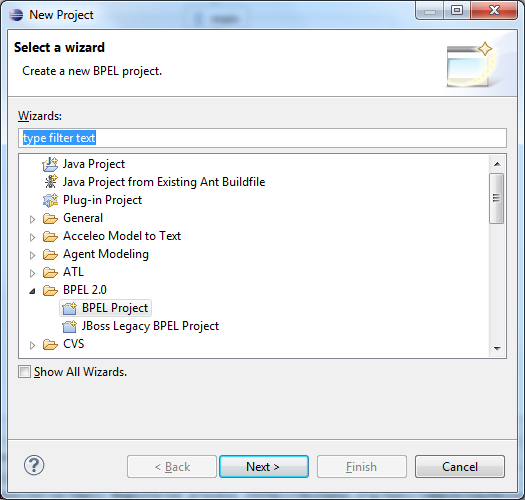
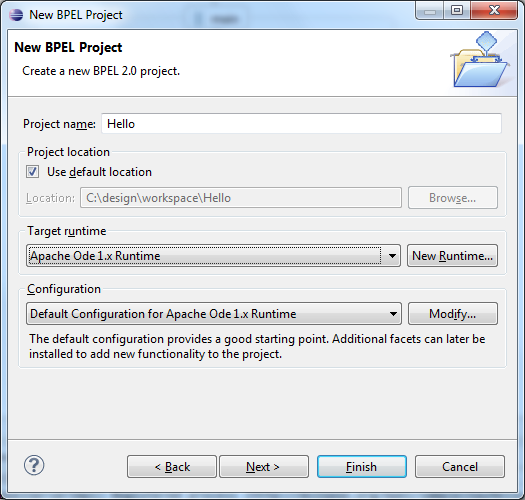
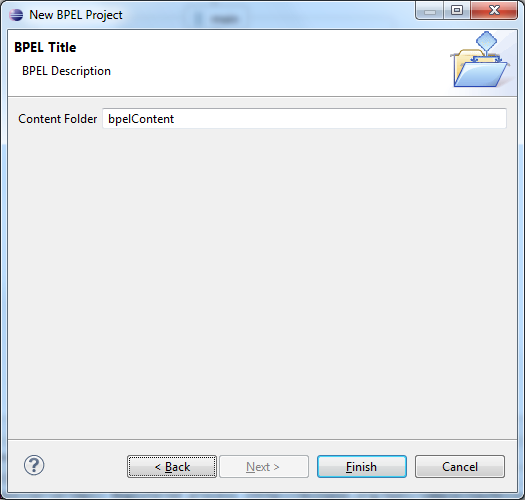
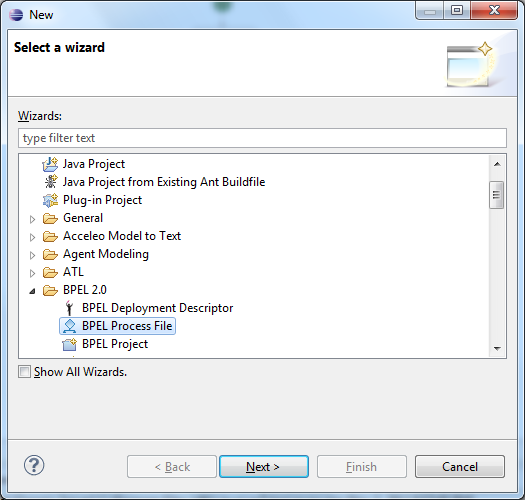
Create a new BPEL project

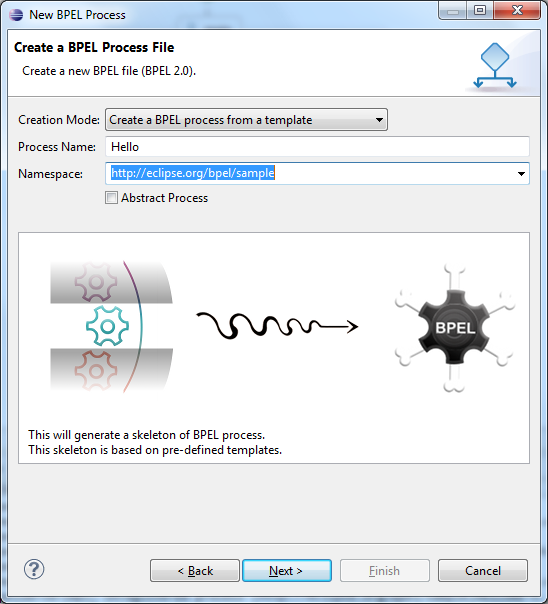


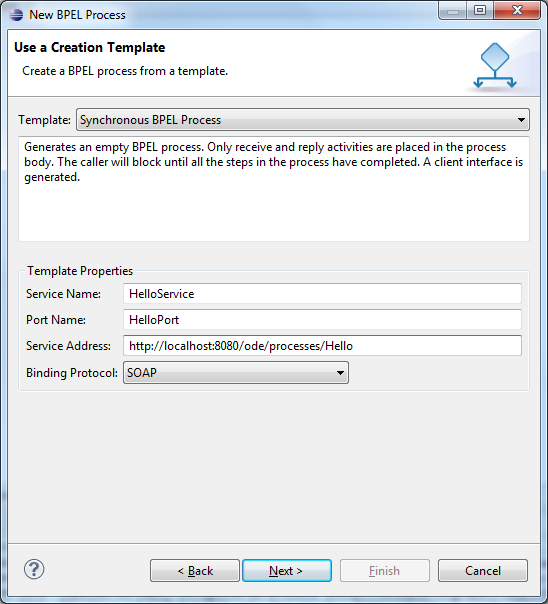


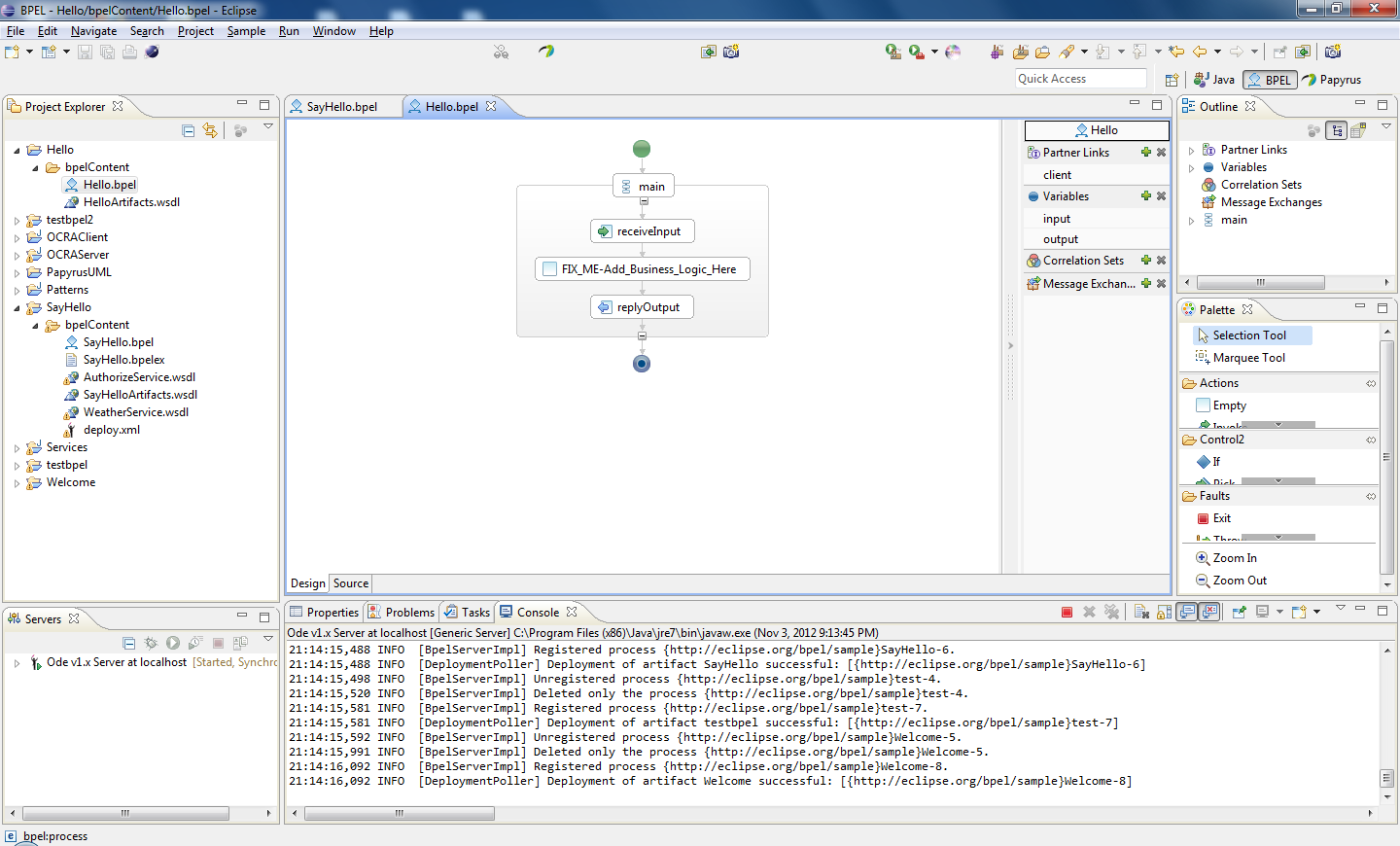


In Project Explorer right click bpelContent folder New->Other.. choose BPEL process file.









In wsdl file change “input” to “location”

<types>

<schema attributeFormDefault=*"unqualified"* elementFormDefault=*"qualified"*

targetNamespace=*"http://eclipse.org/bpel/sample"*

xmlns=*"http://www.w3.org/2001/XMLSchema"*>

<element name=*"HelloRequest"*>

<complexType>

<sequence>

<-- <element name=*"input"* type=*"string"*/> -->

<element name=*"****location****"* type=*"string"*/>

</sequence>

</complexType>

</element>

<element name=*"HelloResponse"*>

<complexType>

<sequence>

<element name=*"result"* type=*"string"*/>

</sequence>

</complexType>

</element>

</schema>

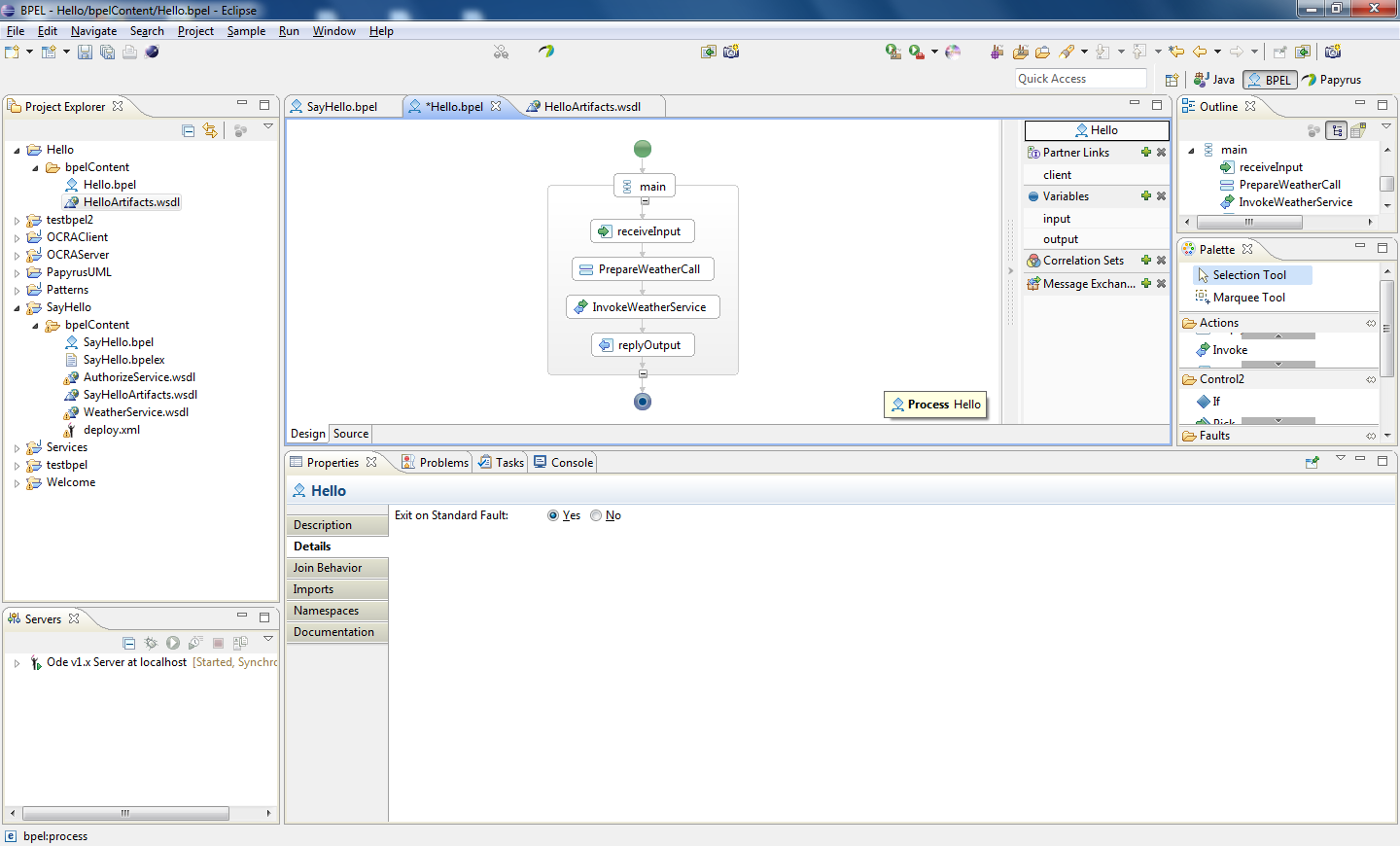
</types>

Remove FIXME activity from designer.

Add Assign and Invoke activities.

Name Assign to PrepareWeatherCall

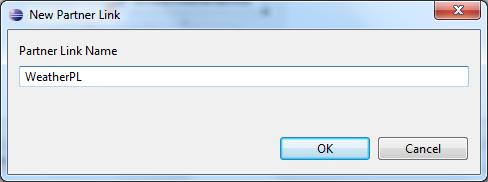
Name Invoke activity to InvokeWeatherService.

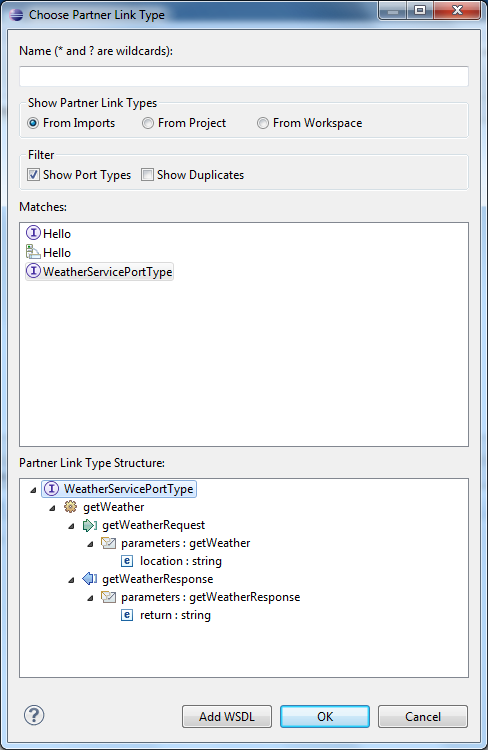


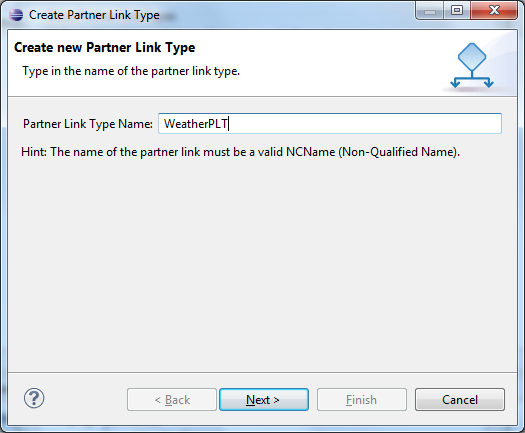
Import weather service wsdl into bpelContent folder

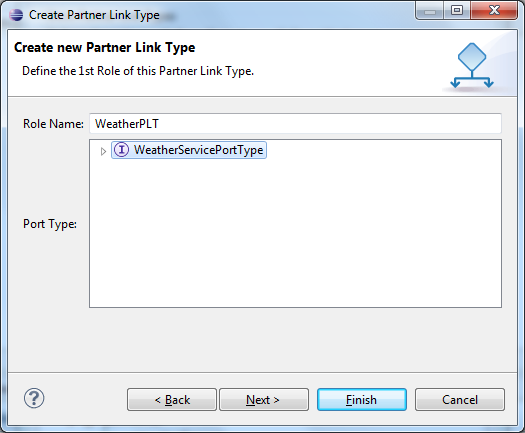
Click InvokeWeatherService -> Properties window -> details

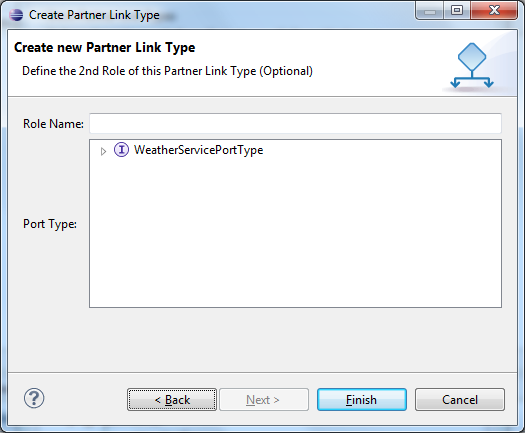
Choose partner link -> Create Global Partner Link



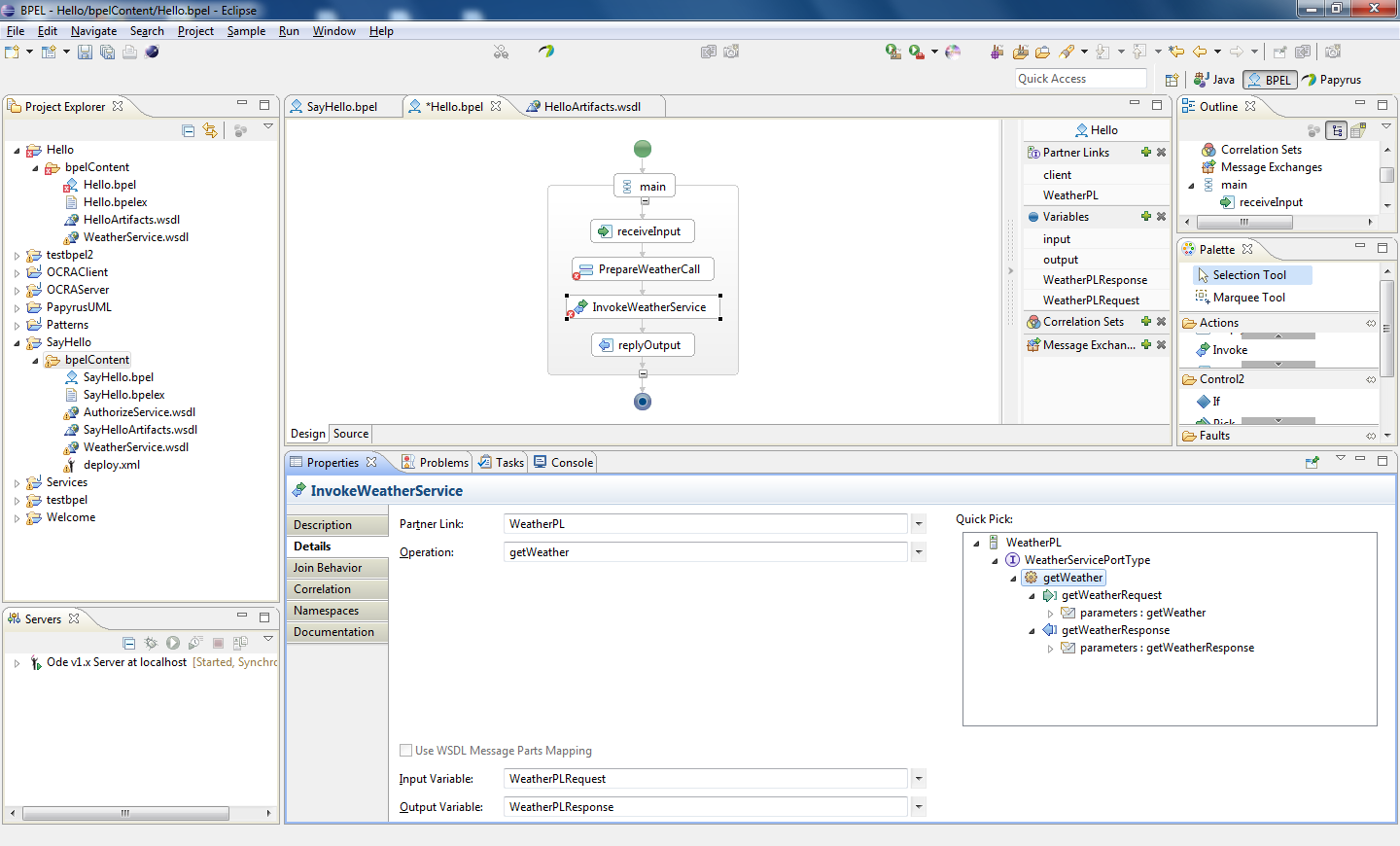




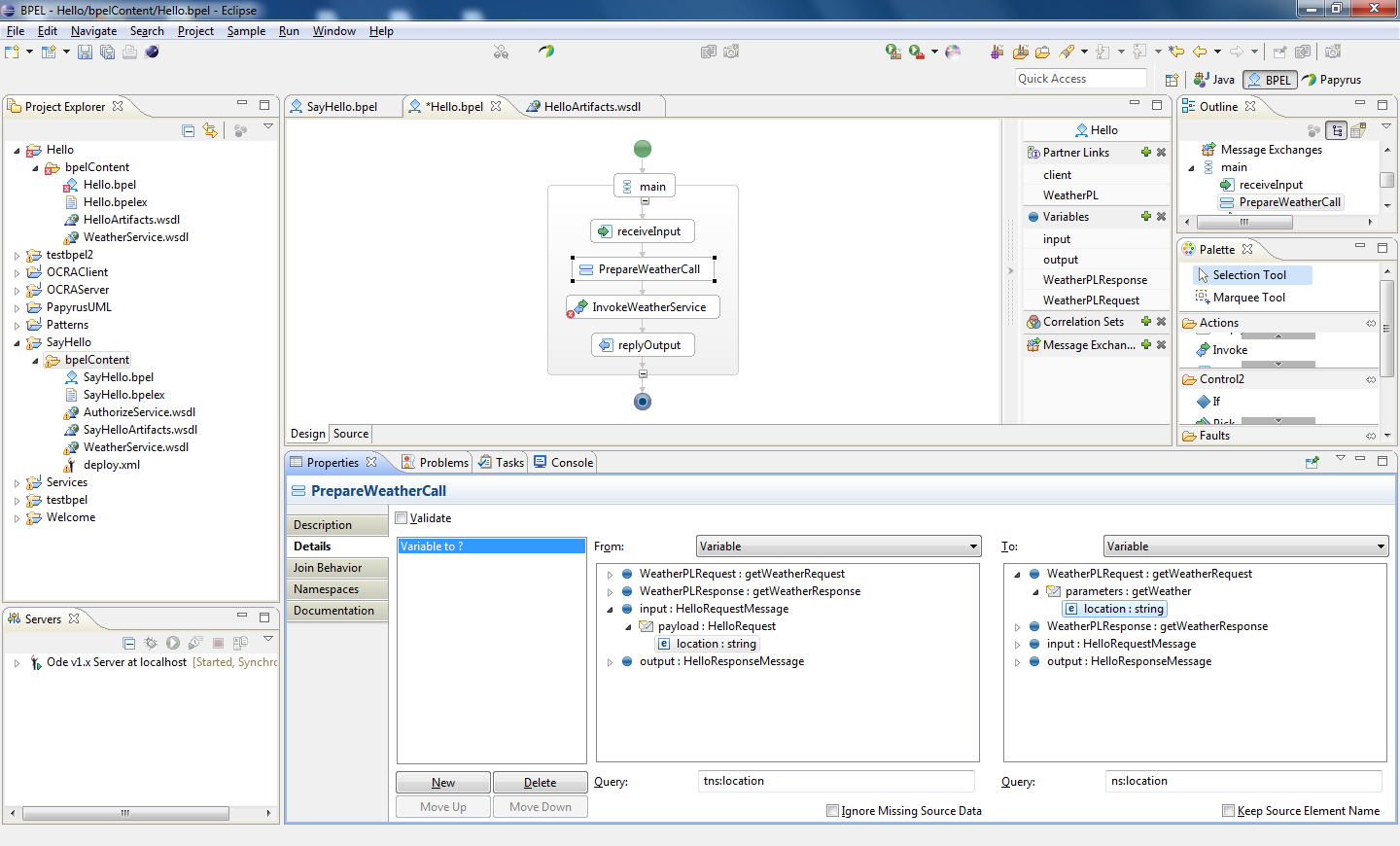


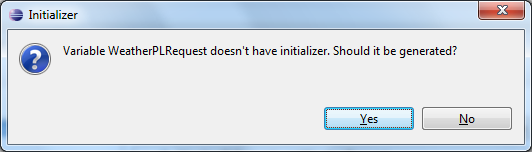


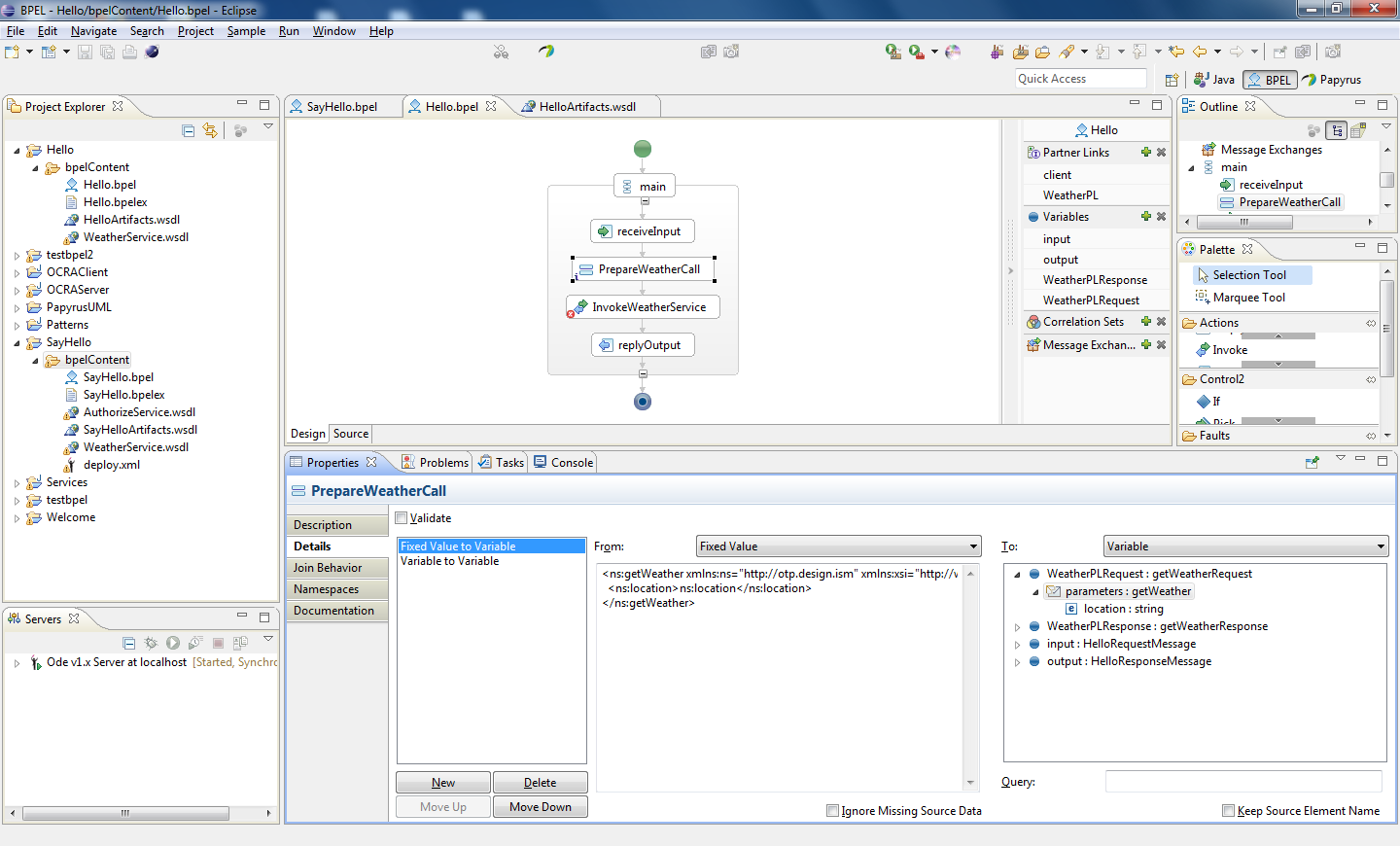
In properties, Details choose operation getWeather and double click getWeather operation in the right. Two variables are created for input and response to/from Weather service.



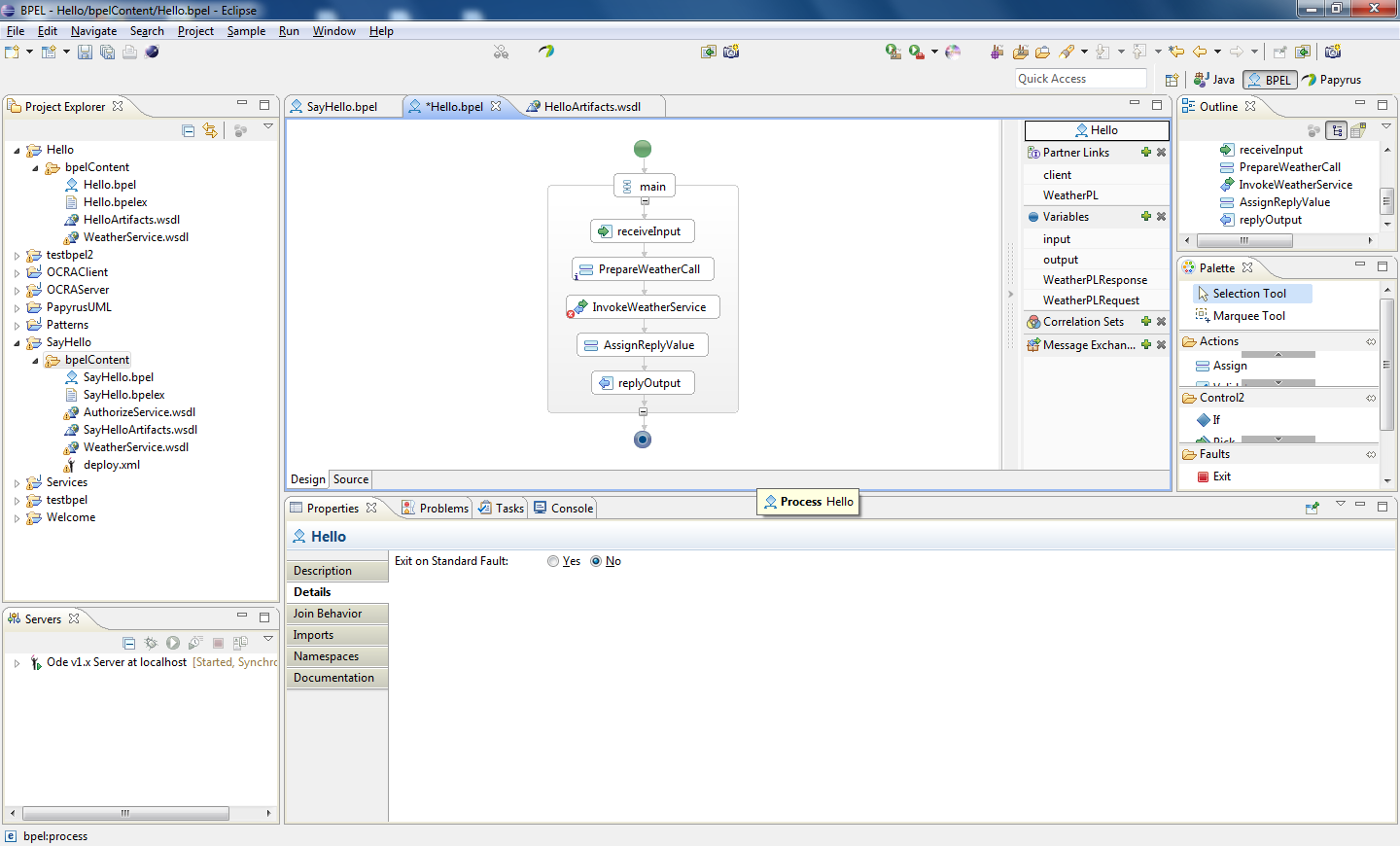
Select PrepareWeatherCall and go to Details in Properties.



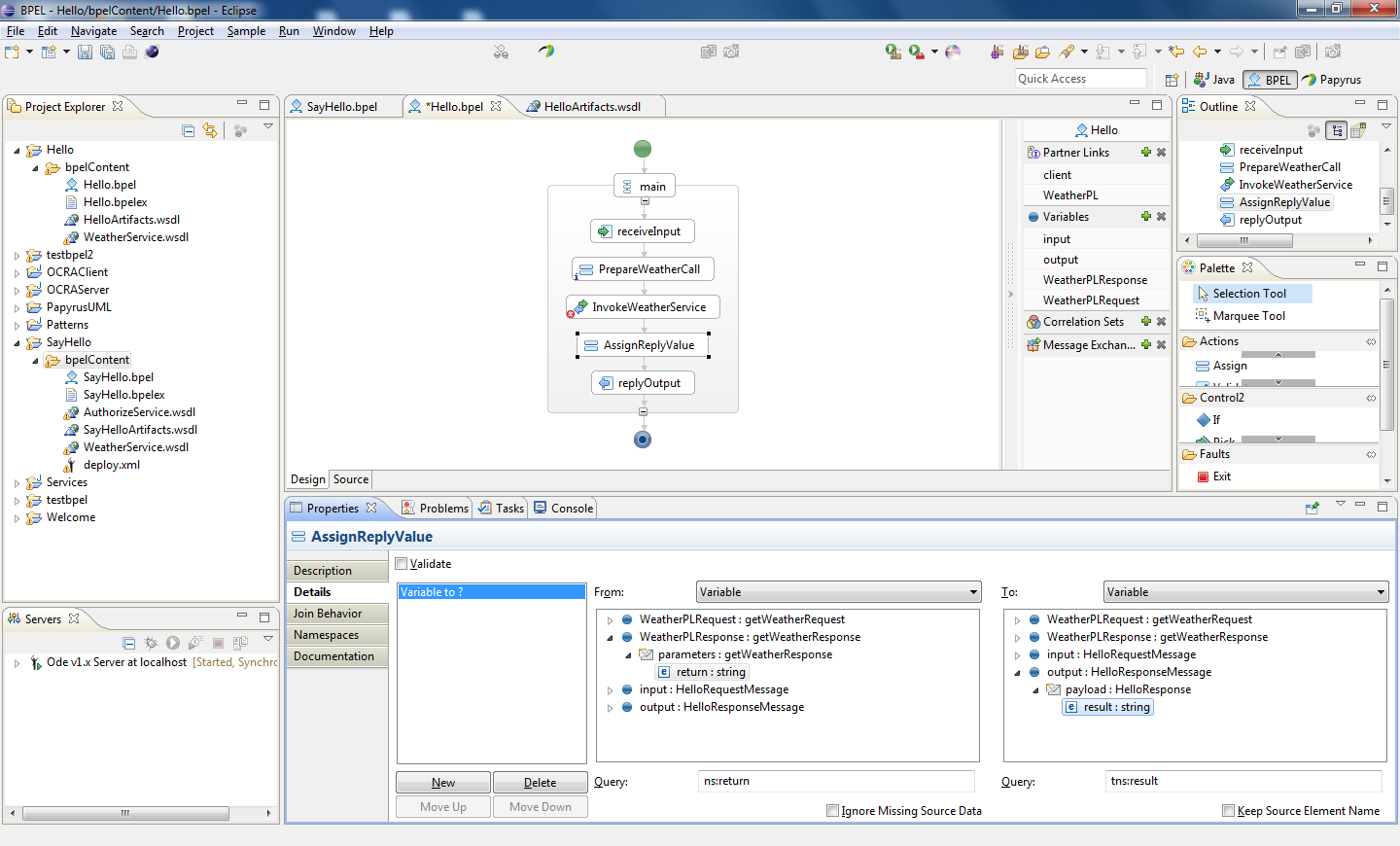


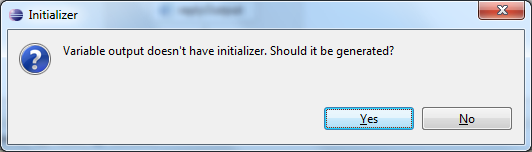


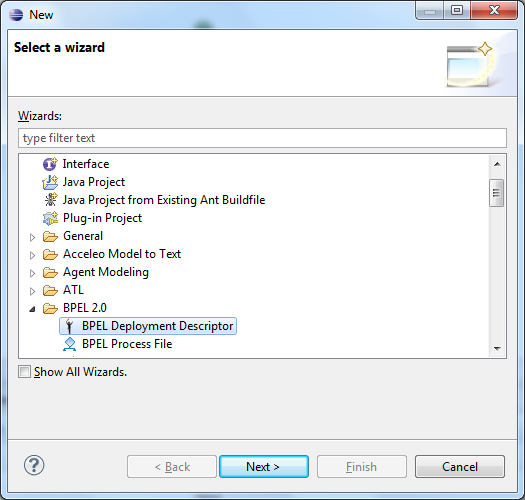
Add a new Assign activity after invoke, named AssignReplyValue.

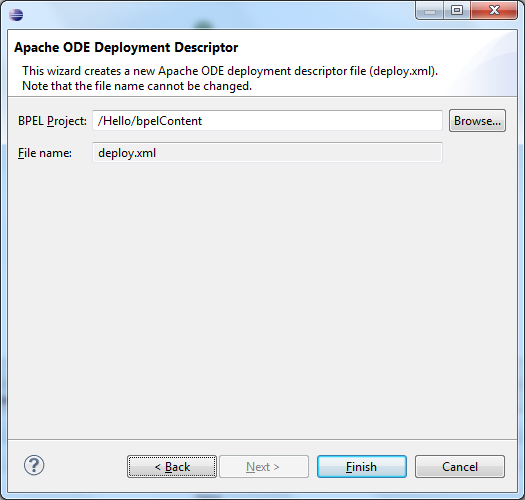


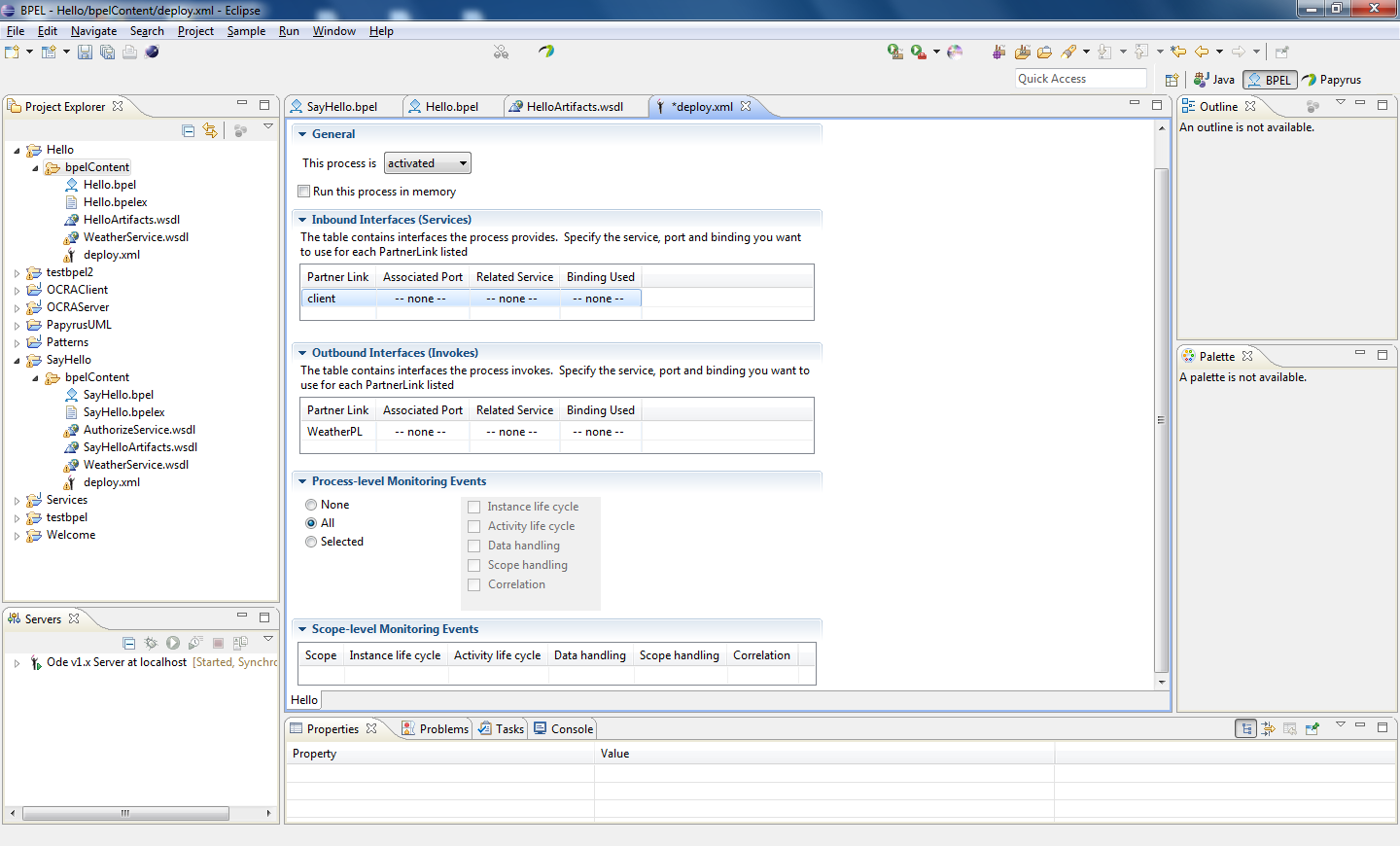
Edit its properties details

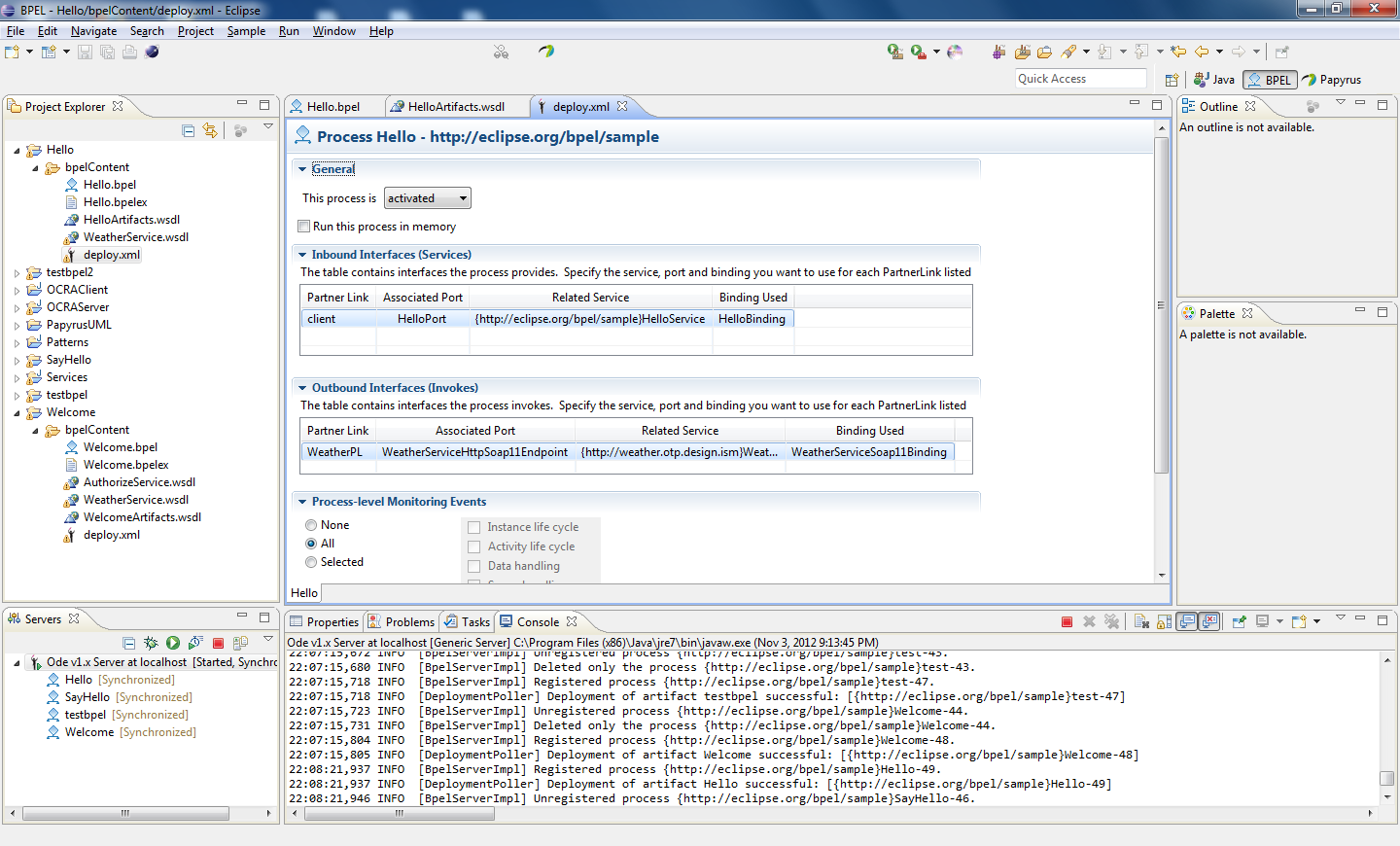


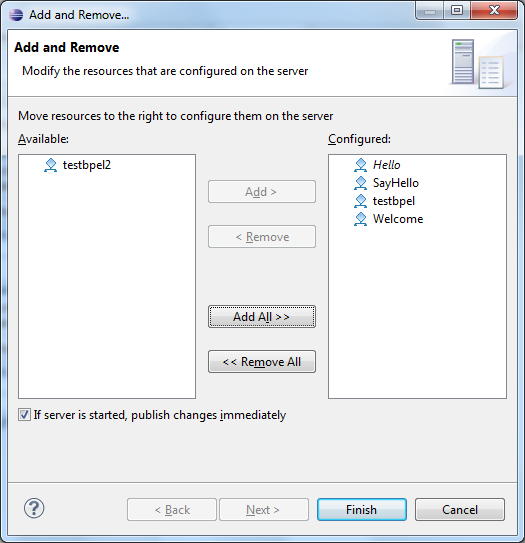




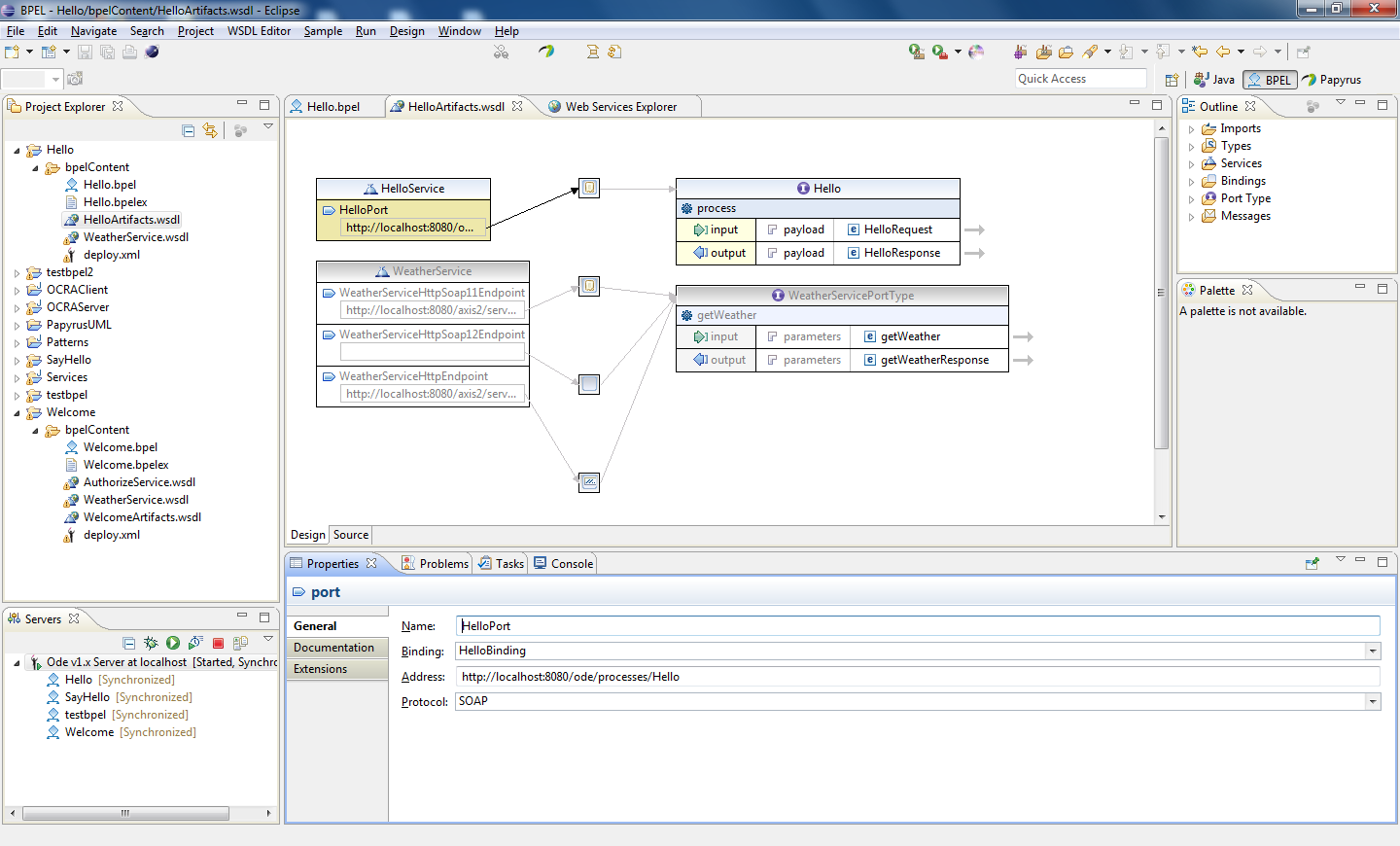


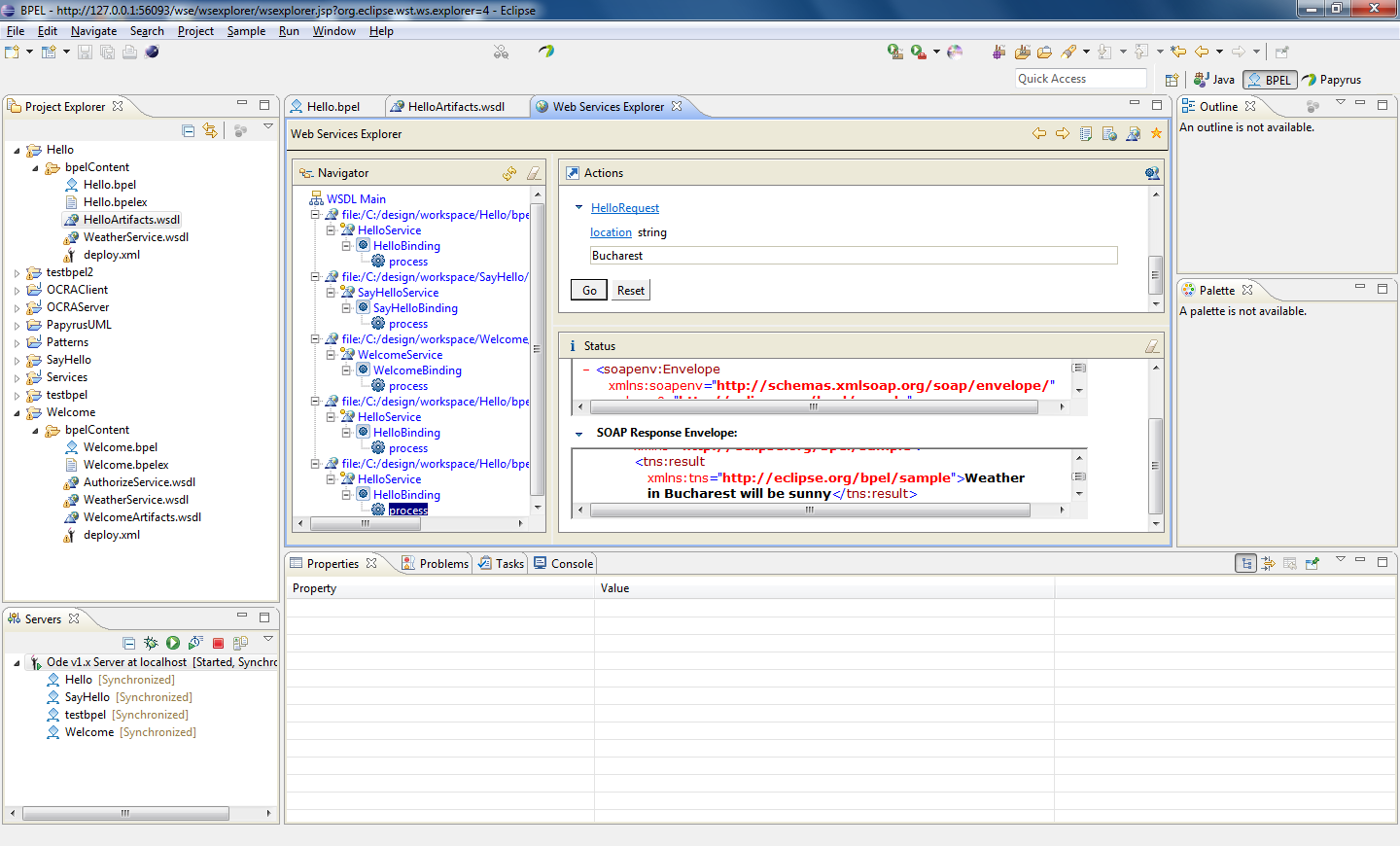






Will get exception while deploying





BPEL FILE Hello.bpel

<!-- Hello BPEL Process [Generated by the Eclipse BPEL Designer] -->

<!-- Date: Sat Nov 03 21:19:59 EET 2012 -->

<bpel:process name=*"Hello"*

targetNamespace=*"http://eclipse.org/bpel/sample"*

suppressJoinFailure=*"yes"*

xmlns:tns=*"http://eclipse.org/bpel/sample"*

xmlns:bpel=*"http://docs.oasis-open.org/wsbpel/2.0/process/executable"*

xmlns:ns=*"http://weather.otp.design.ism"*>

<!-- Import the client WSDL -->

<bpel:import namespace=*"http://weather.otp.design.ism"* location=*"WeatherService.wsdl"* importType=*"http://schemas.xmlsoap.org/wsdl/"*></bpel:import>

<bpel:import location=*"HelloArtifacts.wsdl"* namespace=*"http://eclipse.org/bpel/sample"*

importType=*"http://schemas.xmlsoap.org/wsdl/"* />

<!-- ================================================================= -->

<!-- PARTNERLINKS -->

<!-- List of services participating in this BPEL process -->

<!-- ================================================================= -->

<bpel:partnerLinks>

<!-- The 'client' role represents the requester of this service. -->

<bpel:partnerLink name=*"client"*

partnerLinkType=*"tns:Hello"*

myRole=*"HelloProvider"*

/>

<bpel:partnerLink name=*"WeatherPL"* partnerLinkType=*"tns:WeatherPLT"* partnerRole=*"WeatherPLT"*></bpel:partnerLink>

</bpel:partnerLinks>

<!-- ================================================================= -->

<!-- VARIABLES -->

<!-- List of messages and XML documents used within this BPEL process -->

<!-- ================================================================= -->

<bpel:variables>

<!-- Reference to the message passed as input during initiation -->

<bpel:variable name=*"input"*

messageType=*"tns:HelloRequestMessage"*/>

<!--

Reference to the message that will be returned to the requester

-->

<bpel:variable name=*"output"*

messageType=*"tns:HelloResponseMessage"*/>

<bpel:variable name=*"WeatherPLResponse"* messageType=*"ns:getWeatherResponse"*></bpel:variable>

<bpel:variable name=*"WeatherPLRequest"* messageType=*"ns:getWeatherRequest"*></bpel:variable>

</bpel:variables>

<!-- ================================================================= -->

<!-- ORCHESTRATION LOGIC -->

<!-- Set of activities coordinating the flow of messages across the -->

<!-- services integrated within this business process -->

<!-- ================================================================= -->

<bpel:sequence name=*"main"*>

<!-- Receive input from requester.

Note: This maps to operation defined in Hello.wsdl

-->

<bpel:receive name=*"receiveInput"* partnerLink=*"client"*

portType=*"tns:Hello"*

operation=*"process"* variable=*"input"*

createInstance=*"yes"*/>

<!-- Generate reply to synchronous request -->

<bpel:assign validate=*"no"* name=*"PrepareWeatherCall"*>

<bpel:copy>

<bpel:from><bpel:literal><ns:getWeather xmlns:ns=*"http://weather.otp.design.ism"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*>

<ns:location>ns:location</ns:location>

</ns:getWeather>

</bpel:literal></bpel:from>

<bpel:to variable=*"WeatherPLRequest"* part=*"parameters"*></bpel:to>

</bpel:copy>

<bpel:copy>

<bpel:from part=*"payload"* variable=*"input"*>

<bpel:query queryLanguage=*"urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0"*><![CDATA[tns:location]]></bpel:query>

</bpel:from>

<bpel:to part=*"parameters"* variable=*"WeatherPLRequest"*>

<bpel:query queryLanguage=*"urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0"*><![CDATA[ns:location]]></bpel:query>

</bpel:to>

</bpel:copy>

</bpel:assign>

<bpel:invoke name=*"InvokeWeatherService"* partnerLink=*"WeatherPL"* operation=*"getWeather"* portType=*"ns:WeatherServicePortType"* inputVariable=*"WeatherPLRequest"* outputVariable=*"WeatherPLResponse"*></bpel:invoke>

<bpel:assign validate=*"no"* name=*"AssignReplyValue"*>

<bpel:copy>

<bpel:from><bpel:literal><tns:HelloResponse xmlns:tns=*"http://eclipse.org/bpel/sample"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*>

<tns:result>tns:result</tns:result>

</tns:HelloResponse>

</bpel:literal></bpel:from>

<bpel:to variable=*"output"* part=*"payload"*></bpel:to>

</bpel:copy>

<bpel:copy>

<bpel:from part=*"parameters"* variable=*"WeatherPLResponse"*>

<bpel:query queryLanguage=*"urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0"*><![CDATA[ns:return]]></bpel:query>

</bpel:from>

<bpel:to part=*"payload"* variable=*"output"*>

<bpel:query queryLanguage=*"urn:oasis:names:tc:wsbpel:2.0:sublang:xpath1.0"*><![CDATA[tns:result]]></bpel:query>

</bpel:to>

</bpel:copy>

</bpel:assign>

<bpel:reply name=*"replyOutput"*

partnerLink=*"client"*

portType=*"tns:Hello"*

operation=*"process"*

variable=*"output"*

/>

</bpel:sequence>

</bpel:process>

WSDL FILE HelloArtifacts.wsdl

<?xml version=*"1.0"* encoding=*"UTF-8"* standalone=*"no"*?>

<definitions xmlns=*"http://schemas.xmlsoap.org/wsdl/"* xmlns:plnk=*"http://docs.oasis-open.org/wsbpel/2.0/plnktype"* xmlns:soap=*"http://schemas.xmlsoap.org/wsdl/soap/"* xmlns:tns=*"http://eclipse.org/bpel/sample"* xmlns:vprop=*"http://docs.oasis-open.org/wsbpel/2.0/varprop"* xmlns:wsdl=*"http://weather.otp.design.ism"* name=*"Hello"* targetNamespace=*"http://eclipse.org/bpel/sample"*>

<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

TYPE DEFINITION - List of types participating in this BPEL process

The BPEL Designer will generate default request and response types

but you can define or import any XML Schema type and use them as part

of the message types.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ -->

<plnk:partnerLinkType name=*"WeatherPLT"*>

<plnk:role name=*"WeatherPLT"* portType=*"wsdl:WeatherServicePortType"*/>

</plnk:partnerLinkType>

<import location=*"WeatherService.wsdl"* namespace=*"http://weather.otp.design.ism"*/>

<types>

<schema xmlns=*"http://www.w3.org/2001/XMLSchema"* attributeFormDefault=*"unqualified"* elementFormDefault=*"qualified"* targetNamespace=*"http://eclipse.org/bpel/sample"*>

<element name=*"HelloRequest"*>

<complexType>

<sequence>

<element name=*"location"* type=*"string"*/>

</sequence>

</complexType>

</element>

<element name=*"HelloResponse"*>

<complexType>

<sequence>

<element name=*"result"* type=*"string"*/>

</sequence>

</complexType>

</element>

</schema>

</types>

<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

MESSAGE TYPE DEFINITION - Definition of the message types used as

part of the port type defintions

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ -->

<message name=*"HelloRequestMessage"*>

<part element=*"tns:HelloRequest"* name=*"payload"*/>

</message>

<message name=*"HelloResponseMessage"*>

<part element=*"tns:HelloResponse"* name=*"payload"*/>

</message>

<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

PORT TYPE DEFINITION - A port type groups a set of operations into

a logical service unit.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ -->

<!-- portType implemented by the Hello BPEL process -->

<portType name=*"Hello"*>

<operation name=*"process"*>

<input message=*"tns:HelloRequestMessage"*/>

<output message=*"tns:HelloResponseMessage"*/>

</operation>

</portType>

<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

PARTNER LINK TYPE DEFINITION

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ -->

<plnk:partnerLinkType name=*"Hello"*>

<plnk:role name=*"HelloProvider"* portType=*"tns:Hello"*/>

</plnk:partnerLinkType>

<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

BINDING DEFINITION - Defines the message format and protocol details

for a web service.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ -->

<binding name=*"HelloBinding"* type=*"tns:Hello"*>

<soap:binding style=*"document"* transport=*"http://schemas.xmlsoap.org/soap/http"*/>

<operation name=*"process"*>

<soap:operation soapAction=*"http://eclipse.org/bpel/sample/process"*/>

<input>

<soap:body use=*"literal"*/>

</input>

<output>

<soap:body use=*"literal"*/>

</output>

</operation>

</binding>

<!-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

SERVICE DEFINITION - A service groups a set of ports into

a service unit.

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ -->

<service name=*"HelloService"*>

<port binding=*"tns:HelloBinding"* name=*"HelloPort"*>

<soap:address location=*"http://localhost:8080/ode/processes/Hello"*/>

</port>

</service>

</definitions>