

## ESML Workflow Tutorial

This document gives a detailed description for how to use the ESML extension package. To use this tutorial, the OTIF core and OTIF ESML extension package needs to be installed and the instructions in the “*Getting Started*” and “*Getting Started with the ESML tool chain*” shortcut files should be followed to run the framework.

We start from the point when the OTIF framework has just started.

### ***Look at the ESML workflow to understand how it works***

This part gives a detailed overview about the ESML workflow model and how it works. It describes the different elements of the ESML tool chain.

This section is useful for those who wants to get to know the ESML tool chain and for those who wants to modify the ESML tool chain.

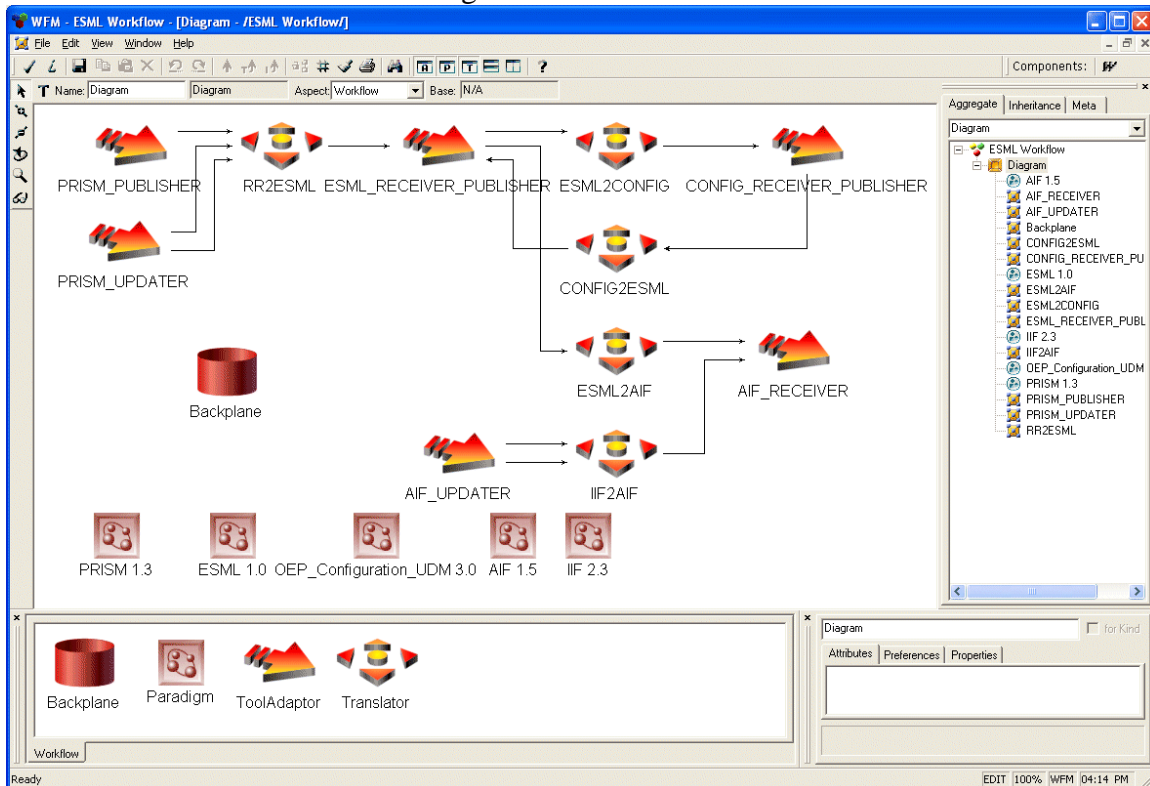
#### **1. Open the ESML workflow model file in GME.**

To start to work with the ESML workflow model please import the *ESML\_Workflow.xml* file from the Documents folder of the OTIF framework into the GME application or open the *ESML Workflow* file under *Start Menu/Programs/OTIF/Workflows* folder. GME is the workflow model editor.

Note: if you have problem with importing the XML file please check that the *WFM.xmp* workflow paradigm registered correctly from the Documents folder of the OTIF framework in GME.

Click on the *ESML Workflow* element in the GME Object Inspector window and open the *Diagram* element.

You should see the following:



**ESML workflow diagram in GME**

The model shows the ESML workflow that contains five translators and six types of tool adaptor.

The tool chain has six tool adaptors:

- (1) *PRISM\_PUBLISHER*,
- (2) *PRISM\_UPDATER*,
- (3) *CONFIG\_RECEIVER\_PUBLISHER*,
- (4) *ESML\_RECEIVER\_PUBLISHER*,
- (5) *AIF\_UPDATER* and
- (6) *AIF\_RECEIVER*.

Note: For each tool adaptor the same, generic tool adaptor (GTA) is used, but started with a different switch, effectively turning it into a different *type* of tool adaptor. Note that GTA is available both as a (Windows) application and as an Eclipse plug-in.

After starting the *Generic\_ToolAdaptor.exe* application or the Eclipse GTA plug-in and logging in as one of the type of the tool adaptors which is defined here in the ESML workflow model, the application or the plug-in will act as this tool adaptor and can do anything like the specified tool adaptor shows in the ESML workflow model.

The arrows on the model represent document flows. Each document flow has a type, which is assigned from the set of available document types enumerated at the bottom as “Paradigm” atoms (e.g. *ESML 1.0*, etc.). In GME, these are sets, containing the document flow edges as elements (and this containment relationship represents the typing information). Using the set-visualization mode of GME, one can select a paradigm, and the window will show the shows belonging to that paradigm.

The *PRISM\_PUBLISHER* can publish *PRISM* documents, which will be translated by the *RR2ESML* translator into *ESML* documents.

The translated documents can be fetched by an *ESML\_RECEIVER\_PUBLISHER* tool adaptor.

The *PRISM\_UPDATER* can publish two documents: a *PRISM* and an *ESML* document. In this case the *RR2ESML* translator will merge the input (published) *ESML* document with the data of the *PRISM* document and produce another *ESML* document which can be the original, input document updated with new data. The updated document can be fetched by the *ESML\_RECEIVER\_PUBLISHER* tool adaptor.

Thus, the *ESML\_RECEIVER\_PUBLISHER* tool adaptor can fetch *ESML* documents translated or updated by the *RR2ESML* translator or produced by the *CONFIG2ESML* translator.

The *CONFIG\_RECEIVER\_PUBLISHER* tool adaptor can publish *OEP\_Configuration\_UDM* documents, which will be translated by the *CONFIG2ESML* translator into *ESML* documents. The translated *ESML* documents can be fetched by an *ESML\_RECEIVER\_PUBLISHER* tool adaptor.

The *ESML\_RECEIVER\_PUBLISHER* tool adaptor can also publish *ESML* documents. These documents will be translated by the *ESML2AIF* and the *ESML2Config* translators.

In this case one *ESML* document will be translated into two different kinds of document (an *AIF* and an *OEP\_Configuration\_UDM* document).

The *AIF\_UPDATER* can publish two documents: an *IIF* and an *AIF* document. The *IIF2AIF* translator generates an updated *AIF* document from the *AIF* document using the *IIF* document.

The *AIF\_RECEIVER* type of tool adaptor will fetch *AIF* documents, which were translated by the *ESML2AIF* translator or updated by the *IIF2AIF* translator.

The *CONFIG\_RECEIVER\_PUBLISHER* type of tool adaptor will fetch *OEP\_Configuration\_UDM* documents, which were translated by the *ESML2Config* translator.

## ***How to use this workflow configuration***

The above description explained how the ESML workflow works. This section describes how to use the OTIF framework elements to execute this workflow.

This section shows how to “interpret” this workflow model and send it to the server.

This model defines a specific configuration of OTIF. To make the Backplane work with this configuration we have to interpret it like any other models in GME.

### **2. To interpret this workflow model to register it in the Backplane.**

To interpret this model, the *Interpret (i)* button must be activated in GME. After the interpretation, a popup window will appear and ask the password of the Backplane server. In the default installation you may leave it empty. But if you changed the password of the server you have to specify the correct password here.

If you are done you may press the OK button to send this workflow configuration to the Backplane server.

After this interpretation and registration process we are ready to use our new workflow model.

## ***We are ready to play with the ESML tool chain***

This section describes how we can use the previously registered ESML tool chain.

### **3. Let's start to use the ESML tool chain.**

We may use the *Generic\_ToolAdaptor.exe* or the *GenericToolAdaptor* Eclipse (GTA) plug-in to publish and fetch documents.

In the tutorial we will use the *Generic\_ToolAdaptor.exe* first and after that the Eclipse GTA plug-in.

## Using the *Generic\_ToolAdaptor.exe*

This section describes the usage of the *Generic\_ToolAdaptor.exe* application in the ESML tool chain.

### 4. Starting the *Generic\_ToolAdaptor.exe*

After installing the ESML extension package we can find a number of shortcut files in the *Start Menu/Programs/OTIF/Tooladaptors/ESML Tool Chain* folder. Every shortcut file starts the *Generic\_ToolAdaptor.exe* application with a different argument, which specifies the type of the tool adaptor.

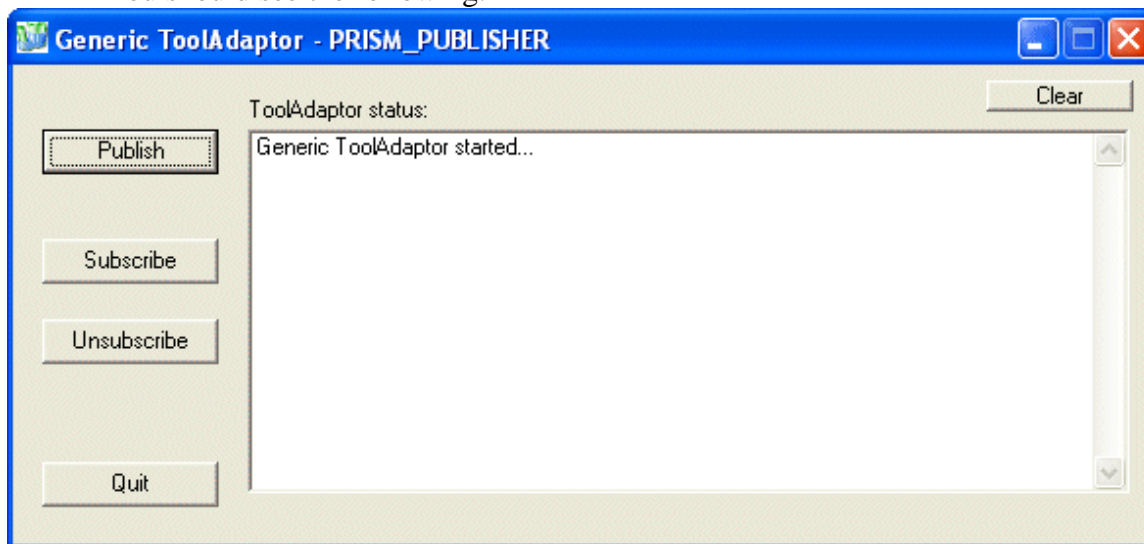
Note: these types are defined in the workflow model.

After starting the *Generic\_ToolAdaptor.exe* with a specific type, the *Generic\_ToolAdaptor.exe* will act as the specified type.

### 5. Publish a document

Let's publish a document into the ESML tool chain. First of all start the “*GTA PRISM\_PUBLISHER*” shortcut file. It starts the *Generic\_ToolAdaptor.exe* as the *PRISM\_PUBLISHER*, which can publish a *PRISM* document as we can see in the *ESML* workflow model.

You should see the following:



*Generic\_ToolAdaptor.exe* acts as *PRISM\_PUBLISHER* type

Click on *Publish* button. Specify the name of the document. You may change other parameters of the document but it is not necessary for now.

Click on *Publish* button and select the *PRISM\_sample.xml* file from the *Samples* directory of the OTIF framework. This is a *PRISM* type of XML file, which will be translated by the *RR2ESML* translator as the ESML workflow model says.

You may close the application now.

## 6. Fetch the translated document

Let us fetch the translated document to see the result. Start the “*GTA ESML\_RECEIVER\_PUBLISHER*” shortcut file to start the *Generic\_ToolAdaptor.exe* as the *ESML\_RECEIVER\_PUBLISHER* type, which can fetch and publish *ESML* documents.

Click on *Subscribe* button because we would like to fetch *ESML* documents. Select the *ESML* paradigm and click on *Subscribe* button.

After subscription we will get a notification when a document with the specific (right now the *ESML*) paradigm is available.

When the notification arrives, click on the *Save* button and specify the name of the file. This document is an ESML document and we want to use it in GME so specify a GME file, for example like the following: “something.mga”.

If the file is a GME file the application will ask you about the GME paradigm of the file. Now choose the ESML paradigm and click OK.

You may close the *Generic\_ToolAdaptor.exe* and open the fetched file in GME.

## 7. Publish a non-UDM conform *OEP\_Configuration\_UDM* XML file

We know how to publish any regular documents into the Backplane server, but what happens if we have a file which is not compatible with the XML tools used (UDM) in OTIF? In this case an explicit, XSLT-based pre- (or post-) processing can be done on the XML file before (or after) it is published (or fetched) in the *Generic\_ToolAdaptor.exe*.

Suppose that our *OEP\_Configuration\_UDM* XML file is non-UDM compliant, but we want to translate it into an *ESML* document and we have an XSLT script called *bs2cfg.xsl* that performs this post-processing.

**Note: the publishable XML file, its DTD file, the XSLT script file and the DTD file of the transformed output XML file should be in the same directory.**

First of all start the “*GTA CONFIG\_RECEIVER\_PUBLISHER*” shortcut file. It starts the *Generic\_ToolAdaptor.exe* as the *CONFIG\_RECEIVER\_PUBLISHER*, which can publish an *OEP\_Configuration\_UDM* document.

Click on *Publish* button. Specify the name of the document. You may change other parameters of the document but it is not necessary for now.

Check in the *XSLT transform...* checkbox to activate the XSLT transformation support.

Click on *Publish* button and select your non-UDM compliant XML file.

After the selection the application will ask you to select the XSLT file for the XSLT pre-processing. Select your *bs2cfg.xsl* XSLT file.

The application will do the transform based on the XSLT script file and publish the transformed file into the Backplane server.

You may close the application.

## 8. Fetch a non-UDM conform *OEP\_Configuration\_UDM* XML file

Some cases you want to fetch your translated file and right after you want to do some XSLT post-processing on the fetched file to get your desired result.

Suppose that we want to fetch a translated file and we have an XSLT script called *cfg2bs.xsl*.

Start the “*GTA CONFIG\_RECEIVER\_PUBLISHER*” shortcut file to start the *Generic\_ToolAdaptor.exe* as the *CONFIG\_RECEIVER\_PUBLISHER*, which can fetch and publish *OEP\_Configuration\_UDM* documents.

Click on *Subscribe* button because we would like to fetch *OEP\_Configuration\_UDM* documents. Select the *OEP\_Configuration\_UDM* paradigm and click on *Subscribe* button.

After subscription we will get a notification when a document with the specific (in this case: *OEP\_Configuration\_UDM*) paradigm is available.

When the notification arrives, click on the “...” button to select the XSLT script (right now our *cfg2bs.xsl*) to do the post-processing and check the XSLT translation checkbox to allow the post-processing. Click on the *Save* button and specify the name of the file to save it.

After the fetching process you will get your fetched and then XSLT-transformed result in this file.

## 9. Try out other types

You may try out other types of tool adaptors. If you want to publish an *ESML* file, start the “*GTA ESML\_RECEIVER\_PUBLISHER*” shortcut file and publish the *esml\_sample.mga* file from the Samples directory of the framework.

**Note:** the *esml\_sample.mga* file should be created first by importing the *esml\_sample.xml* file into GME and save it.

After that you can fetch the translated document. If you need an *AIF* document, you may use the “*GTA AIF\_RECEIVER*”. If you need an *OEP\_Configuration\_UDM* document, you may use the “*GTA CONFIG\_RECEIVER\_PUBLISHER*”.

**Note:** you can start the *Generic\_ToolAdaptor.exe* by itself. In this case it will ask you about its own type. So you don’t have to use the shortcut files.

**Note:** if you change the names of the tool adaptor types or use different kind of ESML workflow model you may not be able to use these shortcuts.



## Using the Eclipse GTA plug-in

This section describes the usage of the Eclipse GTA plug-in in the ESML tool chain.

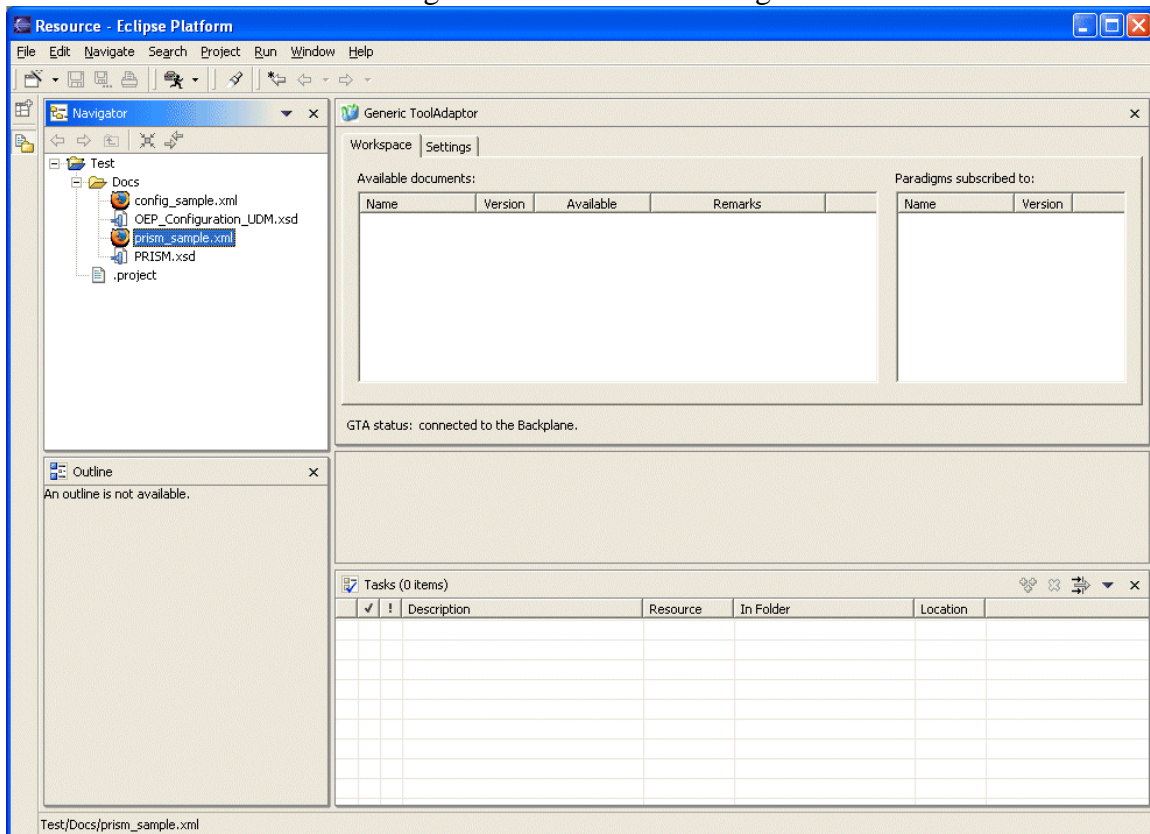
### 10. Start to use the GTA Eclipse plug-in.

Start the Eclipse framework and check that the Eclipse plug-in is installed as the “*Getting Started with the Eclipse plugin*” says.

Create a simple project in the Navigator window. Create a folder under the new project with any name. Drag and drop the files (*PRISM.xsd*, *PRISM\_sample.xml* and *esml\_sample.mga*) from the *Samples* directory of the OTIF framework onto this newly created folder.

**Note:** the *esml\_sample.mga* file should be created first by importing the *esml\_sample.xml* file into GME and save it.

You should see something similar like the following:



**GTA Eclipse plug-in main screen**

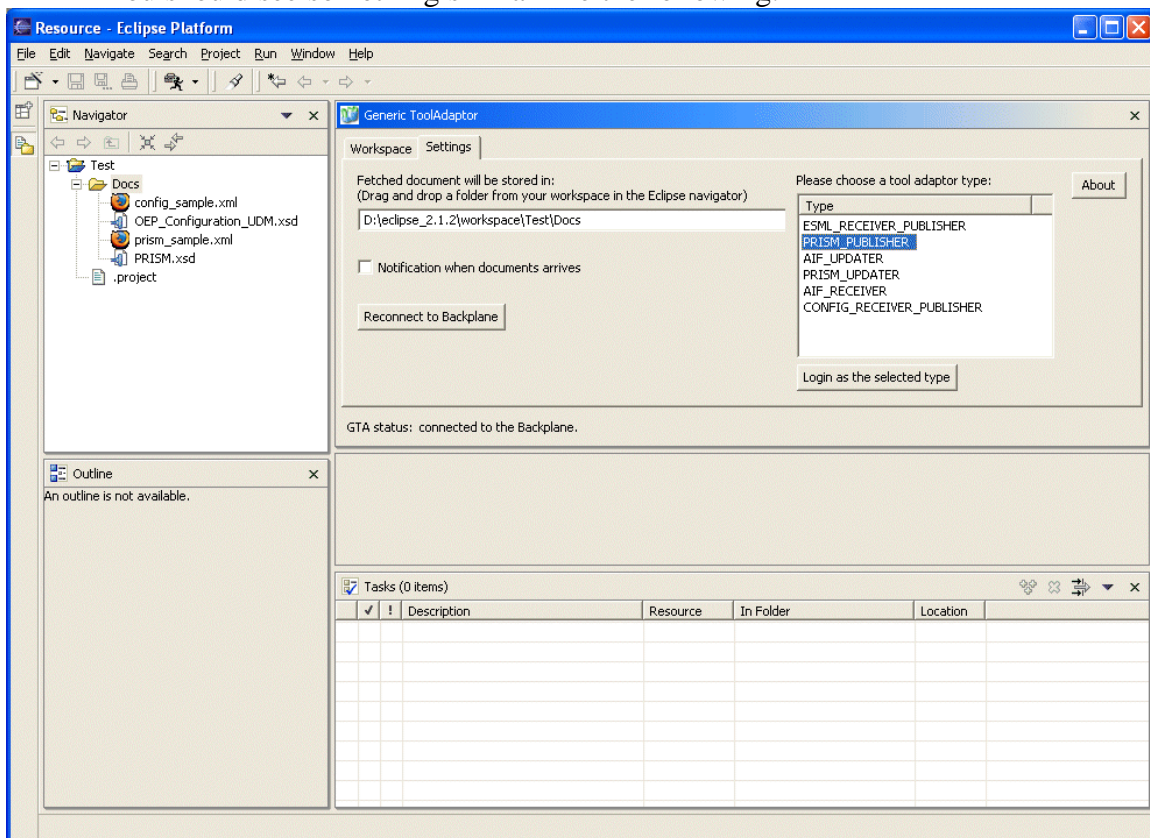
## 11. Setup the Eclipse plug-in.

First of all we have to select our type of tool adaptor and login into the Backplane. After login the generic tool adaptor will act as the selected type of tool adaptor.

Click on the *Settings* tab and select the *PRISM\_PUBLISHER* type and click on the “*Login as the selected type*” button. If everything went all right the GTA status informs us as what kind of tool adaptor the plug-in acts right now.

We may specify our fetch directory. This directory will store our fetched documents. For now just drag and drop the *demo* folder onto the edit box.

You should see something similar like the following:



**GTA Eclipse plug-in Settings**

## 12. Publish our first document into the ESML tool chain.

Please click on the *Workspace* tab. We will publish the *PRISM\_sample.xml* file. This is an *PRISM* file that will be translated by the *RR2ESML* translator.

Drag and drop the file from the *Navigator* window to the *Available documents* list box. Specify the name in the popup window and click *Publish* to send it.

### 13. Fetch our first translated document

The GTA plug-in has to act as another type of tool adaptor so we have to click on the *Settings* tab and select the *ESML\_RECEIVER\_PUBLISHER* type to fetch the translated document. Click on the *Login* button.

We have to subscribe to the *ESML* paradigm to be able to get notifications about available documents.

Right click on the *Paradigms subscribed to* list box and click *Subscribe to ESML* paradigm.

Right after subscription you should get a notification about our previously translated document. Right click on the notification and click on *Fetch*.

You may change the name of the document but it is not necessary but you have to change the extension to *mga* because we will use the document as a GME document.

If you save the document as a GME file you have to specify a GME paradigm for the document.

Right now you have to select the *ESML* paradigm and click on *OK* button.

A new document should appear in our *demo* folder. Right click on the file and open it in GME (you have to register the GME application as an external tool in the Eclipse environment).

### 14. Try out other types

You may try out other types of tool adaptor. If you want to publish an *ESML* file, select the *ESML\_RECEIVER\_PUBLISHER* and publish the *esml\_sample.mga* file.

After that you can fetch the translated document. If you need an *AIF* document, you may select the *AIF\_RECEIVER* type to login and subscribe to the *AIF* paradigm to get a notification. If you need an *OEP\_Configuration\_UDM* document, you may select *CONFIG\_RECEIVER\_PUBLISHER* type to login.

**Note: the Eclipse GTA plug-in also supports XSLT pre - and post-processing.**