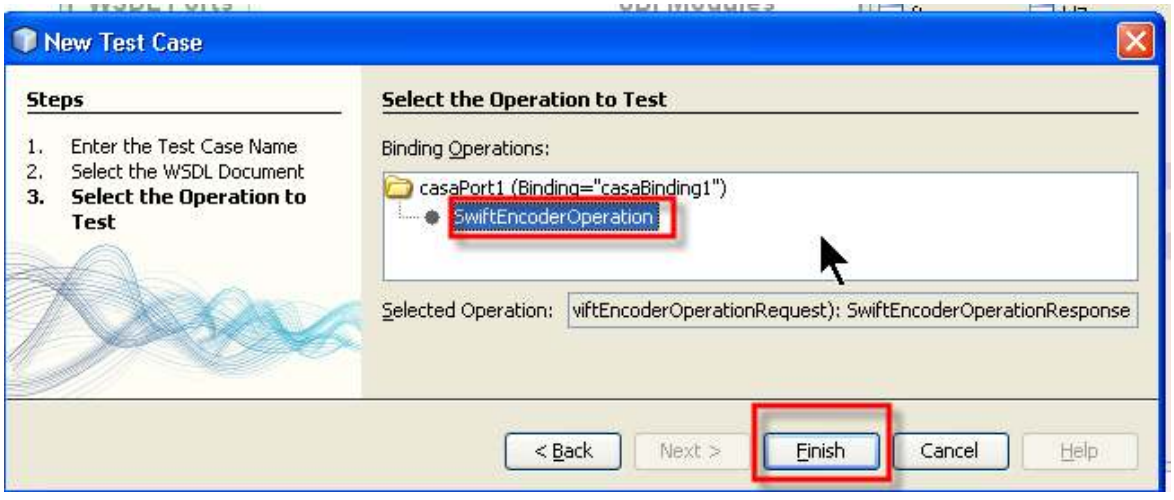
TestCase1 …



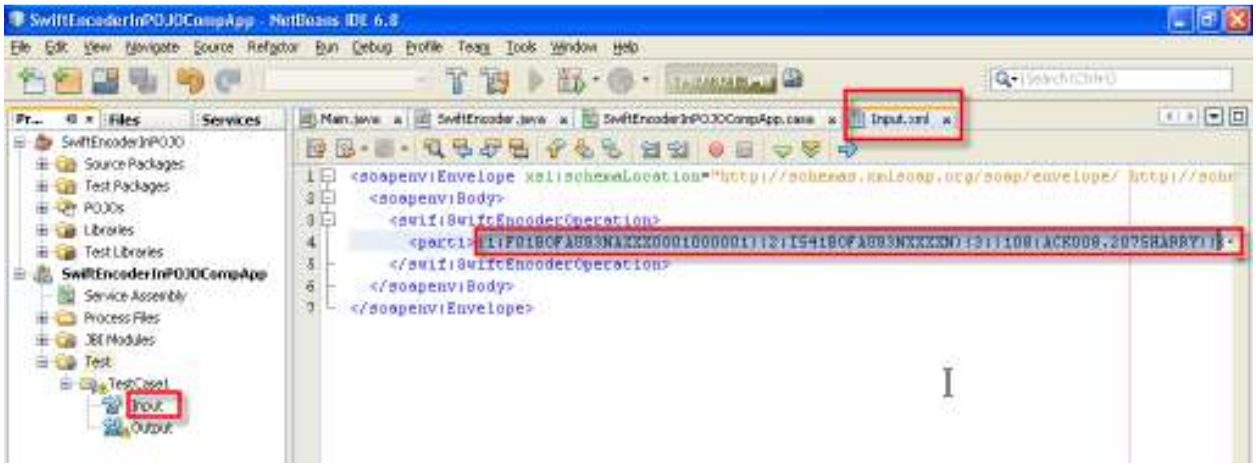
Choose WSDL …



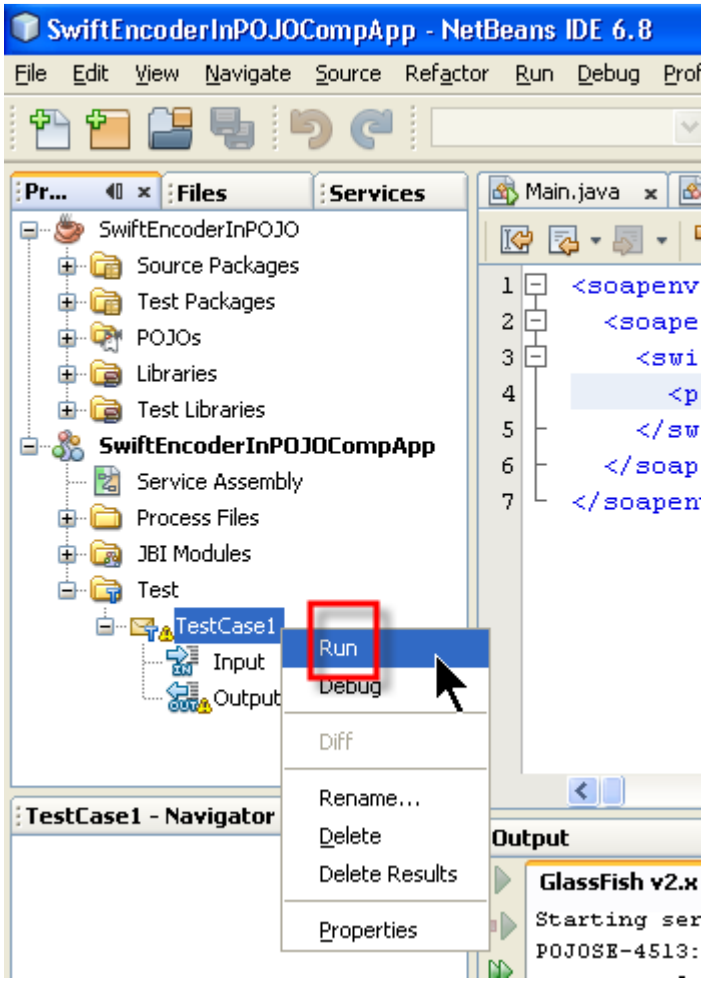
Choose operation.



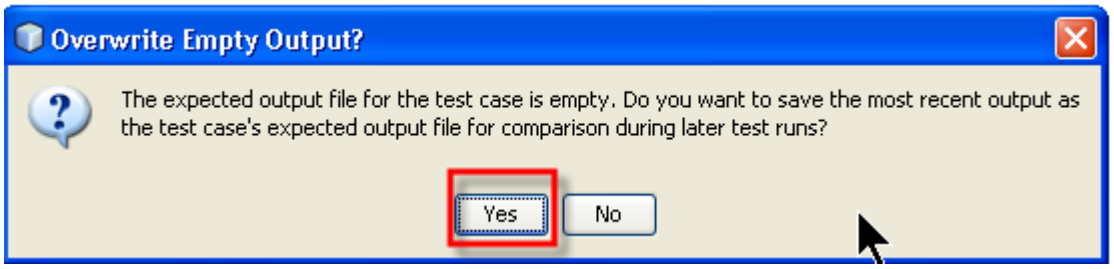
Copy-n-paste the input MT data.



Run the test.

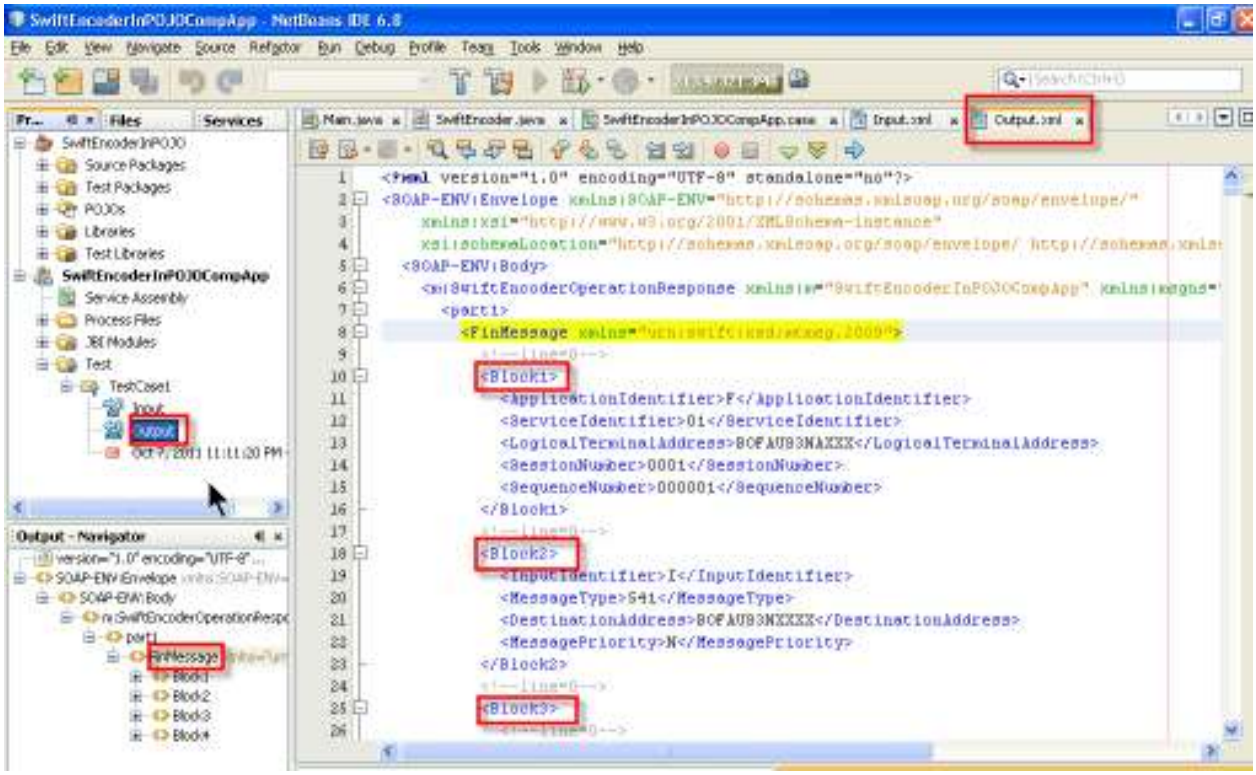


Answer yes because it's the first time run.



Open the output (the test result), you can see the XML.

Note you'll be seeing the entire XML document including block1-block5.



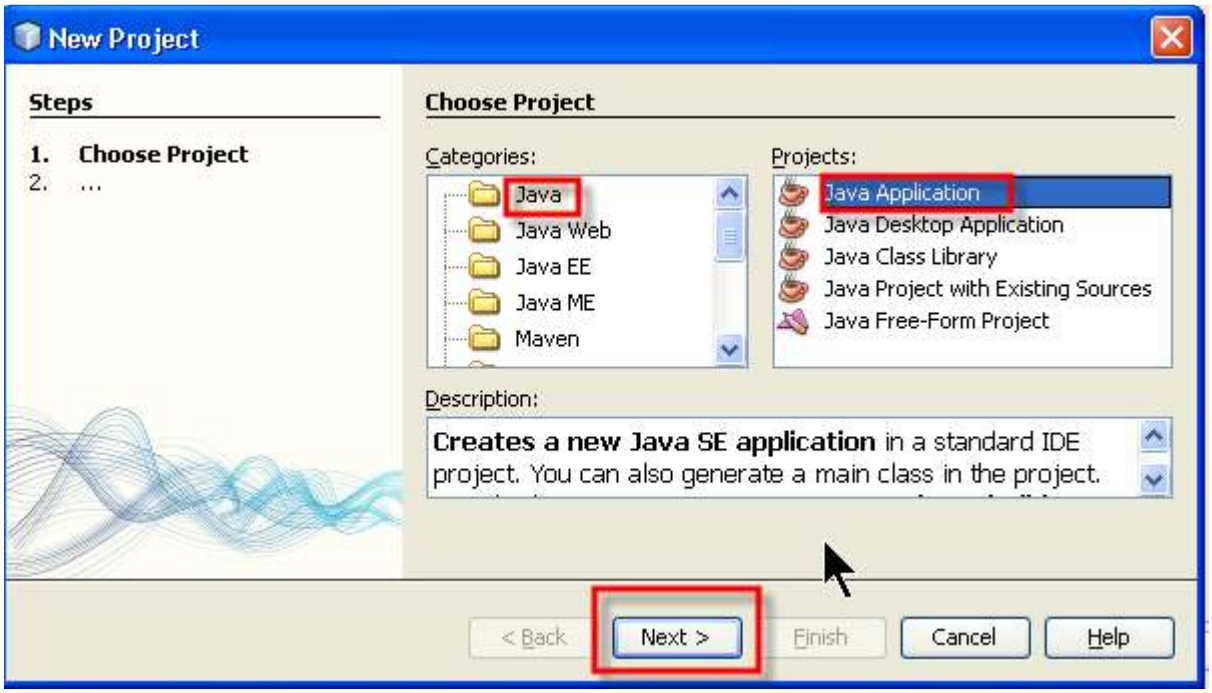
4. Using SWIFT Encoder in Java SE

Demonstrate how to use SWIFT Encoder in Java SE (Standard Edition) by use of a simple example.

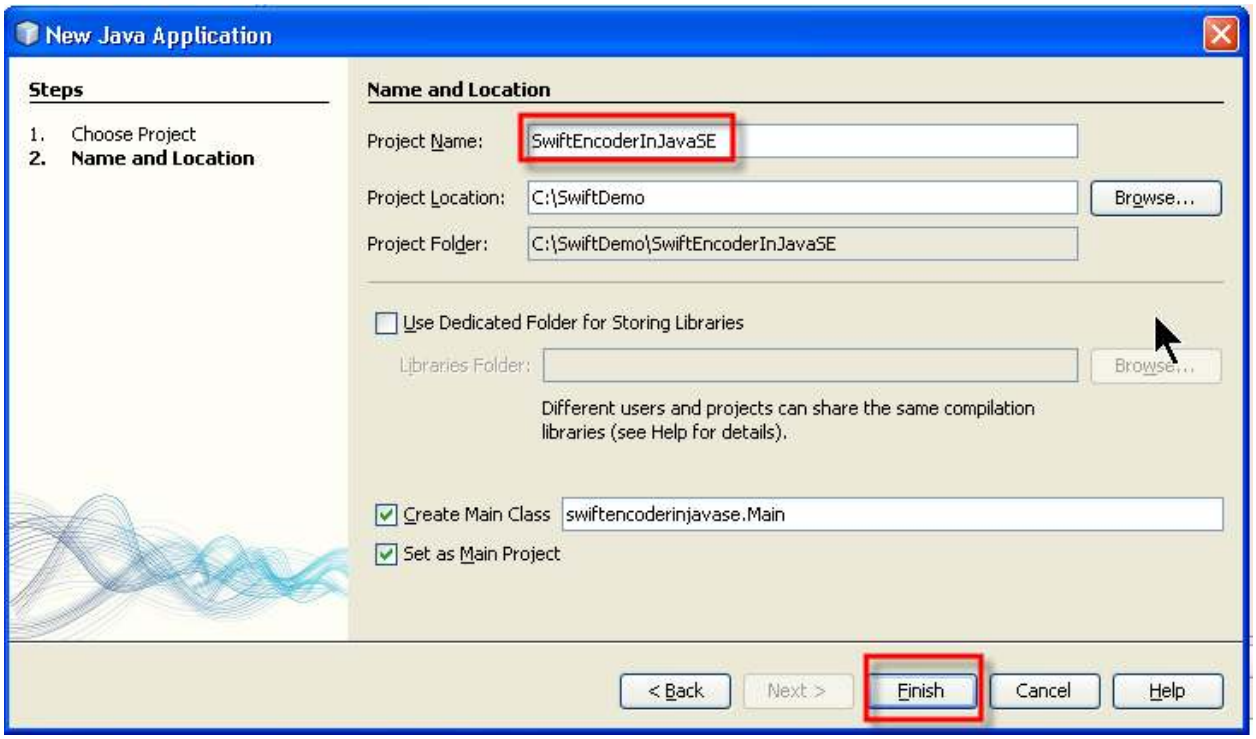
Here is a screen cast:

http://swiftencoder.appspot.com/SwiftEncoderInJavaSE.html

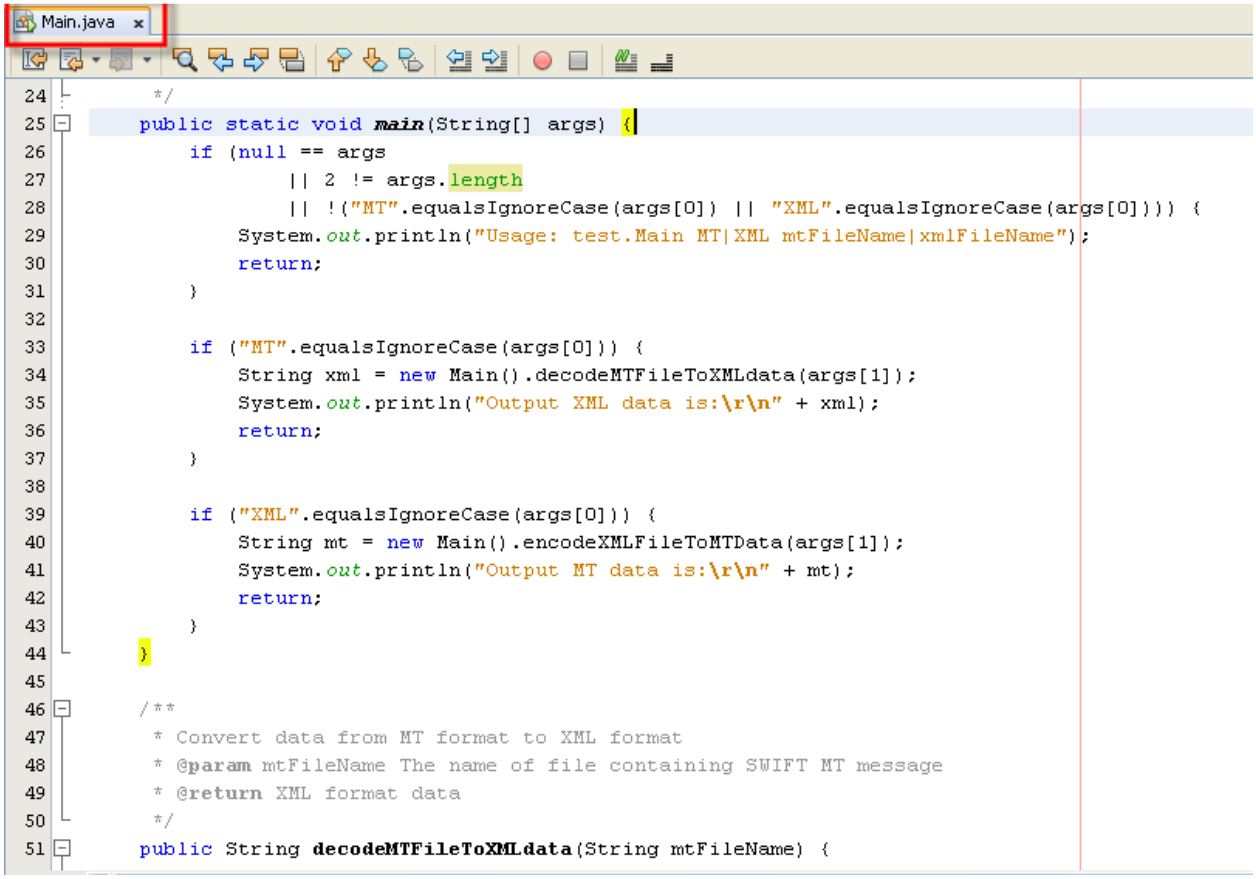
Create a project Java➔Java Application



Name it as "SwiftEncoderInJavaSE"



Here is a screen shot.



Code is similar to POJO sample (actually much simpler than POJO case), I won't describe the details any more.

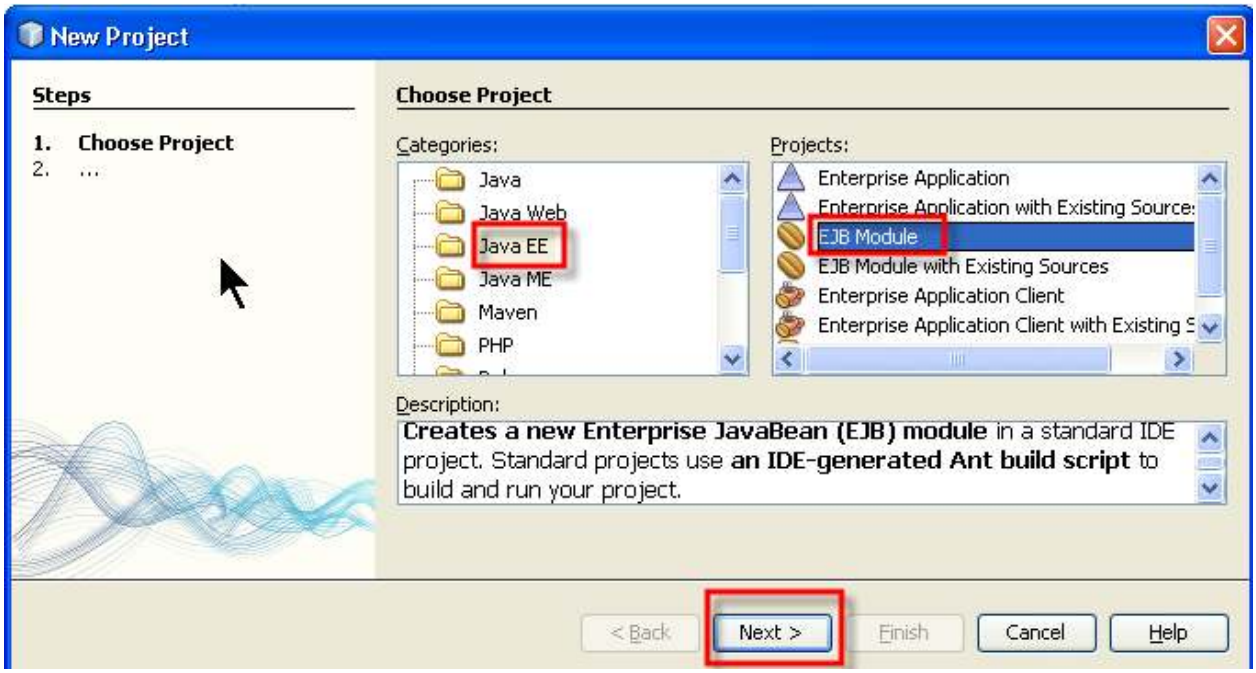
5. Using SWIFT Encoder in Java EE

Demonstrate how to use SWIFT Encoder in Java EE (Enterprise Edition) by use of a simple example.

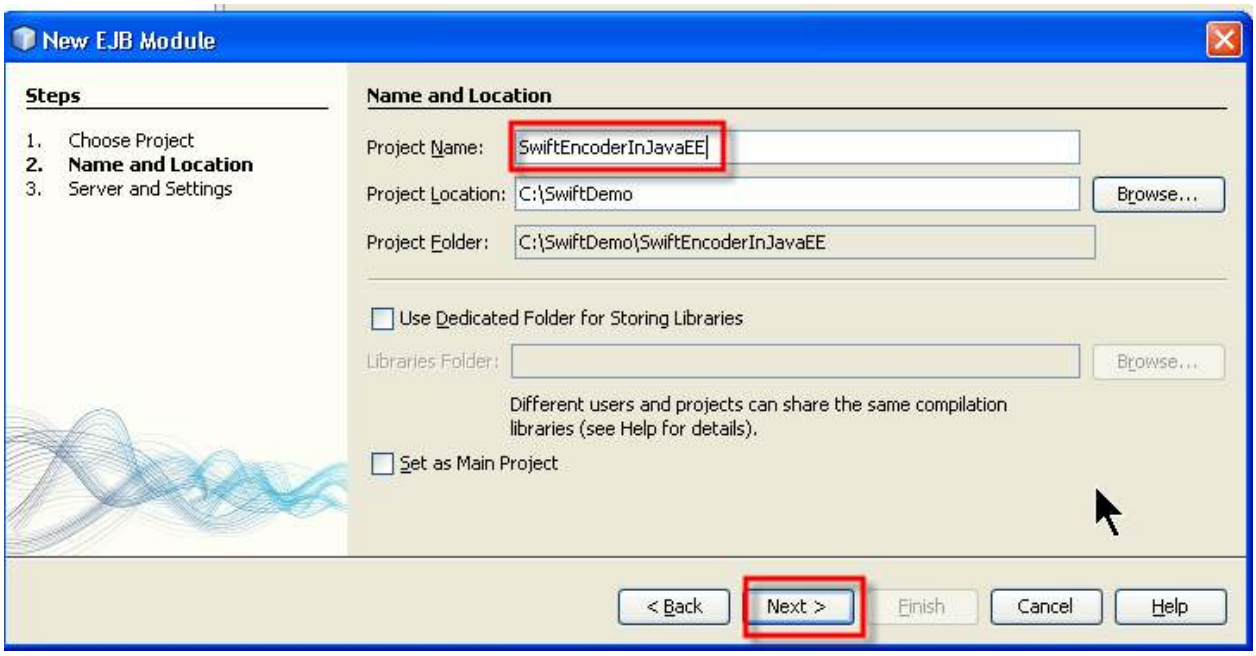
Here is a screen cast:

http://swiftencoder.appspot.com/SwiftEncoderInJavaEE.html

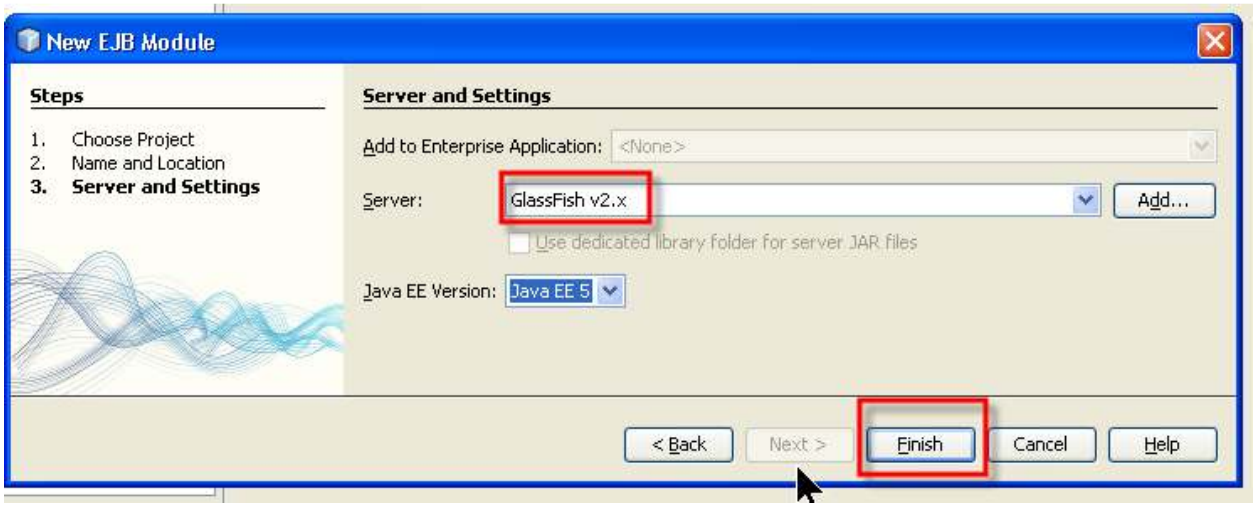
Create a project Java EE➔EJB Module



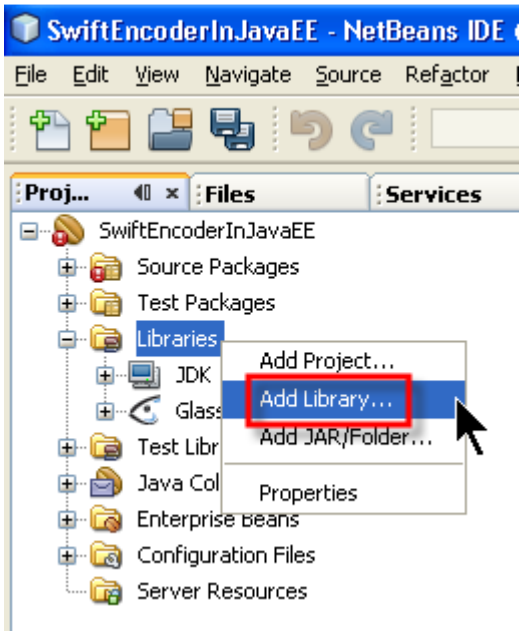
Give name "SwiftEncoderInJavaEE"



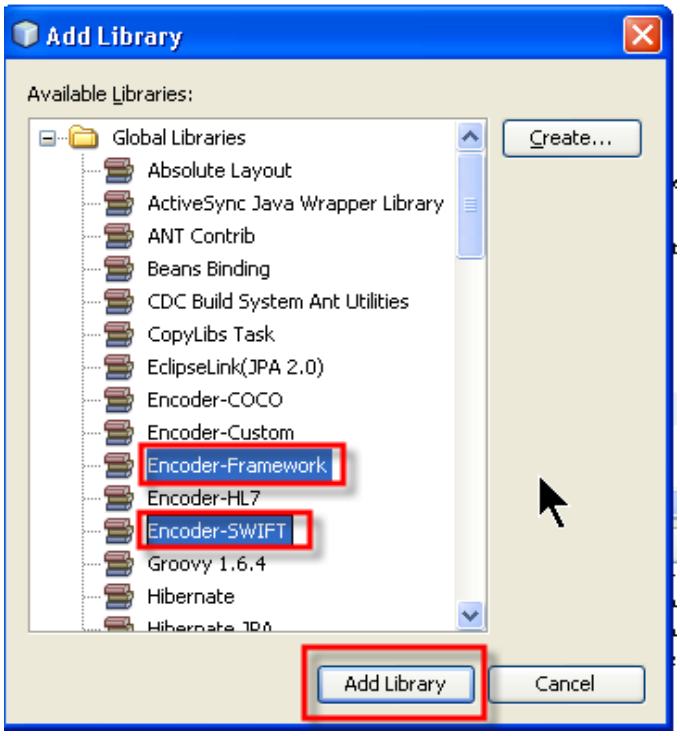
Choose the default server setting.



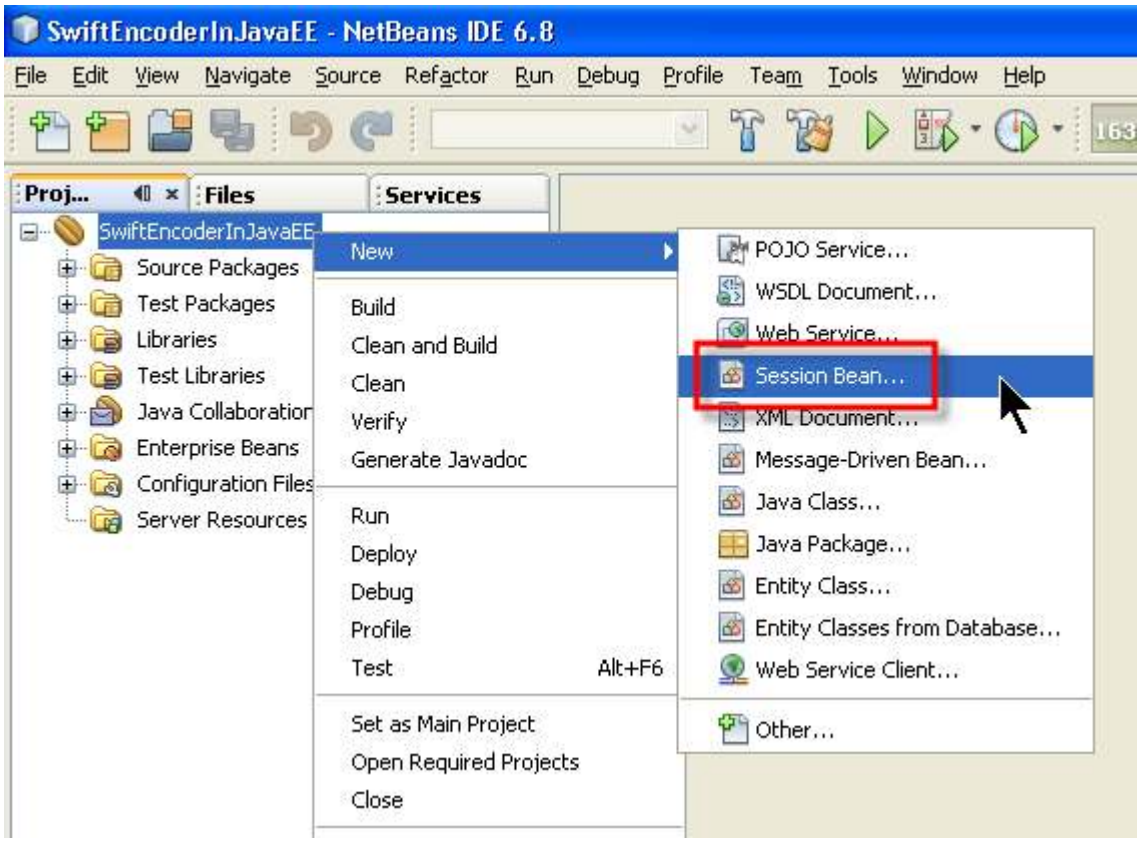
Add Library …



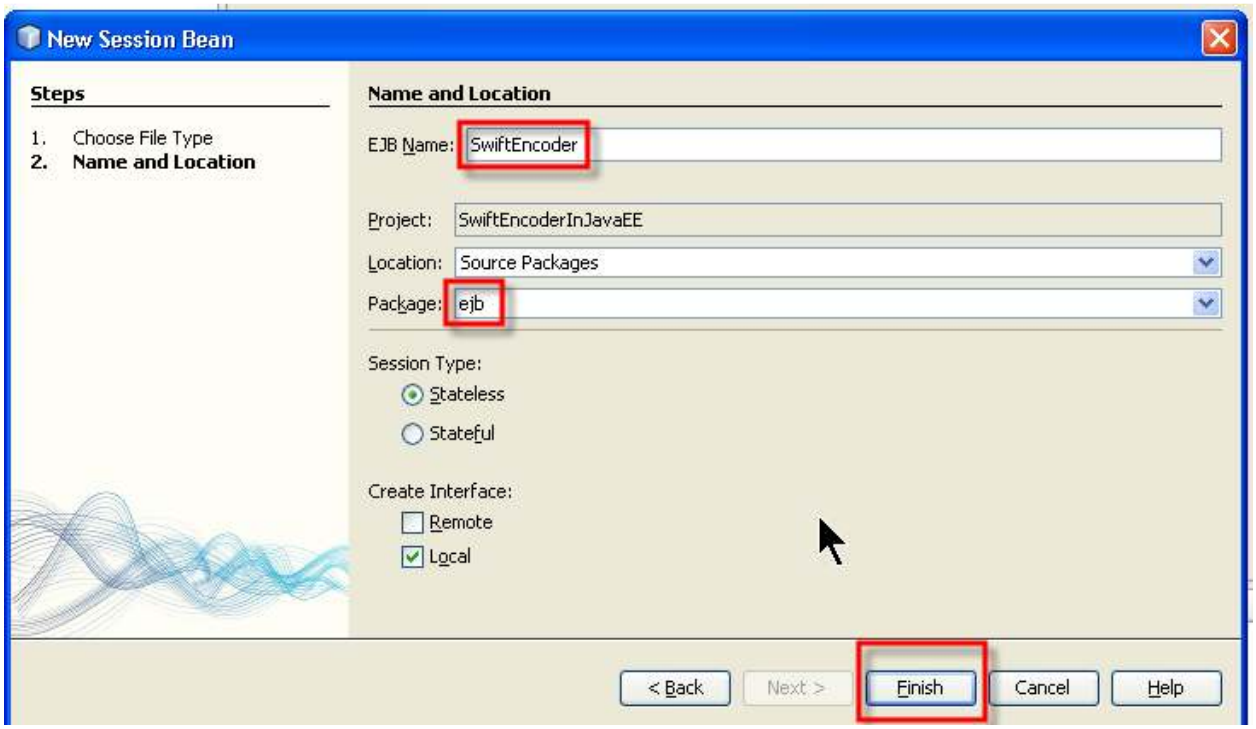
Choose libraries “Encoder-Framework” and “Encoder-SWIFT”.



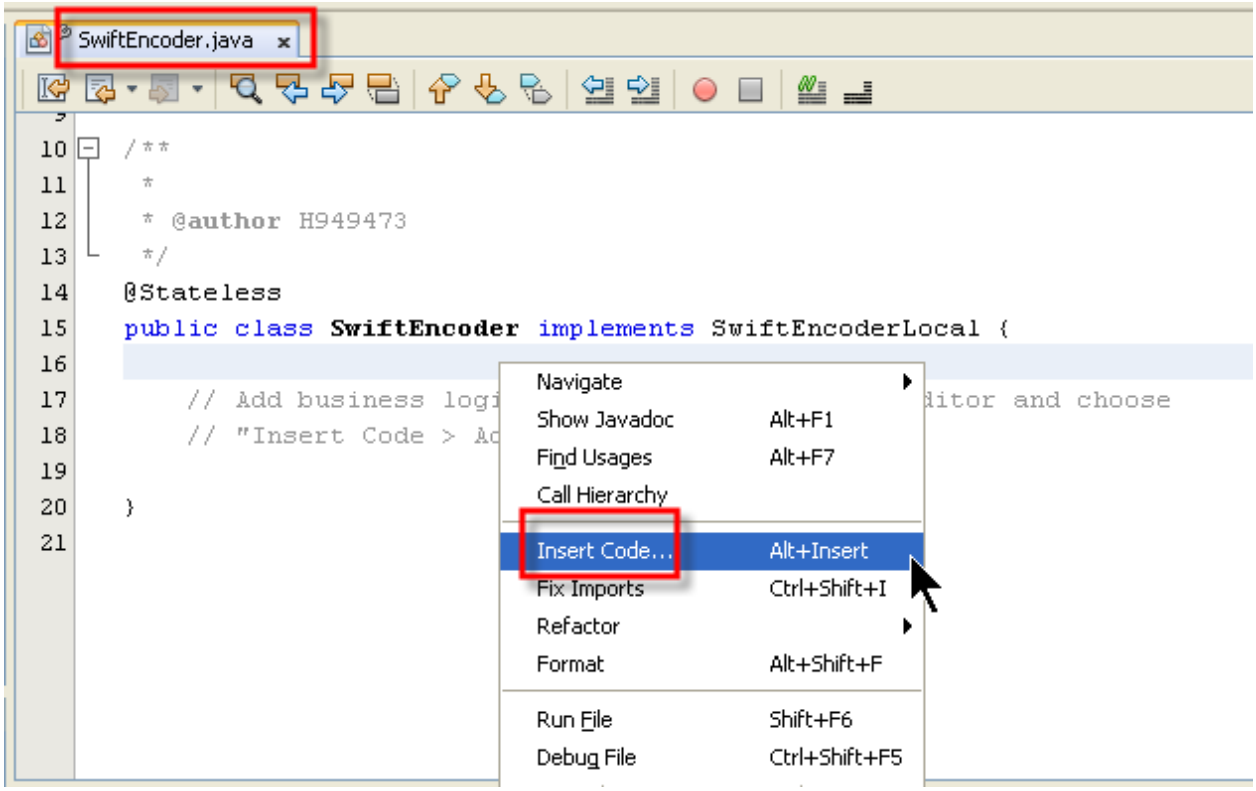
Create a Session Bean …



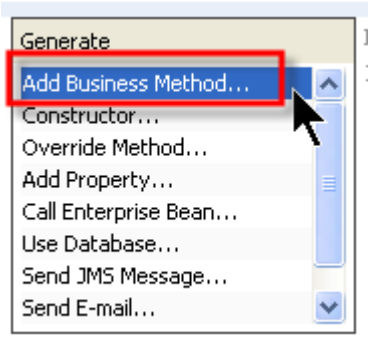
Give Bean name "SwiftEncoder", package "ejb".



Right-click in source panel, select "Insert Code …"

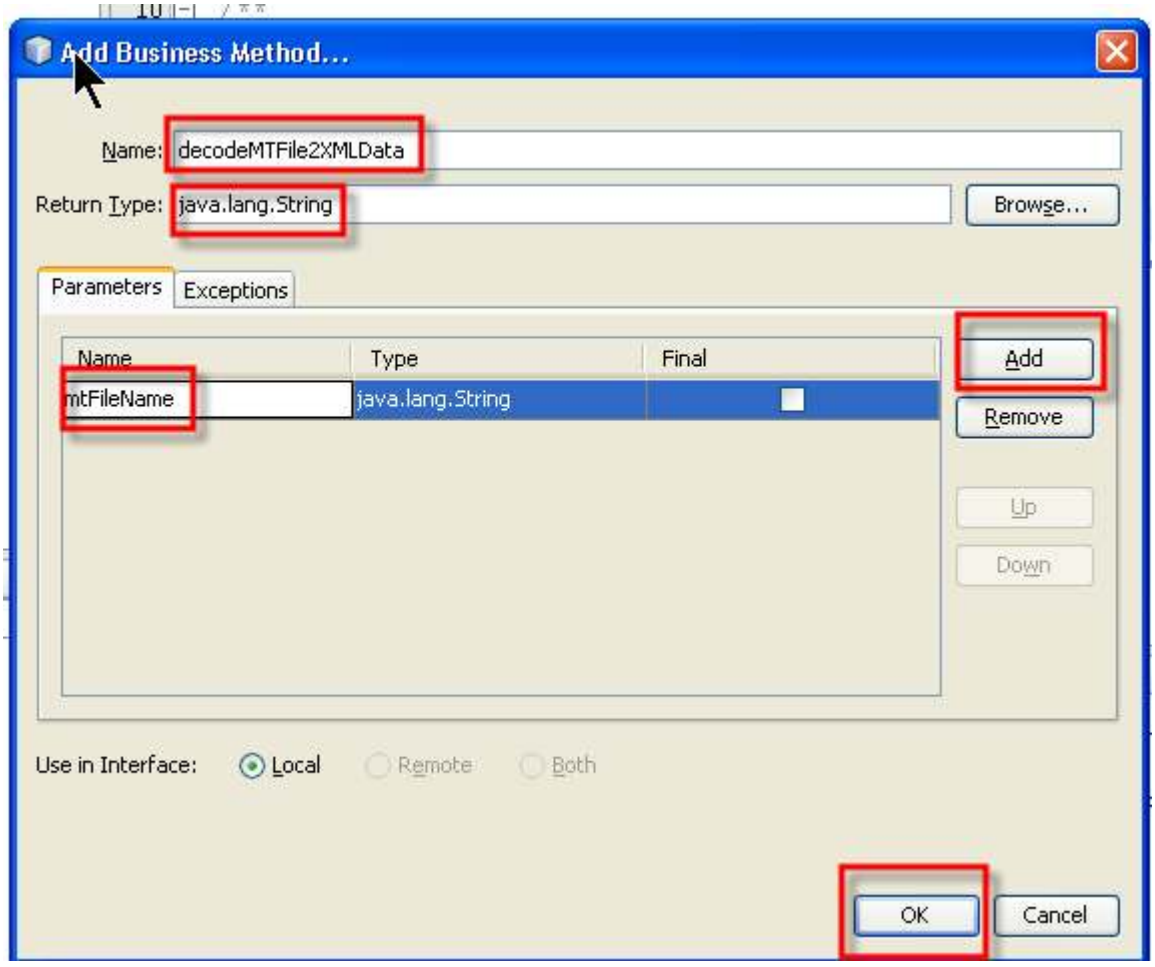


Choose “Add Business Method …” from pop-up menu



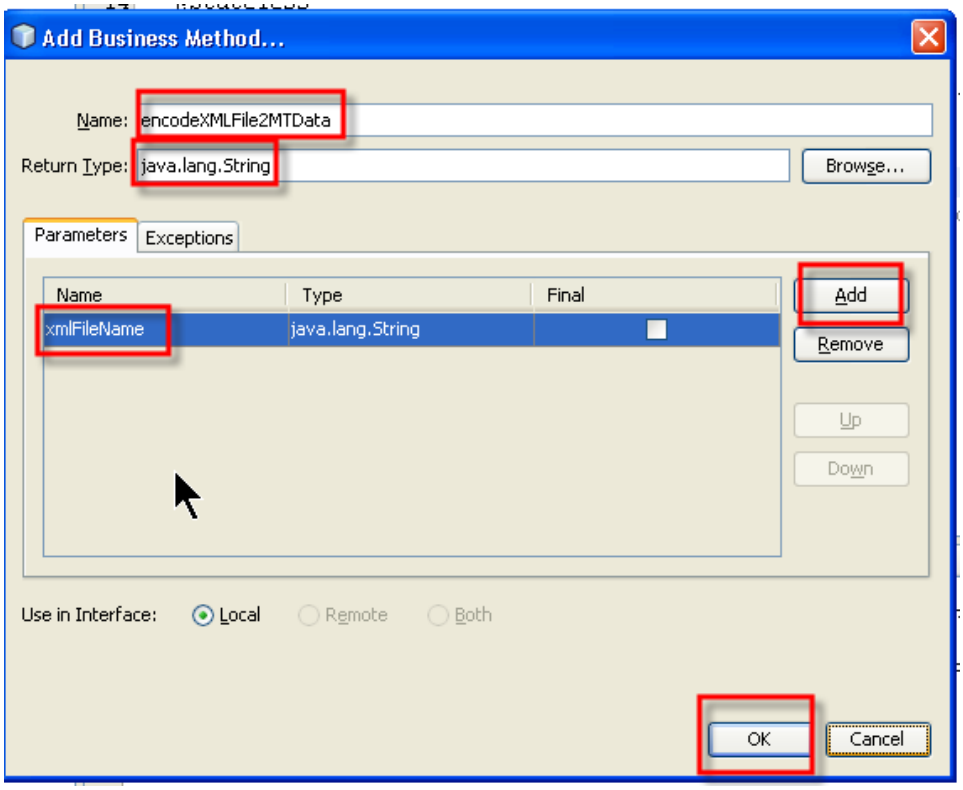
Give name “decodeMTFile2XMLData”, return type “java.lang.String”,

add Parameter “mtFileName” with type “java.lang.String”.

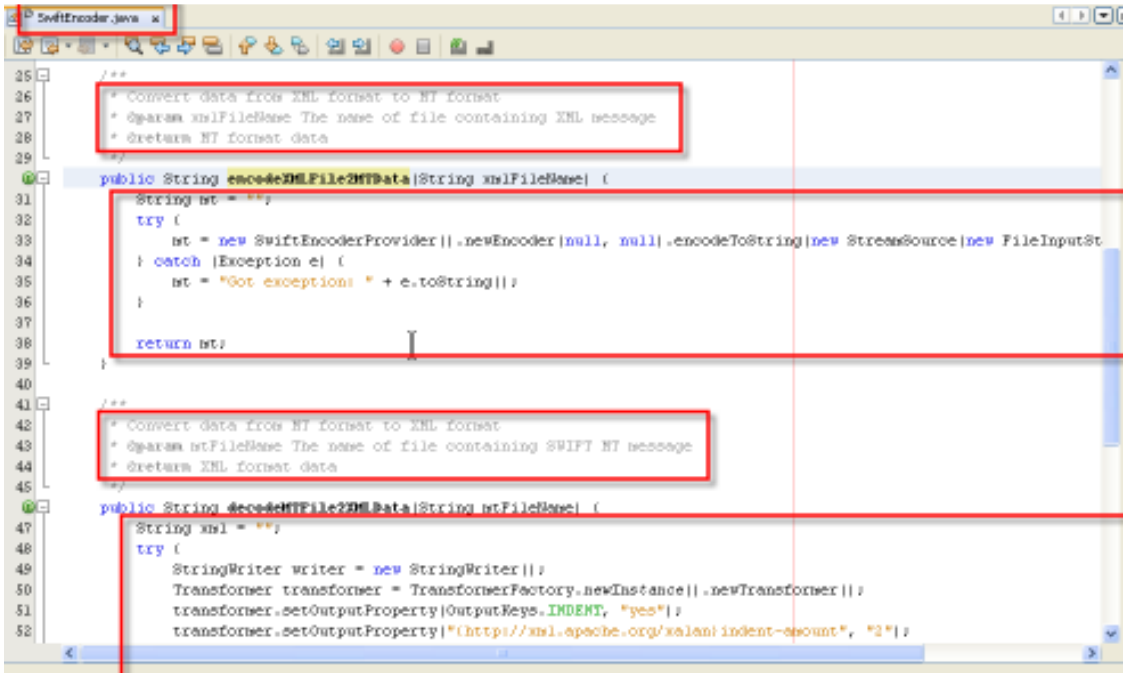


Add another method …

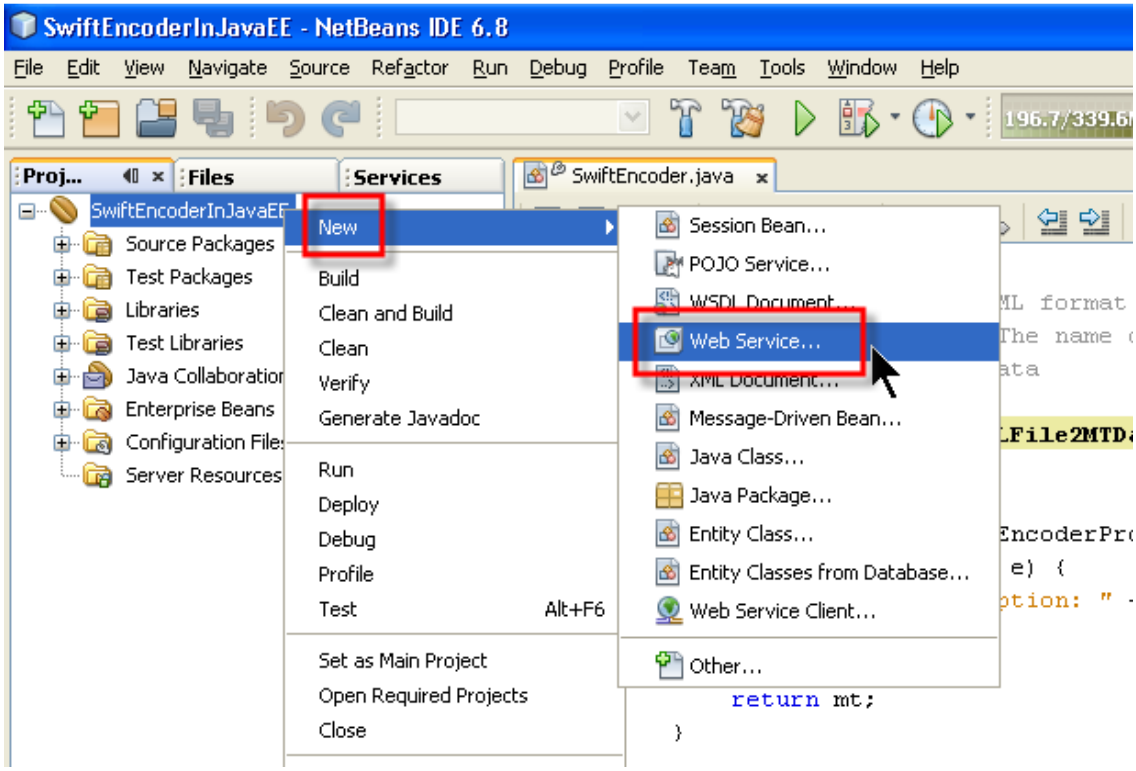
Name “encodeXMLFile2Data”, return type “java.lang.String”, add Parameter “xmlFileName” with type “java.lang.String”.



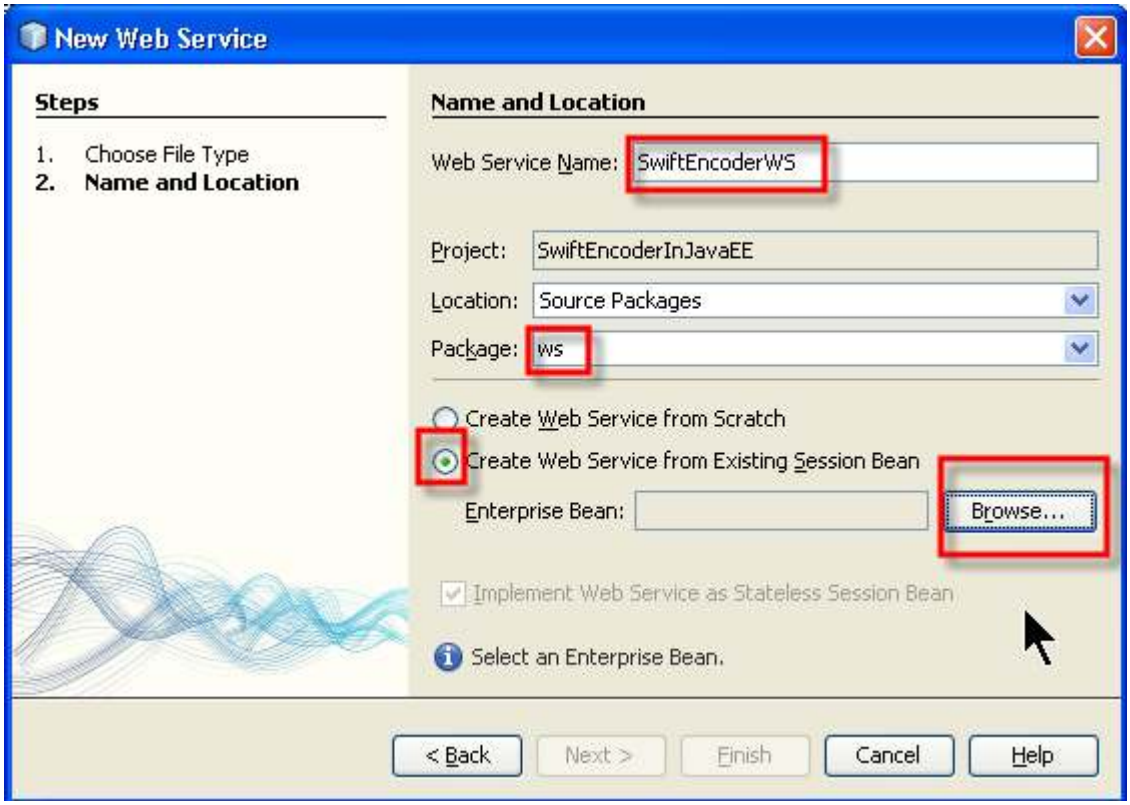
Write codes for two methods …



Create a web-service for Session Bean. New … Web Service …

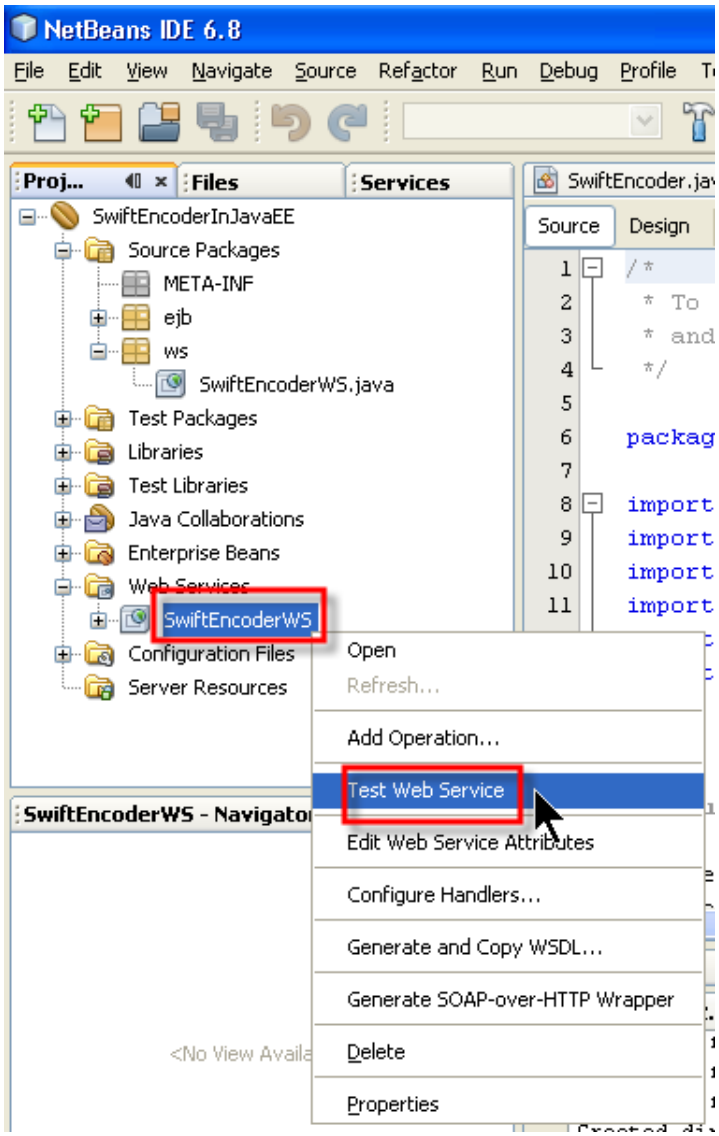


Name "SwiftEncoderWS", package "ws", click "Browse" button.



Select EJB “SwiftEncoder”.

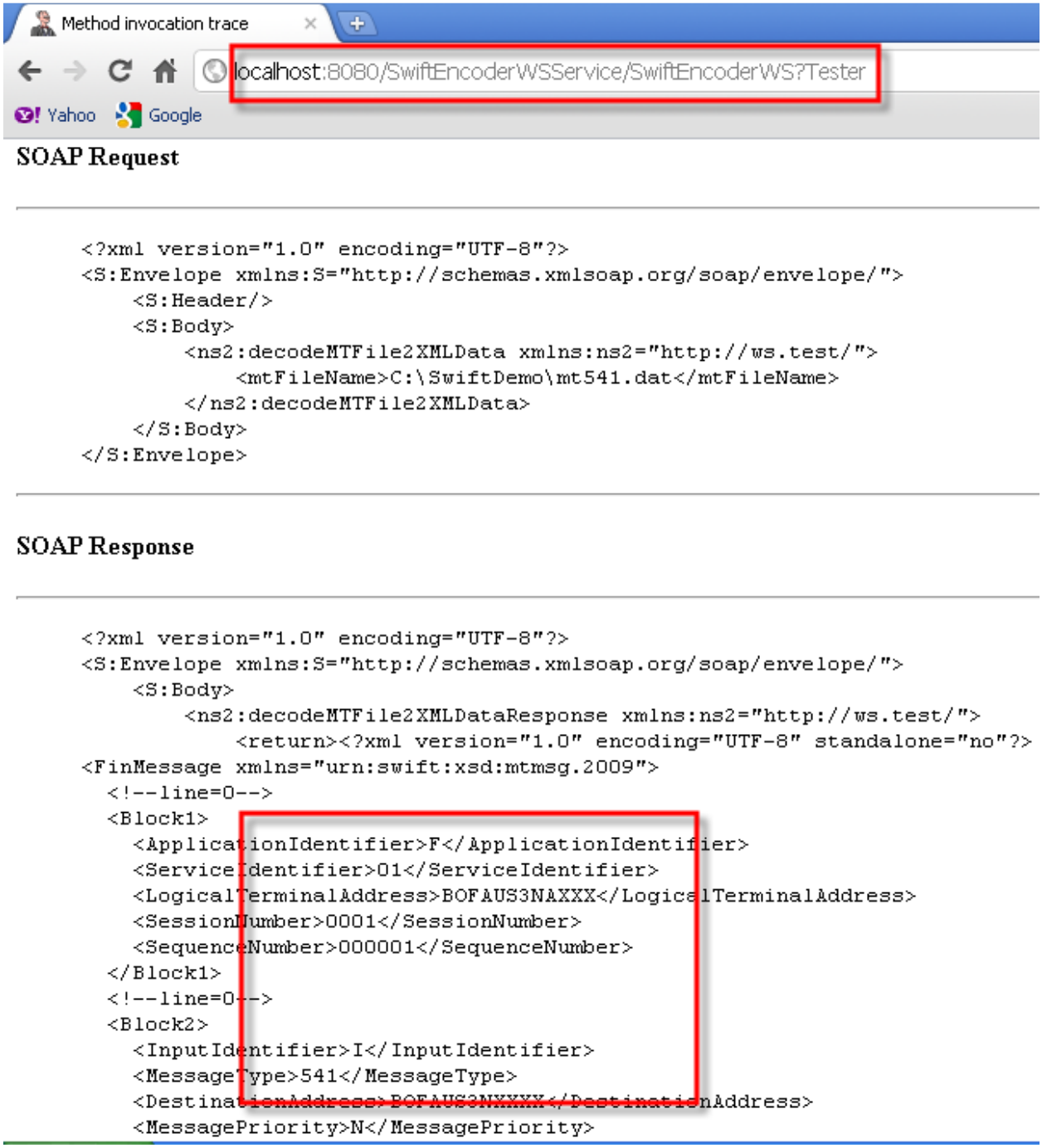
Right-click web-service “SwiftEncoderWS”, select “Test Web Service”



A browser open the Tester Web Service URL, two operations appear as two buttons: “decodeMTFile2XMLData” and “encodeXMFile2MTData”.

Specify a test MT data file in the text box next to button “decodeMTFile2XMLData”, e.g. “C:\SwiftDemo\mt541.dat”, then click button “decodeMTFile2XMLData”.

The decoded XML is returned with the SOAP response.



Now test encode function …

Specify an XML file in the text box next to button "encodeXMFile2MTData", e.g. "C:\SwiftDemo\mt541xml", then click button "encodeXMFile2MTData".

You can see the MT data in SOAP response.