in	G+	f	y
share	+1	share	tweet

Table of Contents [-]

Duration: 20 - 30 minutes

In this tutorial you will create a flat file schema to define a flat file structure and a service to read in and parse that flat file.

Prerequisites

· The only prerequisite is access to Integration Server, a browser and the Designer IDE.

Step 1: Create a Flat File

· **In this step:** You will create a CSV flat file.

· In **Windows Explorer**, navigate to the **C:** directory. Right click in this directory and create a new text Document. Name the text document **ffTutorialInput** (remove the .txt).
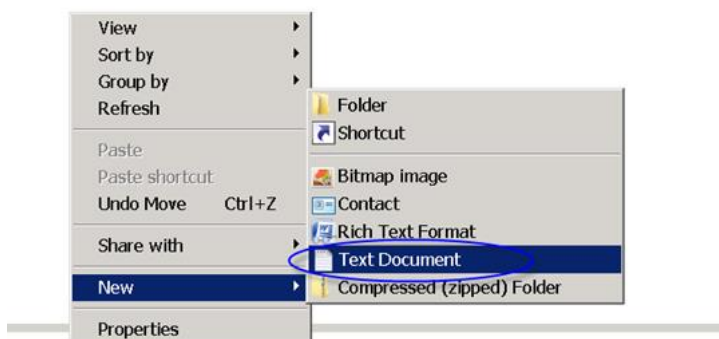
Open **ffTutorialInput** in **Notepad** and copy/paste the 3 rows/records below into the text document, **Save** and close it. This is a Customer CSV consisting of **Customer Number, Name, Address, City, State and Zip** attributes.

12345,John Smith,123 Main St,San Francisco,CA,27262

54321,Bill Jones,3636 Hale Ave.,Chicago,IL,60643

19283,Mary Olson,74746 N. Dodge St,Philadelphia,PA,64722
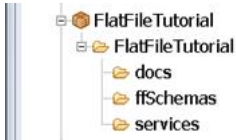
Step 2: Create a Flat File Dictionary

· **In this step:** You will create a flat file dictionary, mapping to the records and attributes in the CSV file.

 If you already have **Software AG Designer** started, just make sure you are in the **Service Development** perspective. If this is the first time starting **Software AG Designer**, open it from the **Start Menu -> All Programs -> Software AG -> Tools -> Software AG Designer 9.5**.

· In the **Software AG Designer Welcome** page, click on the **Open the Service Development Perspective** link:
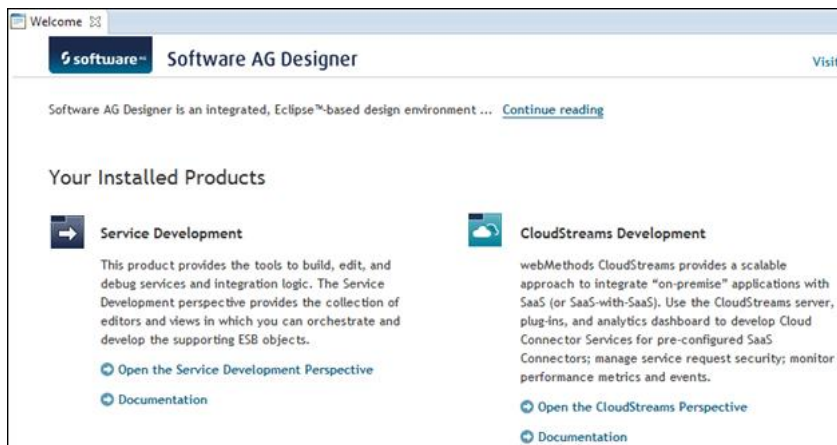
· In the **Package Navigator** view, create a new package called **FlatFileTutorial**. Under the **FlatFileTutorial** package create a folder called **FlatFileTutorial**. Under the **FlatFileTutorial** folder create 3 folders called **docs, ffschemas** and* services. It should look exactly like the following:
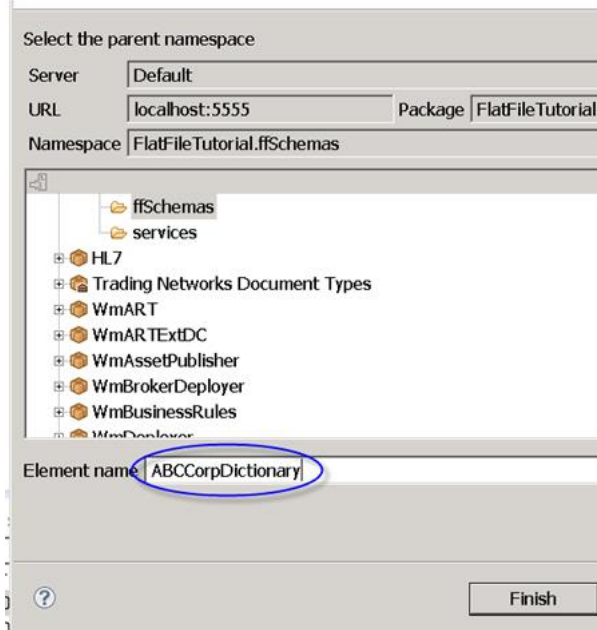

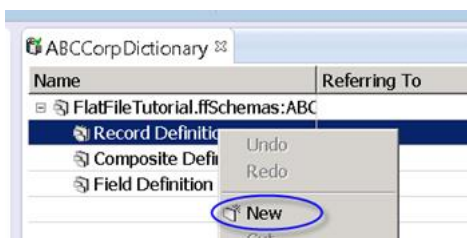· Right click on the **ffschemas** folder and create a new **Flat File Dictionary**
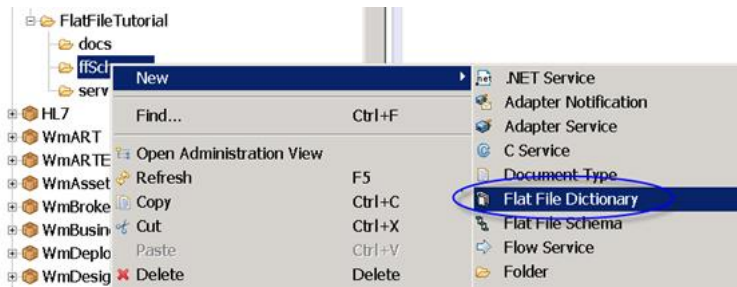
· Name the Flat File Dictionary **ABCCorpDictionary** and click **Finish**.

· Right click on **Record Definition** and select **New**

· Name the new record **Customer** and click **Finish**

· Right click on **Customer** and select **New**. In the wizard, select **Field Definition** and click **Next**.

· In the drop down, select **ExtractorType** of **Nth Field.** Click the **Insert Row** icon 5 times to add 5 new fields to the record definition. You should see entries for 6 rows with no names entered so far.

· Using your cursor enter the values as shown below: **CustNum, Position0** (column indices are 0 based)**,** etc. and click **Finish.**

· Your dictionary should look like the following. **Save** your work **Ctrl-S.**

Step 3: Create a Flat File Schema

· In this step: You will create a flat file schema describing the format of your input file.

· Right click on the **ffSchemas** folder and select **New ->Flat File Schema**:
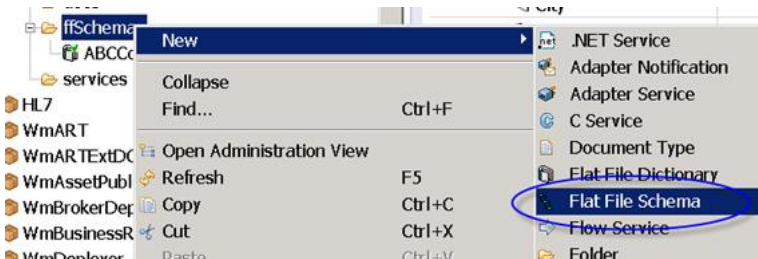
· Name the new Flat File Schema **Customer** and click **Finish.**

New Flat File Schema

Create a New Flat File Schema

Specify the name and location for the flat file schema.

Select the parent namespace

Server	Default
URL	localhost:5555
Package	FlatFileTutorial
Namespace	FlatFileTutorial.ffSchemas

- ffSchemas
  - services
- HL7
- Trading Networks Document Types
- WmART
- WmARTExtDC
- WmAssetPublisher
- WmBrokerDeployer
- WmBusinessRules
- WmDeployer

Element name Customer

Finish

· Select **Delimiter** for **Record parser type**, (you may have to scroll down) carriage return line feed for **Record Character** (delimiter), and enter a **comma** (type into field) for **Field or composite** (delimiter) as shown below. Leave all other fields as they are.

*Customer

Description:

▾ Record parser

Record parser type
- ⊙ Delimiter
- ○ Fixed length
- ○ Variable length
- ○ EDI document type

Record
- ⊙ Character: carriage return line feed
- ○ Character position:

Field or composite
- ⊙ Character: ,
- ○ Character position:

Subfield
- ⊙ Character:
- ○ Character position:

· Select the **Flat File Structure** tab at the bottom.

· In the **Properties** view, to the right, place your cursor in the value box for the **Set** option. This will make the elipsys appear, to the right, as shown below. Click the **elipsys** button.

· Select the **ABCCorpDictionary** entry under **ffSchemas** and click **Next.**

· Select **Customer**, click **Finish** and **Save** your work: **Ctrl-S.**
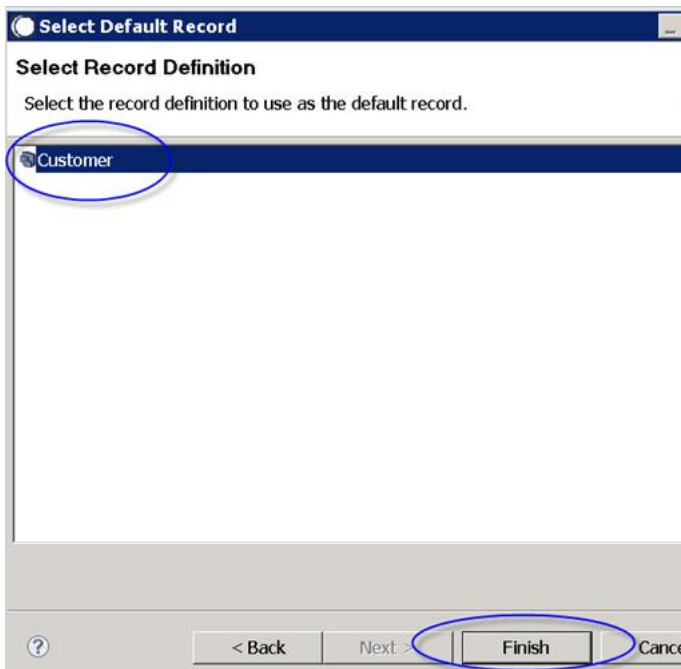
· Click on the **Customer** record and click the **Create Document Type** button along the top tool bar. This will automatically create an Integration Server document type from our schema structure.
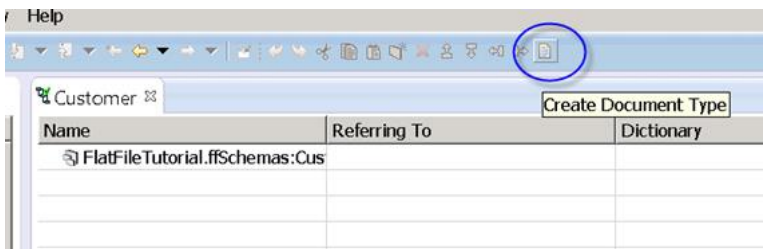
· This creates **CustomerDT** under your **ffSchema** folder. Double click **CustomerDT** to open it and expand your record. We now have an IS doc type matching our flat file schema structure. Drag and drop your newly created **CustomerDT** to your **docs** folder**.**
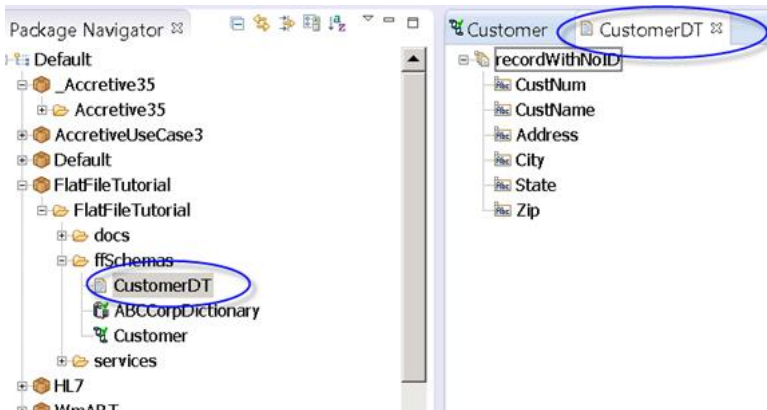
Step 4: Create a Flow service to read and parse your flat file.

· In this step: You will create a flow service to read in your flat file, parse it, and loop over it so it can be processed.

· Right click on the **services** folder, select **New->Flow Service**
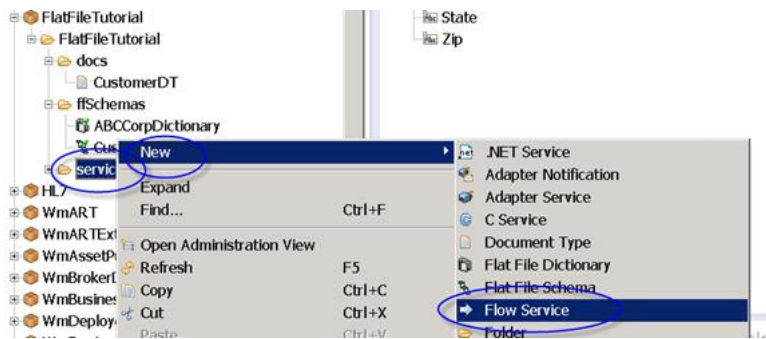
· Name your flow service **processCustomerFF** and click **Finish**.
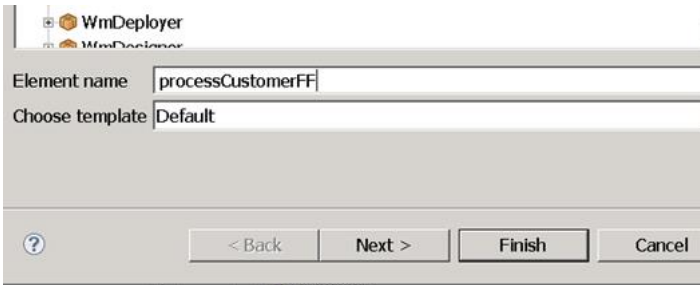
· In the **Package Navigator** view, navigate to the **pub.file:getFile** public service, under the **WmPublic** package. Drag **getFile** into the flow editor of your **processCustomerFF** service as shown below.
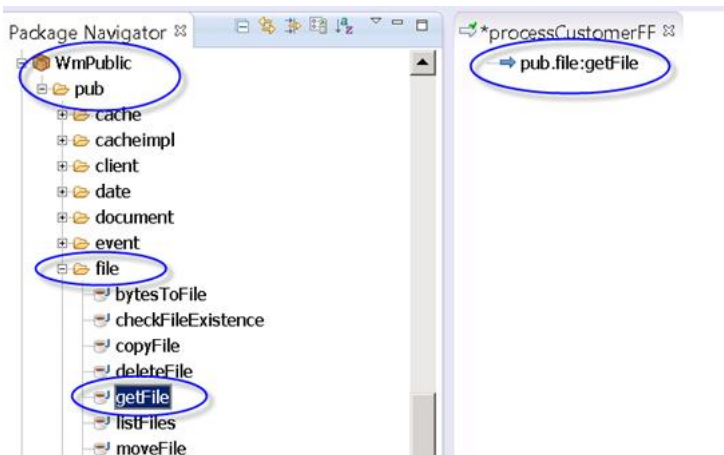
· With **pub.file:getFile** selected, click the **Pipeline** tab, and double click **filename** in the **Service In** column. This will open the **Enter Input for 'filename'** dialog. Enter the file path of the CVS input file; in our example this is **c:\ffTutorialInput**. Click **OK.**
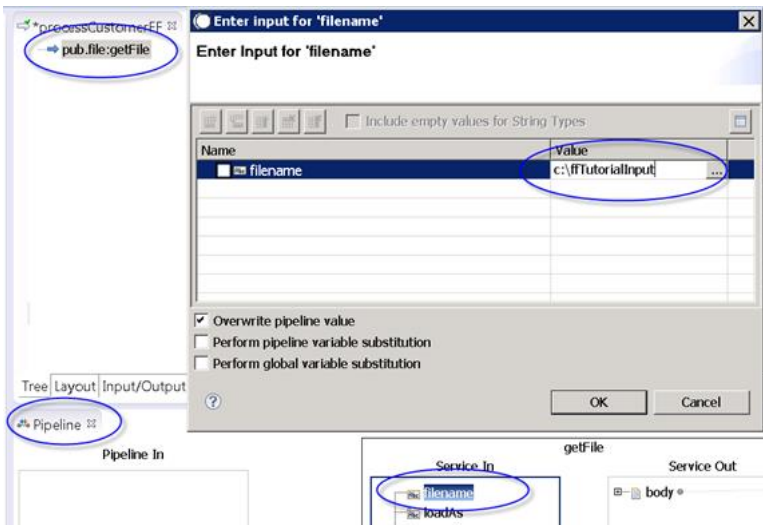
· Navigate to the **pub.flatFile:convertToValues** service under the **WmFlateFile** package. Drag **convertToValues** onto the flow editor of the **processCustomerFF** service, under the **getFile** service, as shown below.
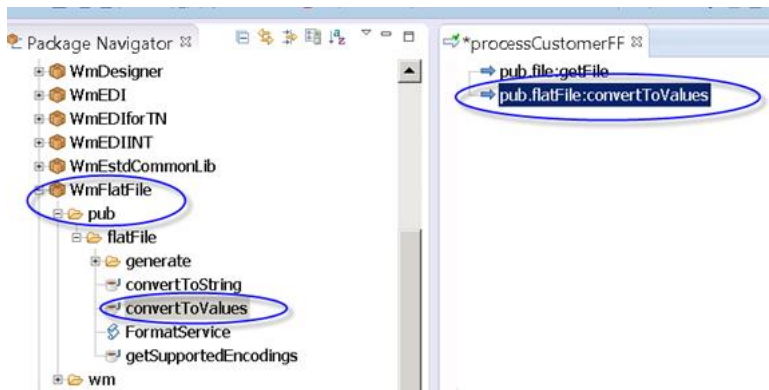
· Under **Pipeline In**, expand **body** and map **bytes** to **ffData** in the **Service In** column, as shown below.
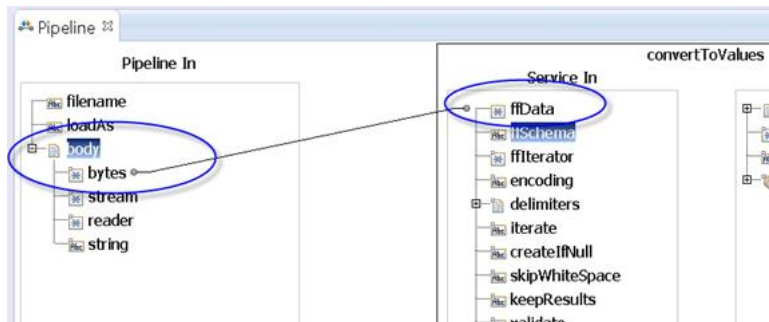
· Right click on the **Customer** flat file schema in the **Package Navigator** view and select **Copy**. Double click **ffSchema** in the **Service In** column. This will open the **Enter Input for 'ffSchema'** dialog. Position your cursor in the **Value** field and paste. This will set the **Customer** schema as input. Click **OK** and **Save.**
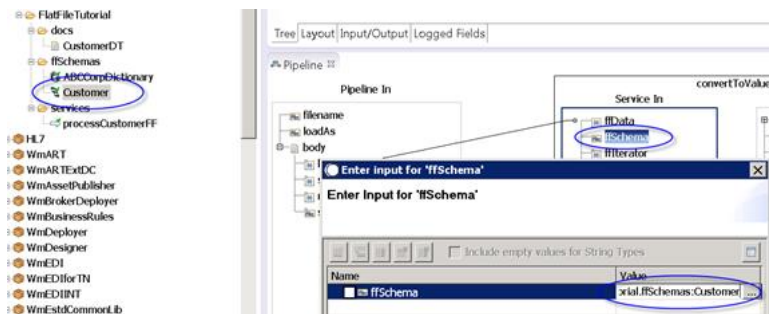
· Map the **convertToValues** output to the **CustomerDT** IS Document Type. In **Package Navigator**, right click on **CustomerDT** (in the **docs** folder) and select **Copy**. Right click the bottom of the **Pipeline Out** column (lower right) and **Paste**. This will copy the **CustomerDT** document with a name of **Untitled**. Change the name to **CustomerDT**, as shown below. Lastly, map **ffvalues** to **CustomerDT** by dragging and dropping. **Save** (**Ctrl-S**).
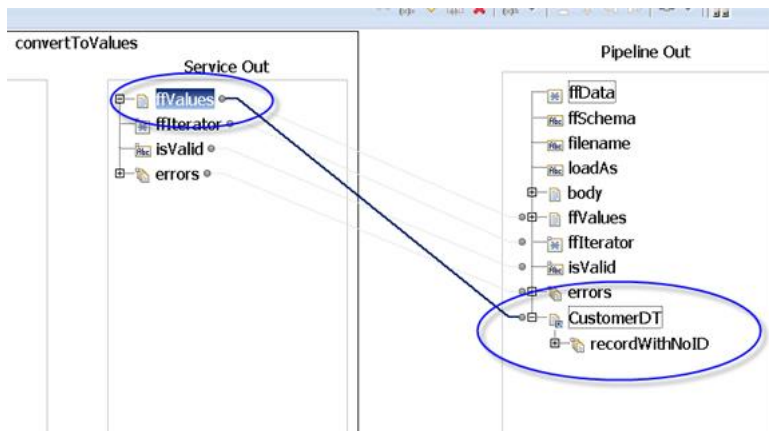
· Drag a **MAP** step from the **Palette** on the right side of the flow editor into your service as shown below. If the **Palette** is not displayed, you can expand/collapse the **Palette** by clicking the small arrow highlighted below. **Save** your work.
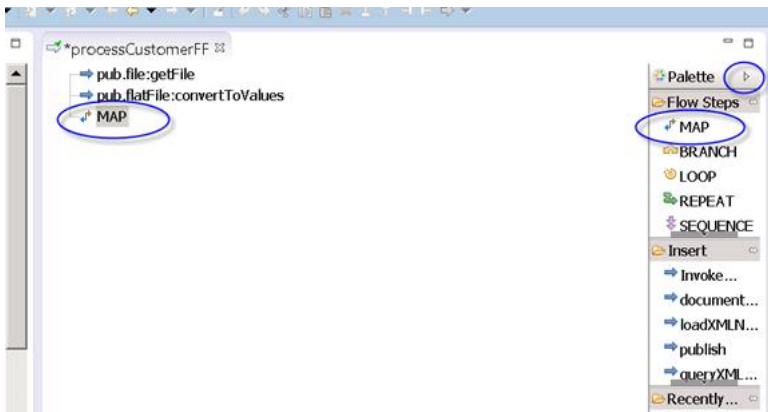
· Your service is now completed and your CSV input file is now converted to an IS doc type. The IS doc type can be mapped as input to any other service within the IS by dragging and dropping – left to right.
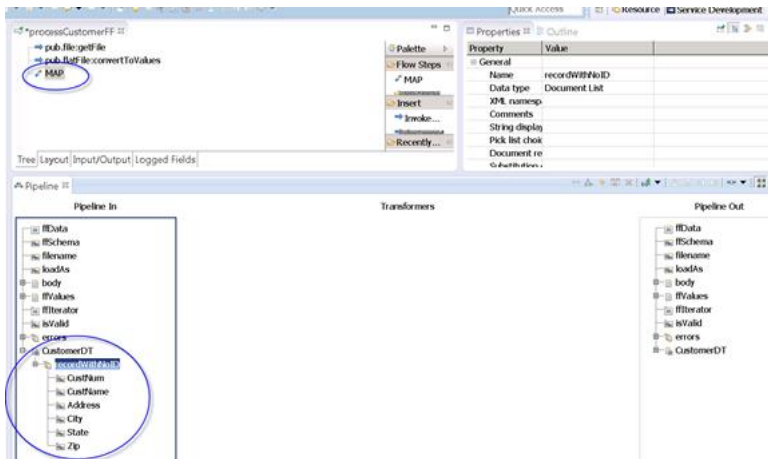
Step 5: Test your Flat File Service

In this step: You will run your service which will go out and read in the flat file and convert it to an IS doc type.

· In the **Package Navigator** view, right-click on your newly created service, **processCustomerFF**, and select **Run As** -> **Run Flow Service** from the popup menu. There are no input parameters to this service so click **OK.**

· Double click on the **Results** tab, to maximize/minimize the view. As you can see the flat file input has been parsed and mapped to an IS document type.

Conclusion

You have created a service to get a CSV flat file from the file system and converted it into an IS document type where it can be mapped into other services/systems. We created a Flat File dictionary containing our record layouts. Because we created our flat file elements in a dictionary they can be shared across all flat file schemas, simply by referencing them from the schema. While this tutorial processed a fairly simple CSV file, the flat file capabilities are very powerful. We can process delimited, fixed length, variable length, hierarchical, EDI and many other type of flat files.

To import the solution of this tutorial download End FlatFileTutorial.zip (http://techcommunity.softwareag.com/ecosystem/communities/public/webmethods/products/esb_and_integration/codesamples/20160704121912812) (http://techcommunity.softwareag.com/pwiki?p_auth=9XnJ4YCR&p_p_id=36&p_p_lifecycle=1&p_p_state=exclusive&p_p_mode=view&p_p_col_id=column-1&p_p_col_pos=1&p_p_col_count=2&_36_struts_action=%2Fwiki%2Fget_page_attachment&p_r_p_185834411_nodeId=11809&p_r_p_185834411_title=Flat+File+Processing+Tutorial&_36_ and follow the directions in the Import an IS Package (http://techcommunity.softwareag.com/pwiki/-/wiki/Main/Import+an+IS+Package) tutorial.

35 Attachments (http://techcommunity.softwareag.com/pwiki?
p_p_id=36&p_p_lifecycle=0&p_p_state=pop_up&p_p_mode=view&_36_struts_action=%2Fwiki%2Fview_page_attachments&p_r_p_185834411_nodeName=Main&p_r_p_185834411_title=Flat+File+Processing+Tutorial&_
%2Fwiki%2FMain%2FFlat%2BFile%2BProcessing%2BTutorial%2Fpop_up%3F_36_viewMode%3Dprint)
54286 Views

Average (1 Vote)
☆ ☆ ☆ ☆ ☆

Comments	⌄