



# Semantic Web Services Tutorial

Hands-on session

First Asian Semantic Web Conference (ASWC 2006),  
3-7 September 2006, Beijing China.



[www.deri.at](http://www.deri.at)



[www.wsmx.org](http://www.wsmx.org)

## **Installations and Configurations**

### **1. WSMX installation:**

In this section we explain a step by step WSMX installation on a Windows operating system. You can download WSMX available on the CVS on the SourceForge server (WSMX hosting site -<http://sourceforge.net/projects/wsmx>).

#### **1.1. Pre-installation:**

Following software is necessary to run WSMX:

##### **1.1.1. J2SE SDK**

**Step 1.** Go to the java directory in tutorial CD to install Java(TM) 2 SDK, Standard Edition 1.5.0.

Step2. Setup the environment variable JAVA\_HOME and point it to where the java has been installed (say c:\java).

**Step 3.** Run the installer j2sdk-1\_5\_0-windows-i586-p.exe

##### **1.1.2. Tomcat**

**Step 1.** Go to the tomcat directory in the tutorial CD to get tomcat.zip file. Unzip the file in your hard drive (say at c:\tomcat) and setup the environment variable CATALINA\_HOME to c:\tomcat

**Step 2.** Run the startup.bat to start Tomcat and the shutdown.bat to stop Tomcat. These files can be run from the command line or found at; C:\tomcat\bin

**Step 3.** Open internet browser and type URL <http://localhost:8080>

#### **1.2. WSMX configuration:**

**Step 1.** Go to the wsmx directory in the tutorial CD to get the wsmx.zip file. Unzip it in your hard drive (say c:\wsmx)

**Step 2.** Go to PATH\wsmx (where you have placed the extracted files)

**Step 3.** Run the start.bat to run wsmx

**Step 4.** Open internet browser and type URL <http://localhost:8081>

If the WSMX server page is visible, WSMX has been installed and running successfully.

**Note:** WSMX is an ongoing development will have many updated versions in near future. The latest versions of the WSMX can be downloaded from <http://www.wsmx.org/downloads.html>

### 1.3. WSMT configuration:

**Step 1.** The wsmt folder in tutorial CD contains wsmt-1.3-gpl-windows-tutorial.zip and unzip it in your hard drive (say c:\wsmt).

**Step 2.** Go to c:\wsmt and run wsmt.exe

**Step 3.** Open SEE perspective of WSMT.

**Step 4.** On the left panel, right click on the "server" and open "properties" to set the IP and ports. In this case, IP is 'localhost', HTTPport is 8080 and servlet port is 8050

---

## 2. Deploying end-point services:

The end-point Web Services provides the implementation of Service Provider (Moon Company in this case). We do not need to go into the implementation details in this tutorial.

The implementation of the end-point services has been provided in a single file "moon.war" and "moonlib.jar" which can be found in the CD at path "usecase\end-point services".

These implementation files are to be deployed on the tomcat server. Following steps are to be followed:

**Step 1.** Start the tomcat server using \TOMCAT\_HOME\bin\startup.bat

**Step 2.** Open internet browser and type <http://localhost:8080> (Please make sure that tomcat is started on port 8081 to avoid any conflict with wsmx running). The Apache tomcat server web console should be visible.

**Step 3.** Click on the Tomcat Manager. Supply the login and password for the tomcat administrator. Note: the tomcat administrator user information can be found in "TOMCAT\_HOME\conf\tomcat-users.xml".

If the user information is not available or to be provided, add following line

```
<user username="omair" password="tomcat" roles="manager,admin"/>
```

After Update, do not forget to restart Tomcat server.

**Step 4.** Go to the "Local WAR file installation" section. Click on browse and select the "moon.war" file from "tutorial CD\end-point services". After selecting the file, click on Deploy.

**Step 5.** Copy the "moonlib.jar" file to \TOMCAT\_HOME\webapps\axis\WEB-INF\lib

**Step 6.** Restart the Tomcat server. Open internet browser and go to URL <http://localhost:8080/moon/services>



**Step 7.** If the OMService, CRMService and RNetService are available in the list of available Web Services, this means end-point services have successfully been deployed.

---

## **Running WSMX and its use-case**

### **3. Browsing ontologies, goals and Web Service Descriptions in WSMT**

After the WSMT is started, on the left side of WSMT GUI, the “Navigator” view shows all the projects in your workspace. For this tutorial session, the WSMT contains all the WSMO Ontologies, Web Services and Goals needed to execute the use case in the project called “WSMO Tutorial”. The workspace also contains some other WSML examples that you can use to understand the WSMT functionality.

There are two main editors for viewing and manipulating WSMO descriptions in the WSML formalism. These are the  WSML Visualizer and the  WSML Text Editor and instructions for using them are provided in sections 3.1 and 3.2.

#### **3.1 WSML Visualizer**

The WSML Visualizer provides a graphical representation of the information contained within a WSML document, using a directed graph. In this part of the tutorial we will look at the Simpsons ontology, which is a schema and some instances related with the Simpsons television show.

**Step 1.** Open “The Simpsons” project by clicking the plus on the left hand side of the folder icon.

**Step 2.** Right click on “the-simpsons-ontology.wsml” in the navigator view, choose “Open With → WSML Visualizer”

In the visualizer you will see the blue node at the centre that represents the ontology contained within this WSML file. The other nodes in the representation are concepts (yellow) and instances (red). The graph is automatically layed-out using a spring layout algorithm however the representation is fully interactive and by clicking on a node and dragging it the representation can be moved to obtain the best layout.

**Step 3.** Manipulate the graph using the dragging feature to get the best representation.

The visualization provided in this editor is fully interactive and allows the user to edit the WSMO Ontology, Web Service, Goal or Mediator being visualized. In the following steps we will add a new concept, some attributes and an instance to the Simpsons ontology. The concept that we shall add will represent the concept of a public park and the instance will represent the park located in Springfield.

**Step 4.** Right click on the yellow concept node called "place" and choose "Add → Sub Concept"

**Step 5.** In the "New Concept ID" dialog enter "public\_park" and select OK (NOTE: a new concept node is added to the visualization).

The visualizer separates the complexities in the ontology into multiple levels, in order to start manipulating the attributes of a concept we need to "step down" to the concept semantic level. We do this by double clicking on a concept.

**Step 6.** Double click on the "public\_park" concept node (NOTE: the view is constrained to information about the selected concept).

**Step 7.** Right click on the "public\_park" concept node and choose "Add → Attribute"

**Step 8.** In the "New Attribute ID" dialog enter "hasLocation" and select OK (NOTE: we now have an error in our ontology as the attribute hasLocation has no range).

**Step 9.** Right click on the brown "hasLocation" attribute node and choose "Add → Attribute Range (Concept)"

**Step 10.** In the "Select a Concept" dialog, expand the "place" concept in "The Simpsons Ontology", choose "town" and select OK. (NOTE: the error in the ontology is now removed).

Now that we have created our concept and added an attribute we can create an instance of that concept. Note that there are two attributes available for use on the instance. The first is the hasLocation that we just created and the second is the hasName attribute inherited from the "place" super concept of "public park".

**Step 11.** Right click on the "public\_park" concept again and choose "Add → Instance"

**Step 12.** In the "New Instance ID" dialog enter "springfield\_public\_park" and select OK

**Step 13.** Double click on the new "springfield\_public\_park" instance node to move to the instance semantic level.

We have created our instance, but we still need to assign values to the attributes of the instance.

**Step 14.** Right click on "springfield\_public\_park" and choose "Add Attribute"

**Step 15.** Choose the "hasLocation" attribute from the "public\_park" concept and select OK. (NOTE: You will automatically be shown the "Select an Instance" dialog).

**Step 16.** Choose the "Springfield" instance from the "town" concept and select OK.

**Step 17.** Right click on "springfield\_public\_park" and choose "Add Attribute"

**Step 18.** Choose the "hasName" attribute from the "place" concept and select OK (NOTE: You will automatically be shown the "Specify a Data Value" dialog).

**Step 19.** In the Data Value text field enter “Springfield Public Park” and select OK.

**Step 20.** Save the ontology by using the Ctrl-S shortcut or “File → Save”.

### 3.2 WSML Text Editor

The WSML Text Editor shows the contents of a WSML document, in the WSML human readable syntax, to the user with syntax highlighting, content assistance and many other features that help advanced users who are familiar with WSML to create and manage their WSML documents more efficiently.

**Step 21.** Right click on “the-simpsons-ontology.wsml” in the navigator view, choose “Open With → WSML Text Editor”

**Step 22.** Use the Ctrl-F shortcut to bring up the find dialog

**Step 23.** In the find enter “public\_park” and click Find.

You can now see the human readable syntax of the concept that we created in the visualizer.

**Step 24.** In the find dialog click Find again.

You can now see the human readable syntax of the instance that we created in the visualizer.

### 3.3 The Moon WSML Files

**Step 25.** Browse the files located in the “WSMO Tutorial” project with the editor of your choice and become familiar with the Ontologies, Web Services and Goals.

---

## 4. Getting familiarized with WSMX

### 4.1. WSMX Discovery

**Step 1.** Run the Web Service Modeling Toolkit (WSMT) by double-clicking the wsmt.exe (if WSMT is already running, this step is not required).

**Step 2.** On the left side of WSMT GUI, select the project named “Discovery Tutorial” in “navigator” dialog, which contains some WSML Web Service descriptions and Goals.

**Step 3.** Select and right click on the Web Service descriptions “WebServiceA.wsml, WebServiceB.wsml and WebServiceC.wsml” in the menu, select the item “SEE” and in the sub-menu, click “Store”. In the same way store all the Web Service descriptions that are required to be available for discovery.















**Step 4.** Select and right click on the Goal “goal.wsml”, in the menu, select the item “SEE” and in the sub-menu, click “Achieve Goal”. This will execute the goal to find


out any Web Service descriptions available according to the requirements provided in the Goal.

## 4.2. WSMX Data Mediation

Mappings are to be created between source ontology and target ontology.

Mapping process description:

- **Step 1:** Open the WSMX Data Mediator tool
  - As the WSMX Data Mediator tool is deployed as an Eclipse plug-in, you might need to activate the *WSMX Data Mediator* perspective. You can do this by going to: Window -> Open perspective -> Other... and choosing "*WSMX Data Mediator*" from the list
- **Step 2:** Load the two ontologies into the mapping tool, i.e. source ontology and target ontology by clicking on  in the Tool bar corresponding to the "Source ontology" view and "Target ontology" view, respectively.
- **Step 3:** Meaning of graphical symbols:
  -  - compound concept
    -  - the compound target concept the selected source is mapped to
    -  - the compound target concept that is suggested by the tool as a good mapping candidate for the selected source
  -  - primitive concept (concept having no attributes)
    -  - the primitive target concept the selected source is mapped to
    -  - the primitive target concept that is suggested by the tool as a good mapping candidate for the selected source
  -  - data type
    -  - the target data type the selected source is mapped to
    -  - the target data type that is suggested by the tool as a good mapping candidate for the selected source
  -  - attribute
    -  - the target attribute the selected source is mapped to
    -  - the target attribute that is suggested by the tool as a good mapping candidate for the selected source
- Select an item in source ontology. Please observe the suggestion made by the tool. Go to the " Status View" situated in the bottom part of the application and study the bases of this suggestion.

Map one item from the source with an item of the target by pressing  in the Tool bar. Please observe the updates in the view (either "Concept2Concept", "Attribute2Attribute", "Concept2Attribute", "Attribute2Concept" situated in the bottom part of the application).

---





## 5. Executing the Goal services

**Step 1.** Please make sure that WSMT, WSMX and Tomcat should be started already.

**Step 2.** Using WSMT, store the all the WSMX Web Services descriptions and Ontologies ("coreelements.wsml", "dictionary.wsml", "MoonOntology.wsml", "purchaseorderconfirmation.wsml", "purchaseorderrequest.wsml" and "WSMoon.wsml") available under "WSMO Tutorial".


**Step 3.** Please create a set of mappings between the "PurchaseOrderRequest" (purchaseorderrequest.wsml) and "MoonOntology" (MoonOntology.wsml) ontologies. The wsml files are available in WSMT workspace, however it can also be found in "TutorialCD\usecase\_files\wsml\_files".

Mapping process description:


- **Step 3.1.** Open the "WSMX Data Mediator perspective"
  - As the WSMX Data Mediator tool is deployed as an Eclipse plug-in, you might need to activate the *WSMX Data Mediator* perspective. You can do this by going to: Window -> Open perspective -> Other... and choosing "*WSMX Data Mediator*" from the list
- **Step 3.2.** Load the two ontologies into the mapping tool
  - Load *PurchaseOrderRequest* as source ontology and *MoonOntology* as target ontology. For this click on  in the Tool bar corresponding to the "Source ontology" view and "Target ontology" view, respectively.
- **Our focus in this tutorial is on mapping the *PhysicalAddress* from *Pip3A4PurchaseOrderRequest* to the *shipping* and *billTo* address from the *CreateOrderRequest***
- **Step 3.3.** Select "*Pip3A4PurchaseOrderRequest*" in the source ontology. Please observe the suggestion made by the tool. Go to the " Status View" situated in the bottom part of the application and study the bases of this suggestion.
- **Step 3.4.** Map "*integer*" from the source with "*integer*" from the target by pressing  in the Tool bar. Please observe the updates in the "Concept2Concept" view (situated in the bottom part of the application).
- **Step 3.5.** Repeat the process for "*string*" data type.
- **Step 3.6.** Return to "*Pip3A4PurchaseOrderRequest*" concept and select it. Observe again the scores for this suggestion.  
*Question:* What is the reason for this change? *Response:* The structural factor plays an important role in the suggestion making mechanism. Because initially no mappings were created, the returned structural factor was 0.
- **Step 3.7.** Map "*Pip3A4PurchaseOrderRequest*" from the source and "*CreateOrderRequest*" from the target
  - Selecting the two concepts
  - Press map button () from the Tool bar

- Please note that the content in the source and target ontology views changed. We say that new contexts, relevant for the mapping stage we are in, were created by applying decomposition.
- **Step 3.8.** Since you are the domain expert ;), you know that the *PhysicalAddress* hides under the following hierarchy in the *PurchaseOrderRequest* (source) ontology: *PurchaseOrder* -> *ShipTo*->*PartnerDescription*->*PhysicalLocation*.

Select the *purchaseOrder* attribute in "*Pip3A4PurchaseOrderRequest*". The tool suggests to directly map it to the *shipTo* attribute in *CreateOrderRequest*. This is a viable solution (in the end you will get to map the actual address's fields) **BUT** it is better to think at reusability: it might be useful to have the *PhysicalAddress* (in *PurchaseOrderRequest* ontology) and the *Address* (in *MoonOntology*) mapped at the concept level. So, let's do the followings:



- Select the *purchaseOrder* attribute in "*Pip3A4PurchaseOrderRequest*" on the source
- Select the *CreateOrderRequest* on the target
- Press map button () from the Tool bar
- Please note that only the content in the source ontology view changed.
- **Repeat** the step for the following attributes from the source (in the target view *CreateOrderRequest* remains selected for each of these steps):
  - *shipTo* in *PurchaseOrder*
  - *partnerDescription* in *ShipTo*
  - *physicalLocation* in *PartnerDescription*
- Finally we arrived at the *physicalAddress* attribute in *PhysicalLocation*. The steps you have followed form what is called the **decomposition process** when several levels of "encapsulation" have to be mapped in order to reach the actual data you are interested in.
- Now, you can follow the tools suggestions and *physicalAddress* attribute from *PhysicalLocation* with *billTo* attribute from *CreateOrderRequest*.
- The decomposition updates again the source and target views and you can start mapping the actual field of the two addresses:
  - *addressLine1* to *street*
  - *cityName* to *city*
  - *globalCountryCode* to *countryCode*
  - *nationalPostalCode* to *postalCode*

As you could notice not all of the suggestions at this level are accurate. This is because the only help we have from the structural factor is that all these attributes are of type string – which does not help too much. So, the rest is done by the lexical factor which is extremely susceptible to things like common substrings for example.

- **Step 3.9.** A couple of bullets before it was something mentioned about reusability. Let's see what does this mean:
  - Return to the previous context for the target by pressing  button in the source and target views toolbar.
  - Try now to map *physicalAddress* attribute from *PhysicalLocation* with *shipTo* attribute from *CreateOrderRequest* (for the Moon scenario the billing and shipping address are the same).
  - The tool tells you that *PhysicalAddress* and *Address* are already mapped and asks you if you want to do some refinements on those mappings. If you are happy with what you did before you can simply press *No*.

Please observe the effect of your mappings in the "Concept2Concept", "Attribute2Attribute", "Concept2Attribute", "Attribute2Concept" views, as well as the generated mappings in the abstract mapping language ("Abstract Mapping Language" view) or in Flora-2 ("Flora-2" view).

The mappings can be stored/retrieved in/from an external repository. For this you have to do the followings:

- o Go in the menus to Windows -> Preferences -> WSMX Data Mediator and choose **HSQLDB (on disk)**. Out of the *repository URL* replace *mediationstorage* with the path on your harddrive where you copied the content of "usecase\_files/mappings" from tutorial CD. For example the URL should look something like "jdbc:hsqldb:file:c:/my\_path/mediationdb".
- o If you want to save your mapping in to the mapping storage (which is here a relational data base) press  from the toolbars of the above mentioned views.
- o To retrieve the complete set of mappings (to load them from db) press  from the same toolbars.

**Step 7.** Select the Goal "MoonGoal.wsml", execute it and observe the Semantic Web Service discovery, data mediation and choreography in WSMX console.

--- END ---