



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

# Εισαγωγή στα Δίκτυα Υπηρεσιών

**Assisting Lecture 11 - Apache ODE, Eclipse BPEL  
Designer Examples**

Μύρων Παπαδάκης  
Τμήμα Επιστήμης Υπολογιστών

# **Introduction to Service Networks**

## **CS-592 – Spring 2015**

Assisting Lecture : Eclipse BPEL Designer

Myron Papadakis (myrpap@gmail.com)

# Outline

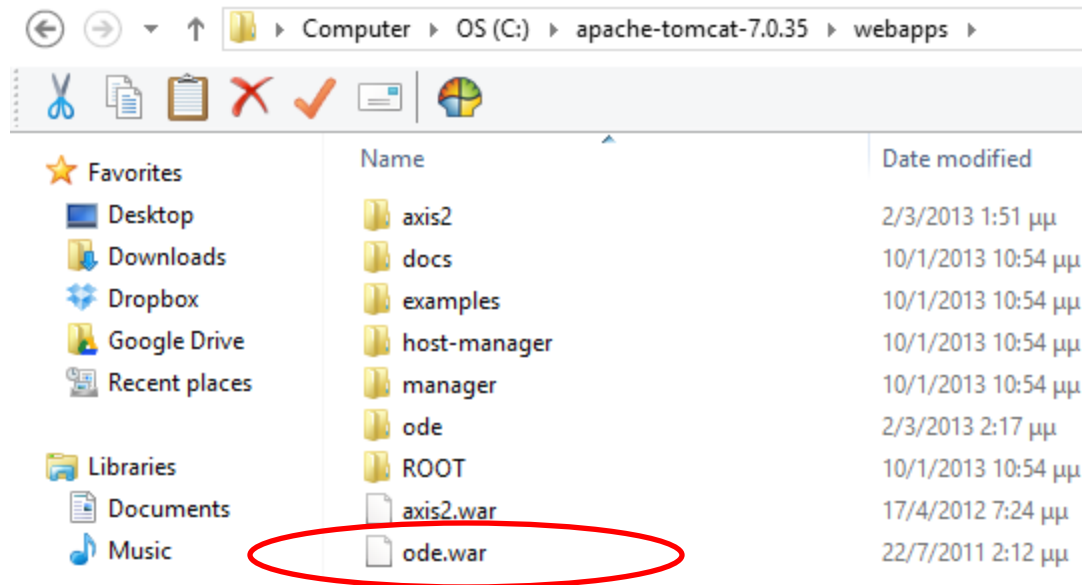
- Apache Ode
- Eclipse BPEL Designer

# Apache Ode

- Apache ODE (Orchestration Director Engine) is an open source BPEL engine.
- Full support for WS- BPEL 2.0 OASIS standard.
- Hot-deployment of processes.
- Management interface for processes, instances and messages.

# Apache Ode

- Download latest stable release: ODE 1.3.5 war distribution from <http://ode.apache.org/getting-ode.html>
- Installation:
  - Unzip package. Copy this file to Tomcat's **webapp**.
  - Start Tomcat and ODE should be up and running.



# Testing Apache Ode

<http://localhost:8080/ode>



The screenshot shows a web browser window with three tabs: 'Apache Tomcat/7.0.35', 'Apache Tomcat/7.0.35', and 'Apache ODE | Home'. The address bar shows 'localhost:8080/ode/'. Below the browser window, the Apache ODE web interface is displayed. The page has a dark header with the text 'Apache ODE' and navigation links for 'Apache ODE', 'User Guide', and 'Developer Guide'. A navigation menu below the header includes 'Home', 'Processes', 'Instances', 'Deployment', and 'Deployment Browser'. The main content area features a large blue banner with the text 'Apache ODE' and a description: 'Apache ODE (Orchestration Director Engine) executes business processes written following the WS-BPEL standard. It talks to web services, sending and receiving messages, handling data manipulation and error recovery as described by your process definition. It supports both long and short living process executions to orchestrate all the services that are part of your application.' To the right of the text is a silhouette of a person holding a red umbrella. Below the banner are three icons: a house for 'Apache ODE', a document for 'User Guide', and a document for 'Developer Guide'. At the bottom of the page, there is a section titled 'Summary of Instances'.

# Deploying a process to Apache Ode

- <http://ode.apache.org/creating-a-process.html>
- Each deployment is a directory with a relevant deployment artifacts
  - One or more processes (.bpel)
  - WSDL and XSD files
  - *Deployment descriptor (**deploy.xml**)*: configuration of services that processes uses. For each process,
    - partnerlinks are bind to concrete WSDL services
    - <receive> is correlated to <provide> element
    - <invoke> is correlated to <invoke> element
- To deploy, copy the **whole** directory containing your artifacts in the
  - “<tomcat-home>/webapps/ode/WEB-INF/processes”

# Eclipse

The Eclipse logo features a dark blue circle with a white ring around its center, set against a light blue background with radiating lines. The word "eclipse" is written in white lowercase letters across the center of the circle.

eclipse

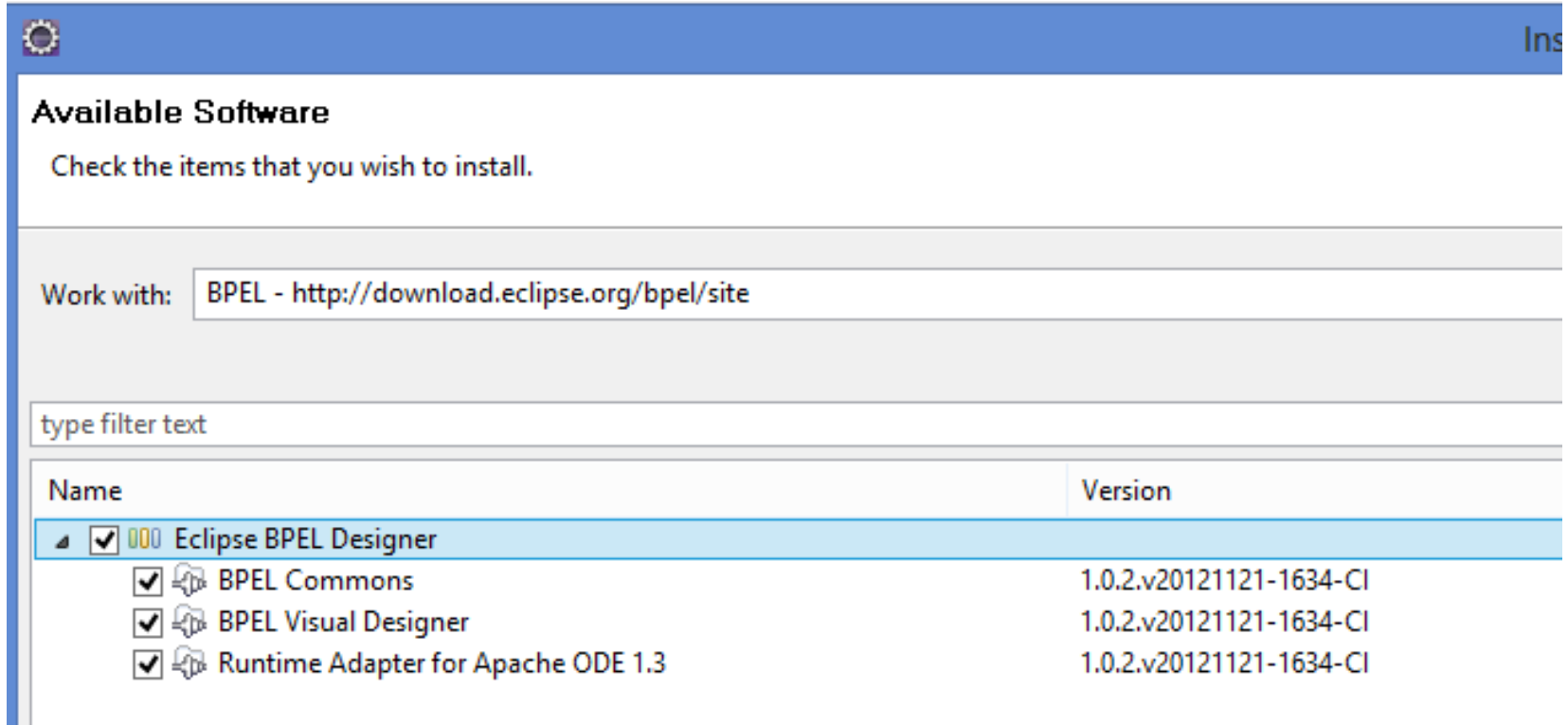




# Eclipse and BPEL

- Installing and Configuring Eclipse with the BPEL Designer
- For installing plug-ins and additional packages go to: **Help → Install New Software**
- In the text field with label “Work With” enter <http://download.eclipse.org/bpel/site> and from the list select “**Eclipse BPEL Designer**”
- After installation of new packages, **Eclipse IDE must be restarted.**
- In order to check installed software, plug-ins and tools, try: “**Help**” → “**About Eclipse**” and click “**Installation Details**”

# Eclipse BPEL Designer







**Available Software**

Check the items that you wish to install.

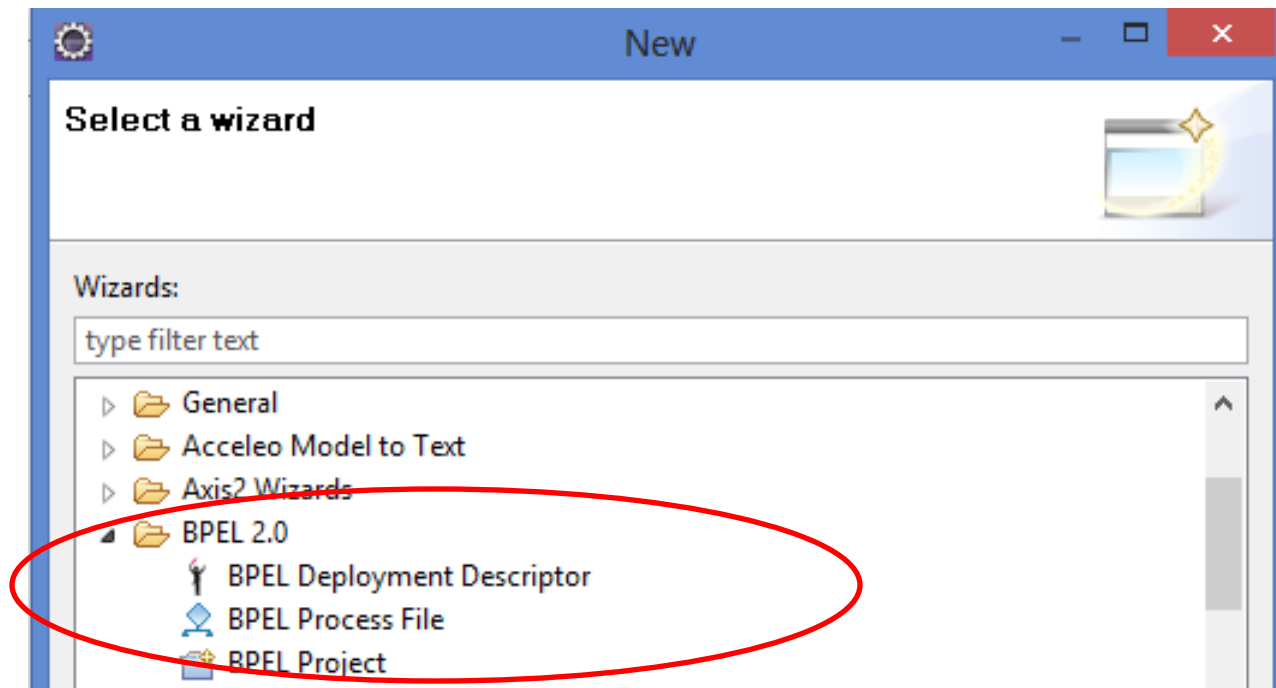
Work with: BPEL - <http://download.eclipse.org/bpel/site>

type filter text

Name	Version
<input checked="" type="checkbox"/>  Eclipse BPEL Designer	
<input checked="" type="checkbox"/>  BPEL Commons	1.0.2.v20121121-1634-CI
<input checked="" type="checkbox"/>  BPEL Visual Designer	1.0.2.v20121121-1634-CI
<input checked="" type="checkbox"/>  Runtime Adapter for Apache ODE 1.3	1.0.2.v20121121-1634-CI

# Eclipse and BPEL

- To verify the installation has been successful, after the restart, click on File → New → Other (in order to create a new project)
- Verify that there is a BPEL 2.0 Entry in the Wizard



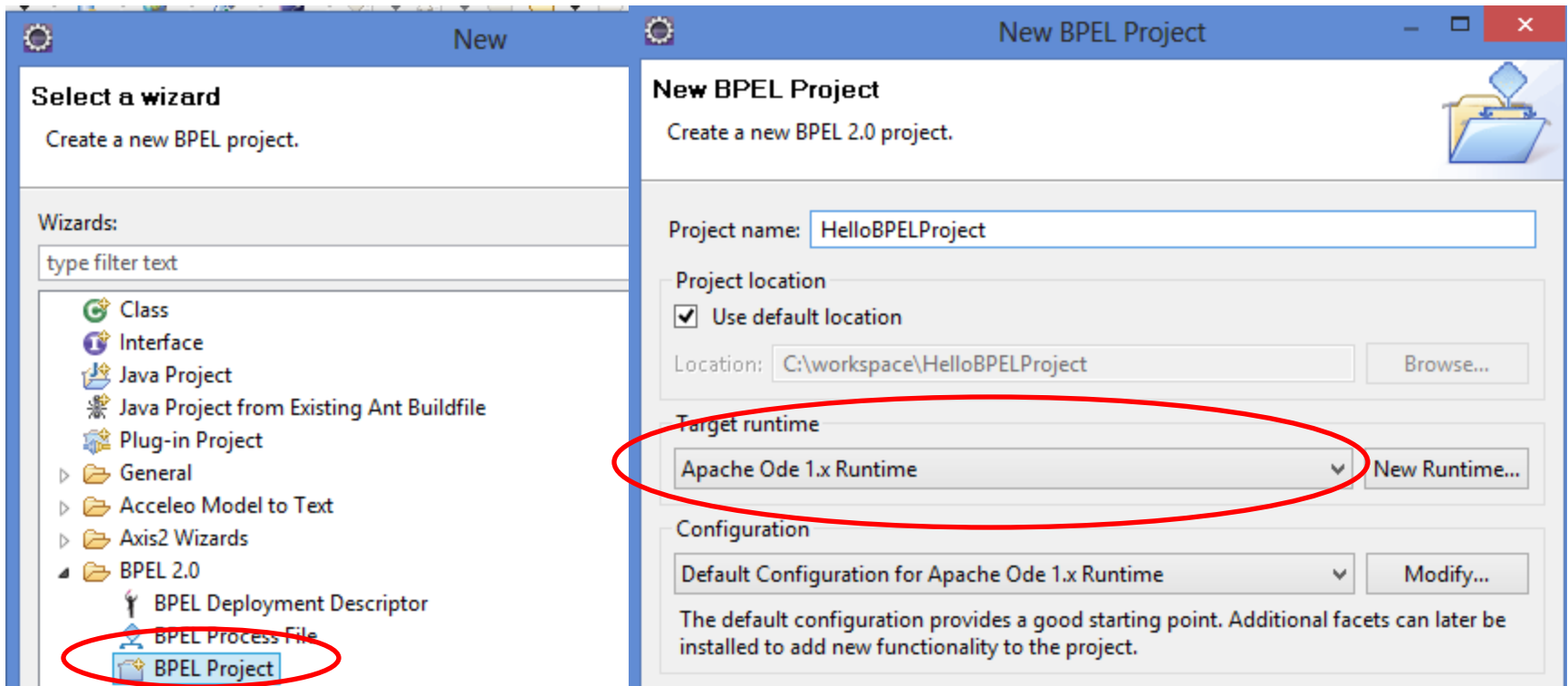
# Eclipse and Apache Ode

- Setup a ODE Server inside Eclipse
  - “Window→**Preferences**->**Server**->**Server Runtime**” and “**Add**”

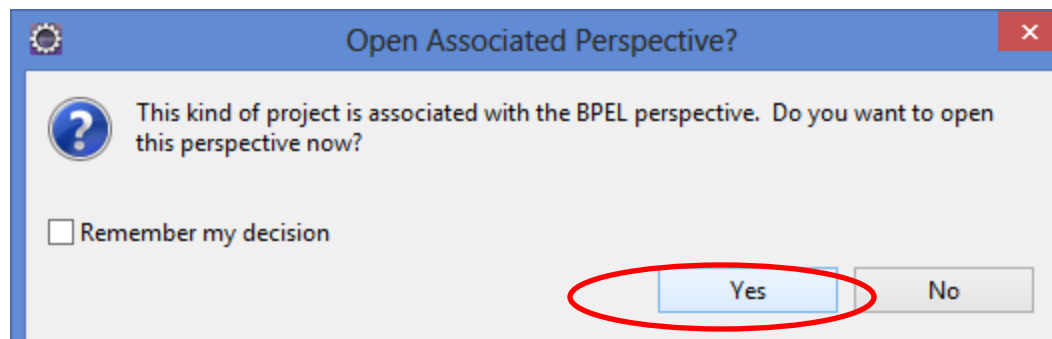
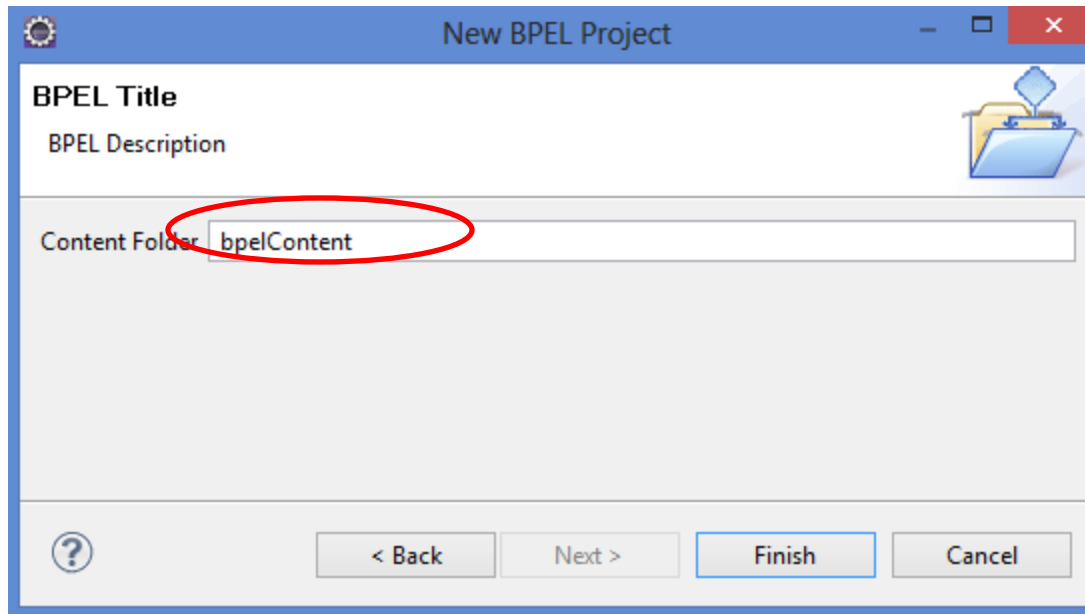
The screenshot displays the Eclipse IDE interface during the setup of a new server runtime environment. The main window is titled "New Server Runtime Environment" and is divided into two panes. The left pane, titled "New Server Runtime Environment", shows a tree view of runtime environments under the "Apache" folder, with "Apache Ode 1.x Runtime" selected. The right pane, titled "New Apache Ode 1.x Runtime Runtime", shows configuration fields for the new runtime environment. The fields are: "JRE" set to "Default JRE", "ODE's home directory" set to "C:/apache-tomcat-7.0.35/webapps/ode", "Tomcat's home directory" set to "C:/apache-tomcat-7.0.35", "Server address" set to "localhost", and "Port" set to "8080". A red circle highlights the "ODE's home directory" and "Tomcat's home directory" fields. Below the dialog, the "Servers" view shows a list of servers: "Ode v1.x Server at localhost [Stopped]" and "Tomcat v7.0 Server at localhost [Started, Sy]".

# Eclipse BPEL Project

- File → New → Other → BPEL 2.0 → BPEL Project

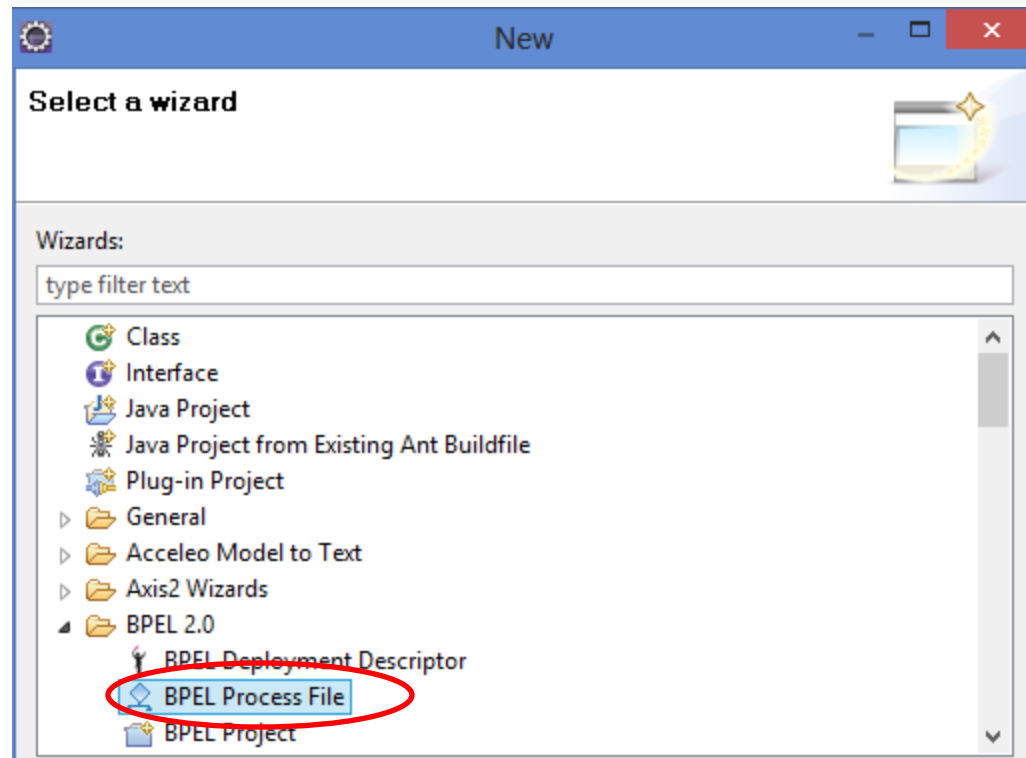


# Eclipse BPEL Project



# BPEL File

- Right click on the folder of your choice (i.e /bpelcontent)
  - ***New → Other → New BPEL Process File***



# BPEL File

**Create a BPEL Process File**  
Create a new BPEL file (BPEL 2.0).

Creation Mode: Create a BPEL process from a template

Process Name: HelloWorld

Namespace: http://helloworld

Abstract Process

Enter or select the parent folder:  
HelloBPELProject/bpelContent

- HelloBPEL
- HelloBPELProject
  - .settings
  - bpelContent
- Servers
- Test
- TravelReservation

File name: HelloWorld.bpel

Advanced >>

**Use a Creation Template**  
Create a BPEL process from a template.

Template: Synchronous BPEL Process

Generates an empty BPEL process. Only receive and reply activities are placed in the process body. The caller will block until all the steps in the process have completed. A client interface is generated.

Template Properties

Service Name: HelloWorldService

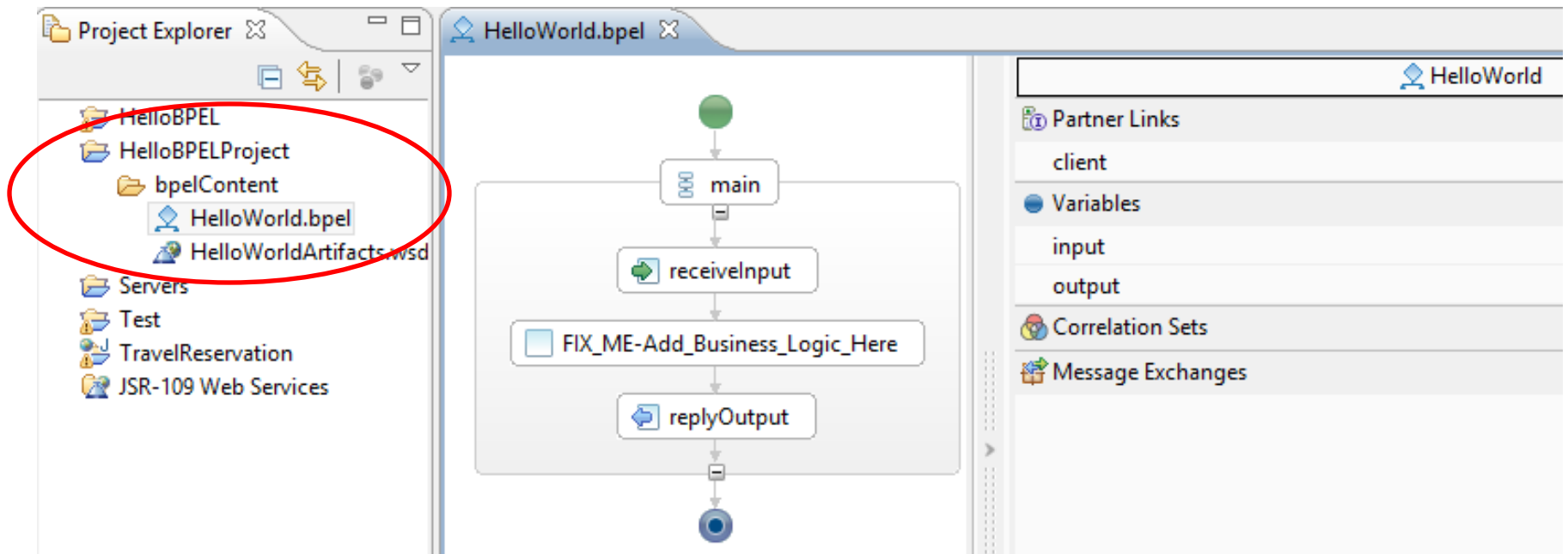
Port Name: HelloWorldPort

Service Address: http://localhost:8080/HelloWorld

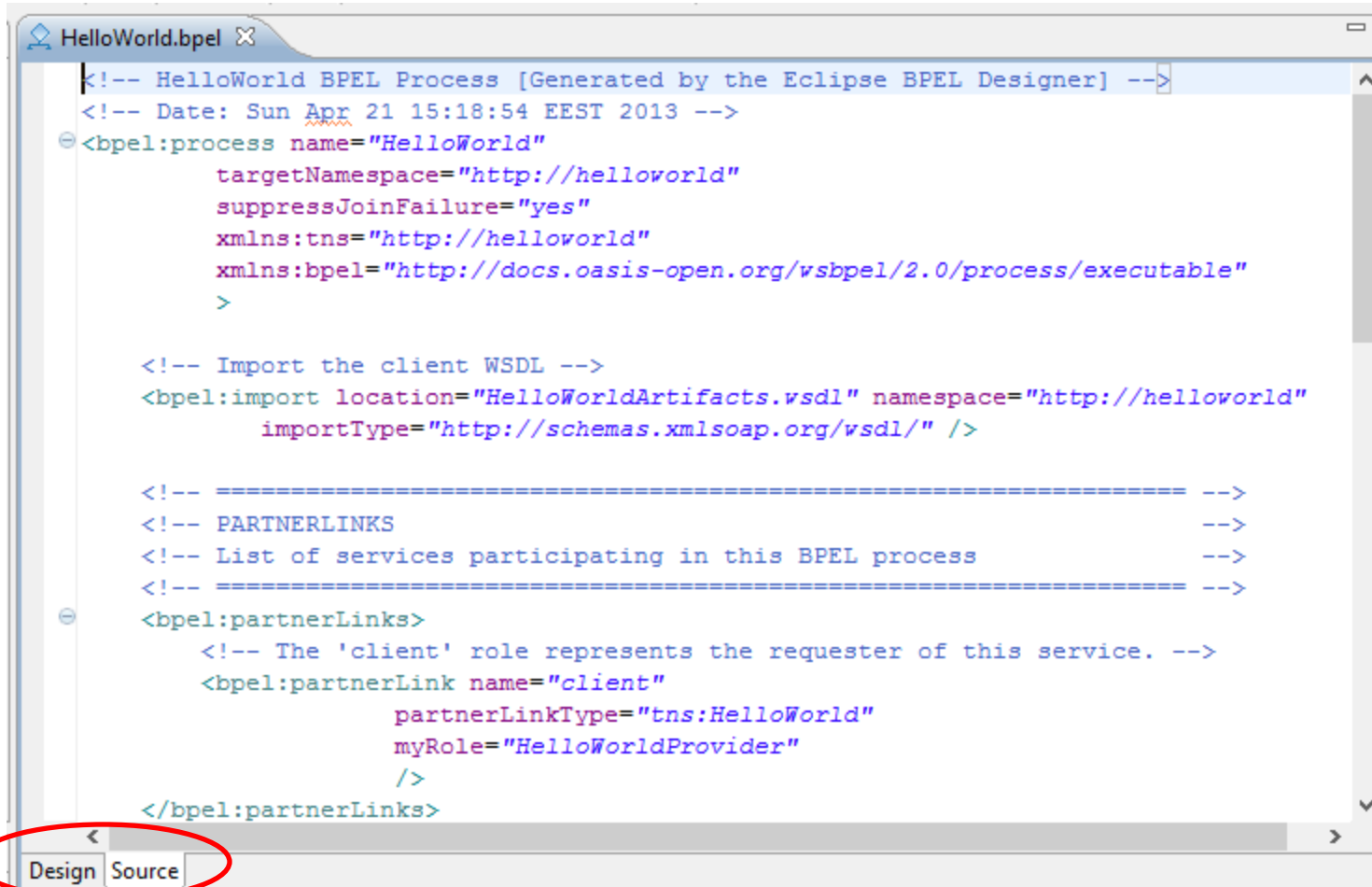
Binding Protocol: SOAP



# BPEL File



# BPEL File



```
<!-- HelloWorld BPEL Process [Generated by the Eclipse BPEL Designer] -->
<!-- Date: Sun Apr 21 15:18:54 EEST 2013 -->
<bpel:process name="HelloWorld"
  targetNamespace="http://helloworld"
  suppressJoinFailure="yes"
  xmlns:tns="http://helloworld"
  xmlns:bpel="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
>

  <!-- Import the client WSDL -->
  <bpel:import location="HelloWorldArtifacts.wsdl" namespace="http://helloworld"
    importType="http://schemas.xmlsoap.org/wsdl/" />

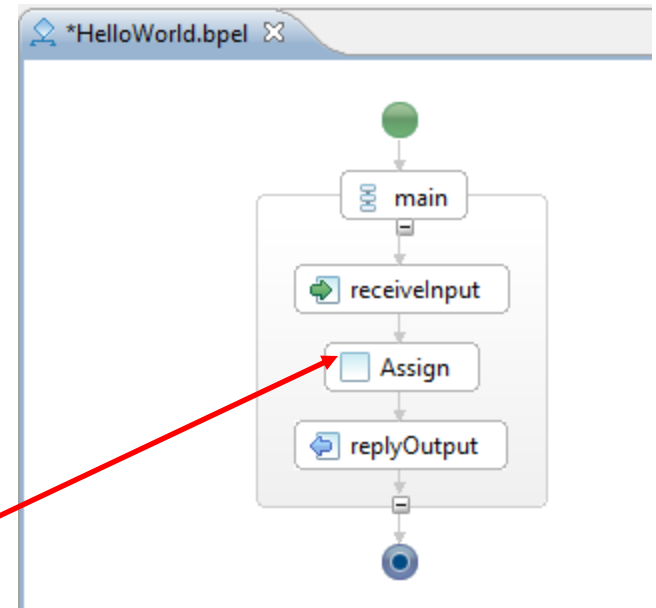
  <!-- ===== -->
  <!-- PARTNERLINKS -->
  <!-- List of services participating in this BPEL process -->
  <!-- ===== -->

  <bpel:partnerLinks>
    <!-- The 'client' role represents the requester of this service. -->
    <bpel:partnerLink name="client"
      partnerLinkType="tns:HelloWorld"
      myRole="HelloWorldProvider"
    />
  </bpel:partnerLinks>
```

Design Source

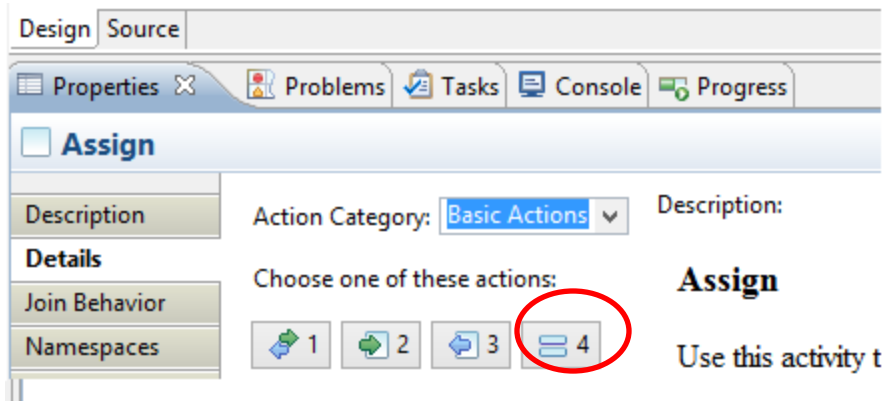
# Example

- We want to design an application that responds the client the same string that it send to the service provider.
- Therefore, we have to add an activity which assign the received input to the output.
- To do this, click on the Assign activity on the Actions folder of the Palette and drag it to the space between the receiveInput and replyOutput activities (if it is not already present)
- IF it is present simply rename the assign activity from “FIX\_ME-Add\_Business\_Logic\_Here” to “Assign”



# Example

- Right click on the **Assign** activity. Select **Show in properties**.
- In the Properties tab, select **Details**. Click on the **New** button to define an assign operator.
- Assign from **input->payload->input** to **output->payload->result**. An **Initializer** popup dialog appear. Click **Yes** to initialize the output variable and save the process.



# Example

Properties Problems Tasks Console Progress

**Assign**

Validate

Description

**Details** Variable to ? From: Variable To: Variable

- input : HelloWorldRequestMessage
- payload : HelloWorldRequest
  - input : string
- output : HelloWorldResponseMessage

- input : HelloWorldRequestMessage
- output : HelloWorldResponseMessage
  - payload : HelloWorldResponse
    - result : string

**Initializer**

Variable output doesn't have initializer. Should it be generated?

Yes No

Properties Problems Tasks Console Progress

**Assign**

Validate

Description

**Details** Fixed Value to Variable From: Fixed Value To: Variable

Variable to Variable

```
<tns:HelloWorldResponse xmlns:tns= "http://helloworld" xmlns:xsi= "  
<tns:result>tns:result</tns:result>  
</tns:HelloWorldResponse>
```

- input : HelloWorldRequestMessage
- output : HelloWorldResponseMessage
  - payload : HelloWorldResponse
    - result : string

# WSDL

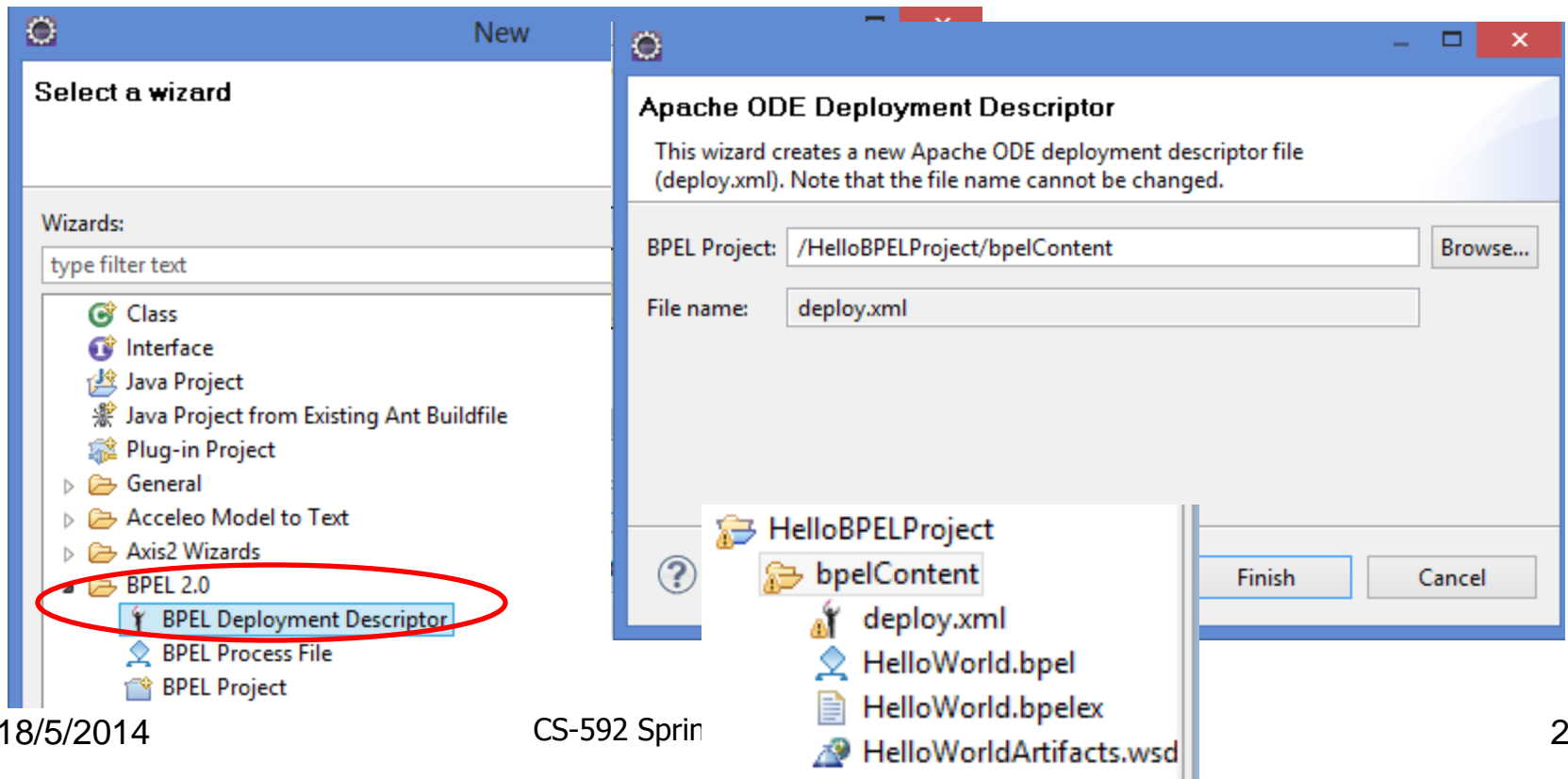
- The BPEL process is designed.
- Now we have to define the Service, Binding type and PortType address to run the HelloWorld service.
- Right click on the **HelloWorldArtifacts.wsdl** file, select **Open With->WSDL Editor**.

The screenshot displays an IDE interface with two windows. The top window, titled 'HelloWorldArtifacts.wsdl', shows a diagram where a 'HelloWorldService' box contains a 'HelloWorldPort' box with the address 'http://localhost:8080/H...'. An arrow points from this port to a 'HelloWorld' process box. The process box has an 'input' section with a 'HelloWorldRequest' payload and an 'output' section with a 'HelloWorldResponse' payload. The bottom window, titled 'port', shows the 'Properties' view for the 'HelloWorldPort'. The 'Address' field is circled in red and contains the text 'http://localhost:8080/processes/HelloWorld'. Other fields include 'Name: HelloWorldPort', 'Binding: HelloWorldBinding', and 'Protocol: SOAP'.

Section	Field	Value
General	Name:	HelloWorldPort
	Binding:	HelloWorldBinding
	Address:	http://localhost:8080/processes/HelloWorld
	Protocol:	SOAP

# Deploy BPEL in Apache Ode

- To let the Apache ODE know your declarations and designs.
- You have to add a descriptor. Right click on the BPEL\_HelloWorld project, select **New->Others->BPEL 2.0->Apache ODE Deployment Descriptor**.



# Deploy BPEL in Apache Ode

- In the Inbound Interfaces table, select the Partner Link as **client**, Associate Port as **HelloPort**. Click on Related Service then other fields will be automatically filled. Save the "deploy.xml" file.

Process HelloWorld - http://helloworld

General

This process is **activated**

Run this process in memory

**Inbound Interfaces (Services)**

The table contains interfaces the process provides. Specify the service, port and binding you want to use for each PartnerLink listed

Partner Link	Associated Port	Related Service	Binding Used
client	-- none --	-- none --	-- none --

**Outbound Interfaces (Invokes)**

The table contains interfaces the process invokes. Specify the service, port and binding you want to use for each PartnerLink listed

Partner Link	Associated Port	Related Service	Binding Used
--------------	-----------------	-----------------	--------------

**Inbound Interfaces (Services)**

The table contains interfaces the process provides. Specify the service, port and binding you want to use for each PartnerLink listed

Partner Link	Associated Port	Related Service	Binding Used
client	HelloWorldPort	{http://hello...	HelloWorld...



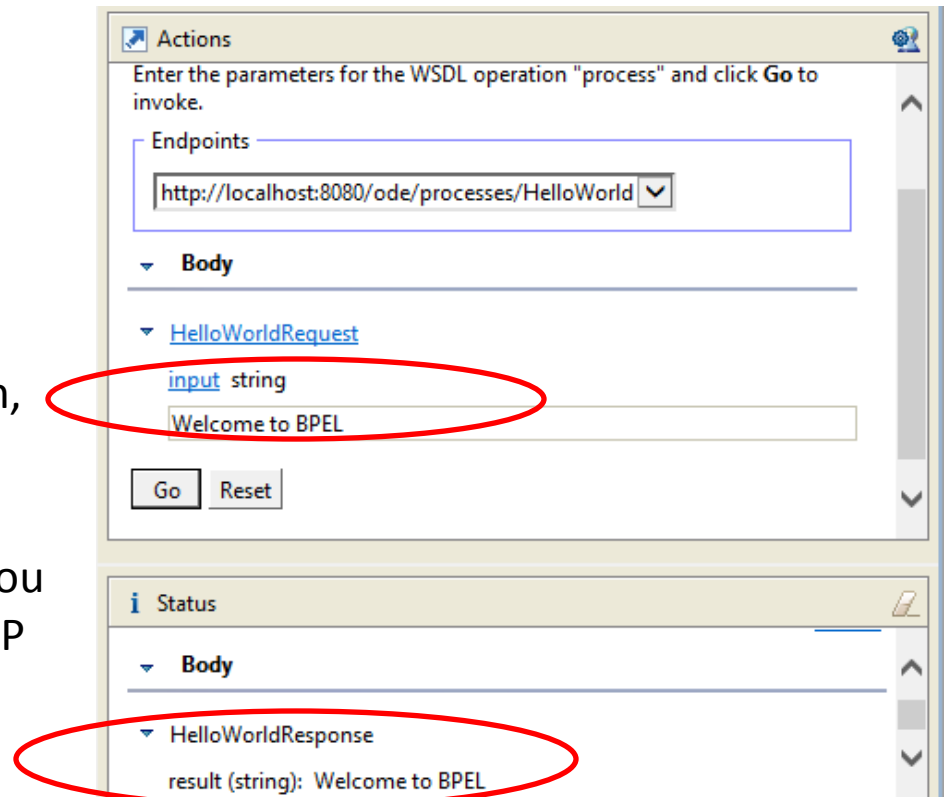
# Time to run the process

- Now you have finished your designs and it's time to run the process.
- However, the ODE Server on the Eclipse could not recognize your BPEL process because it is designed after the server's set up.
- Therefore you have to **restart your Eclipse** (File → Restart)
- After the restart, right click on the **Ode v1.x Server at localhost**, select **Add and Remove Projects**
- Select the **BPEL\_HelloWorld/HelloWorld** in the Available projects box and click **Add**. Then, click **Finish**.
- Start the ODE server. If the deployment was successful, you will see something similar to the following output in the console tab.

```
15:57:19,709 INFO [DeploymentPoller] Deployment of artifact HelloBPELProject successful: [{http://helloworld}HelloWorld-6]
```

# Testing the process

- We will test the "Hello World" deployment with Web Services Explorer provided by Eclipse.
- Right click on the **HelloWorldArtifacts.wsdl** file. Select **Web Services->Test with Web Services Explorer**.
- Select the operation **process**. Then, type **"Welcome to BPEL"** in the input textbox.
- If your deployment was success, you will get the same string in the SOAP response



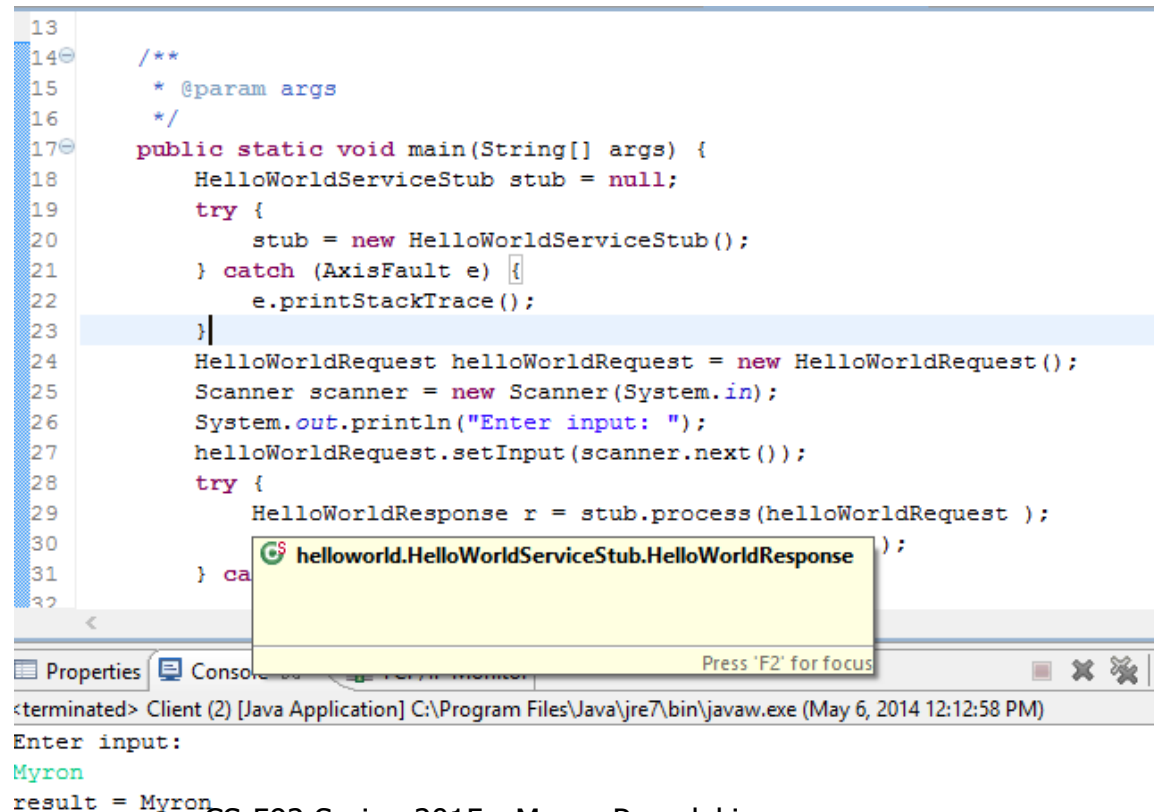
## Option 2

- Deploy to ODE in Tomcat
- Create a folder in **tomcat/webapps/ode/WEB-INF/processes**
- Copy the **.bpel** file, **.wsdl** file and the ODE **deploy.xml**
- Restart Tomcat and visit **http://localhost:8080/ode**
- An extra file will be created with the extension **.deployed** inside

# Clients

- Since the bpel process is itself a web service, clients can be created in the same way as demonstrated in the previous labs (stubs,etc)

```
13
14- /**
15-  * @param args
16-  */
17- public static void main(String[] args) {
18-     HelloWorldServiceStub stub = null;
19-     try {
20-         stub = new HelloWorldServiceStub();
21-     } catch (AxisFault e) {
22-         e.printStackTrace();
23-     }
24-     HelloWorldRequest helloWorldRequest = new HelloWorldRequest();
25-     Scanner scanner = new Scanner(System.in);
26-     System.out.println("Enter input: ");
27-     helloWorldRequest.setInput(scanner.next());
28-     try {
29-         HelloWorldResponse r = stub.process(helloWorldRequest);
30-     } catch (AxisFault e) {
31-         e.printStackTrace();
32-     }
33- }
```

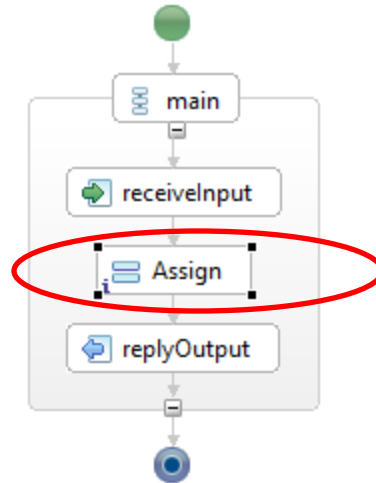


Properties Console Press 'F2' for focus

<terminated> Client (2) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (May 6, 2014 12:12:58 PM)

Enter input:  
Myron  
result = Myron

# Modifying the Hello BPEL Project



The screenshot shows the configuration window for the Assign activity. The window has tabs for Properties, Console, and TCP/IP Monitor. The Assign activity is selected, and the configuration is as follows:

- Description:**  Validate
- Details:** Fixed Value to Variable, Expression to Variable (selected)
- Join Behavior:** (empty)
- Namespaces:** (empty)
- Documentation:** (empty)
- From:** Expression
- To:** Variable
- Edit the associated XPath Expression:** `concat("Hello ", $input.payload/tns:input)` (highlighted with a red oval)
- Query:** tns:result
- Ignore Missing Source Data:**
- Keep Source Element Name:**

Buttons for New, Delete, Move Up, and Move Down are visible at the bottom of the configuration area.

# Modifying the Hello BPEL Project

The screenshot shows an IDE with the following project structure on the left:

- HelloBPELProject
  - bpelContent
    - deploy.xml
    - HelloWorld.bpel
    - HelloWorld.bpelex
    - HelloWorldArtifacts.wsdl
  - HelloBPELProjectClient
    - JAX-WS Web Services
    - Deployment Descriptor: HelloB
    - Java Resources
      - src
        - client
          - Client.java
        - helloworld
        - HelloWorldServiceS
      - Libraries
      - JavaScript Resources
      - build
      - WebContent
      - build.xml

The main editor displays the code for `Client.java`:

```
1 package client;
2
3 import java.rmi.RemoteException;
4
11
12 public class Client {
13
14     /**
15      * @param args
16      */
17     public static void main(String[] args) {
18         HelloWorldServiceStub stub = null;
19         try {
20             stub = new HelloWorldServiceStub();
21         } catch (AxisFault e) {
22             e.printStackTrace();
23         }
24         HelloWorldRequest helloWorldRequest = new HelloWorldRequest();
25         Scanner scanner = new Scanner(System.in);
26         System.out.println("Enter input: ");
27         helloWorldRequest.setInput(scanner.next());
28         try {
29             HelloWorldResponse r = stub.process(helloWorldRequest);
30             System.out.println("result = "+r.getResult());
31         } catch (RemoteException e) {
```

Properties Console TCP/IP Monitor  
<terminated> Client (2) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (May 6, 2014 12:27:43 PM)

```
Enter input:
Myron
result = Hello Myron
```

# References

- <http://www.eclipse.org/bpel/users/pdf/HelloWorld-BPELDesignerAndODE.pdf>
- [http://www-inf.int-evry.fr/cours/WebServices/TP\\_BPEL/helloworld.html](http://www-inf.int-evry.fr/cours/WebServices/TP_BPEL/helloworld.html)
- <http://www.soapui.org/Working-with-soapUI/getting-started.html>

# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης





# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Δια Βίου Μάθηση**» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



**Σημειώματα**

# Σημείωμα αδειοδότησης

- Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>

## •Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

•Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

# Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Κρήτης, Μύρων Παπαδάκης. «**Εισαγωγή στα Δίκτυα Υπηρεσιών. Assisting Lecture 11 - Apache ODE, Eclipse BPEL Designer Examples**». Έκδοση: 1.0. Ηράκλειο/Ρέθυμνο 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://elearn.uoc.gr/course/view.php?id=416/>