

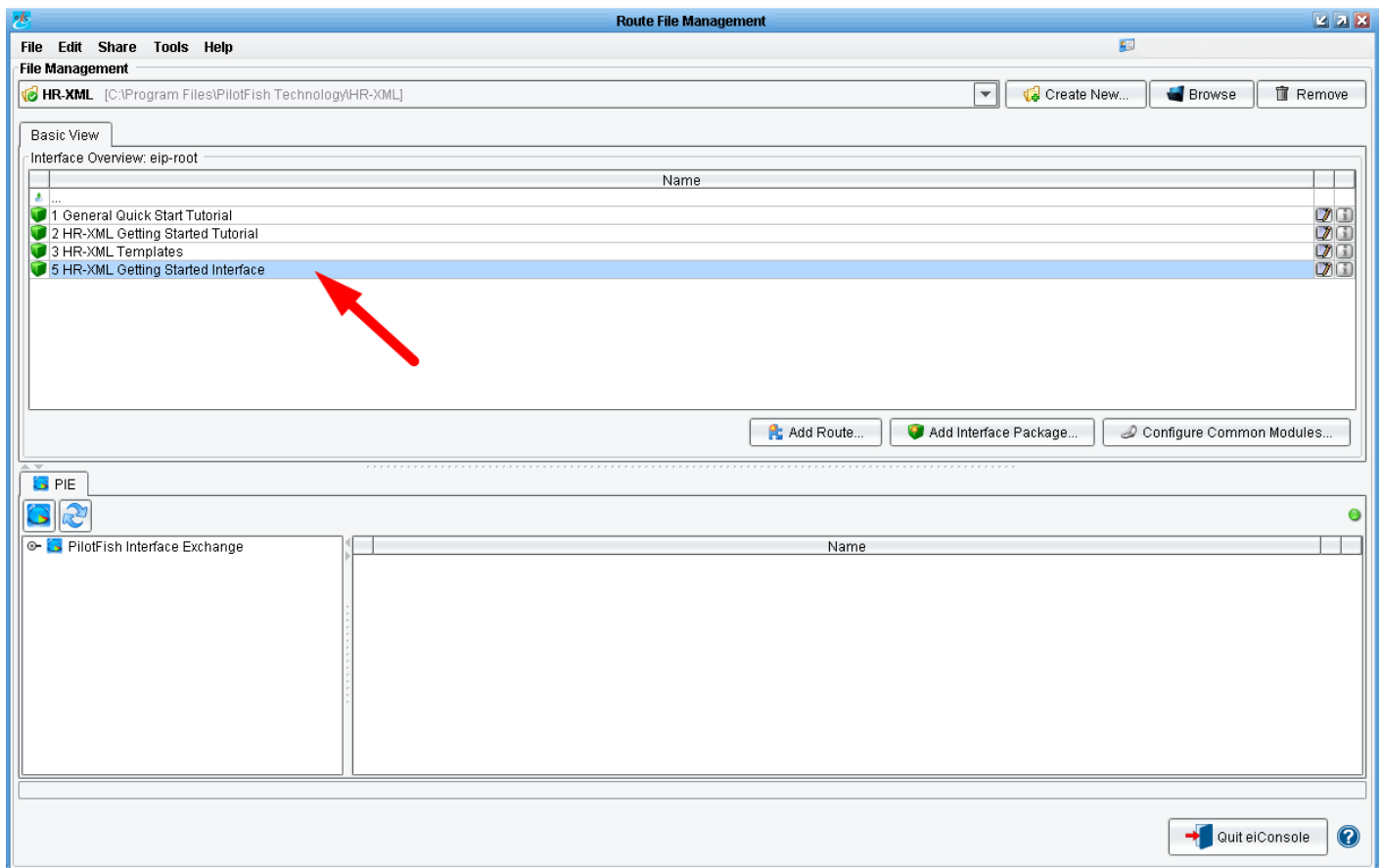
2 HR-XML Getting Started Tutorial

<https://cms.pilotfishtechnology.com/node/4601>

Welcome to the eiConsole for HR-XML Getting Started Tutorial. New users should complete the [General Quick Start Tutorial](#) to learn the basic principles and terminology used in the eiConsole before proceeding to this more advanced lesson. The HR-XML Getting Started Tutorial builds on concepts learned in the previous tutorial that are not repeated here and are required to complete this tutorial.

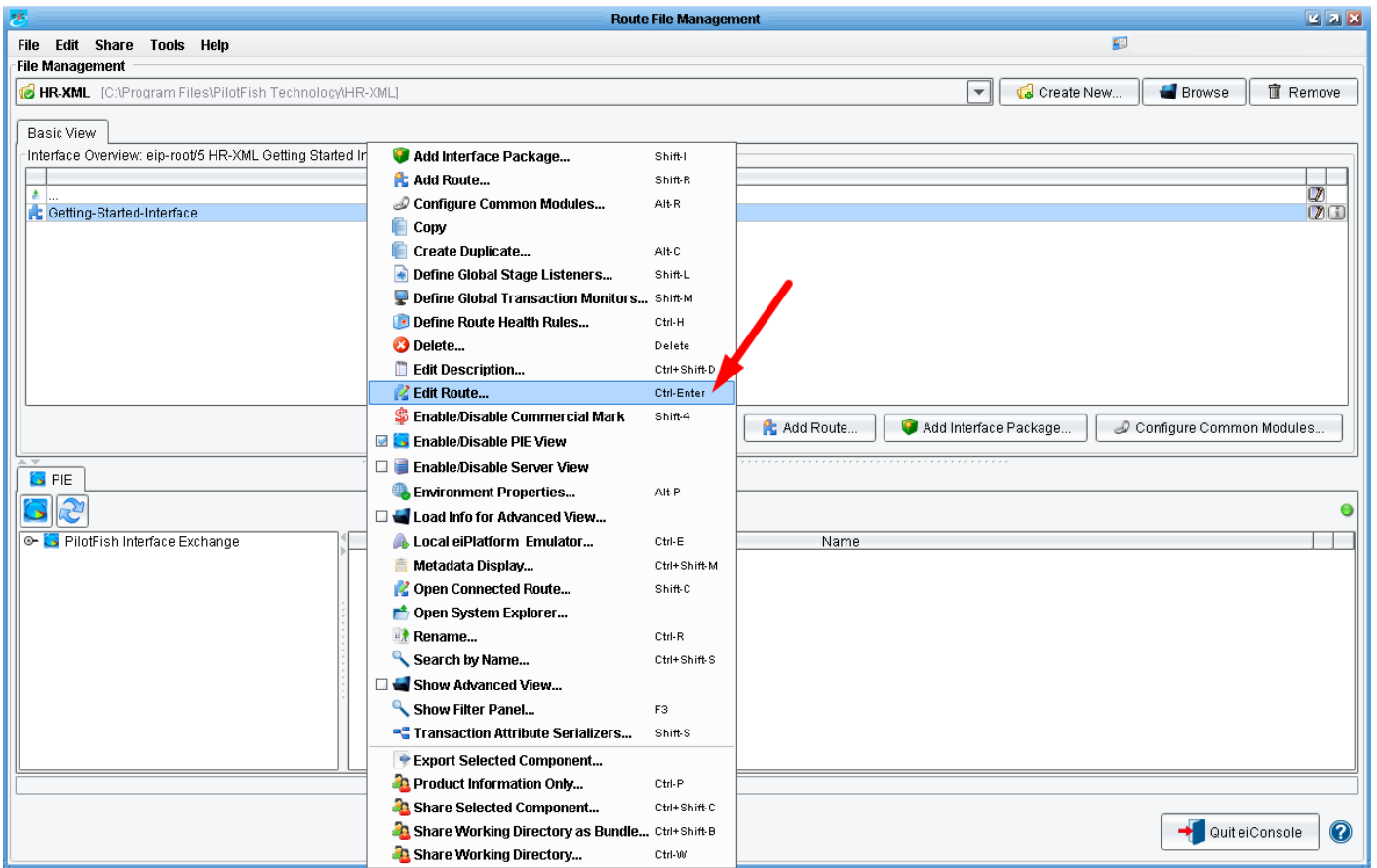
In this tutorial you will build a simple Process Screening Order [interface](#). This interface will accept a list of candidates in a csv file and create a corresponding request for a background and/or credit check in the HR-XML 3.0 [format](#). This interface will take the average user about 1 hour to complete.

Be sure to click on the row, 3 HR-XML Templates, to view the extensive selection of BOD templates which are included with this bundle. These can be used as the starting point in the eiConsole's paint-by-numbers process for developing [interfaces](#).

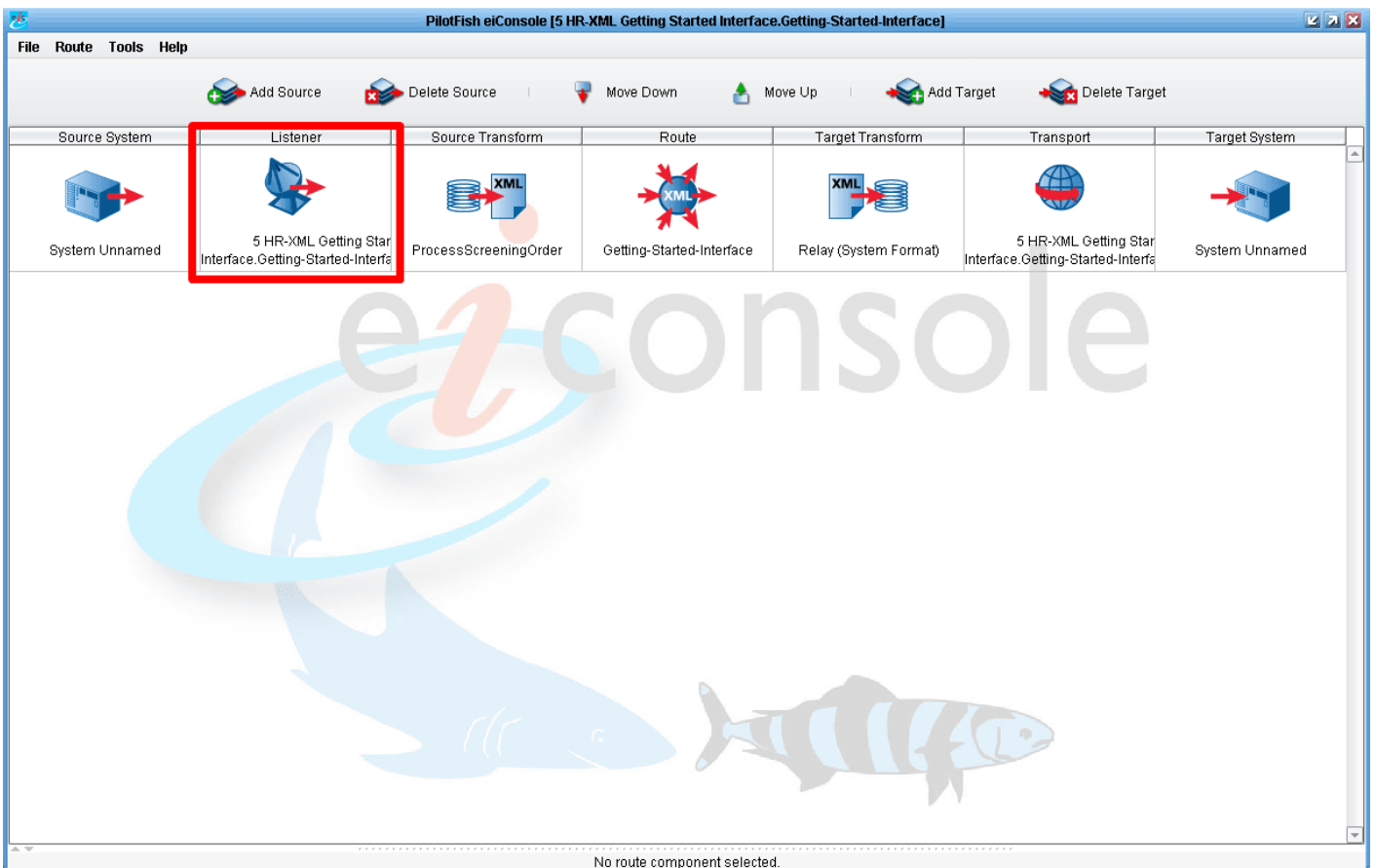


When first opening the eiConsole for HR-XML bundle, you will see 4 [Interface Packages](#) in the [Route File Management](#) window. These are: 1 General Quick Start Tutorial, 2 HR-XML Getting Started Tutorial, 3 HR-XML Templates and 5 HR-XML Getting Started Interface. If you completed the Quick Start Tutorial you should also have the 4 General Quick Start [Interface Package](#).

Let's get started! Double click the **5 HR-XML Getting Started Interface** row.



When you open the package you will see a configured Route/Interface. For this tutorial you'll be tweaking this Route, the **Getting Started Interface** message. Right click on the row and select **Edit Route** from the dropdown.

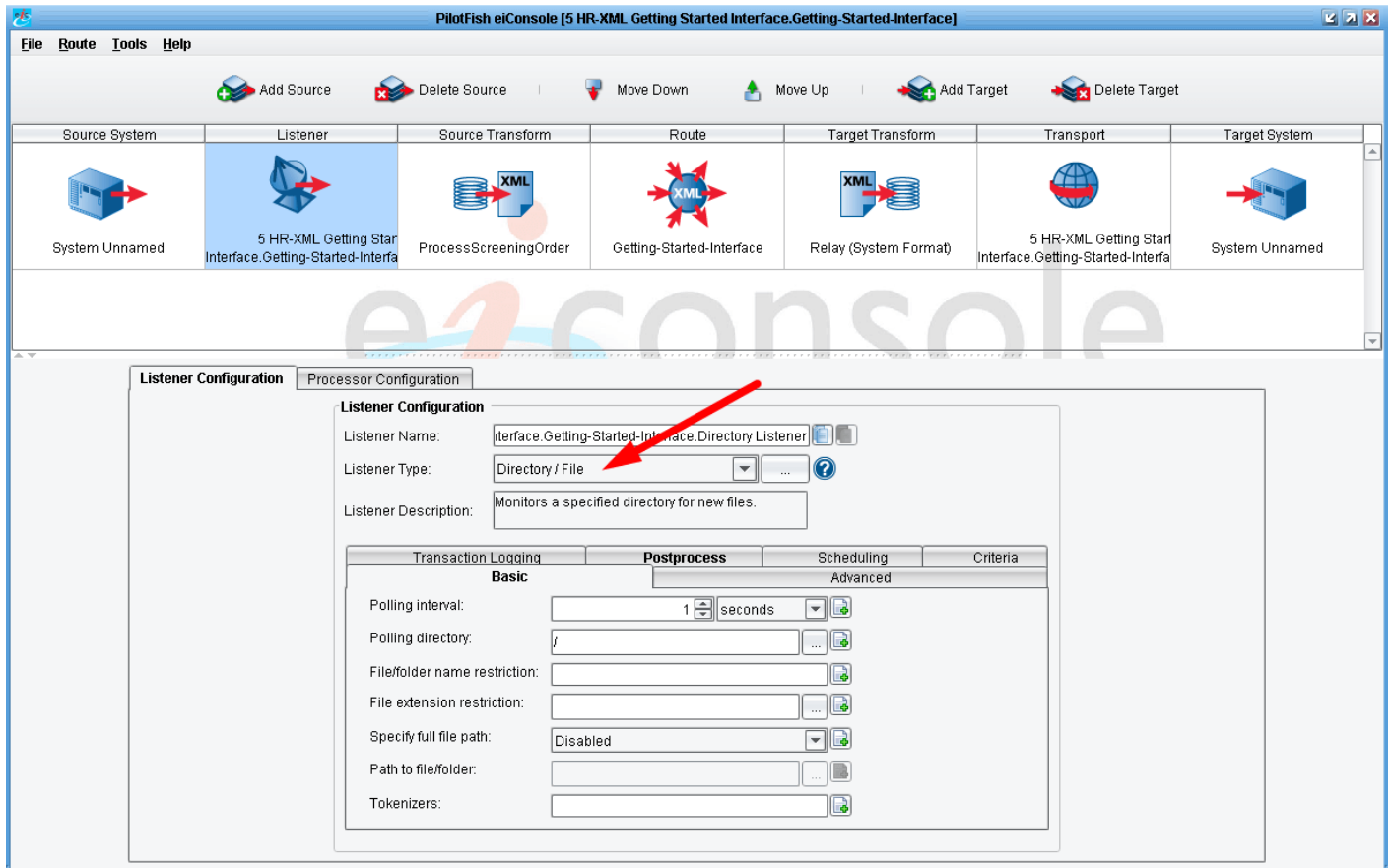


This will bring you to the Main Route grid. The main route grid depicts the flow of data from a Source System to a [Target](#) System. Your job will be to configure each one of the [stages](#) in between including the:

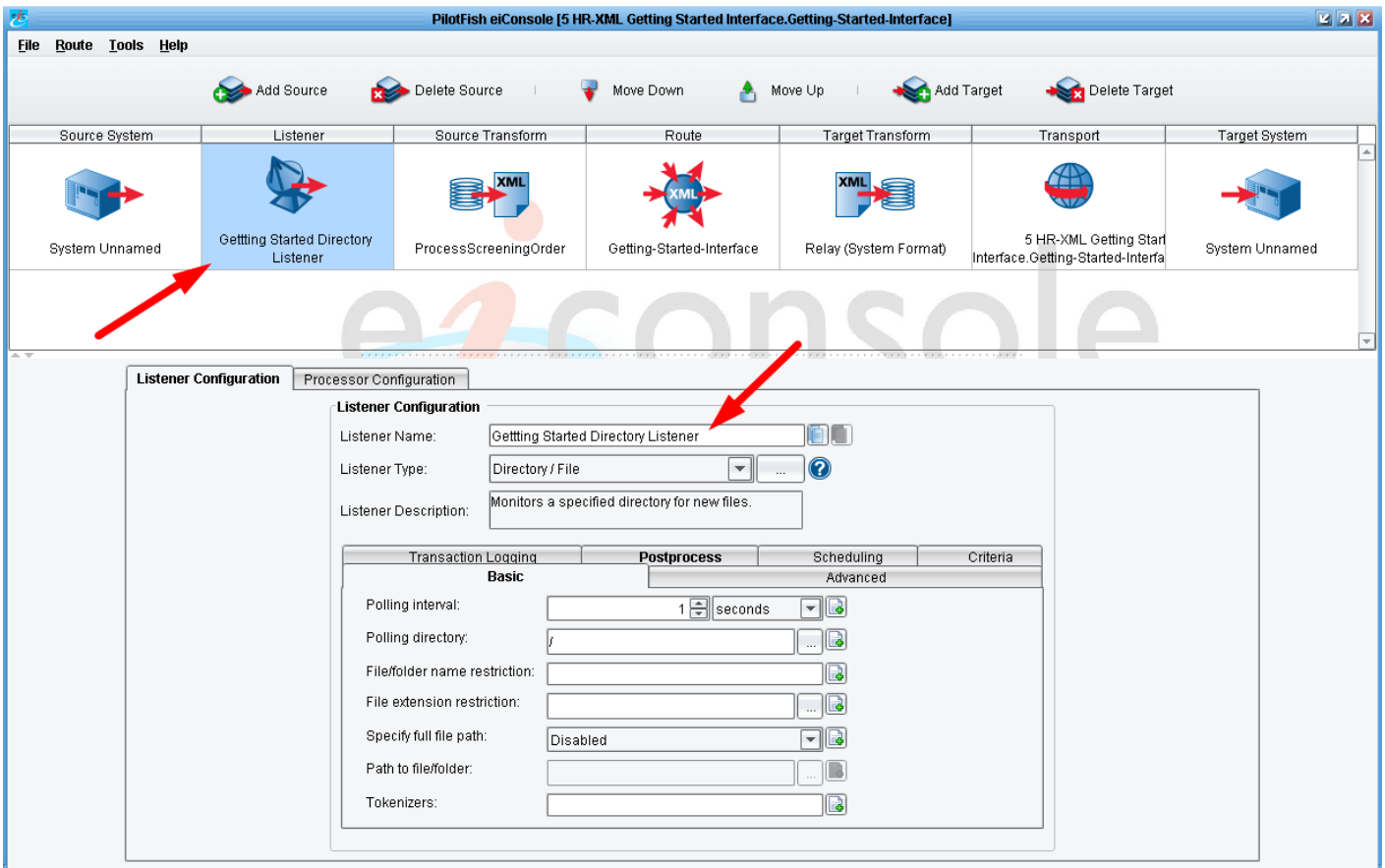
- [Listener](#): handles the connectivity to the Source System;

- [Source Transform](#): from a proprietary format into your HR-XML canonical model;
- [Route](#): any routing rules, if you had multiple [targets](#) you could make a decision as to which data should be sent to which Target systems;
- [Target Transform](#): this allows you to further transform data into a format that the Target System can consume;
- [Transport](#): this allows you to handle the connectivity to the Target System. You'll begin with the Listener.

Click the **Listener** stage in the main route grid.



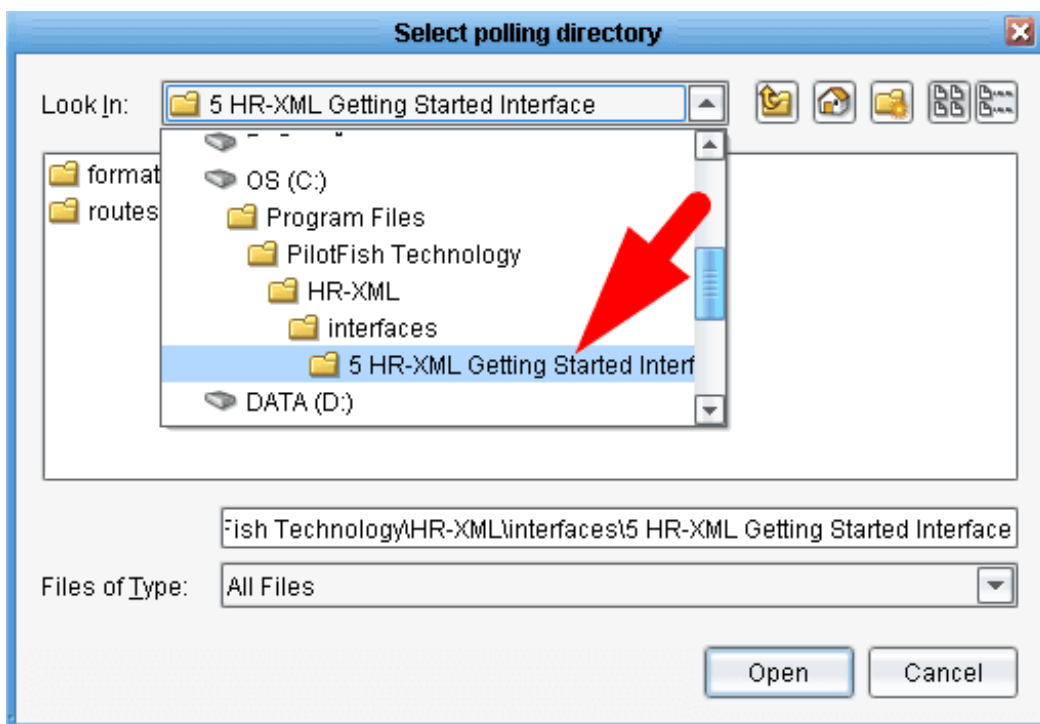
Each time a stage in the upper grid is selected, a set of configuration panels will appear below. With the Listener selected, you'll see the Listener Configuration area. This is where you choose how you expect the Source System to send data. You would do this by selecting the appropriate Listener type from the Listener Type dropdown. Here you see this template has already specified a Directory / File Listener. This means that the system will poll a directory at a specified interval and look for a particular file.

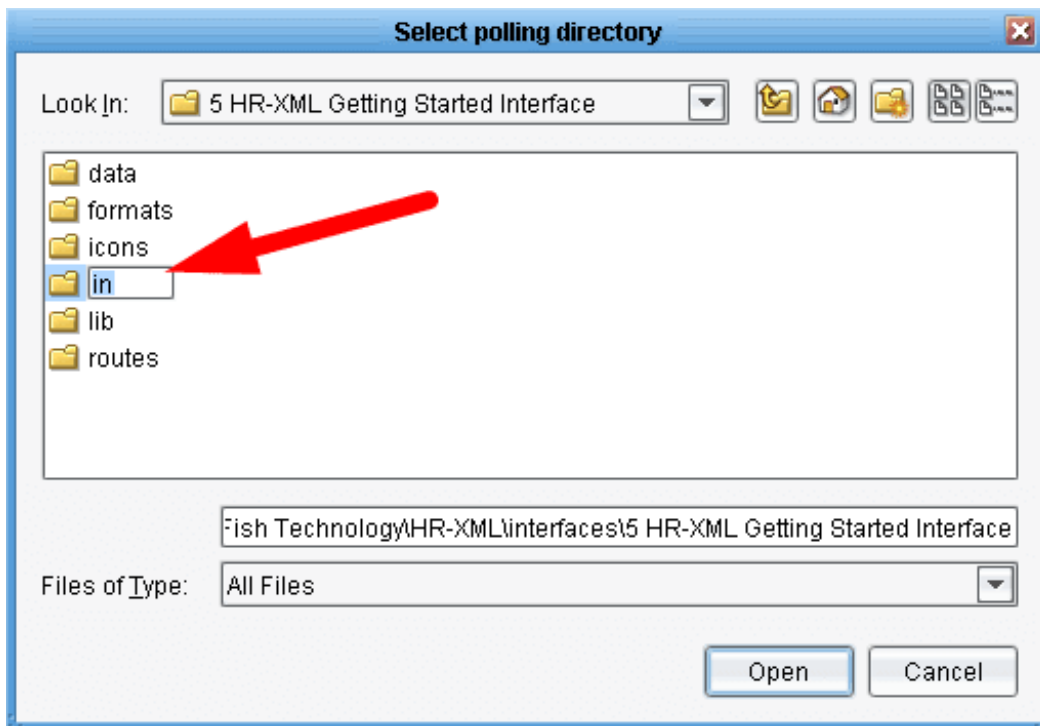


Once the configuration panel opens edit the **Listener Name** to read "Getting Started Directory Listener" or give it some other meaningful name.

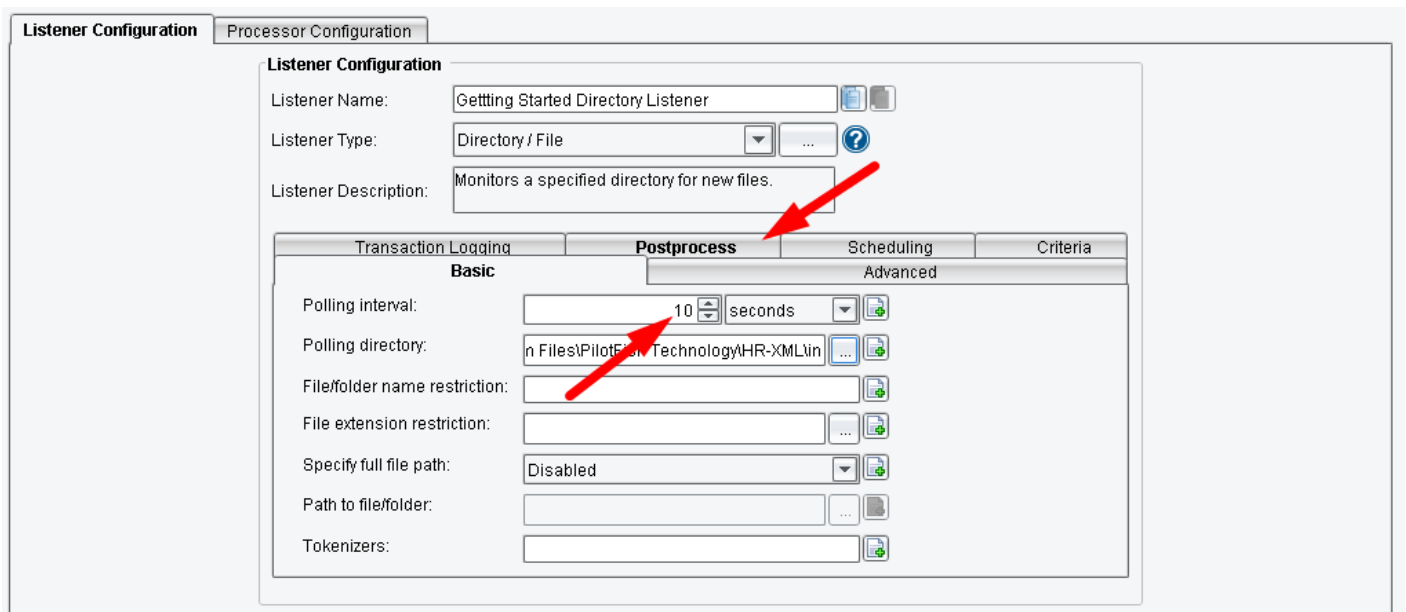
*Note: within a **Working Directory** you cannot use the same Listener Name for two different **routes**. This creates ambiguity in the system. So for example, if you tried to name your Listener "Source Directory Listener" you would be prompted to enter another name because Source Directory Listener is already being used in the 2 HR-XML Getting Started Tutorial sample interface.*

In this case you'll keep the Listener as a Directory / File Listener, but you'll change the Polling directory. To do this, first click the red **Elipsis** button next to the **Polling directory** configuration item.

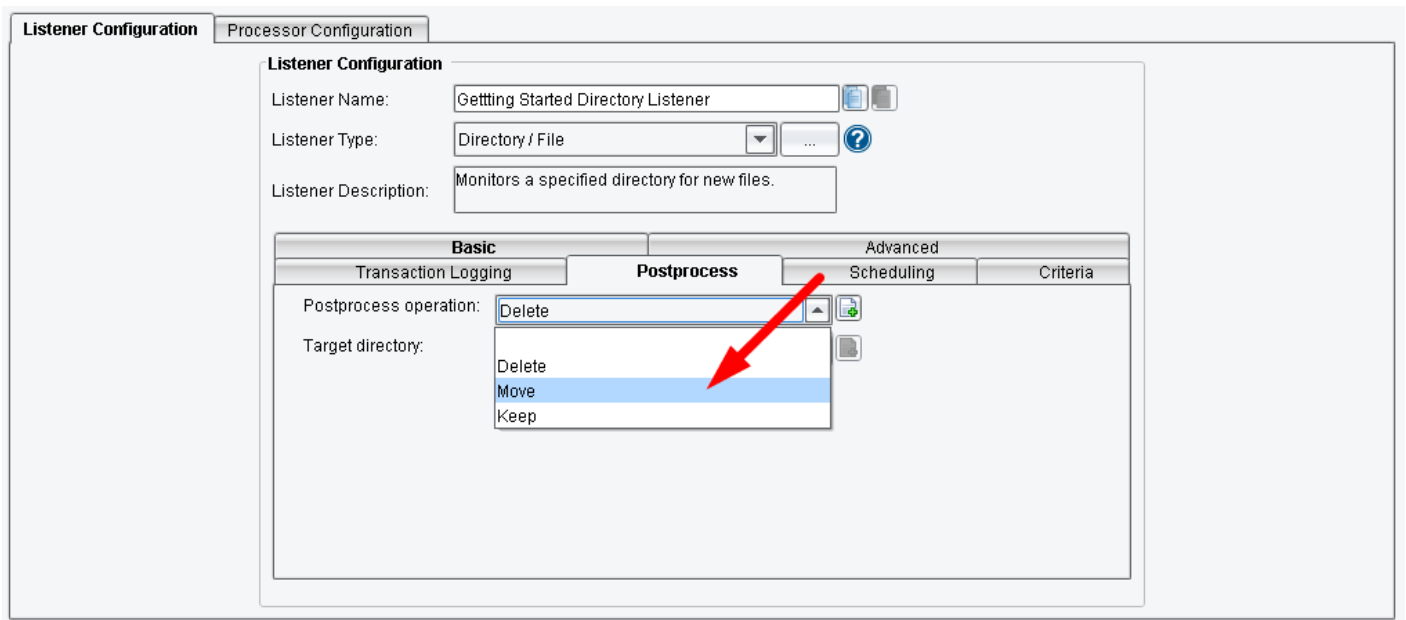




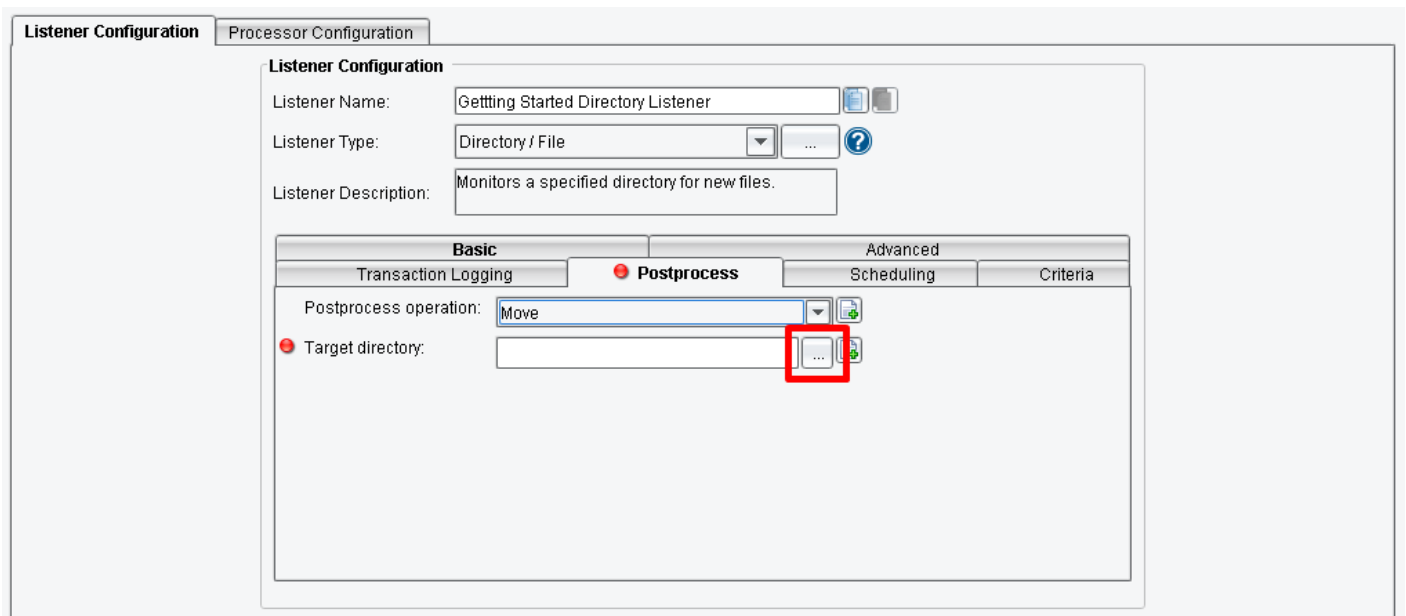
Which will bring up a file selection dialogue. Navigate to the **5 HR-XML Getting Started Interface** folder. Create a new folder called **"in"**, select it and click **Open**.



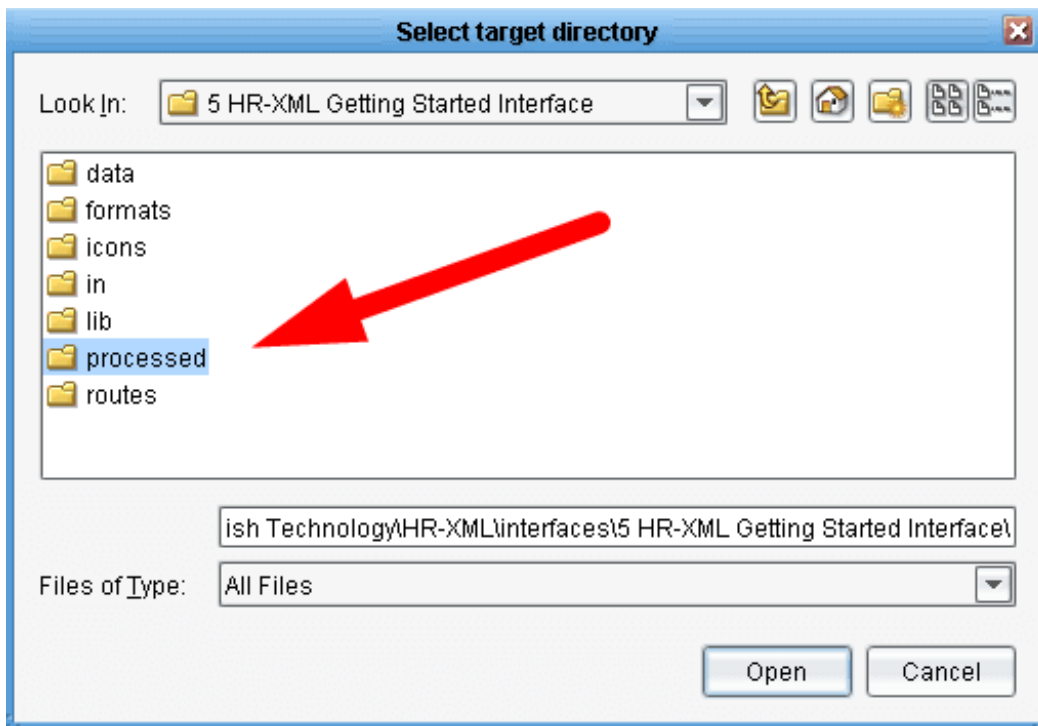
Now, when this Listener runs, it will poll the newly created drop directory every 1 second for any input file that you provide. Next change the polling interval. Select the current **Polling interval** of 1 second and enter the value **"10"**. You will also want to configure how the system behaves after it picks up the file. To do this, click on the **Postprocess** tab in the configuration area.



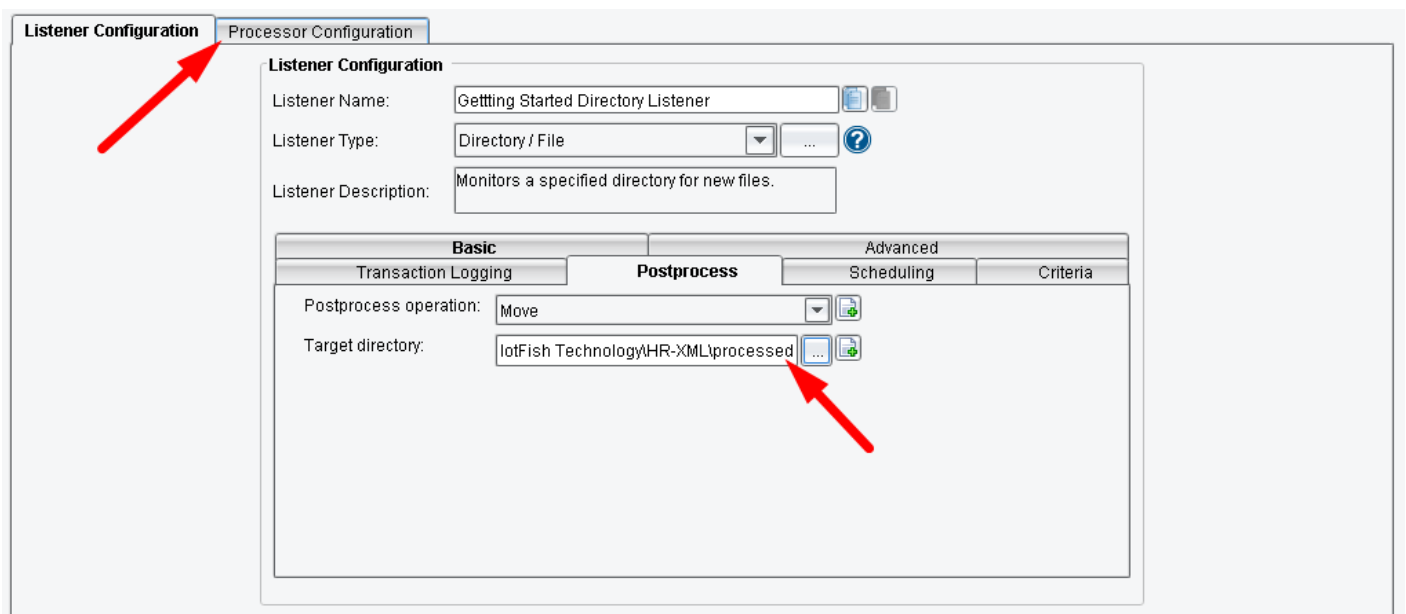
You'll note that the current Postprocess operation is to Delete the file. However, let's say you wanted to keep a backup, so you'd select **Keep**. Here, we will want to select **Move** from the Postprocess operation dropdown.



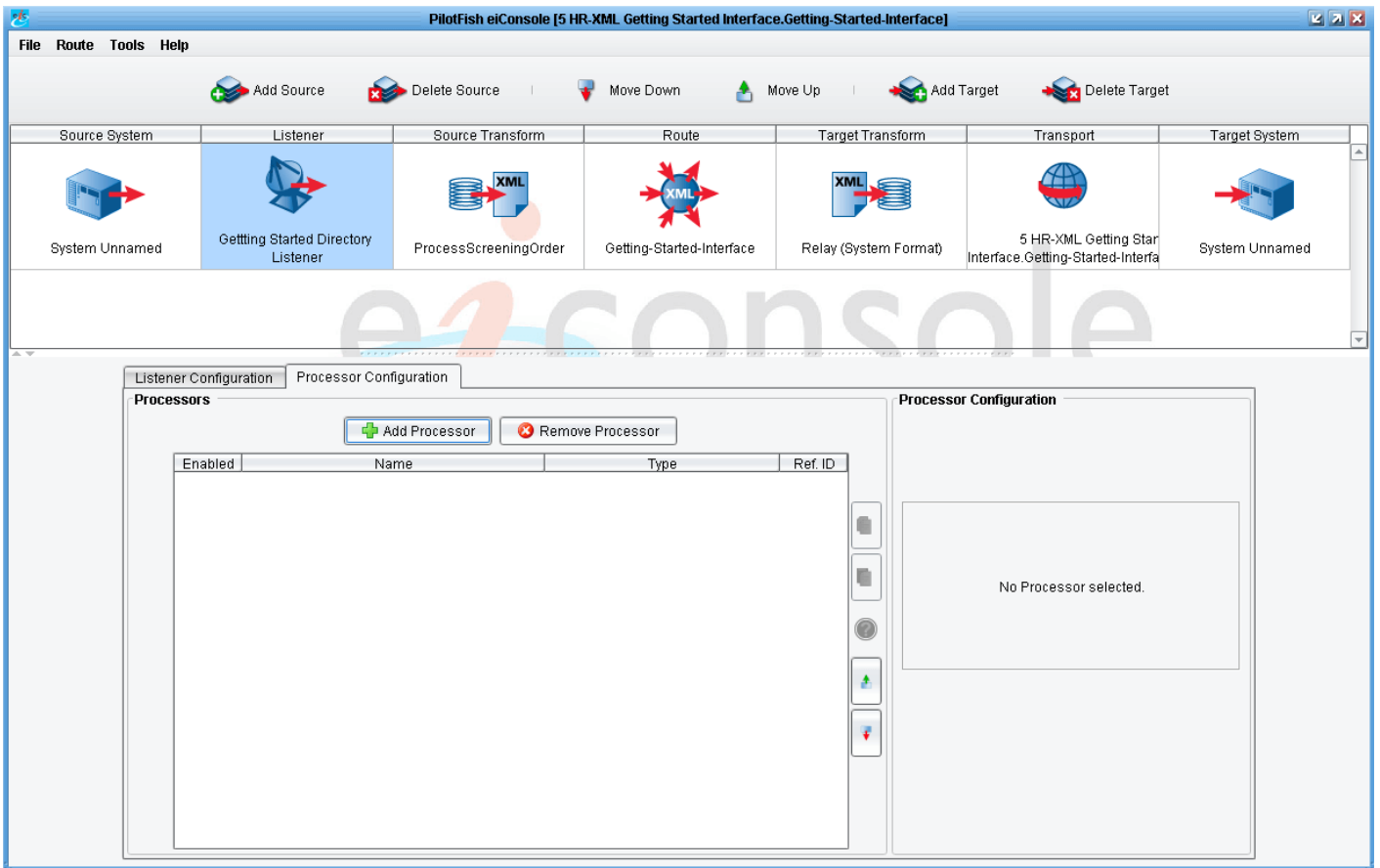
Now we will want to specify the Target Directory. To do so, click the **Ellipsis** button.



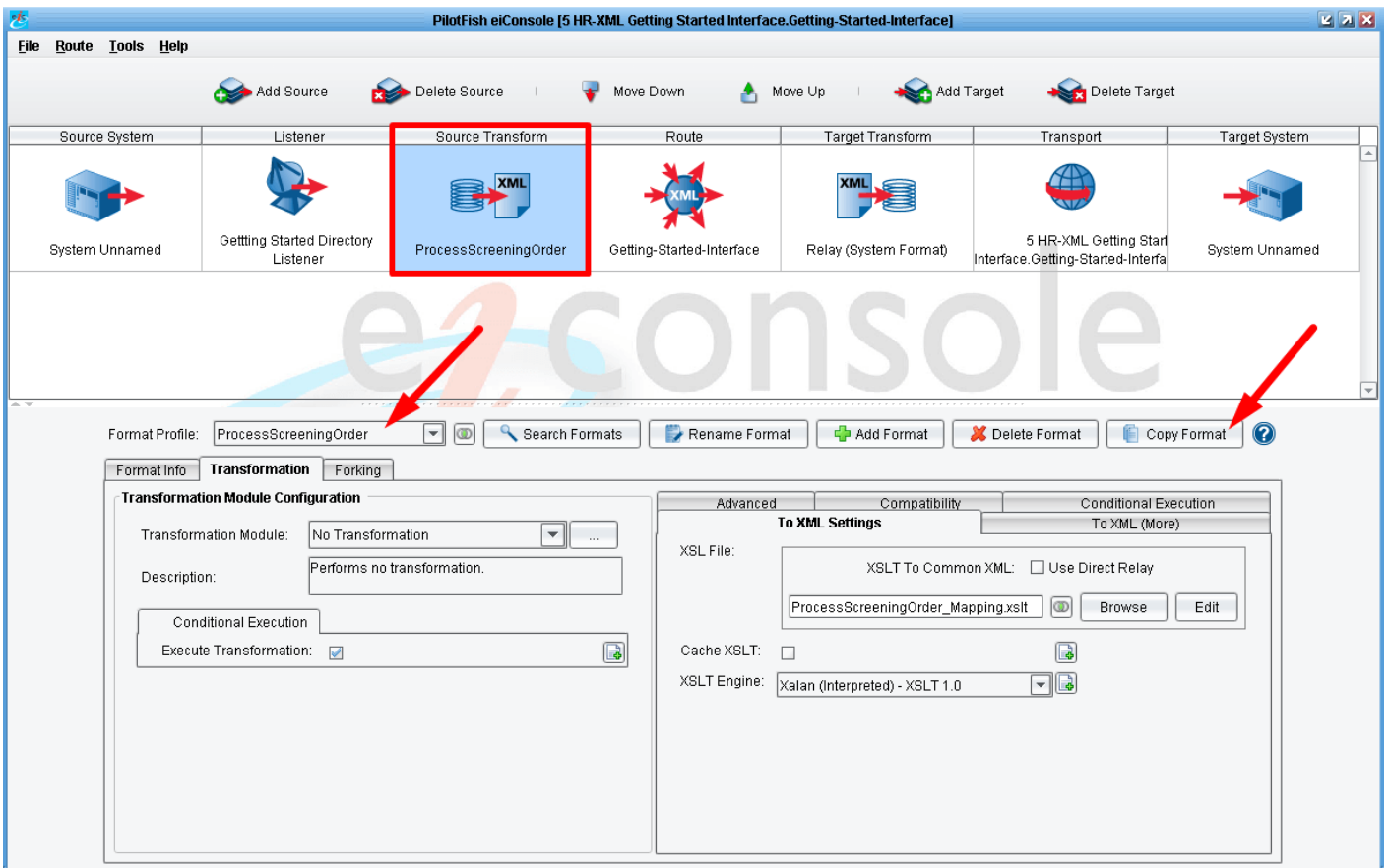
Once again this opens up a file selection dialogue. In the same folder as before create a new folder and name it "**processed**", then click **Open**.



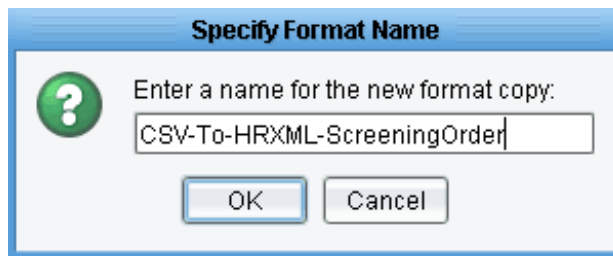
Now, as files are processed out of your in folder, a copy will be moved to the out folder. Let's move on. Select the tab next to the Listener Configuration tab, the **Processor** Configuration tab.



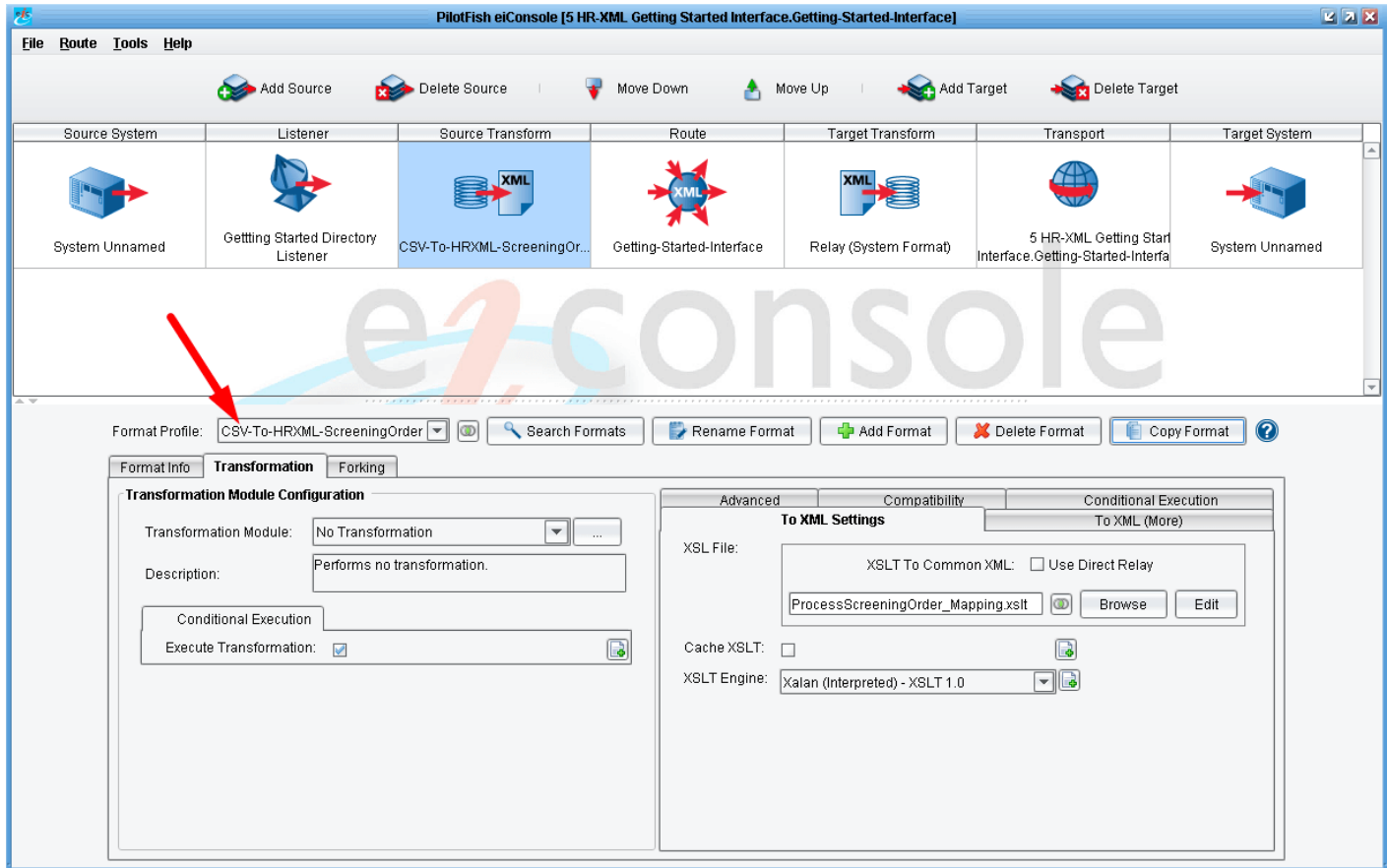
[Processors](#) allow you to do general work with the data stream using a collection of widgets shipped with the eiConsole. In this Route, none of this processing is necessary and we will move on to the next stage.



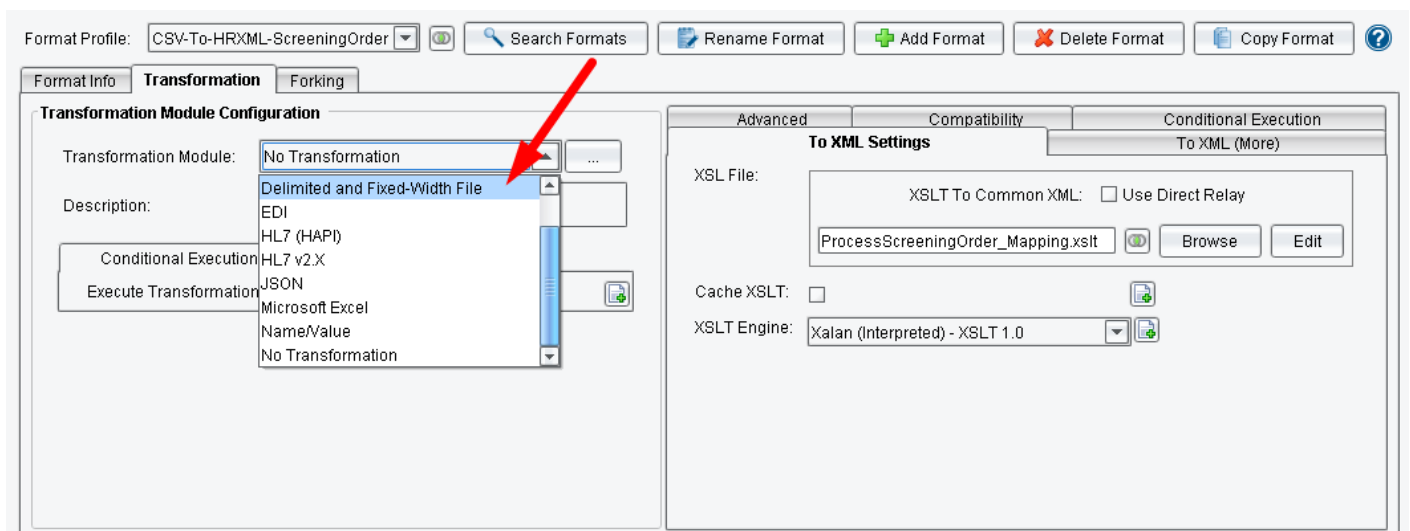
Now click on the **Source Transform** stage in the main route grid. You'll see below that the format ProfileProcessScreening order has been created. This ProfileProcessScreening order format profile is a reference to the existing template for the screening order. Because you don't want to modify the template, but instead want to work off of a copy, you'll need to create a copy of this format. Click the **Copy Format** button to the right of the Format Profile dropdown.



When prompted to provide a name we will call it "**CSV-To-HRXML-ScreeningOrder**" and then click **OK**.



You'll note that the Format Profile selected will change to the newly created format. In the Transformation tab you'll see 2 subsections. The [Transformation Module](#) Configuration and the [XSLT](#) Configuration. The Transformation Module is used to convert the data from its inbound format into XML if it hasn't yet been translated into an XML representation. The eiConsole supports this translation through a variety of [Transformation Modules](#). The most commonly used, and the one you'll use in this case, is the Delimited and Fixed-Width File Transformation Module.



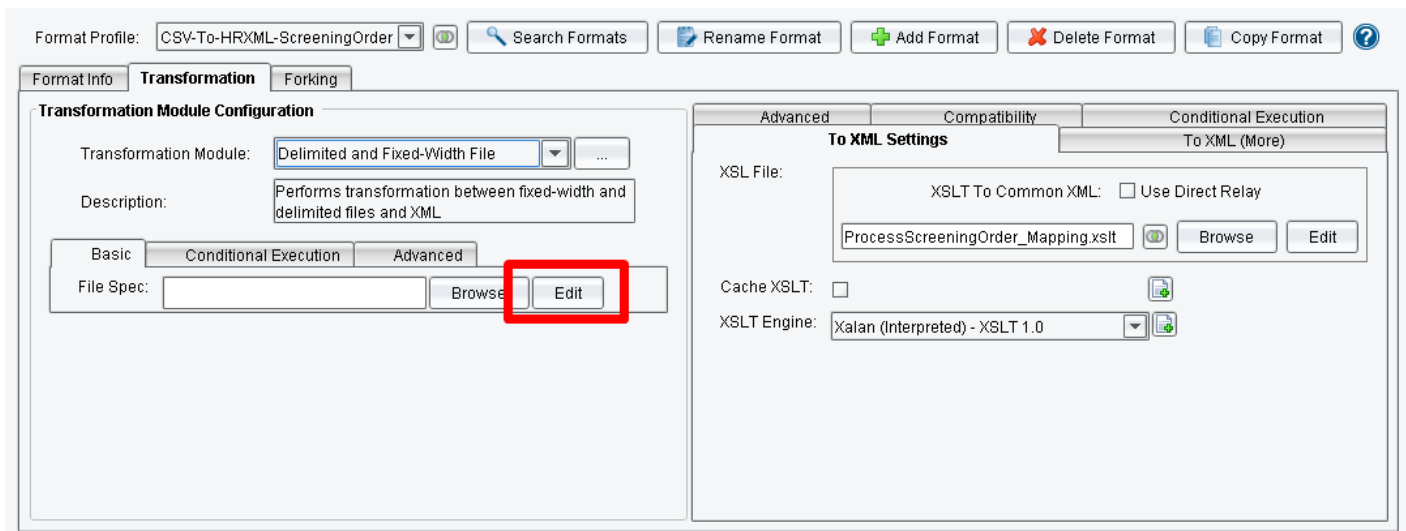
So you'll want to select the **Delimited and Fixed-Width File** Transformation Module from the dropdown.

```
candidates.csv
1 FirstName,MiddleName,LastName,Addr1,Addr2, City, State, ZIP, DOB, Credit, Criminal
2 Mick,E,Moose,1 Castle Court,,Orlando,FL,32801,11/18/1928,X,
3 Nick,N,Pole,1225 Main Street,,Santa Claus,IN,42759,2/17/1928,,X
4
```

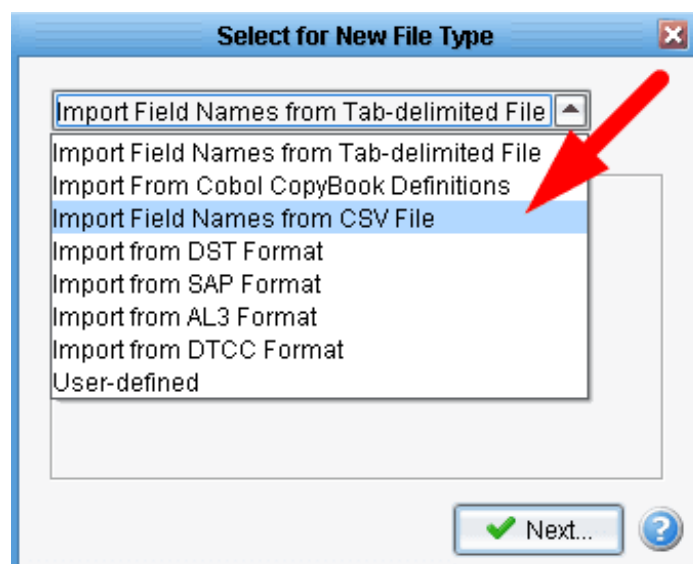
length : 204 lines : 4 Ln: 1 Col: 1 Sel: 0|0 Dos\Windows UTF-8 INS

The Delimited and Fixed-Width File Transformation Module will allow you to convert data from a delimited or fixed-width file format into an XML representation that can be further transformed. In this case, the input will be a csv file. Before you go any further, let's take a look at that file. You'll find the file **candidates.csv** in the sample tutorial, **2 HR-XML Getting Started Tutorial** within the **"in"** folder or in the 3 HR-XML Templates (interfaces3 HR-XML Templatesdatatutorialinput). Open this in a text editor.

When you open this file you'll see it's relatively simple. It's a csv file where the first row indicates column headers and each row thereafter is a data row including some basic information about the candidate. The last two columns are used to indicate whether you want to perform a credit check or criminal background check for each of these candidates.

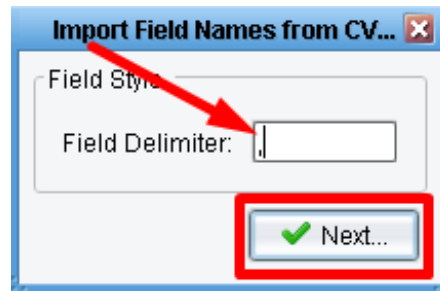


To describe the structure of the file to the eiConsole click **Edit** next to the File Spec configuration item in the Transformation Module Configuration section.

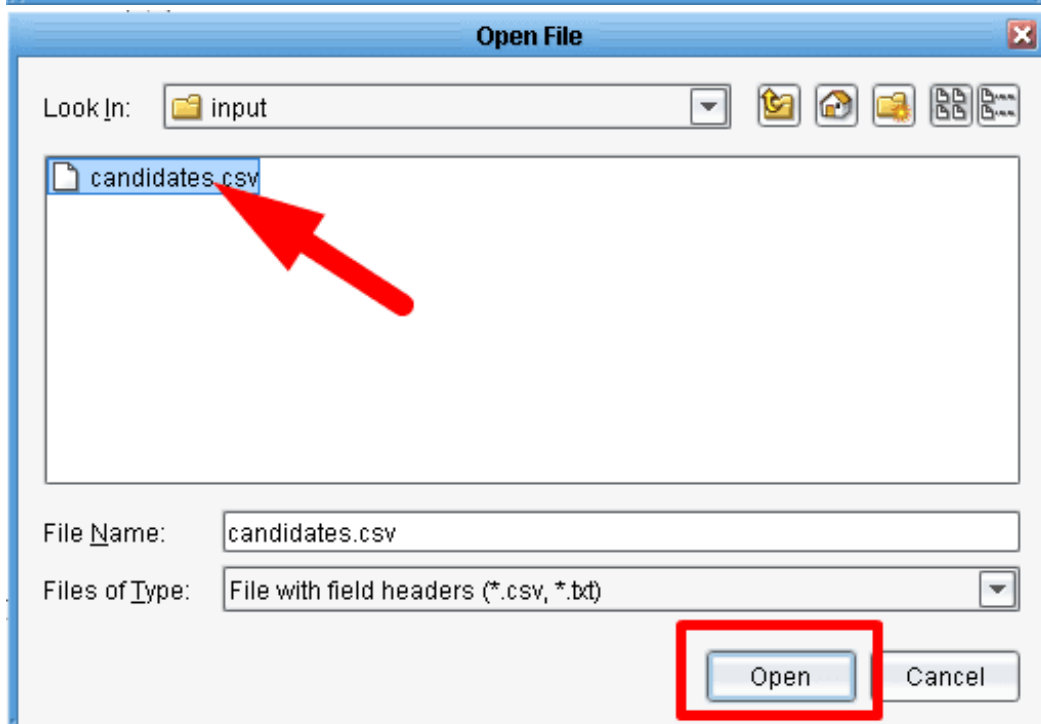
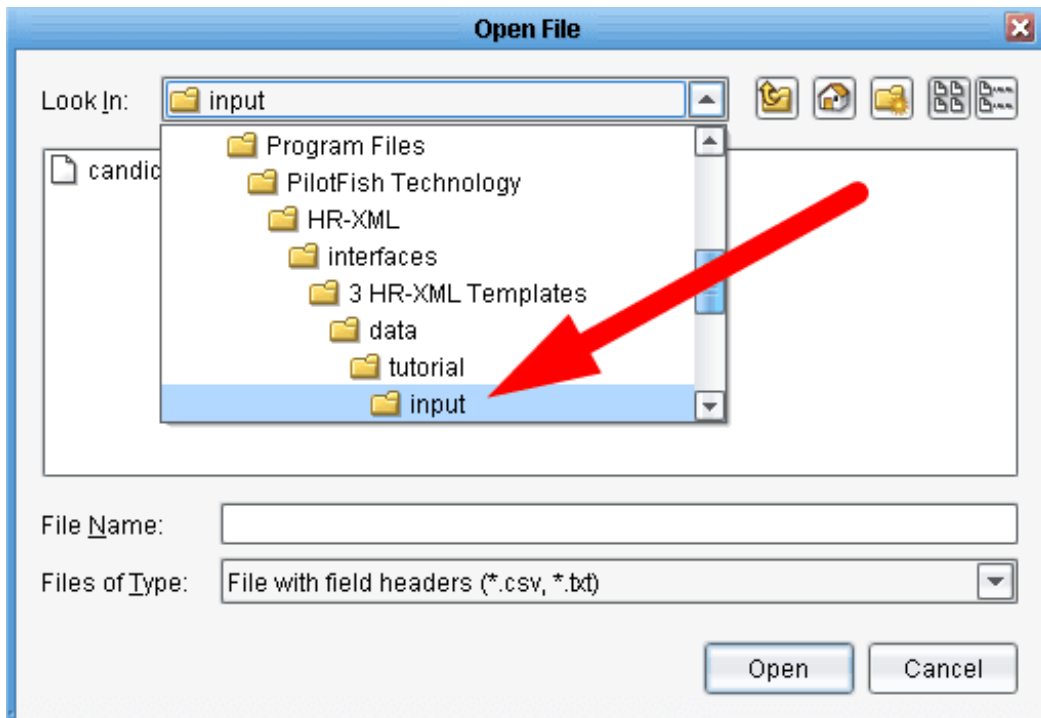


This will launch the [File Specification Editor](#). It will also launch the new file specification wizard. In the Select for New File Type dropdown you'll need to select **Import Field Names from CSV File**.

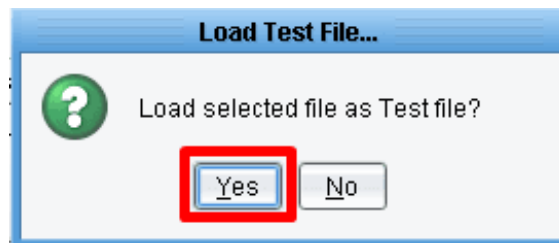
Click that entry in the dropdown and then click the **Next** button.



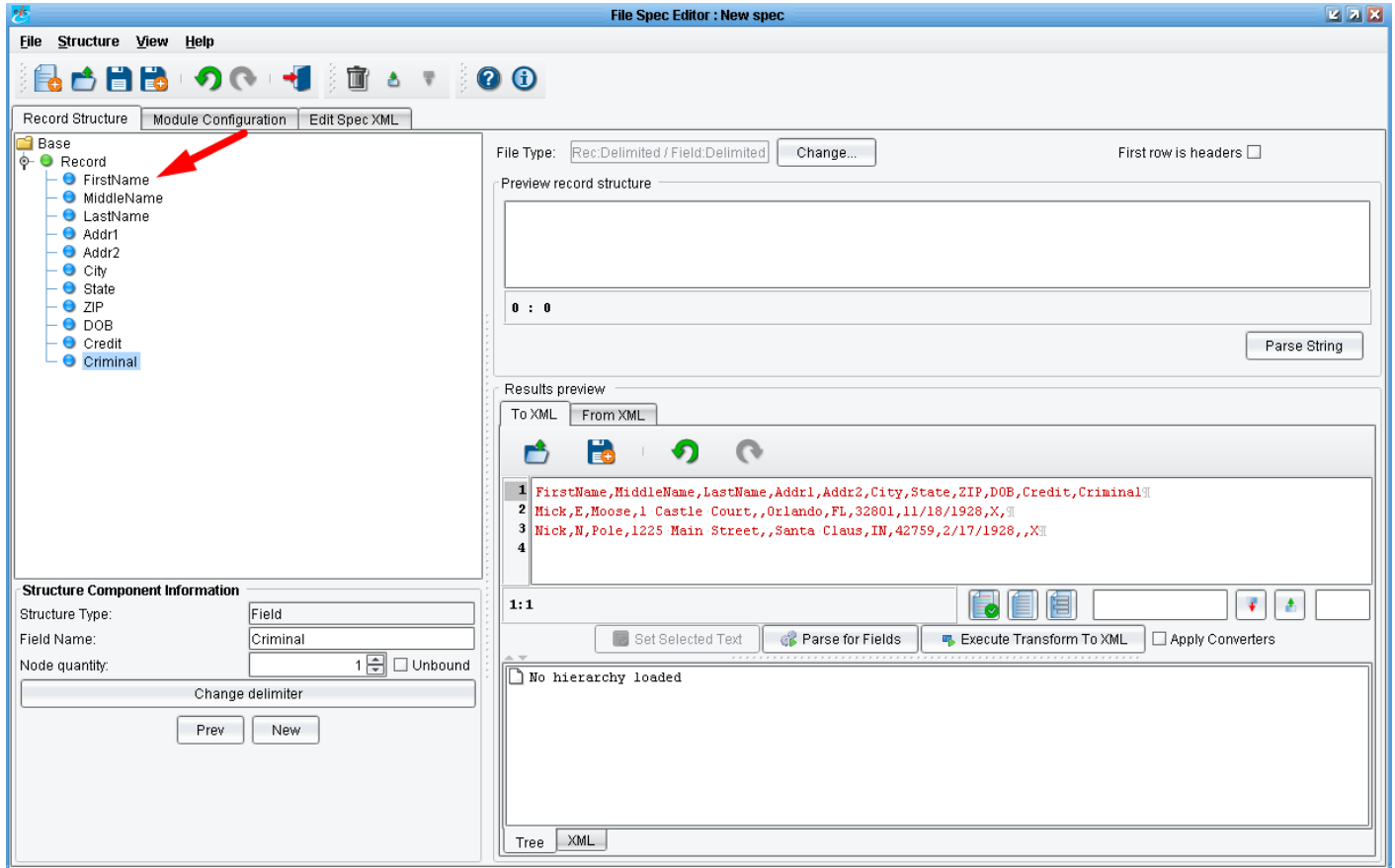
You'll see that the Field Delimiter is correctly set to a comma, so then click **Next** again.



Next you'll need to provide the sample file from which it will read the format. Once again, navigate to the **2 HR-XML Getting Started Tutorial** open the "in" folder and then select the **candidates.csv** file and click **Open**.

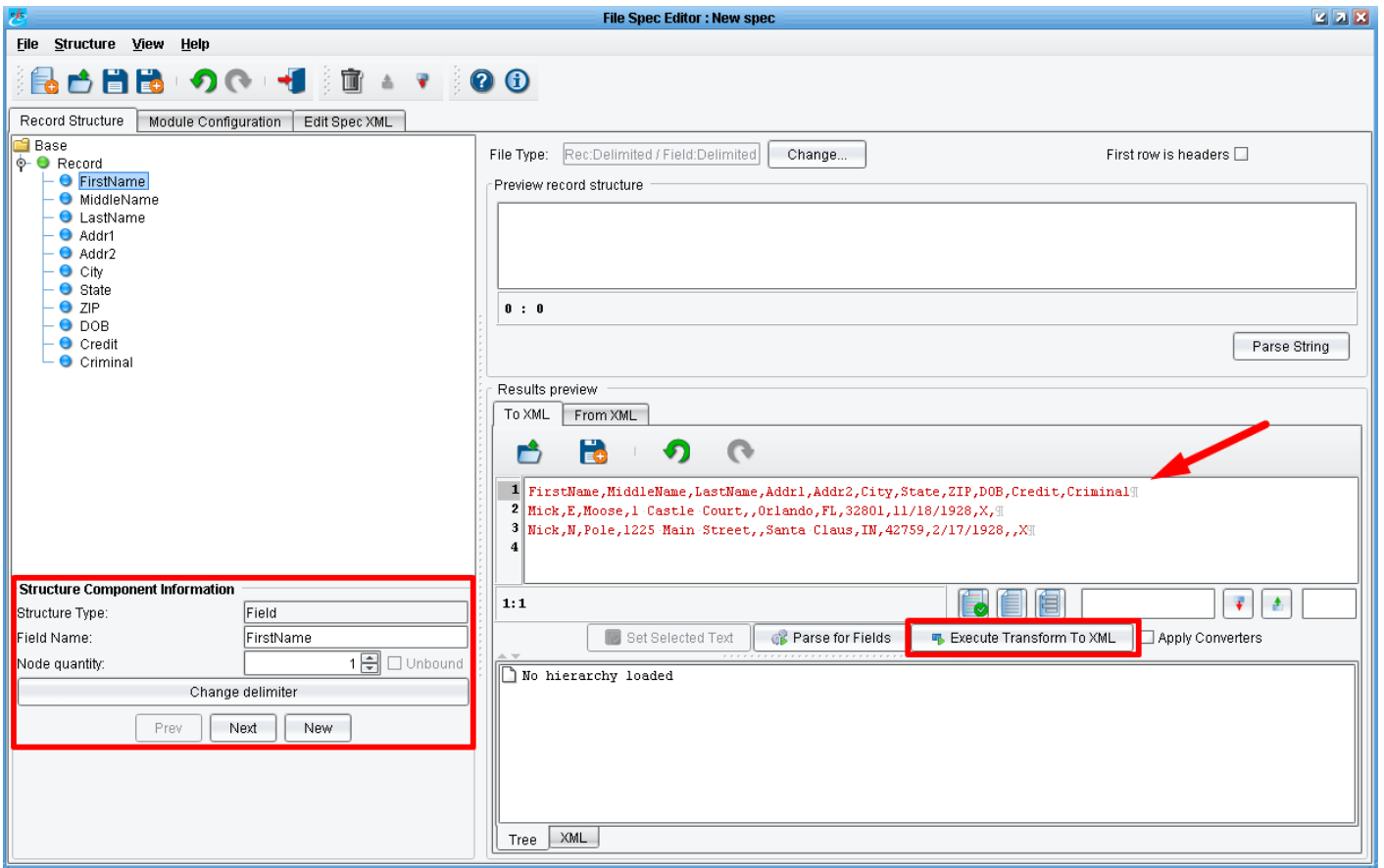


When prompted to **Load Test File** as the Test file, click **Yes**.



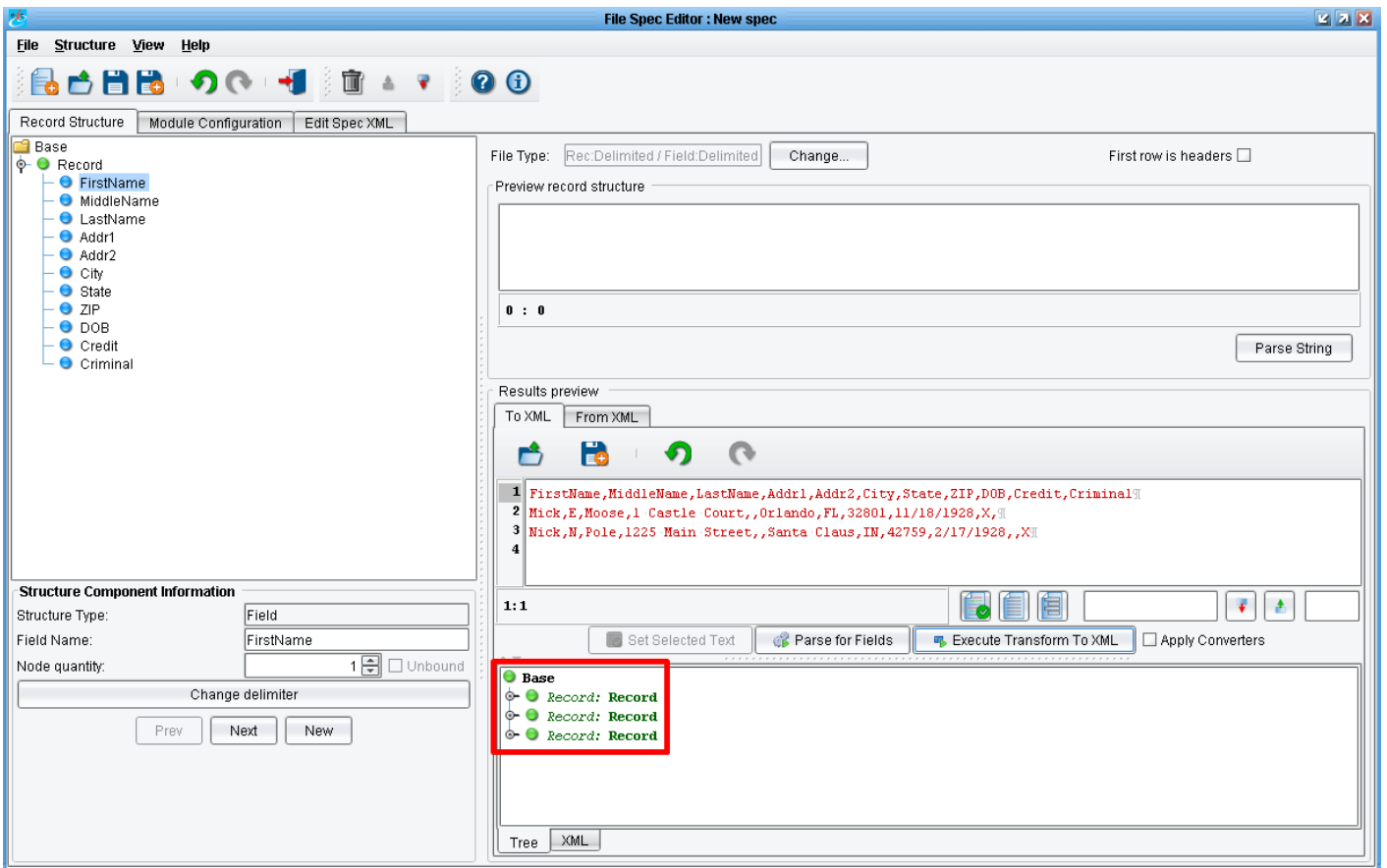
You'll notice that the left hand side of the screen is now populated with a tree that represents the format of that csv file. You have one Record type denoted by a green [node](#) that's name has been defaulted to record. You then have a number of different fields, each represented as a blue node in the list.

Select the **First Name** node.

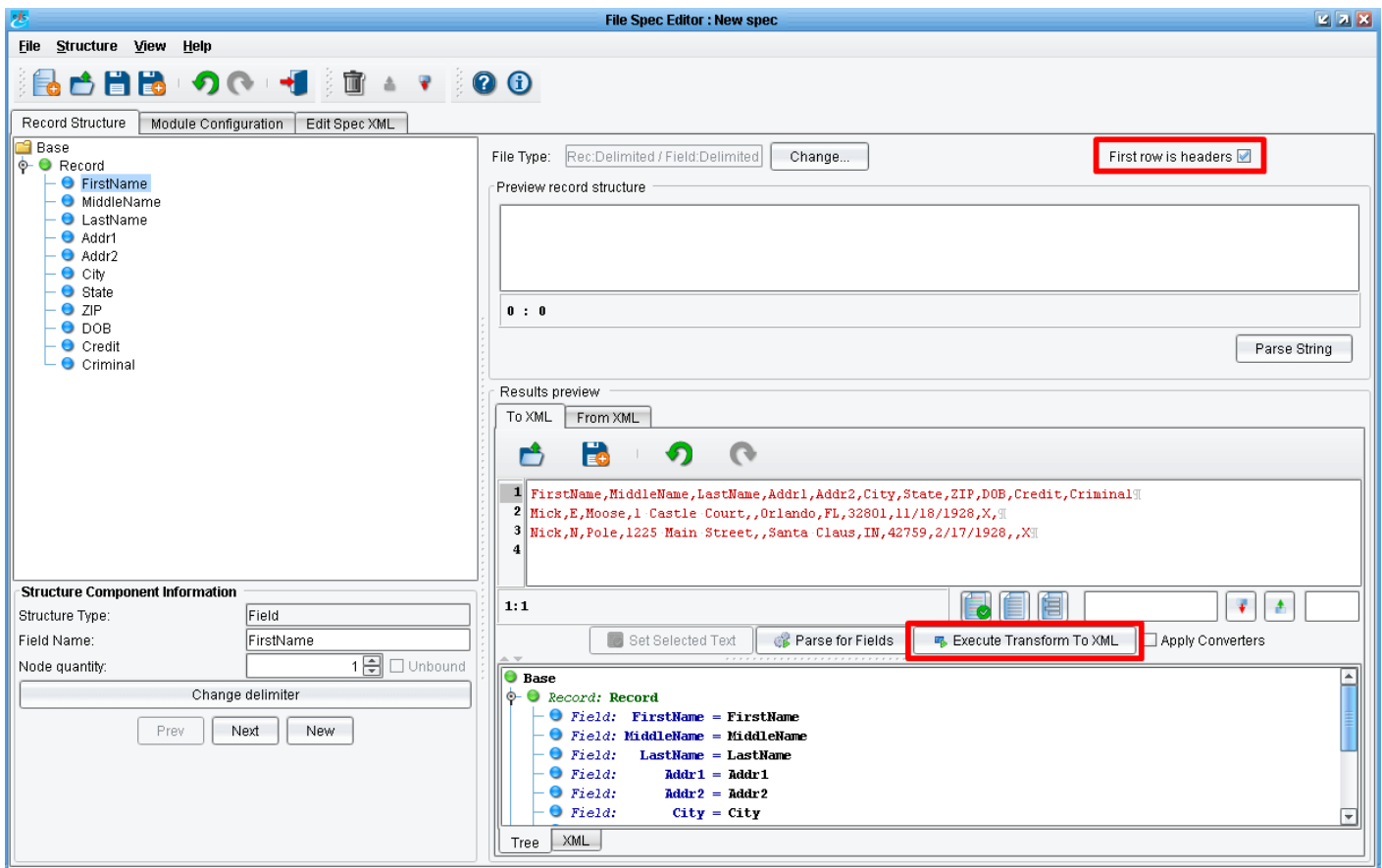


Details about the field appear in the panel underneath the tree.

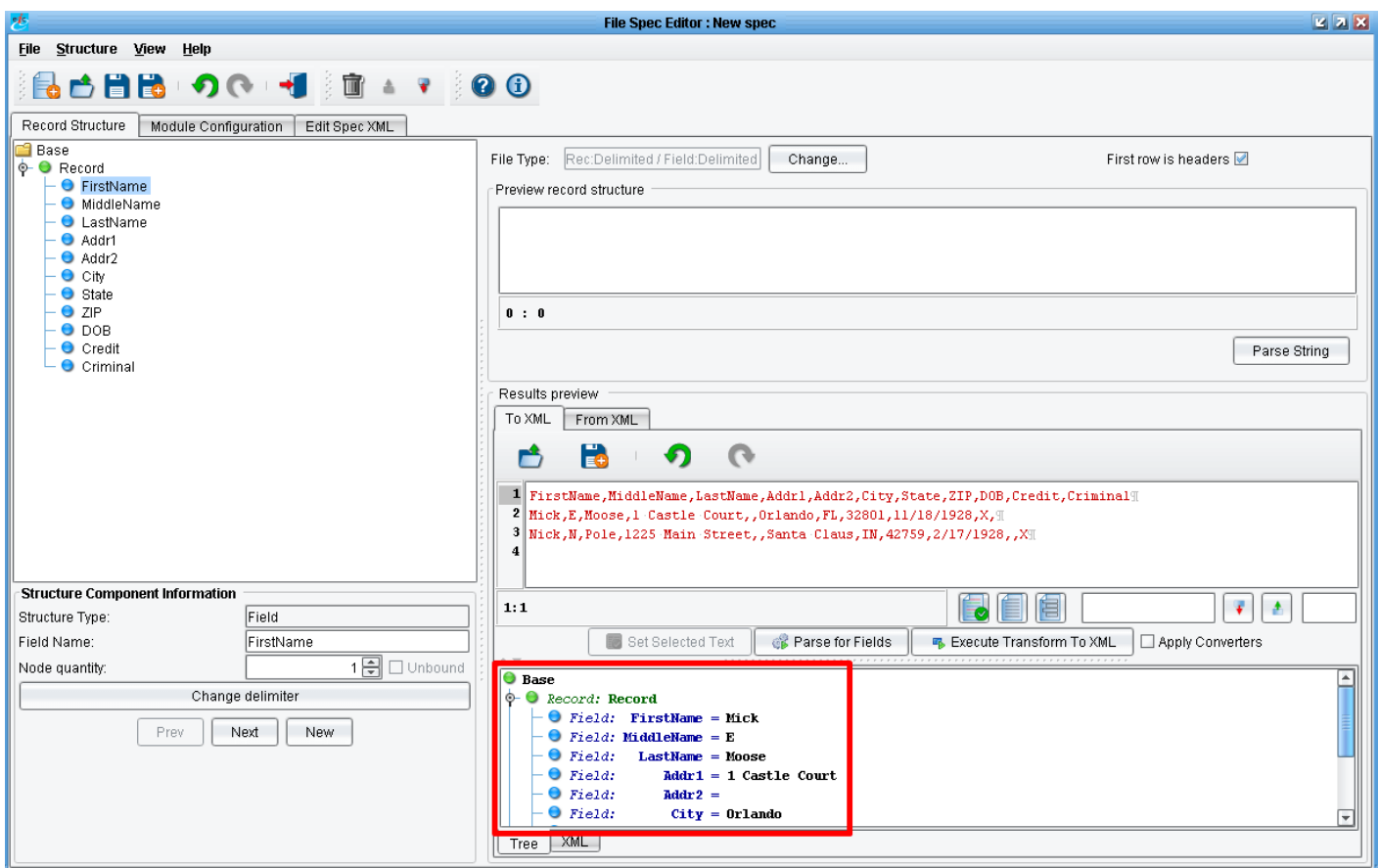
The sample file appears in your **Results preview** window. To preview the transformation of this data into a parsed structure, click the **Execute Transform to XML** button.



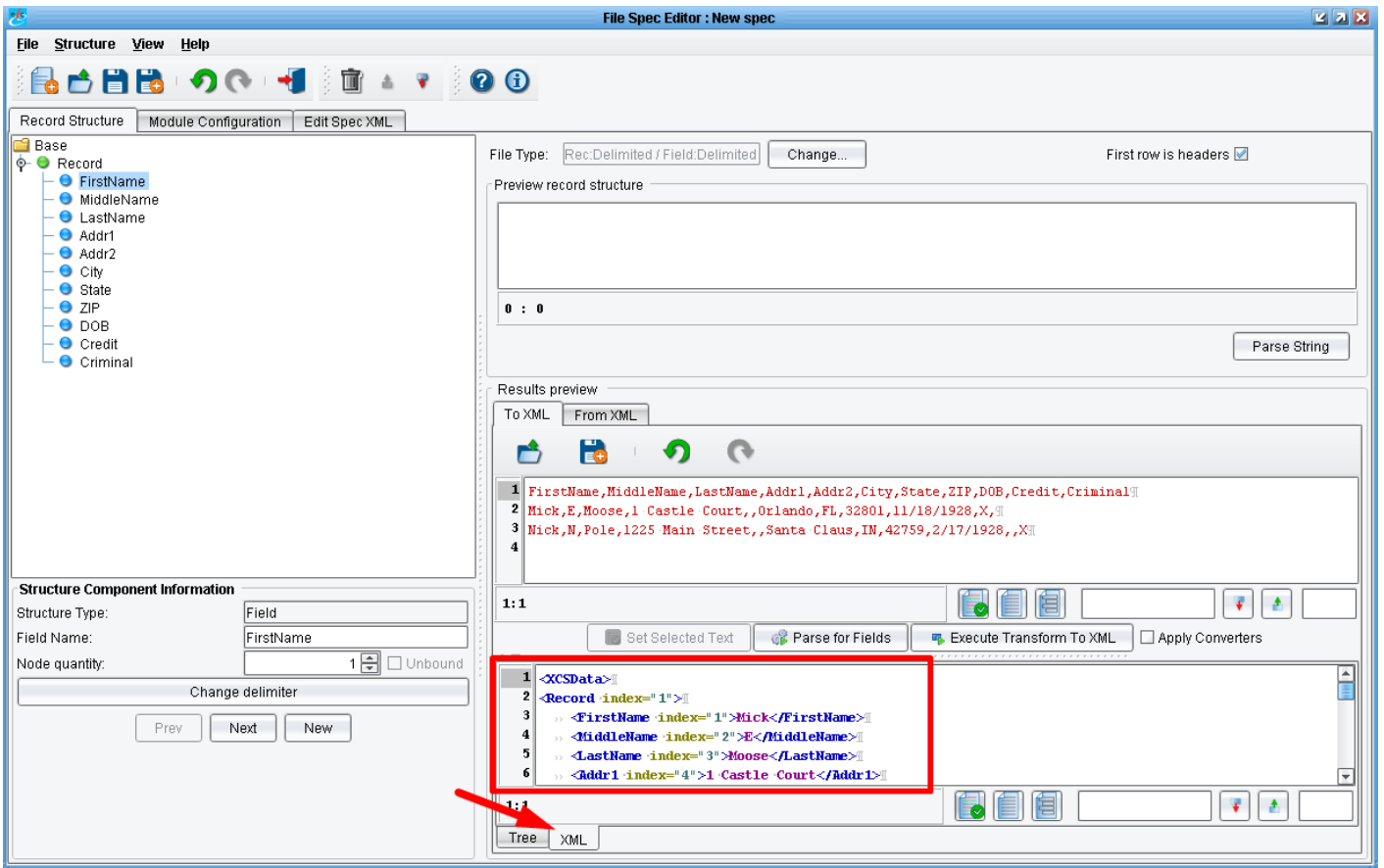
You'll see a new tree appear directly underneath the Results preview area. This tree can be expanded to show how the system parsed the sample file above. Click the [nodes](#) to expand the tree.



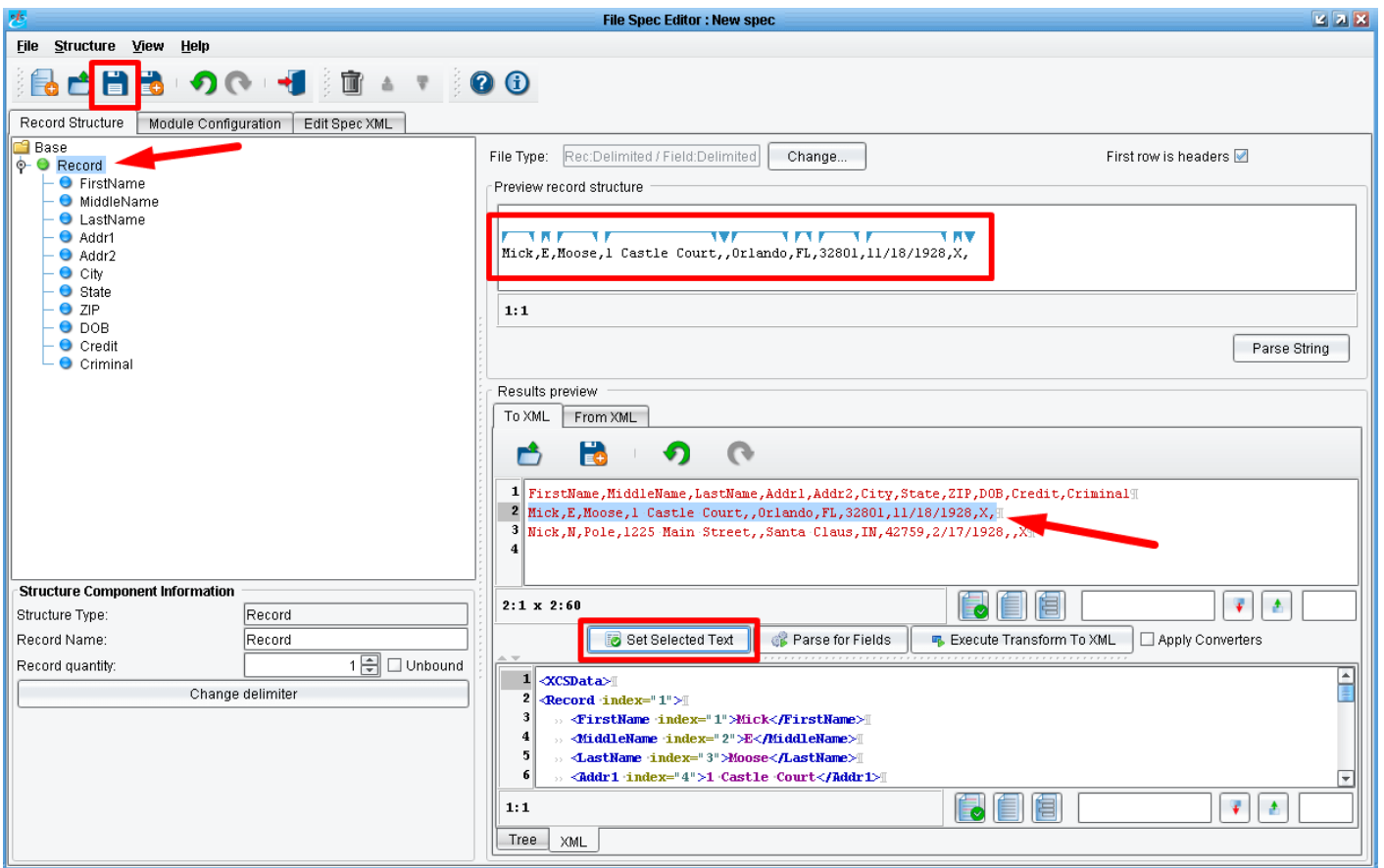
You'll see that the system has incorrectly identified the first row here as data. Instead, this should be ignored as it contains field headers. Check the **First row is headers** checkbox at the top of the screen. And then click **Execute Transform to XML** again.



Click the nodes to expand the tree. You'll see that now the system has correctly found and parsed the two records contained in your data file.

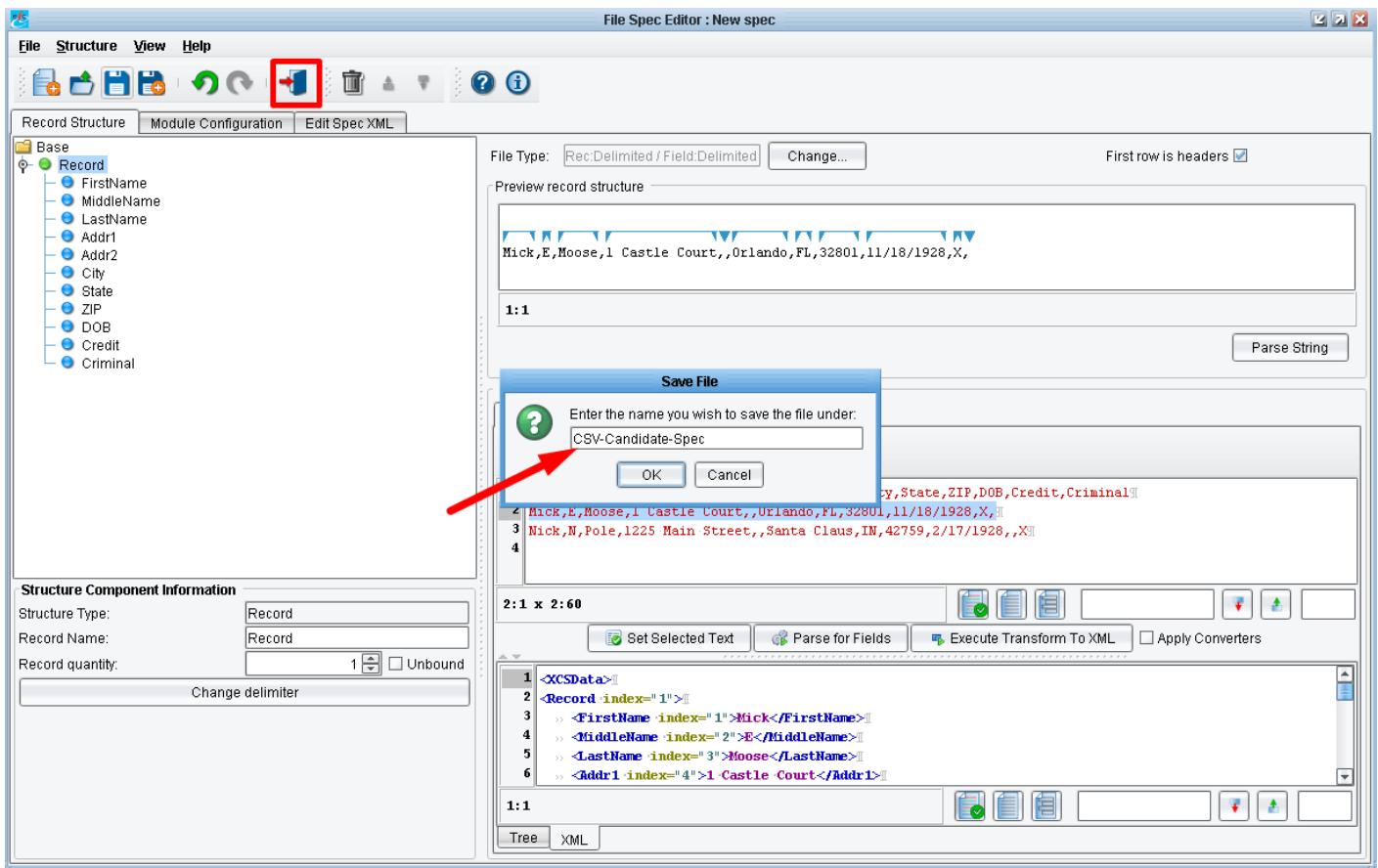


If you click on the XML tab, you'll see the XML structure that the File Specification will generate.



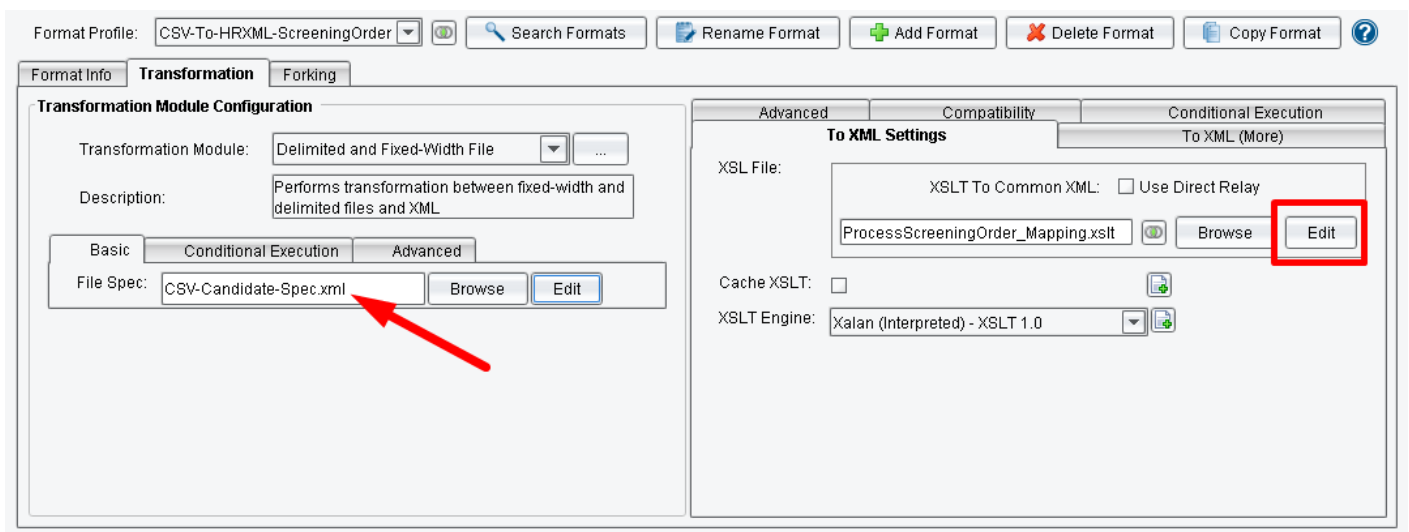
You can also validate how a particular line within the file is parsed. To do this, select a line of the file, select a corresponding Record on the left, and click the **Set Selected Text** button.

If all looks in order you can save your File Specification and move on. Click the **Save** icon in the upper left menu bar.



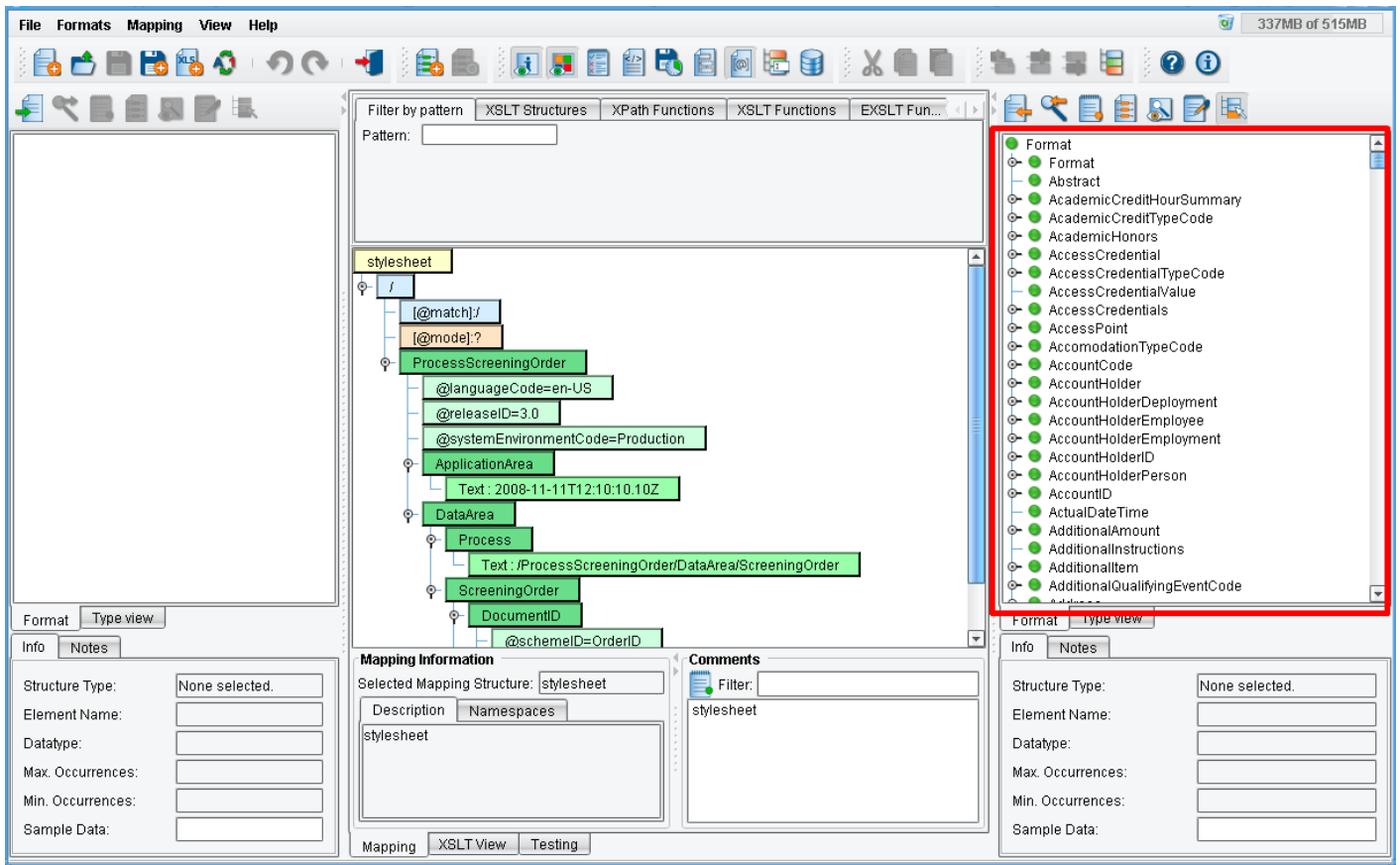
And give the File Specification a name. Let's call it "CSV-Candidate-Spec" and click OK.

Then click the **Return to Console** button in the menu bar.

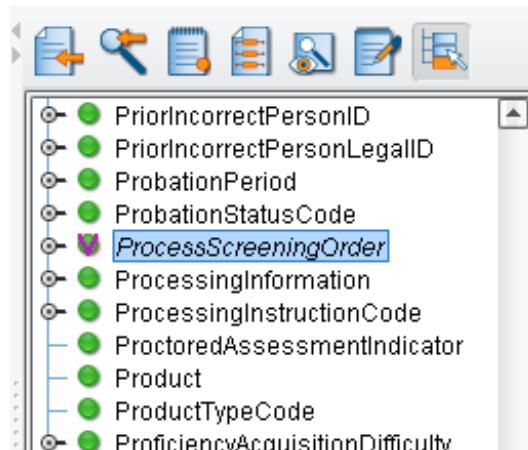


You've now returned to the main route grid where you see that the File Specification has been populated with XML file generated by the File Specification Editor.

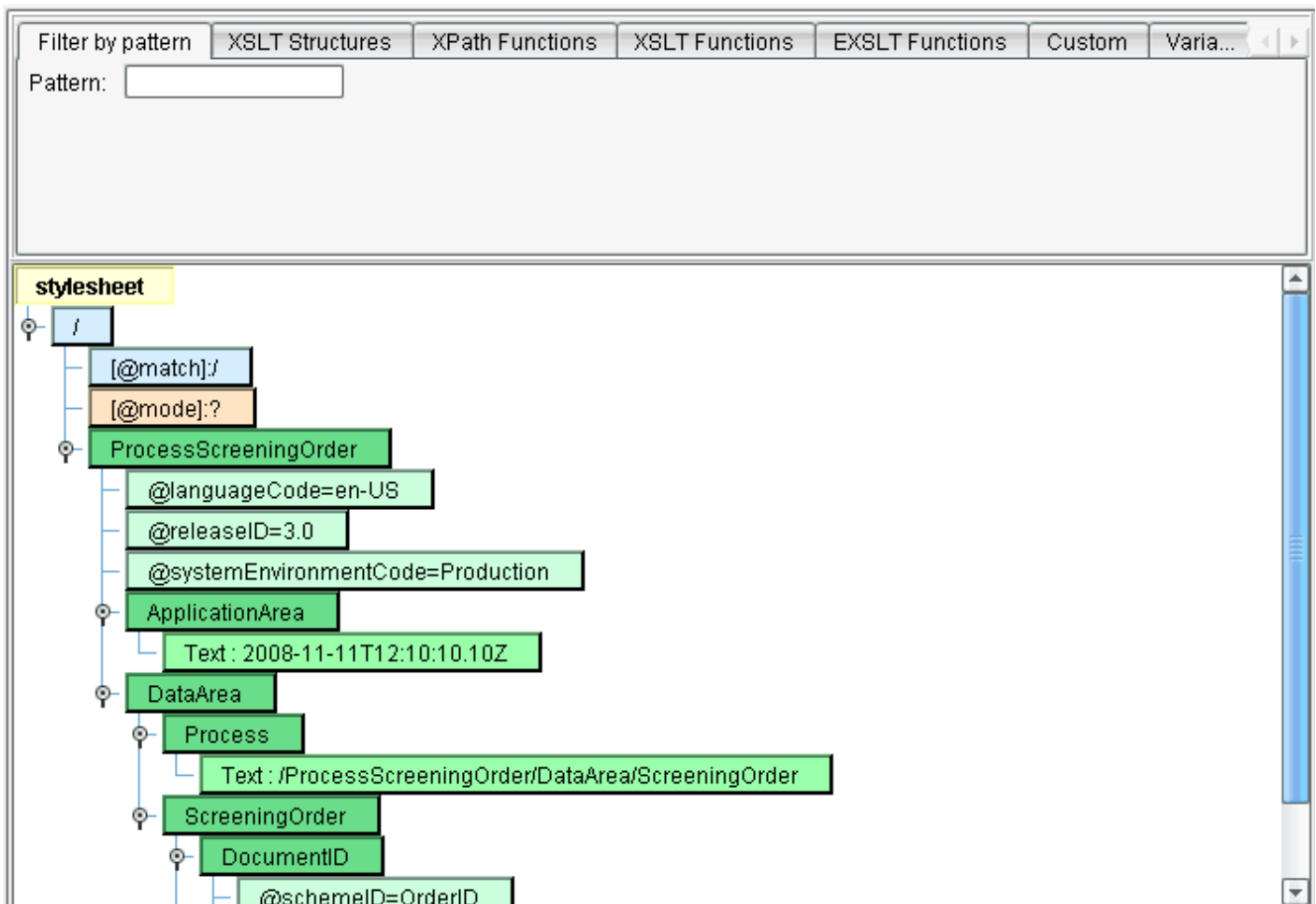
Now that the CSV file can be converted into XML, your next step is to generate the XSLT that will map that XML onto the ProcessScreeningOrder HR-XML BOD. To do this, click the **Edit** button next to the XSLT name on the **To XML** tab.



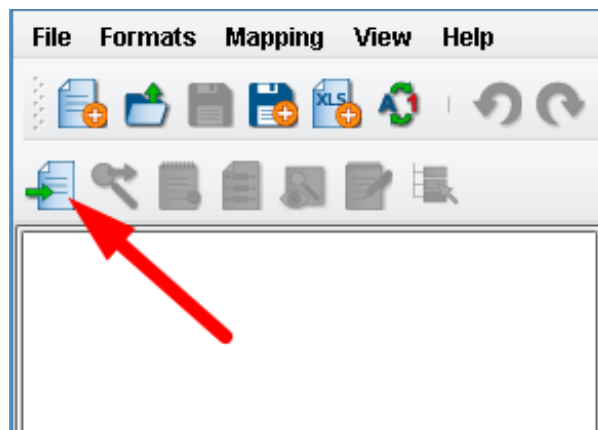
This launches the [Data Mapper](#). The Data Mapper is the tool used within the eiConsole to create logical mappings between 2 different data [formats](#). You'll see here that when you load a template, the panel on the left hand side is blank. This will need to be populated with the format of the data that you wish to map from. The tree on the right hand side has been populated with a tree representing the structure of the HR-XML model.



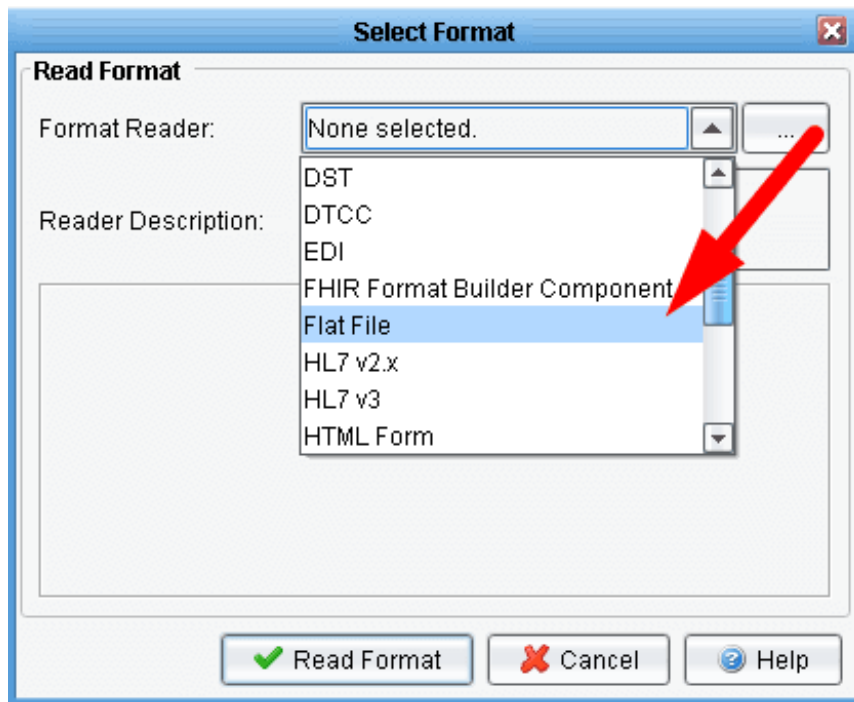
You'll note that the **ProcessScreeningOrder** node has been italicized and has a checkmark. This indicates that this is the particular element, or one of the elements, that has been used in the mapping.



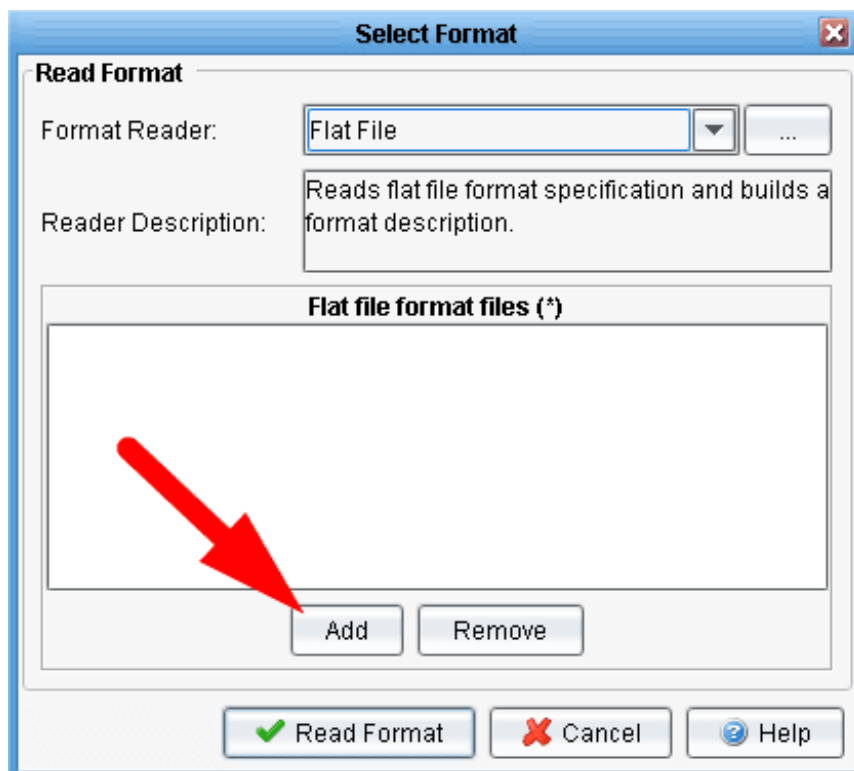
The middle panel contains a set of green nodes, and will contain a set of blue nodes as well. The green nodes indicate XML elements that will be in your output. Here you see the bare bones requirements for the ProccecssScreeningOrder message. There are currently no blue nodes because no values from the Source side have been mapped in yet. Your job will be to load the format of the Source and then map those onto the correct ProcessScreeningOrder template.



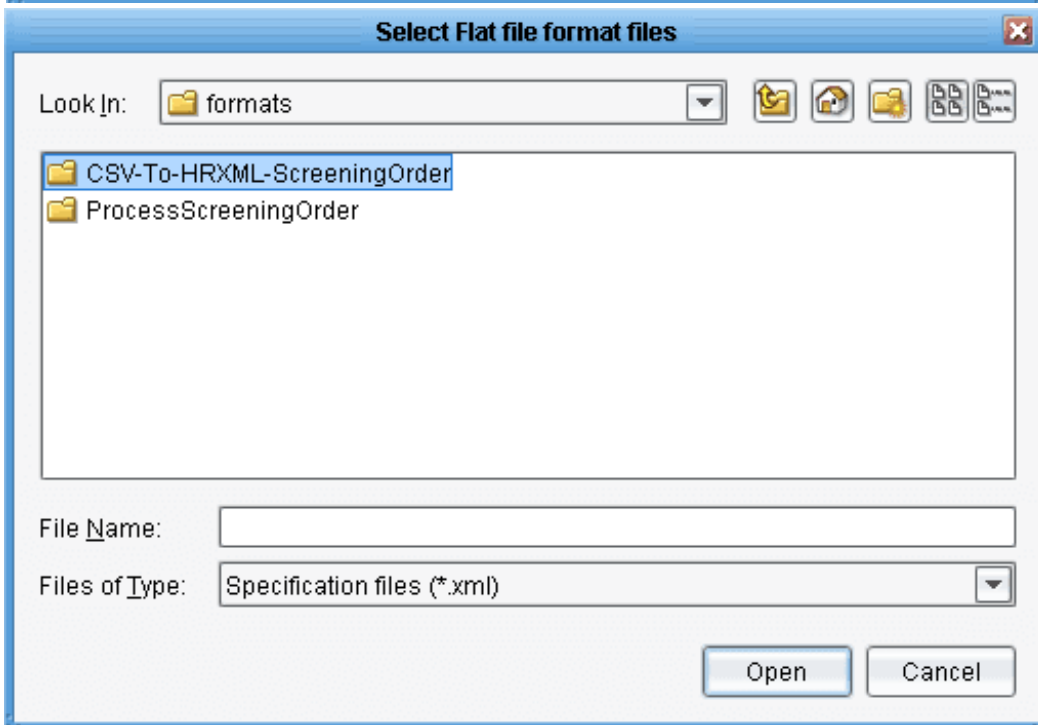
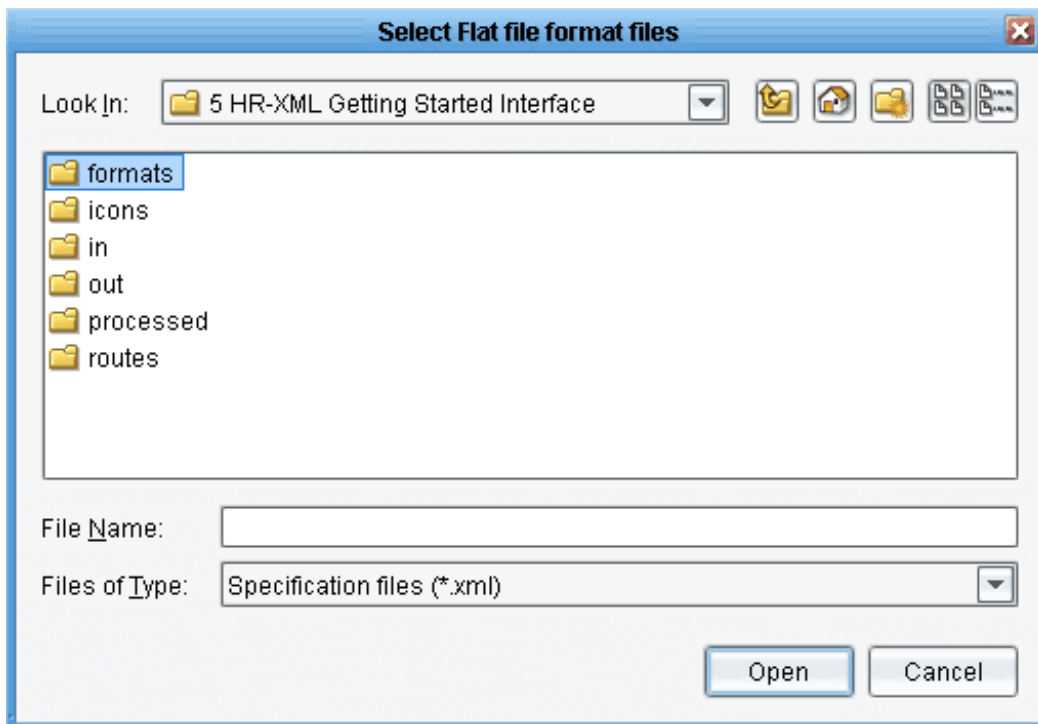
To begin this process, load the File Specification as your Source format. To do this, click the **Open Source Format** button in the top menu bar.



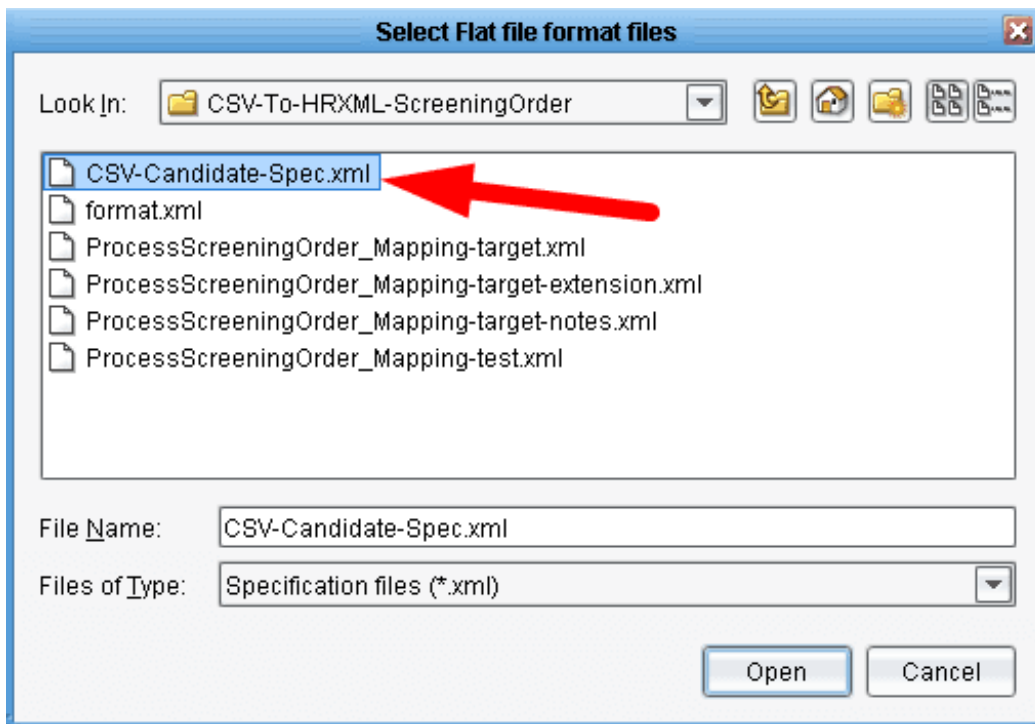
When the Select Format dialogue appears, choose the **Flat File** format reader.



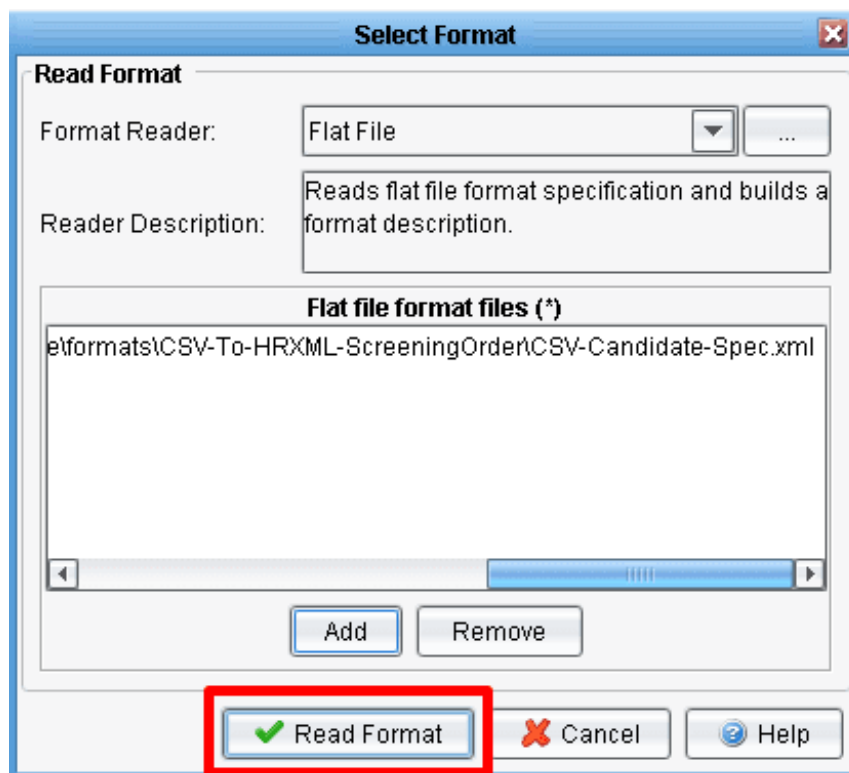
Once selected from the Format Reader dropdown you'll need to select the specific File Specification file. To do that, click the **Add** button underneath the Flat file format files area.



You'll now need to navigate to the file specification you created in the previous step. To do this, locate the formats folder underneath your distribution. A new format folder will have been created to match the name that you provided when you created a copy of the **ProcessScreeningOrder** template. Find that folder called **CSV-To-HRXML-ScreeningOrder** and double click it.



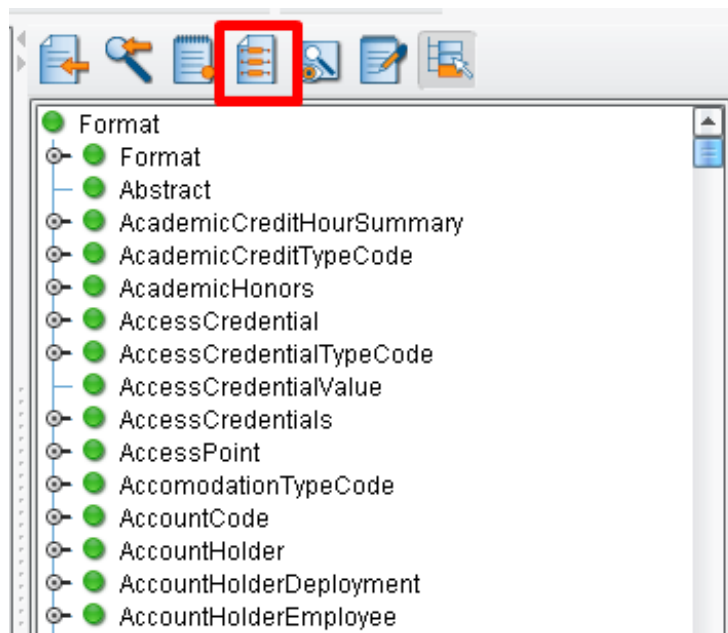
Inside this folder you should see the File Specification that you created, the **CSV-Candidate-Spec**. Select this file and click **Open**.



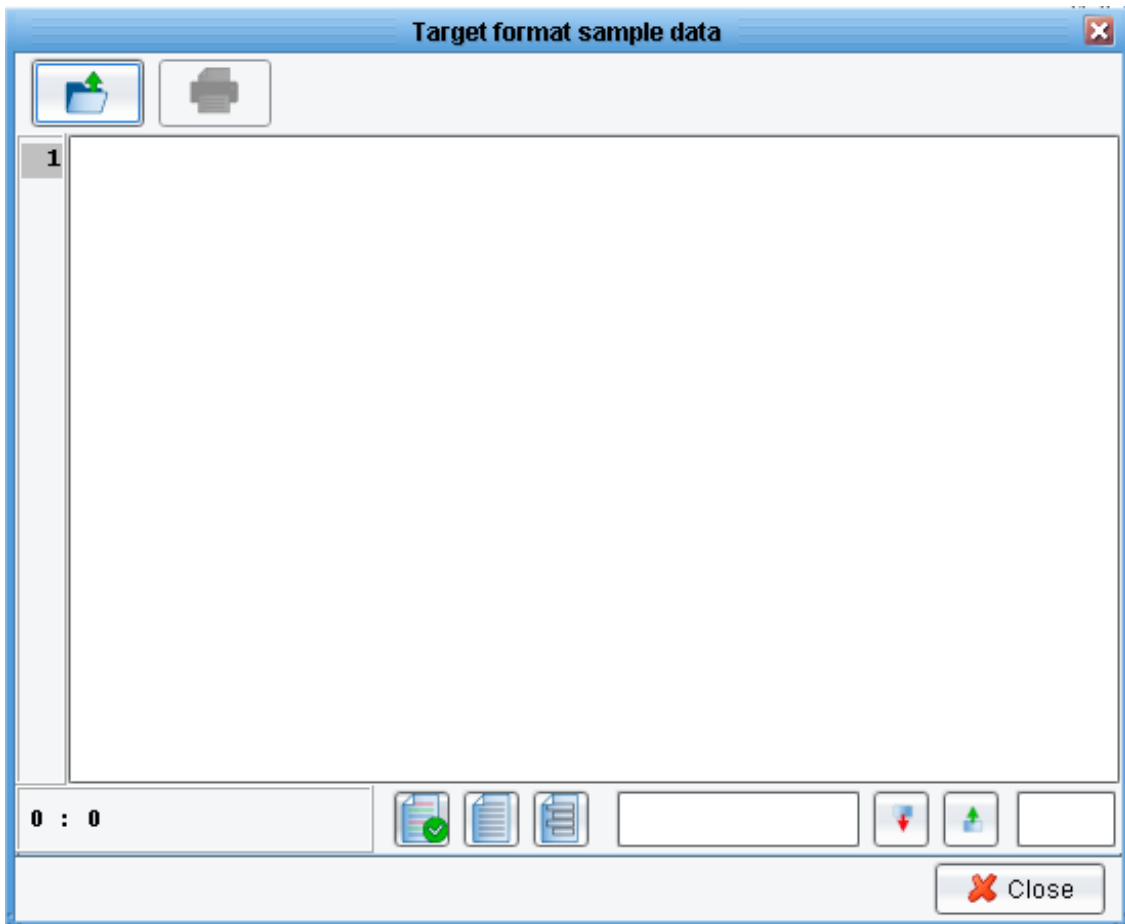
Then click **Read Format**.



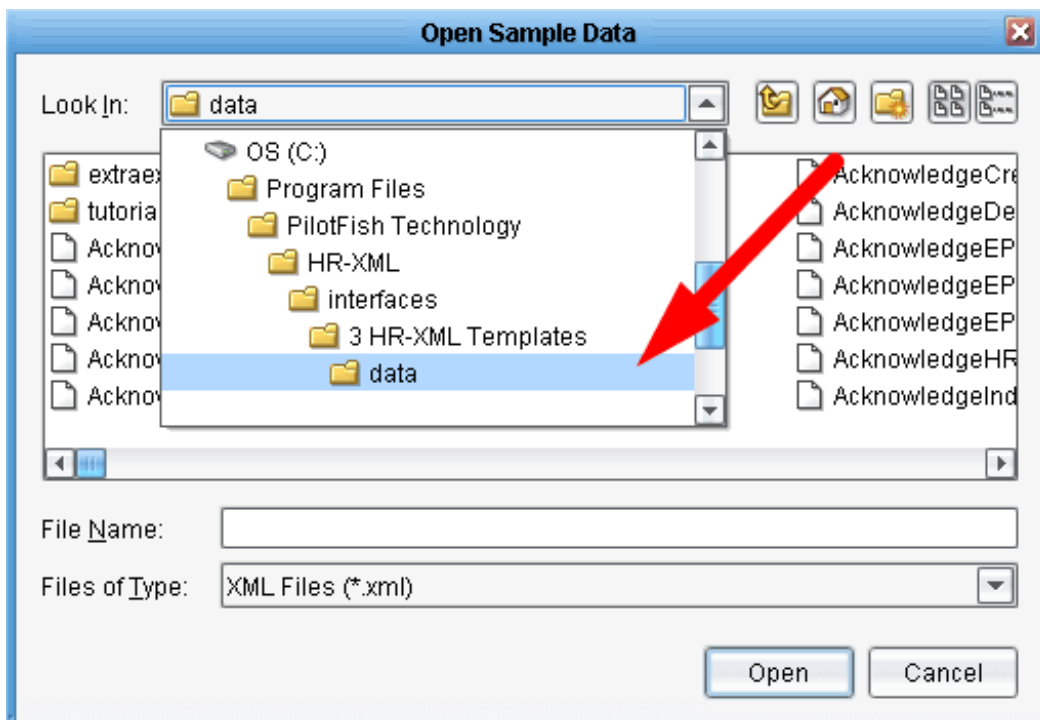
The left hand panel will be populated with a tree very similar to the one you just saw in the File Specification Editor. The Route node is XCSData. Each Record is represented with a Record node. Each individual field is a leaf node.

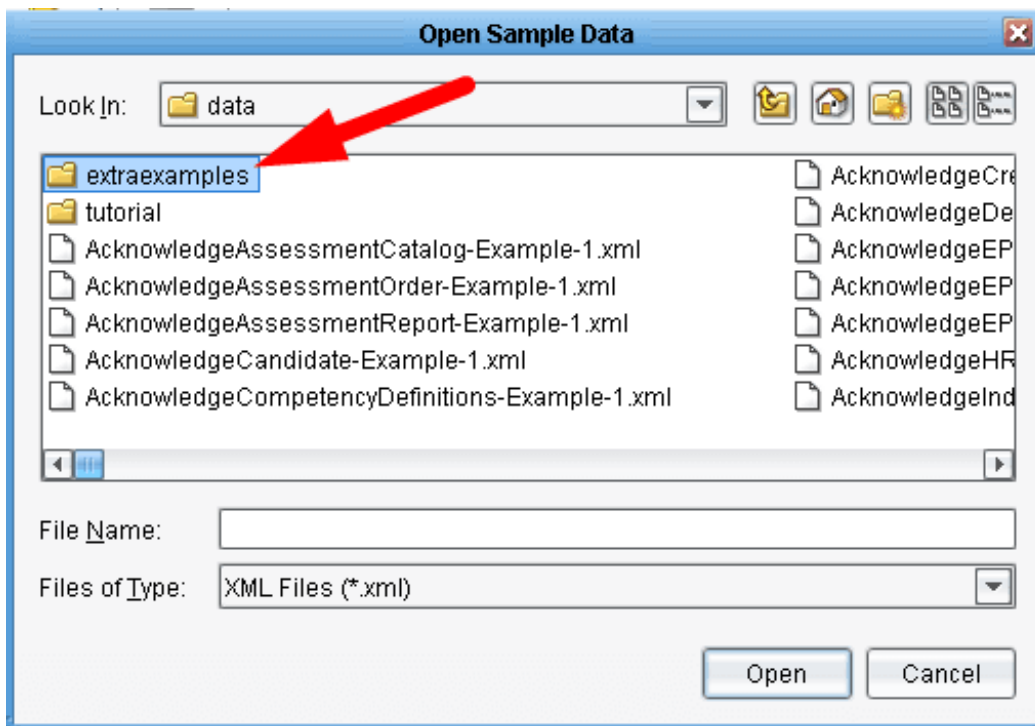


The next thing you'll want to do is load in a sample of the **ProcessScreeningOrder** template that you want to create. Instead of using the lightweight template that's there, you'll want to build this from an HR-XML provided sample file. To do this, click the **Use Sample Data** Icon above the TargetFormat tree.

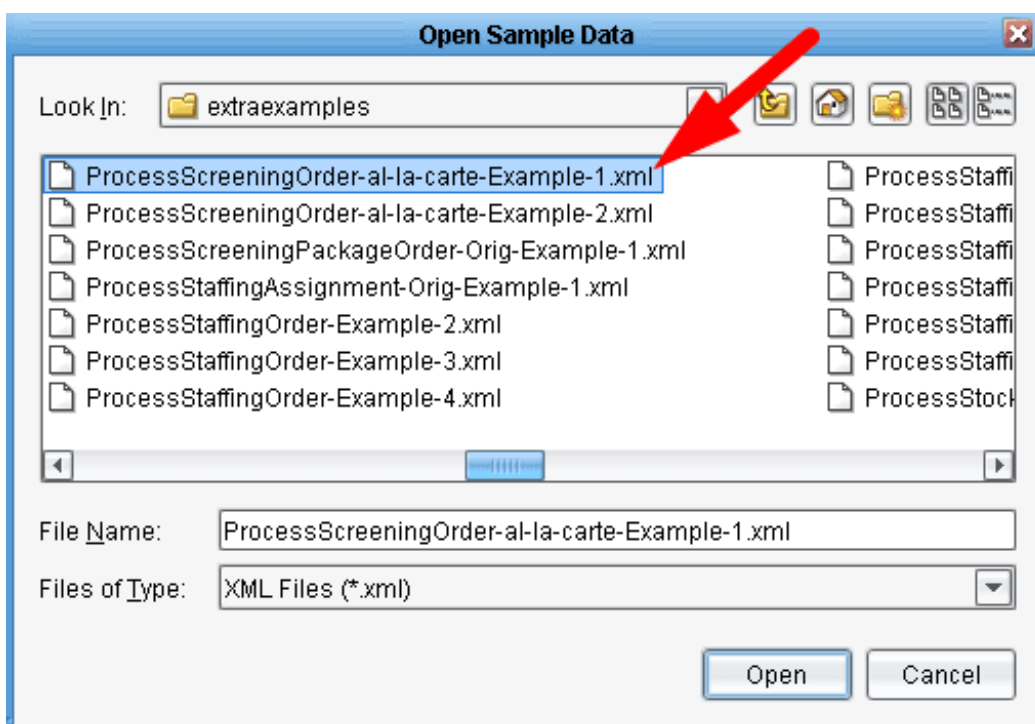


This will launch the Target format sample data dialogue. Click the **folder** icon in the upper left, which will allow you to load sample data from a file.

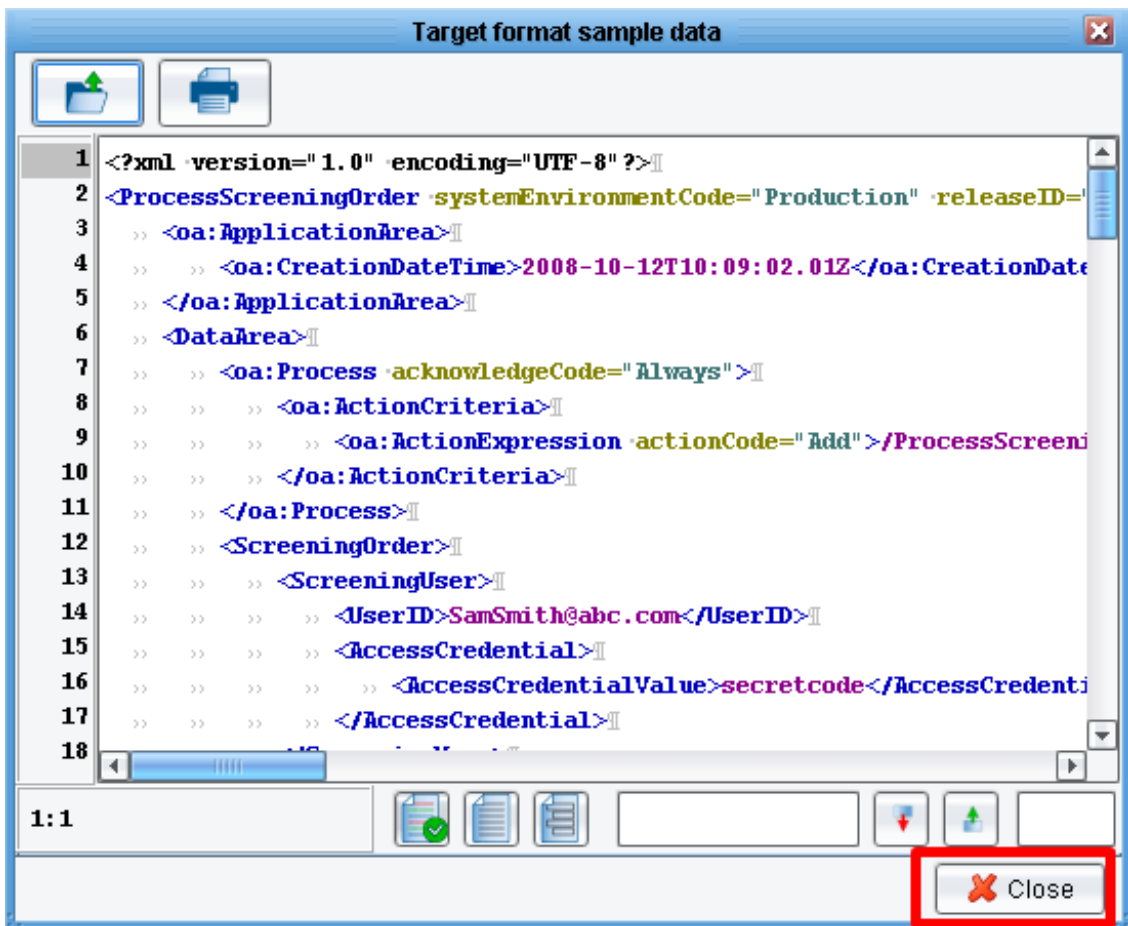




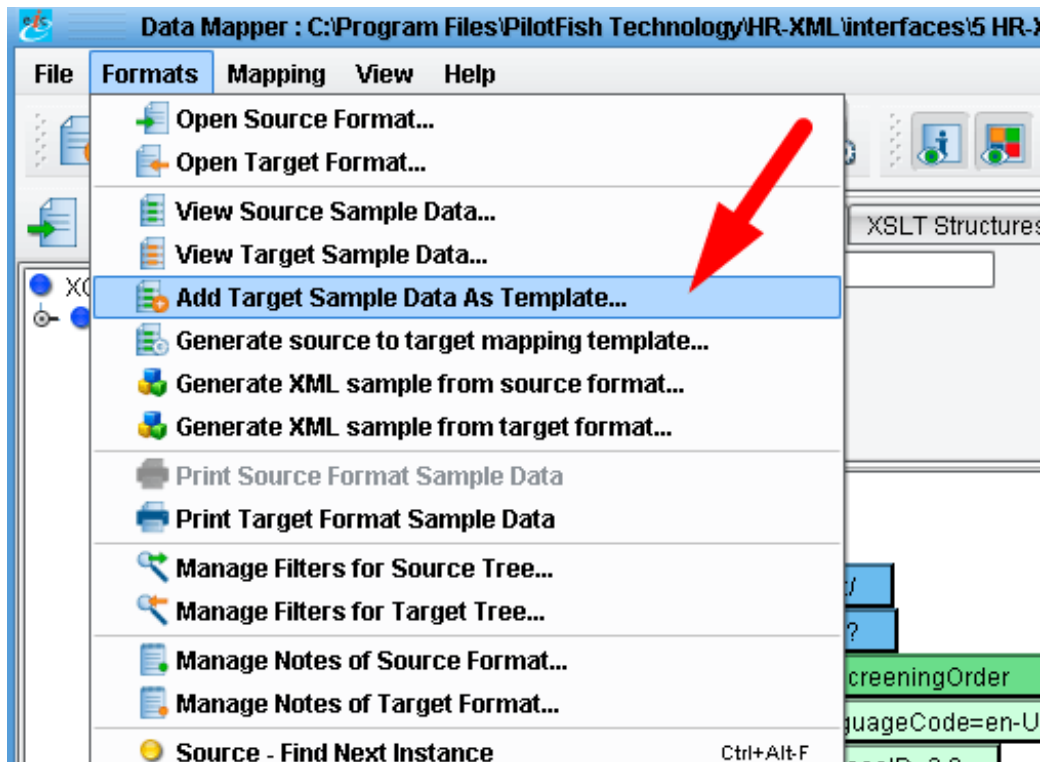
Navigate to the **extraexamples** folder of your distribution and double click it. It can be at **"/interfaces/3 HR-XML Templates/data/extraexamples/"**.



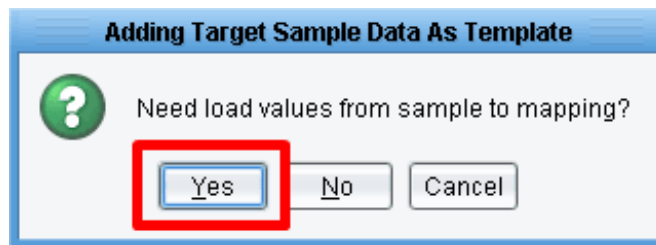
For this tutorial you'll use the **ProcessScreeningOrder-al-la-carte-Example-1.xml**. Select this file and click **Open**.



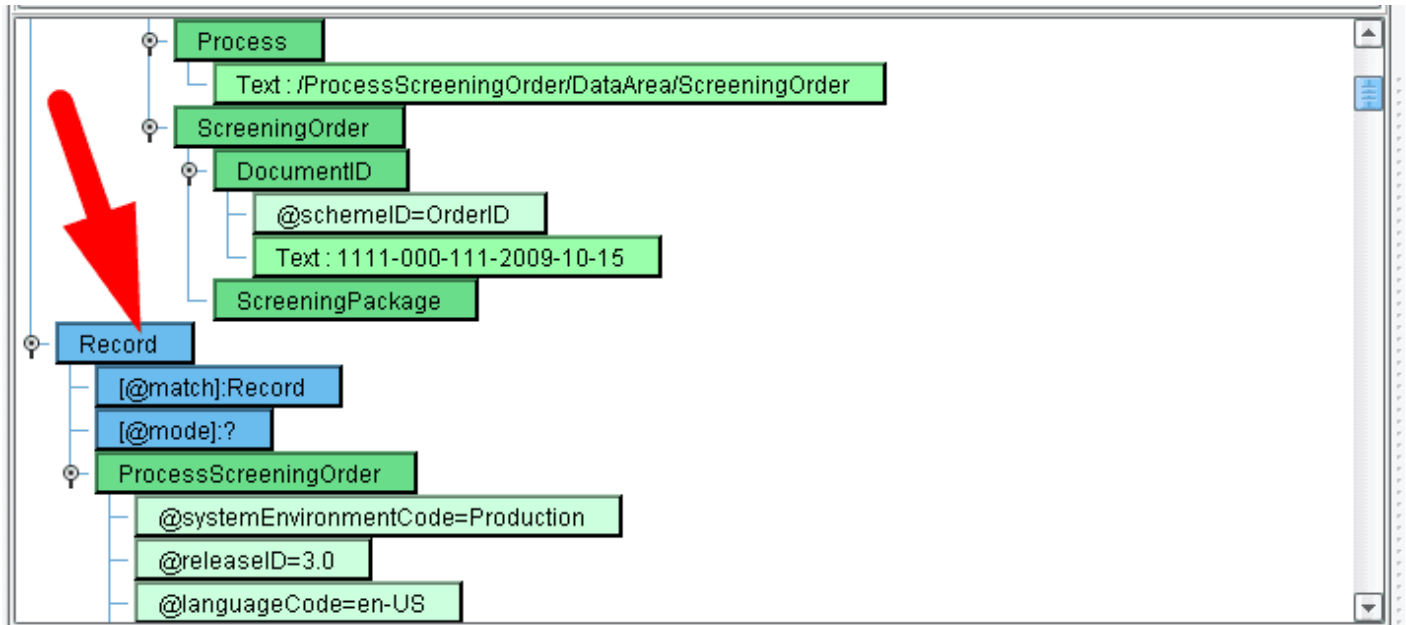
The contents of the file will appear in the Target format sample data area. Next click the **Close** button.



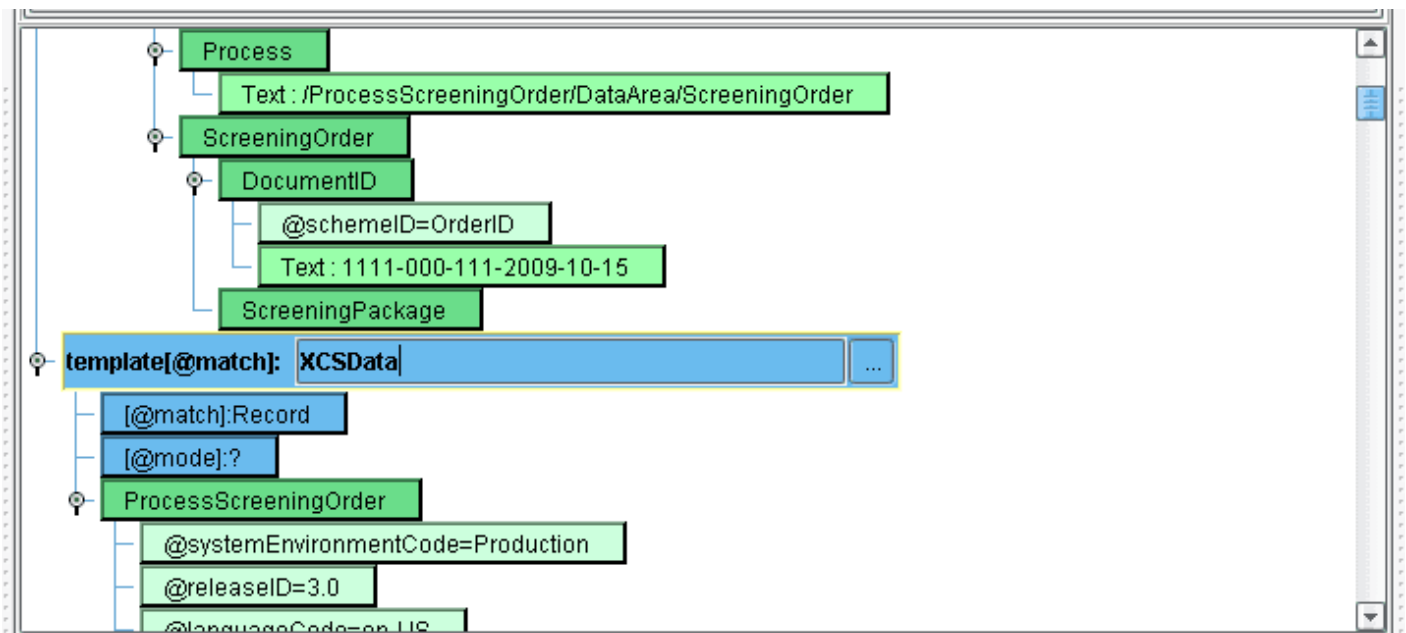
You'll use this template to provide information about the fields that you'll map to. To do this, in the Formats menu select **Add Target Sample Data As Template**.



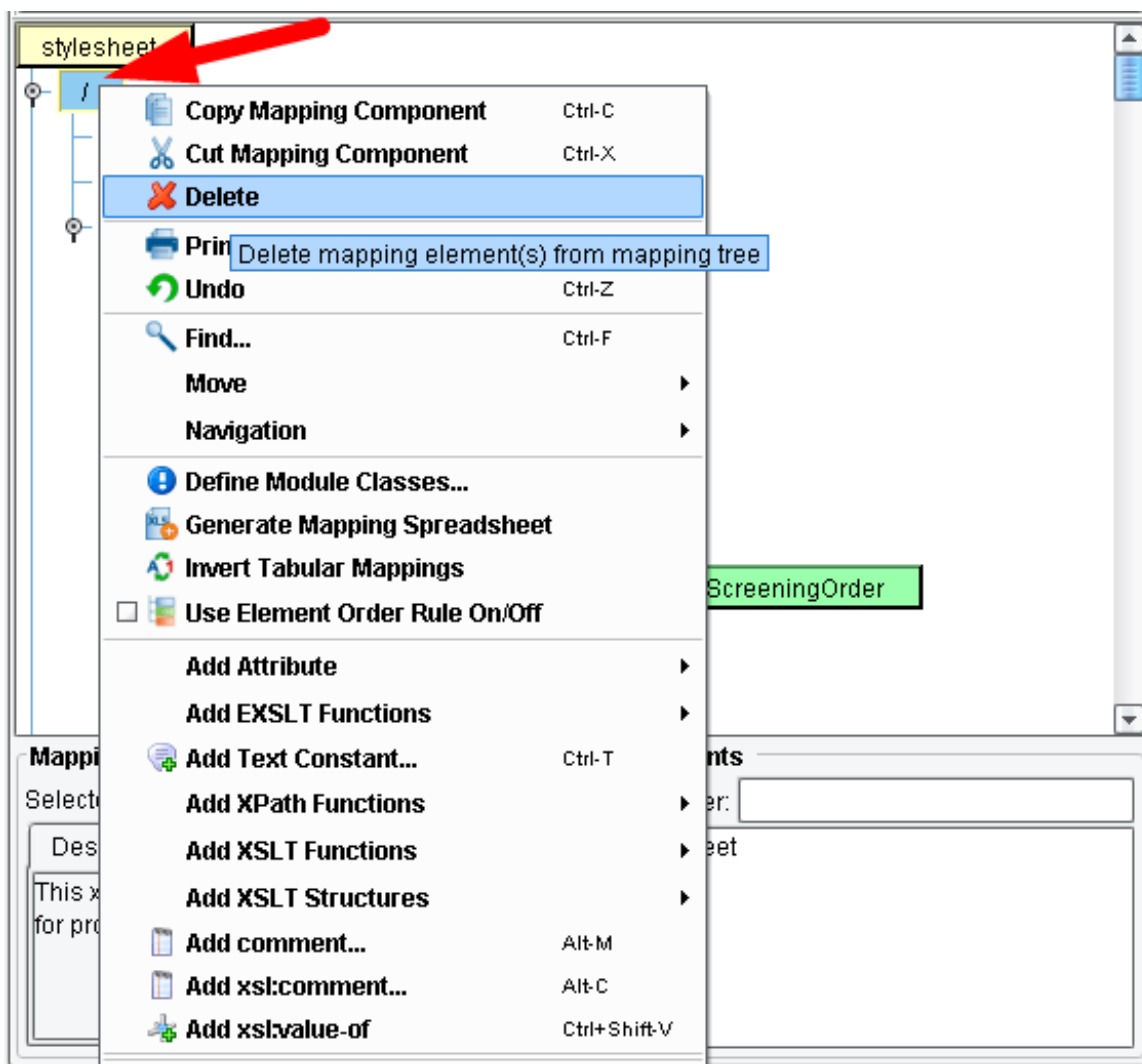
When prompted to load values from the sample to the mapping click **Yes**.



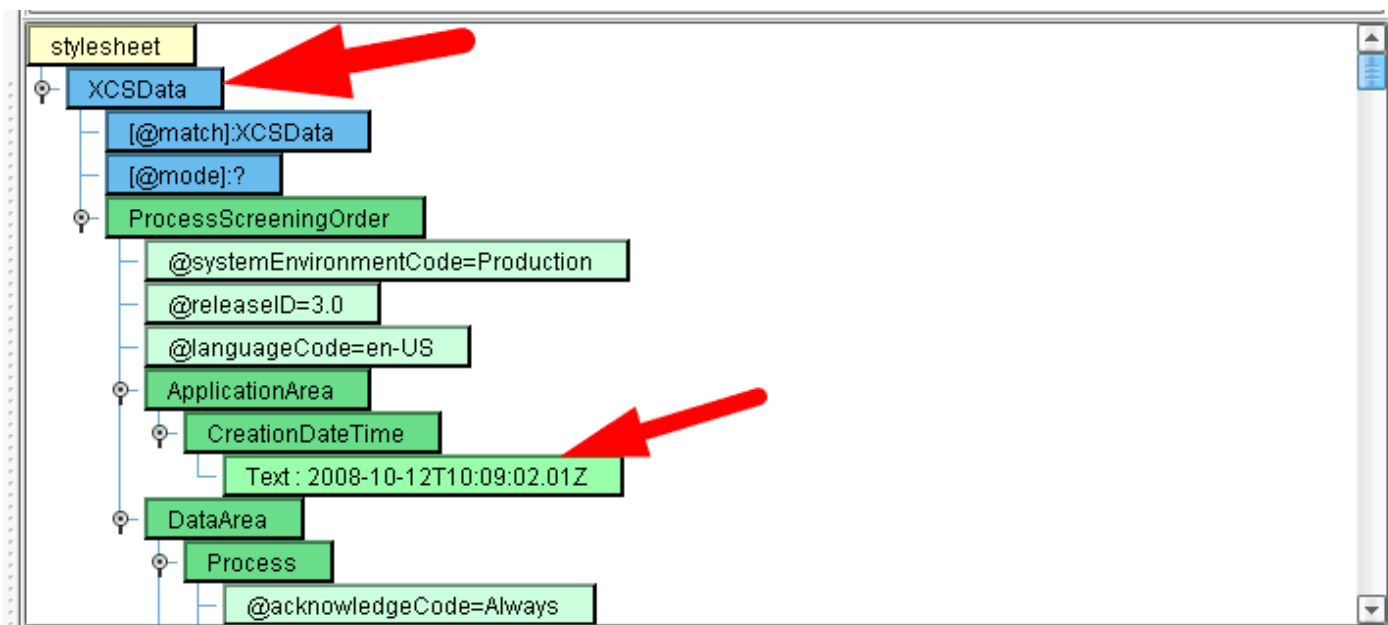
A series of new green nodes will appear in the center panel. Scrolling to the top of that, you'll see a blue node indicating Record.



Now change Record to **XCSData**. Input it and hit **Enter**. We've got the template you'll wish to map to.

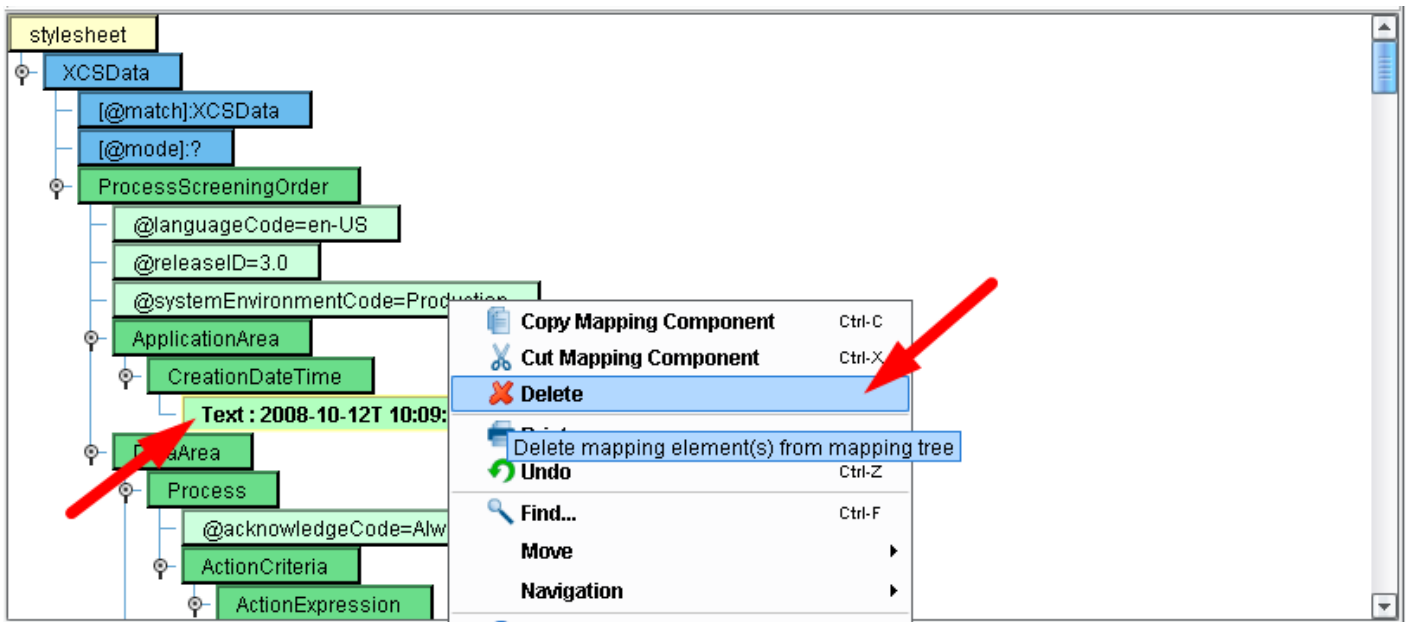


In order to delete the previous template, select "/" (the blue node) at the very top of the center panel. Right click and choose **Delete**.

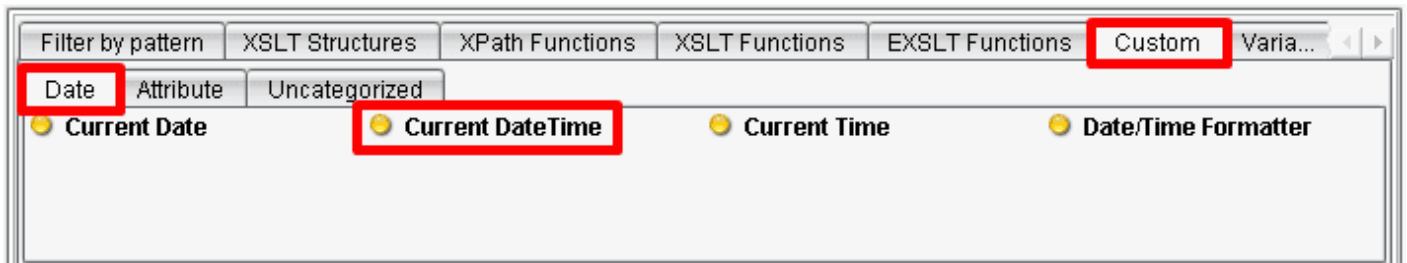


XCSDData will now appear as the only blue node in the mapping. Your next step is to work your way through the center panel, mapping fields from the left into the center as appropriate.

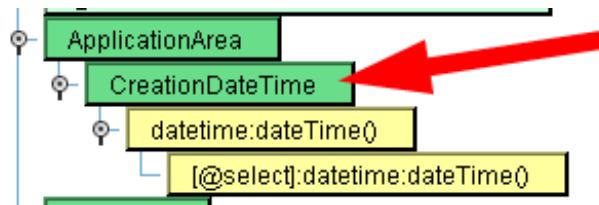
Let's begin by populating the Creation Date and Time. Select the **CreationDateTime** green node underneath **ProcessScreeningOrder** in the **ApplicationArea**. You'll see that it's currently populated with some hard coded text that was pulled from the sample file. You'll want to replace this with the current date and time.



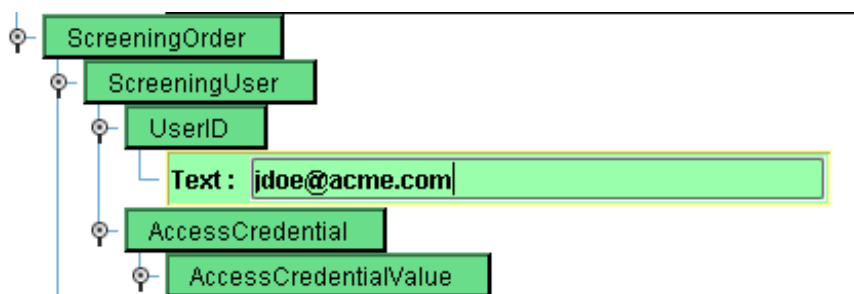
To do this, left click on the text node. Then right click and choose **Delete**.



Now, from the pallet above the main mapping area, select the **Custom** tab and the **Date** sub-tab. Click on the **Current DateTime** node and drag that on top of the **CreationDateTime** node in your main grid.

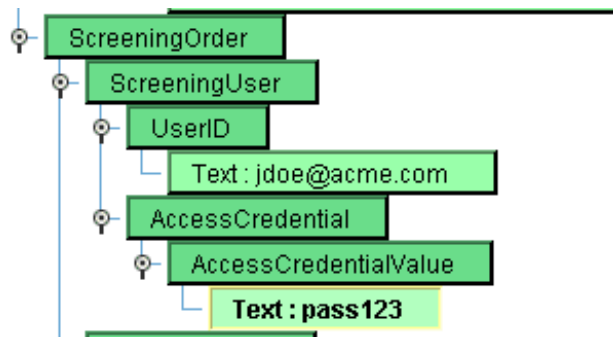


When the **CreationDateTime** node is highlighted with a yellow border, release the mouse and a new yellow node will be created which will populate the **CreationDateTime** element with the current Date and Time.

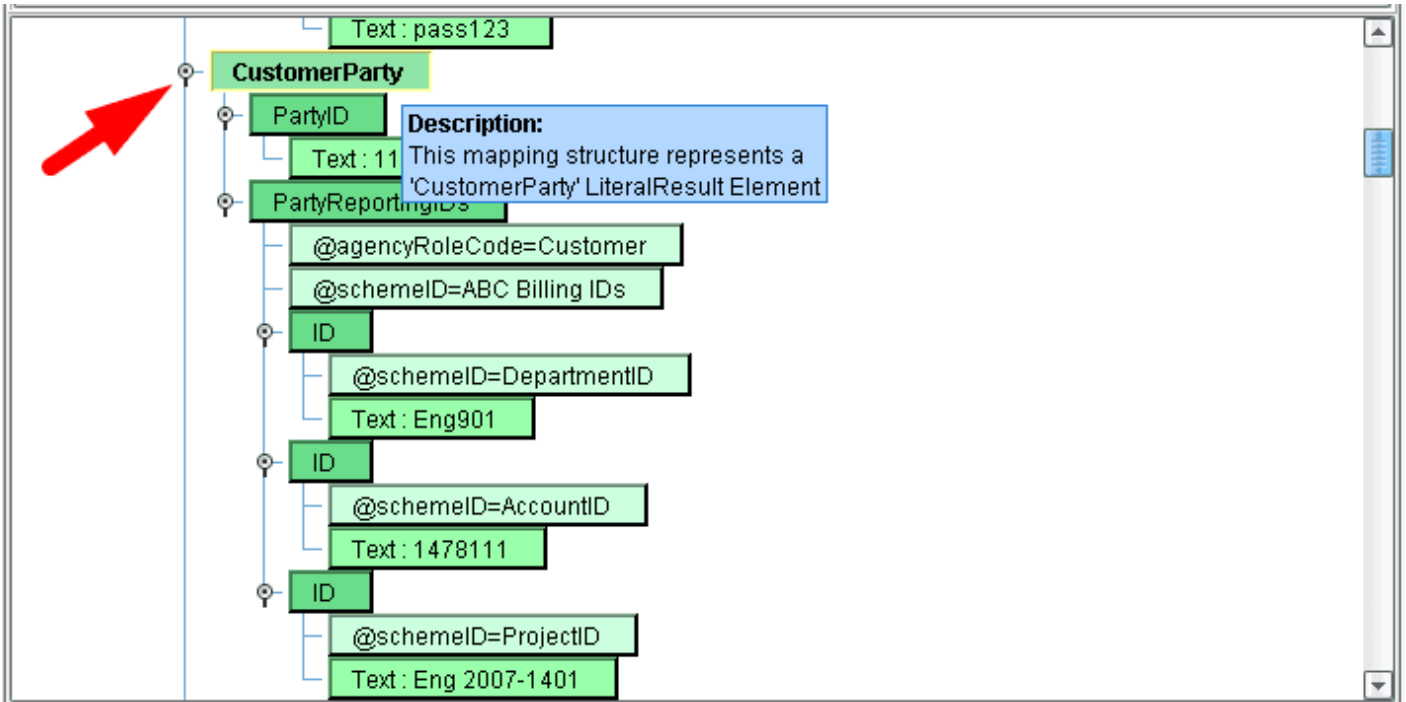


Scrolling further through the mapping you'll leave the **DataArea** – Process alone and move into the **ScreeningOrder**. You'll see there's a **ScreeningUser** section with a **UserID** and an **AccessCredentialValue** hard coded. Let's assume you want to keep this hard coded, but change it to populate it with your own value. To do this, double click first on the **Text** node underneath **UserID**. The text will become editable. Enter in "jdoe@acme.com". Enter the text and then hit the **Enter** key.

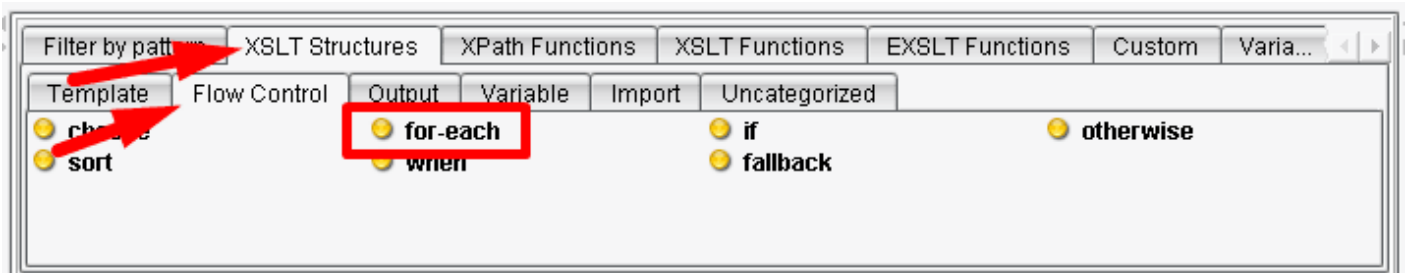
Note: it's important when using the Console to hit Enter after editing the nodes value. Hitting Esc or simply clicking off of the node will reset the node to its original value.



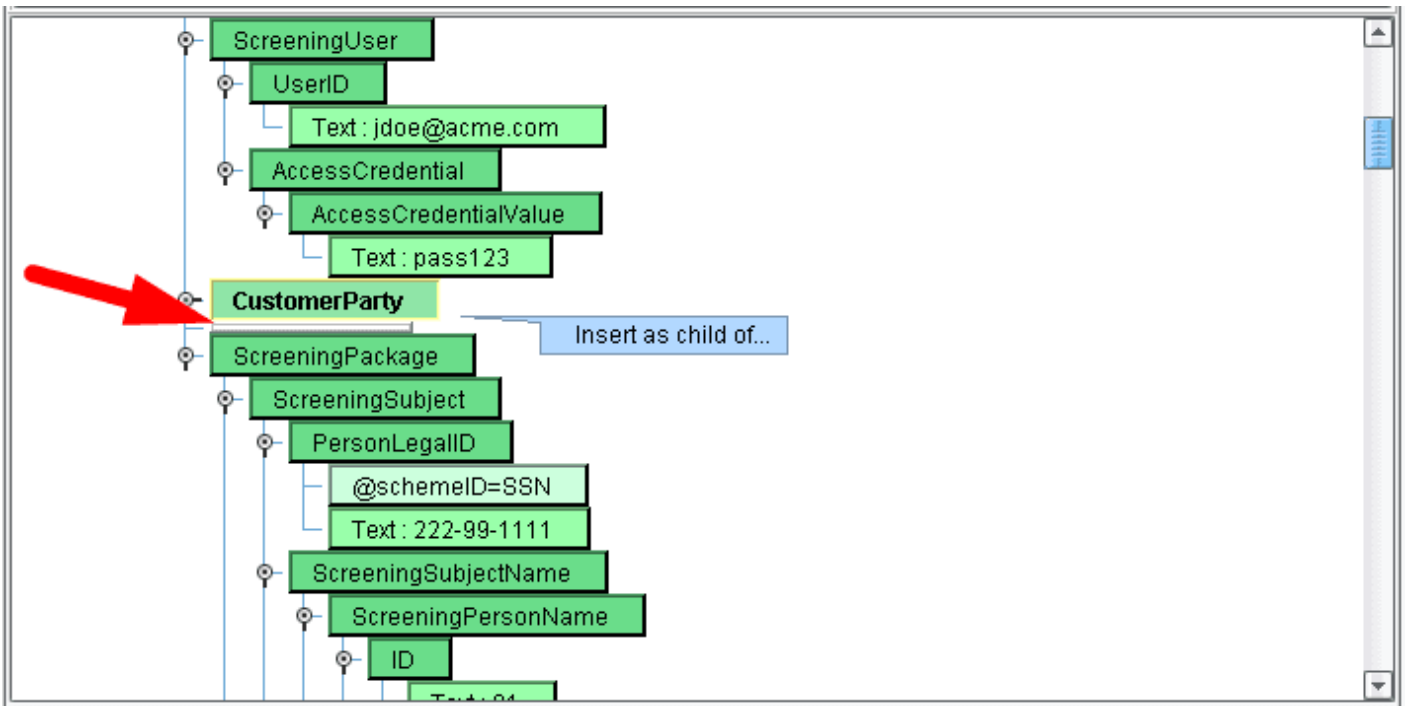
Repeat this process with the **AccessCredentialValue**. Changing secretcode to the simple text "**pass123**" and click **Enter**.



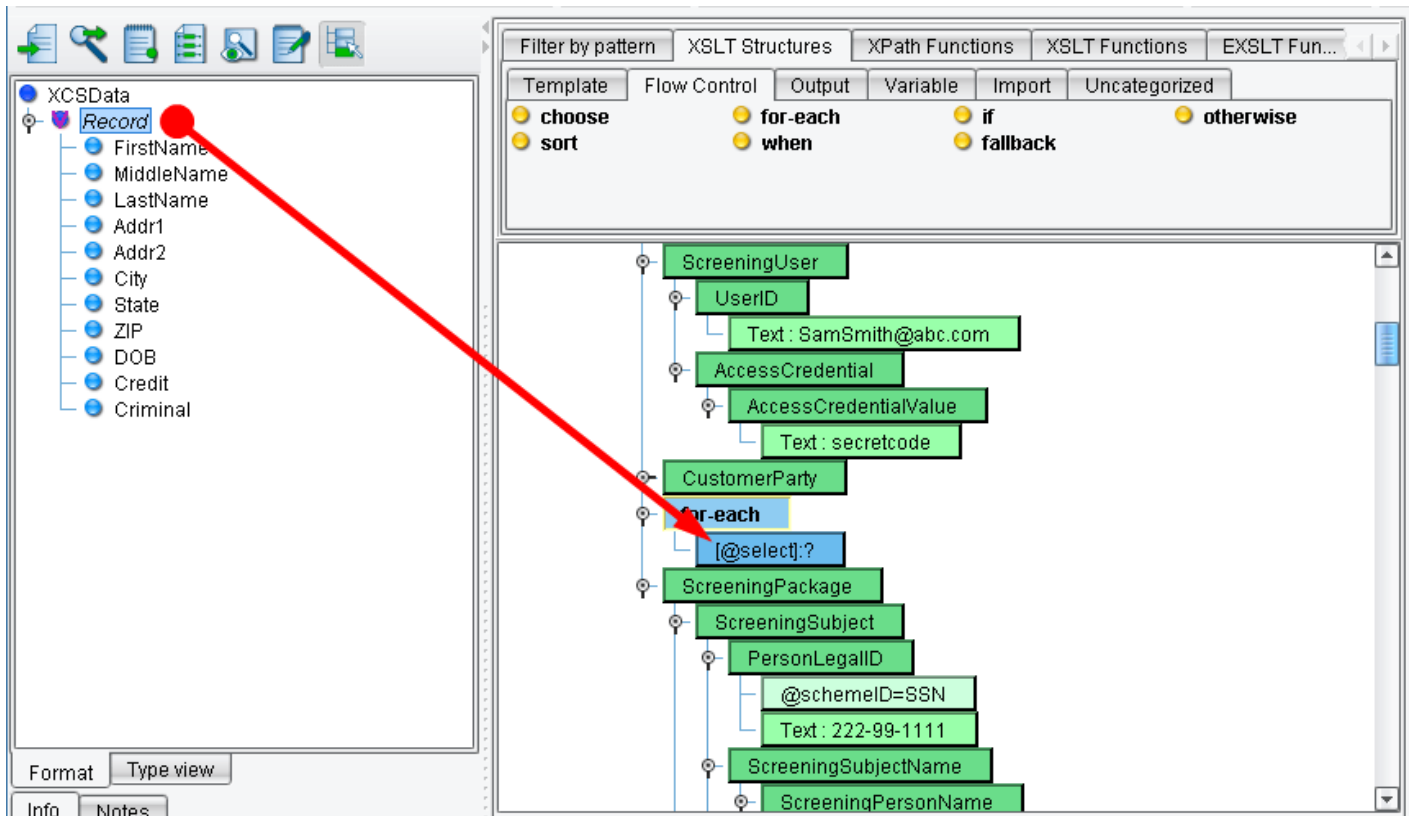
For the purposes of this demonstration, leave the **CustomerParty** section alone. To collapse this node click the collapse icon next to the **CustomerParty**. This will roll up all of the children underneath **CustomerParty** to hide them.



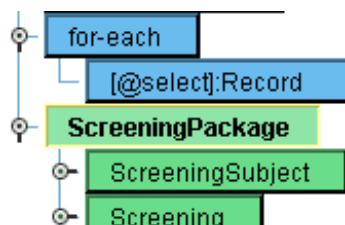
Next, move into the **ScreeningPackage** area. The **ScreeningPackage** has two child nodes, **ScreeningSubject** and then a set of **Screening** nodes. Because you may have multiple records in a file representing multiple candidates, let's assume there could be multiple **ScreeningPackages** within this message. You'll want to create one of them for each record that you encounter. To do this, underneath the **XSLT Structures** tab, select the **Flow Control** sub-tab. Then click the **for-each** node...



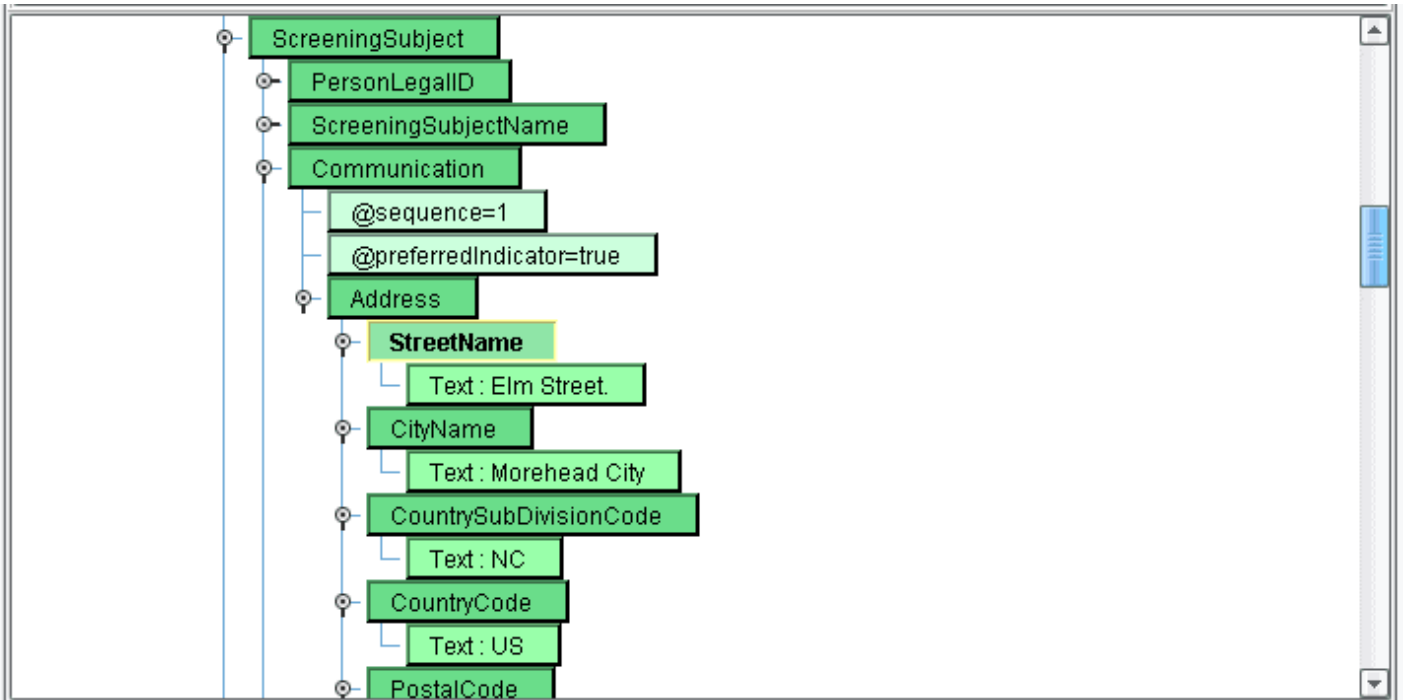
...and drag it carefully above the **ScreeningPackage**. You'll want the gray line to appear between **CustomerParty** and **ScreeningPackage**. **ScreeningPackage** should not be highlighted.



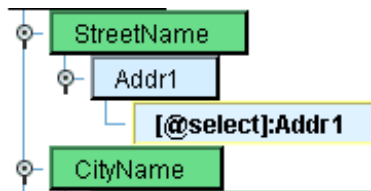
When you release the mouse a blue **for-each** node should appear. You'll want to create a **ScreeningPackage** for each record that you encounter. To choose what you want to iterate over, choose the **Record** node underneath **XCSDData** from the Source tree and drag it on top the **for-each**. When the **for-each** is highlighted with the yellow border release the mouse. You'll now see that anything you create underneath the **for-each** will occur each time you encounter a Record in the Source.



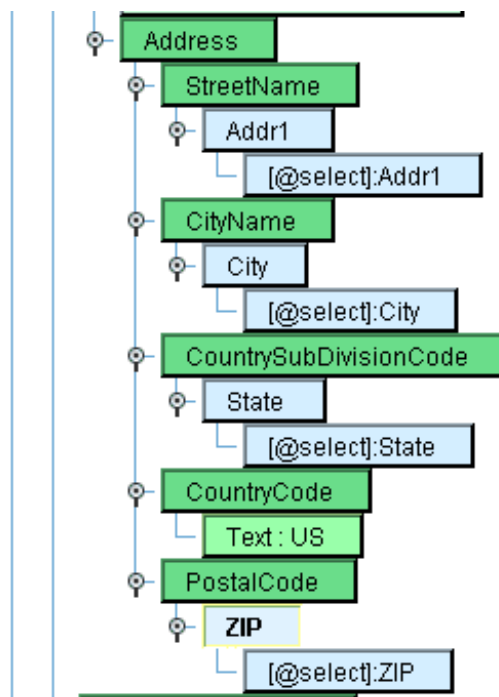
You'd like to create a **ScreeningPackage** in this scenario. To do this you'll make **ScreeningPackage** a child of the **for-each** node. To accomplish this, click on the **ScreeningPackage** node and drag in on top of the **for-each**. When you release the mouse, **ScreeningPackage** will now be a child of the **Record**.



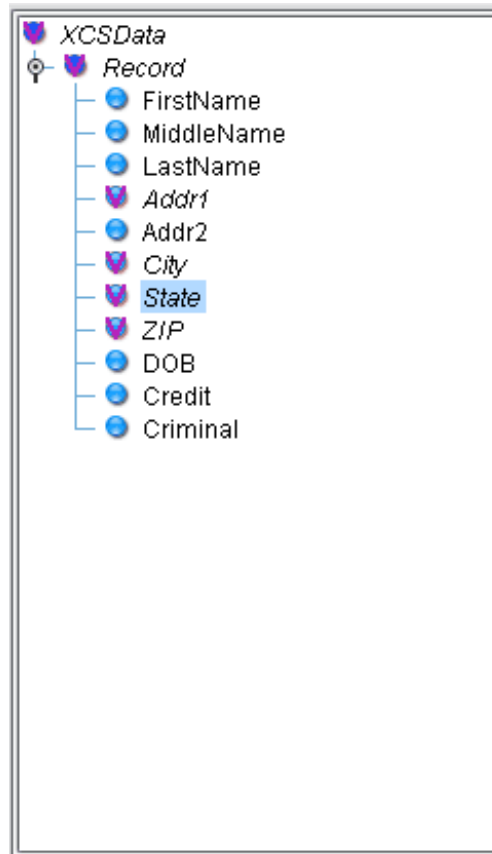
Now, you'll work on the **ScreeningSubject**. Let's map the address area. You'll see that you have a set of hard coded values for the **CountryCode**, **StreetName**, **CityName**, **CountrySubDivisionCode**, and **PostalCode**.



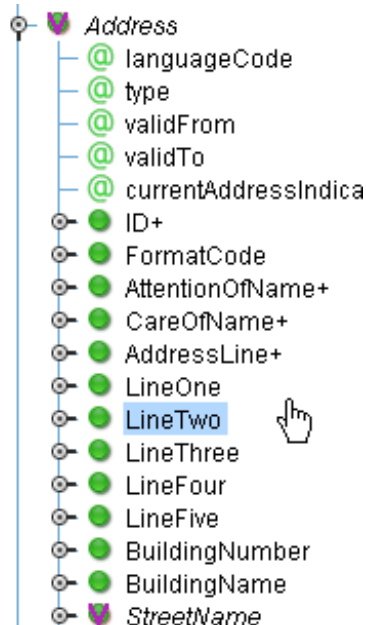
Examining your Source, you'll see for address information you have fields called **Addr1**, **Addr2**, **City**, **State**, and **Zip**. Assume this is a US based candidate, so keep the default **CountryCode** as **US**. However, delete the **StreetName** text **Elm Street** and replace that with **Addr1** from the sample file. Click on the **Text** node underneath **StreetName** and right click for the **Delete** key. Then drag **Addr1** under the **Record** node on top of **StreetName**. You've now created a one-to-one mapping between the **Addr1** node from the Source and the **StreetName** element in your Result.



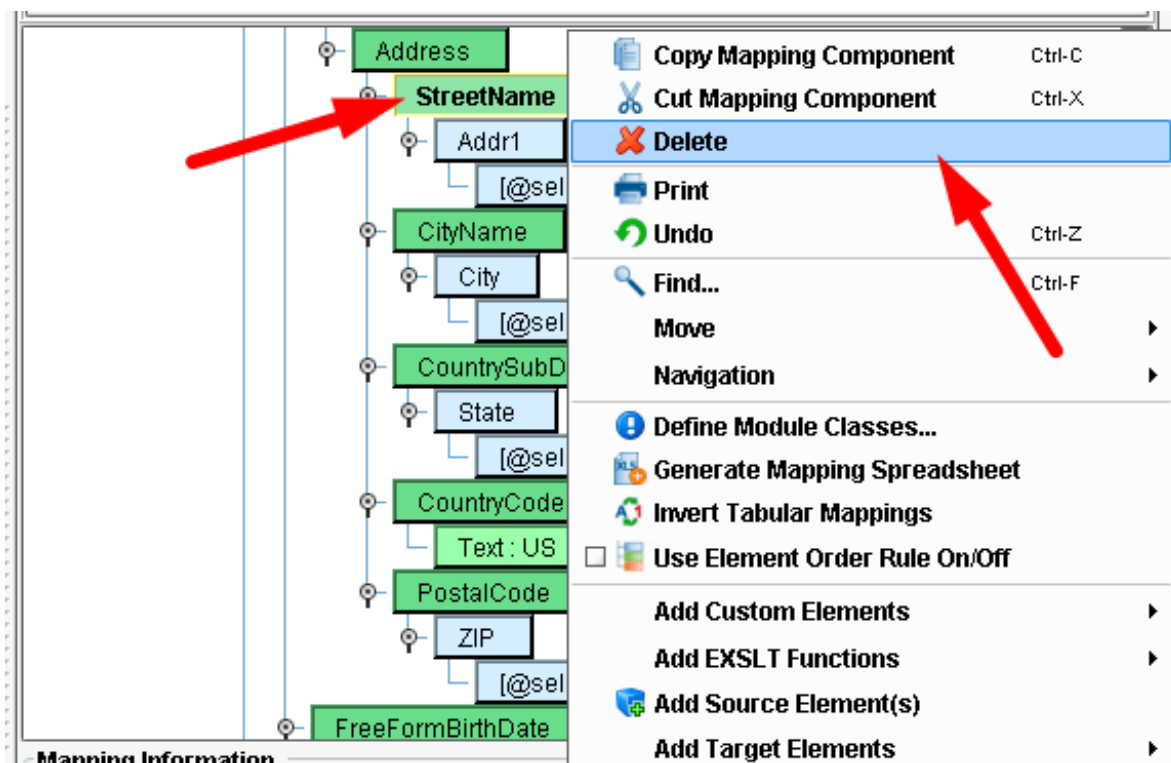
Repeat this process with **CityName** and **CountrySubDivisionCode**. Lastly, do the same thing with **PostalCode**.



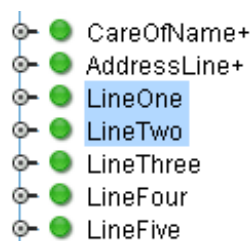
You'll note that **Addr2** doesn't have a check mark. It hasn't yet been mapped. So you'll want to find an additional element legally in the HR-XML specification that you can add to the address.



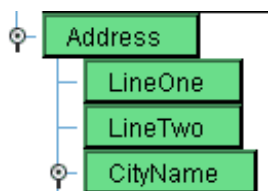
To do this, navigate to the **Address** area in the tree (**Format/ ProcessScreeningOrder/ DataArea/ ScreeningOrder+/ ScreeningPackage+/ ScreeningSubject+/ Communication+/ Address/**) on the right hand side. Once you've done that you'll see a list of all of the allowable children with those that have been used checked off. You'll note that **StreetName** has been mapped; however, **LineOne** and **LineTwo** would be appropriate for mapping **Addr1** and **Addr2**.



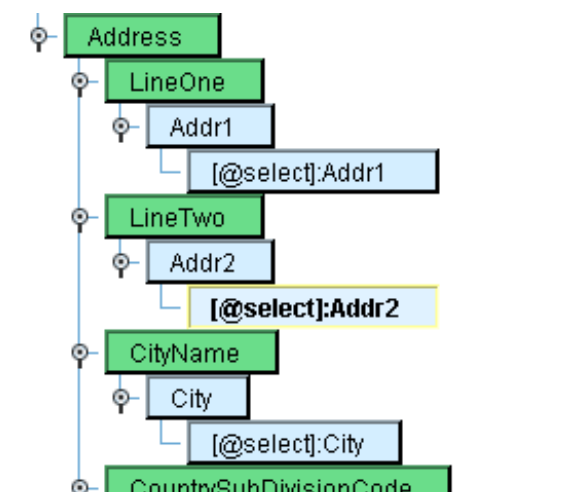
Let's delete the **StreetName** mapping. Click on the **StreetName** node. Right click and choose **Delete**.



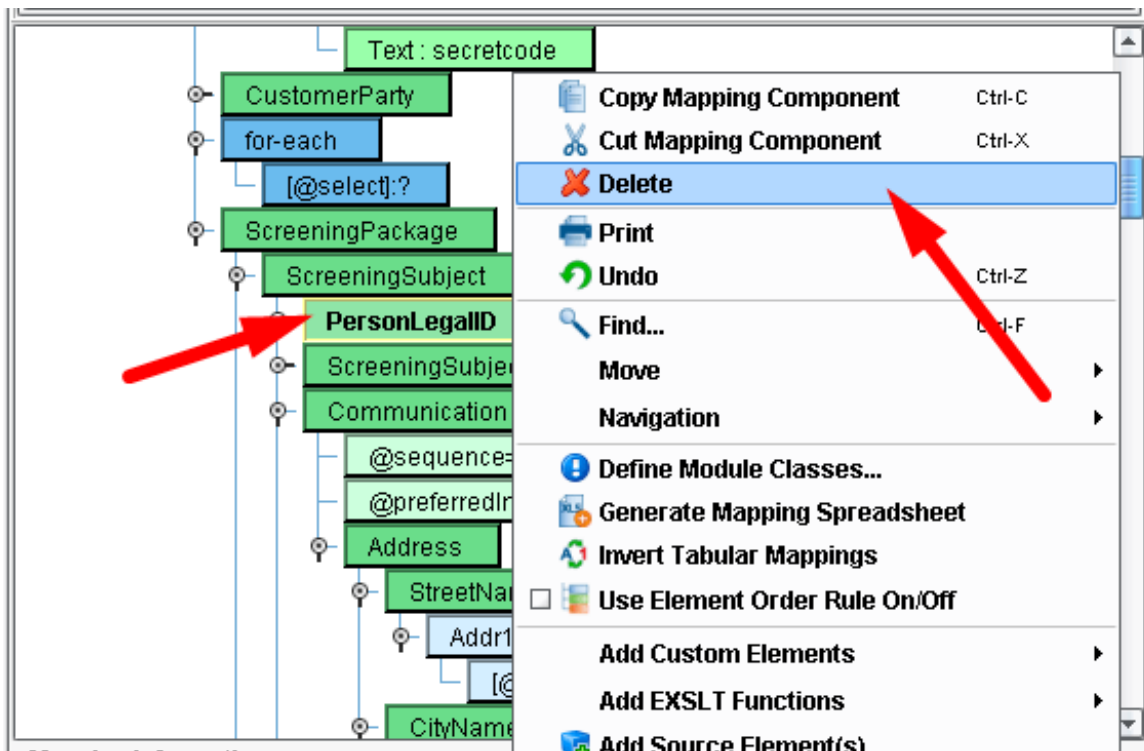
Let's replace that with **LineOne** and **LineTwo**. Multi-select **LineOne** and **LineTwo** by selecting each while holding the Control key.



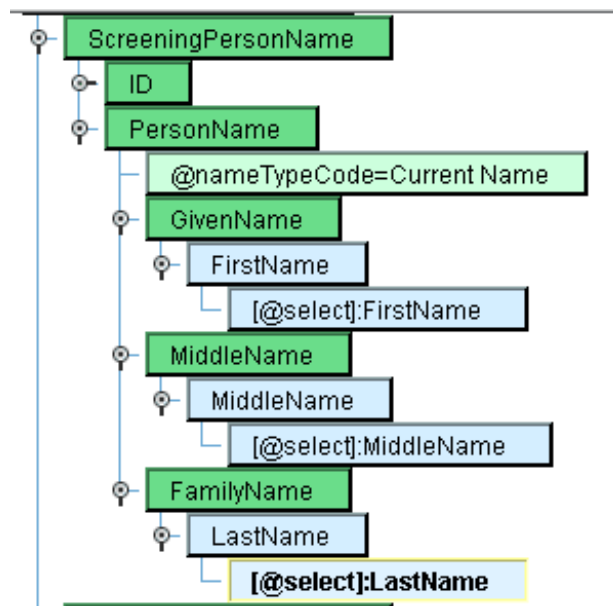
Drag these nodes on top of their parent the **Address** node. You'll see that **LineOne** and **LineTwo** green nodes now appear in the correct element order underneath **Address**.



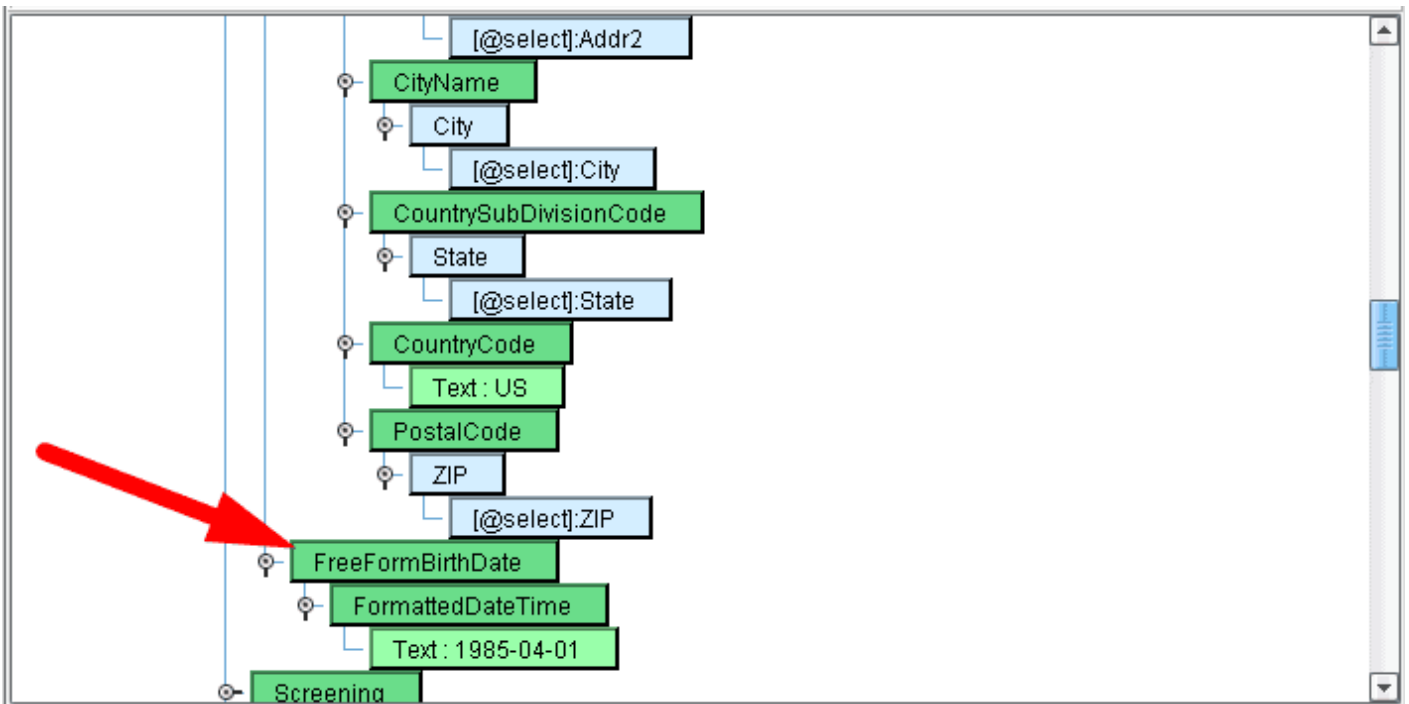
Now, map **Addr1** and **Addr2** to those nodes respectively.



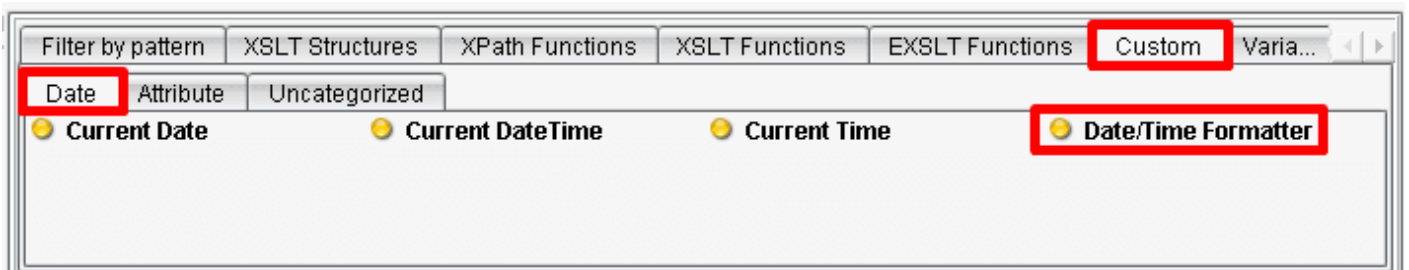
Having done that let's move on. You don't have social security number information, so you're going to remove the **PersonLegalID**. Select that node, right click and hit **Delete**.



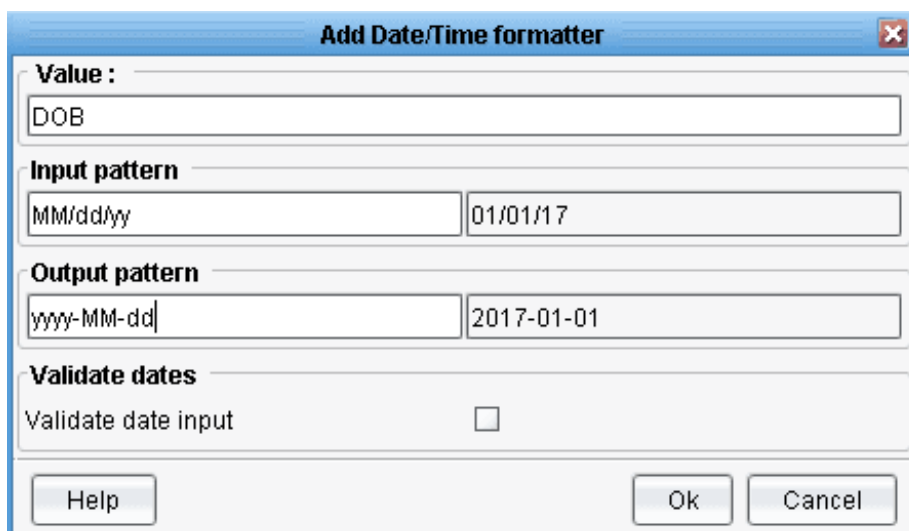
In **ScreeningSubjectName**, you'll only include the first **ScreeningPersonName**. Delete all other child nodes. Using the same process as above you'll map **FirstName** to **GivenName**, **MiddleName** to **MiddleName**, and **LastName** to **FamilyName**. Delete each of the existing text nodes and then replace them by dragging and dropping the corresponding fields from the Source.



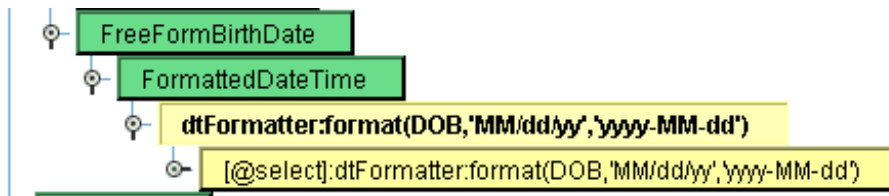
Save your work. Click the **Save** disk icon in the upper left. This will save your XSLT and allow you to continue. Now work on the birth date. Under the **FreeFormBirthDate** node you'll find the **FormattedDateTime** node containing a hard coded birth date. Delete this as you've done several times before and drag the DOB element onto the **FormattedDateTime** node.



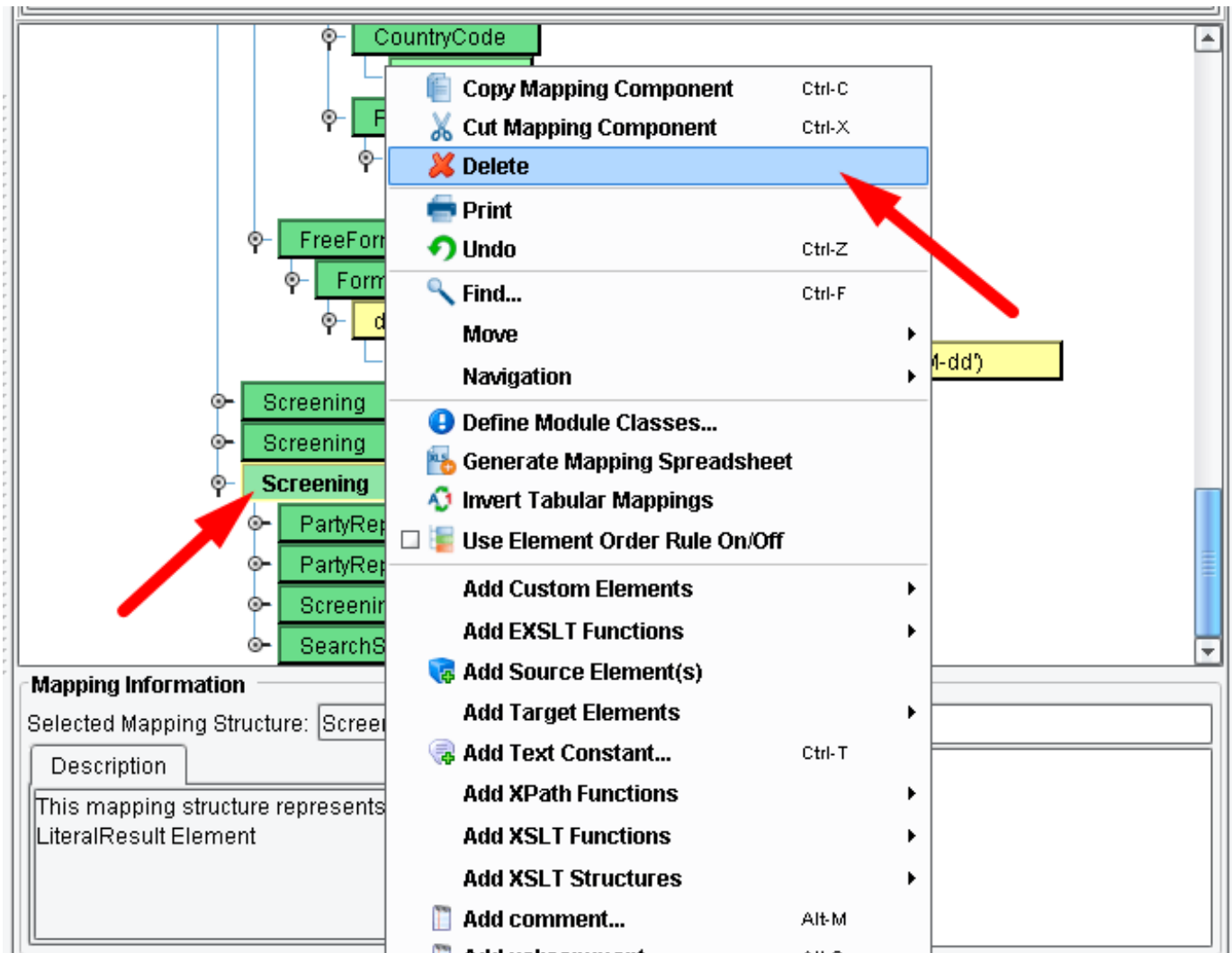
However, thinking back to your sample, the CSV had data in **MM/dd/yy** format. Here, you'll want the format as **yyyy-MM-dd**. To handle such formatting you can use the Date/Time Formatter. Under the **Custom** tab in the pallet and the **Date** sub-tab, find the **Date/Time Formatter** tool and drag it onto the DOB node.



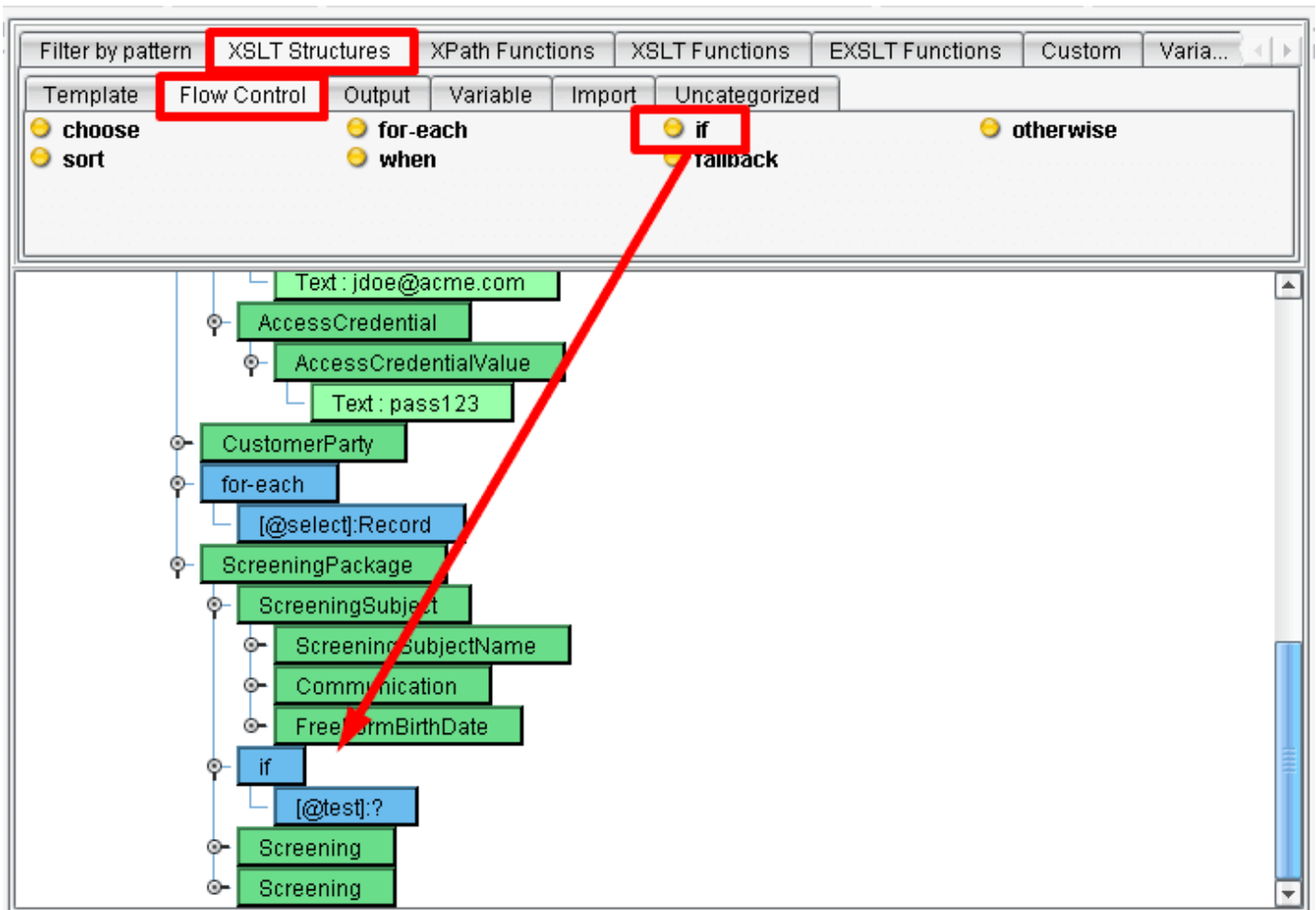
When the mouse is released the **Add Date/Time Formatter** dialogue will appear. Enter your desired Input pattern (**MM/dd/yy**) and replace that with your desired output pattern (**yyyy-MM-dd**). Samples of how this should appear are displayed to the right using the current day. Click **OK**.



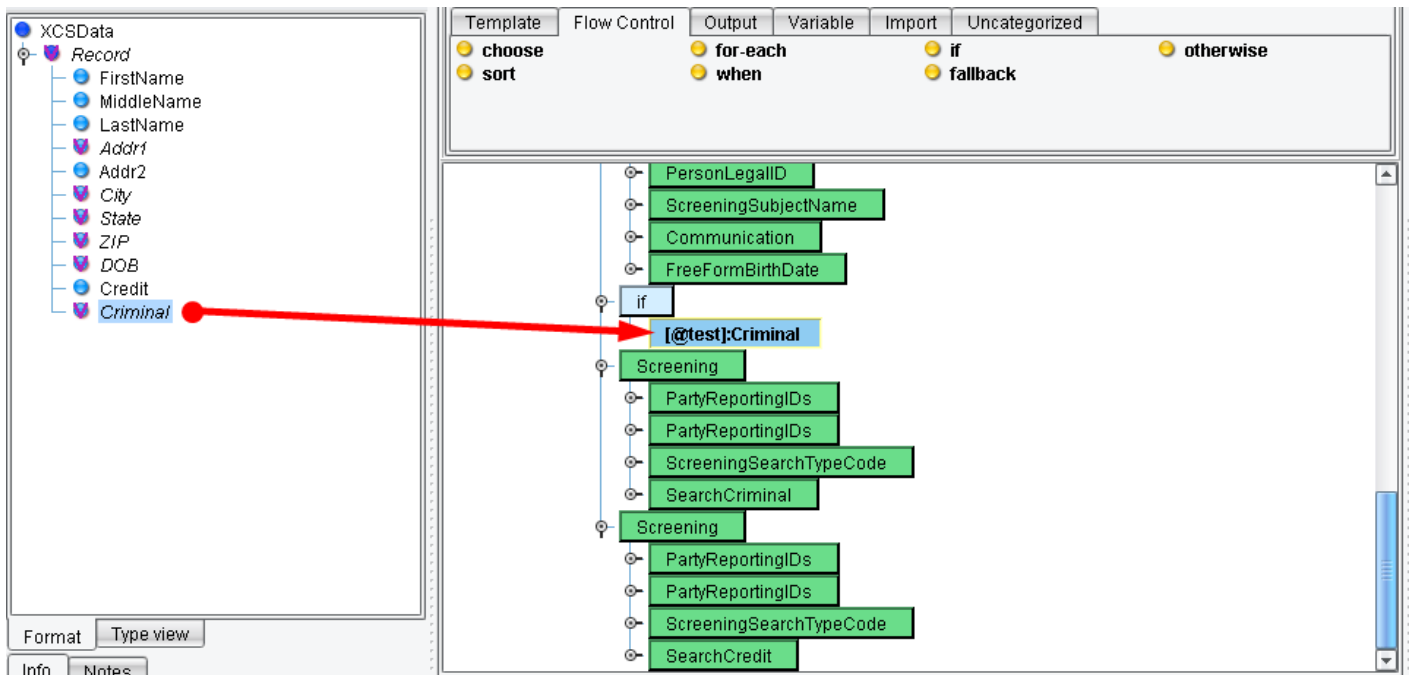
The node will now become yellow with all of the logic having been input into the mapping.



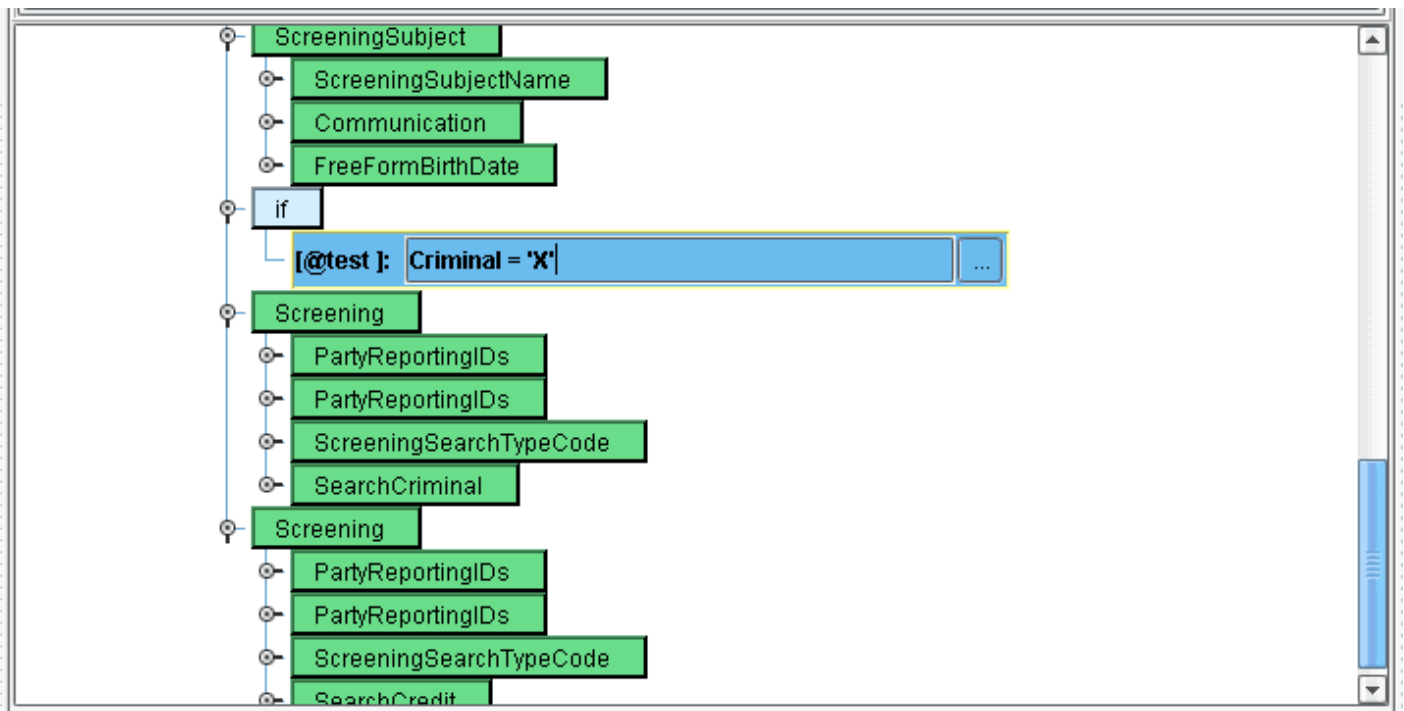
You now have the **Credit** and **Criminal** fields to work with. Remembering back to your sample file you'll see that the **Credit** and **Criminal** fields were either blank or contained an X. An X indicates this is a check that you'd like to perform. This is an example of where you'd want to create a conditional mapping based on the content of the fields. Examining the first two screening nodes you'll see that one represents a criminal check, the second represents a credit check. The third one you can delete. Do that now.



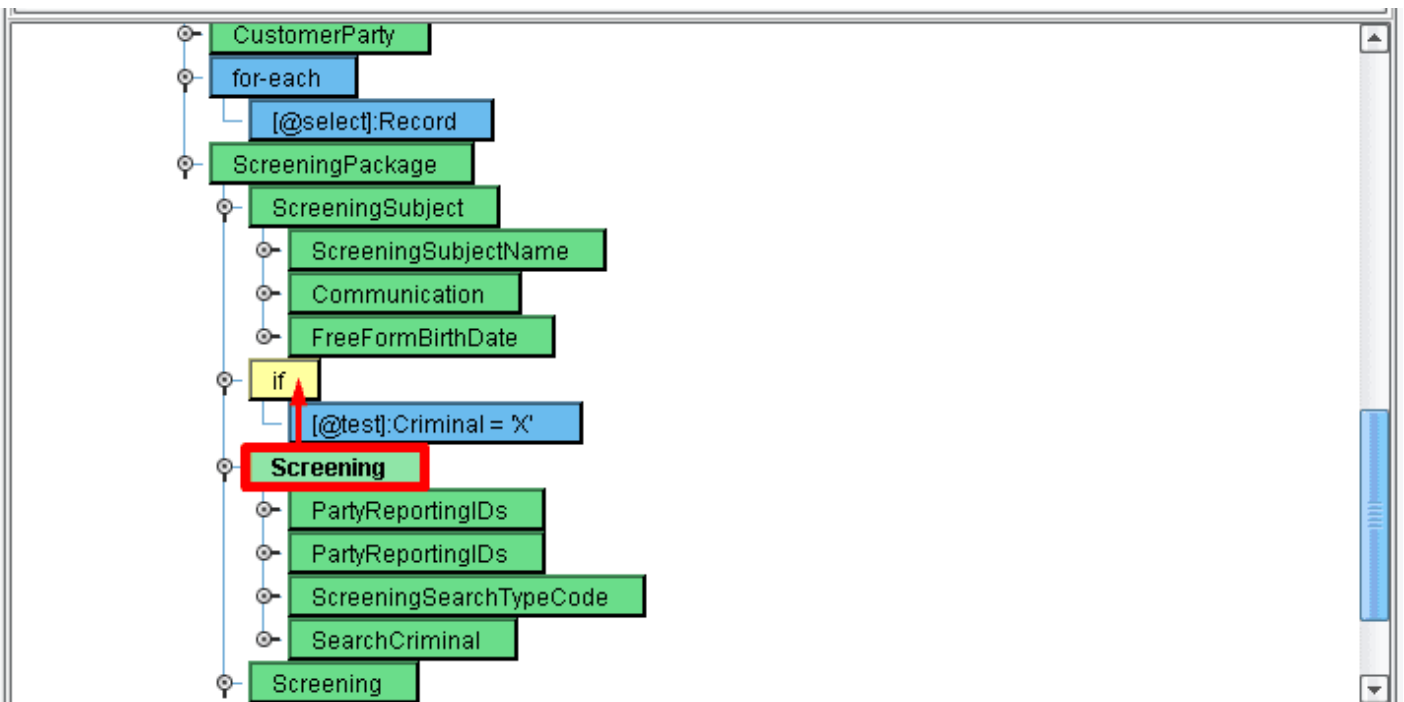
If the credit field has an X you'll want to create a structure very similar to the Screening representing the credit check. Similarly if criminal has an X you'll want to create the other screening structure. To implement this conditional logic click the **XSLT Structures** tab and choose the **Flow Control** sub-tab. Select the **if** tool. And drag it directly above the Screening.



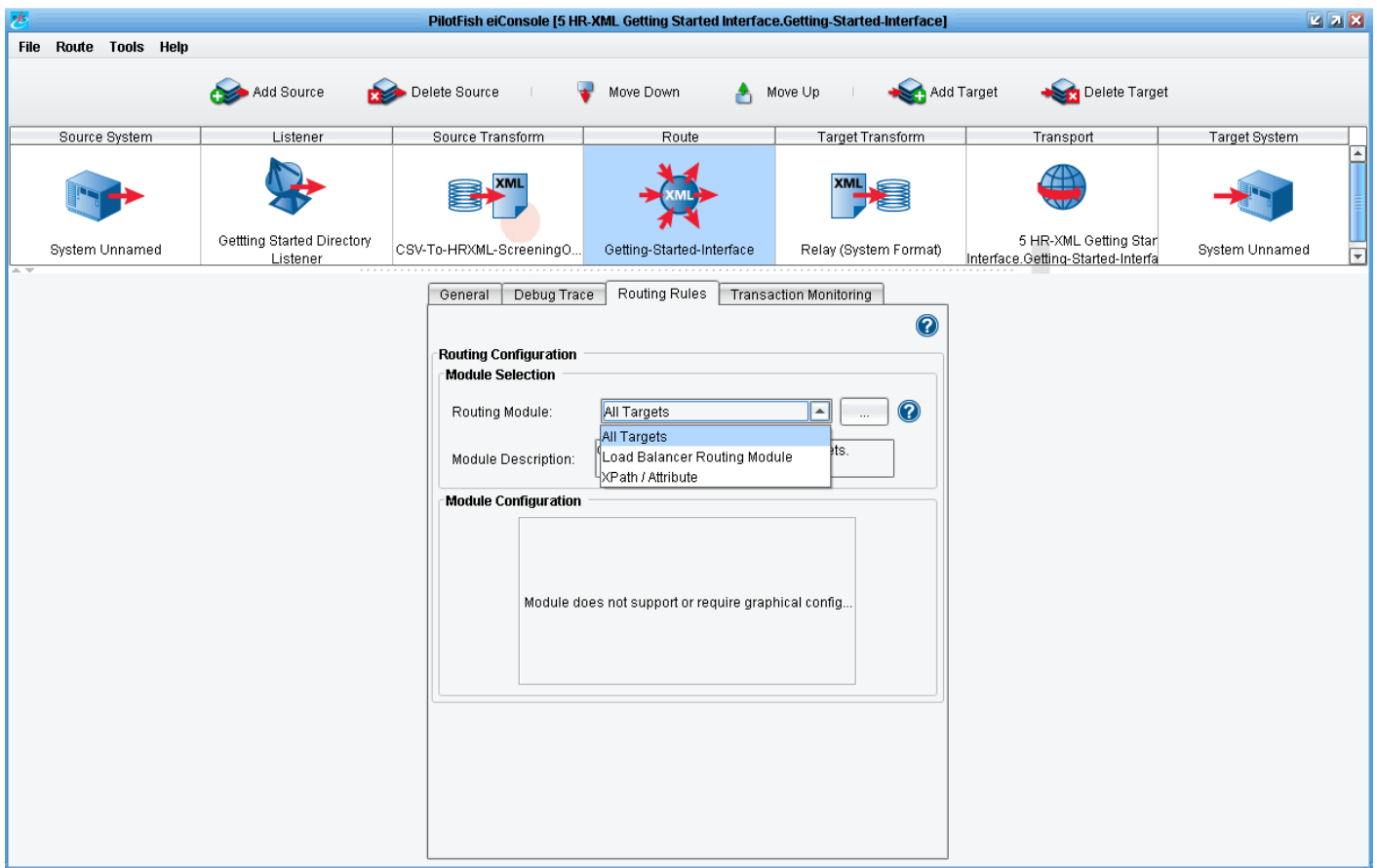
The first check that you'll do is for the criminal X. Select the **Criminal** node and drag it onto the test.



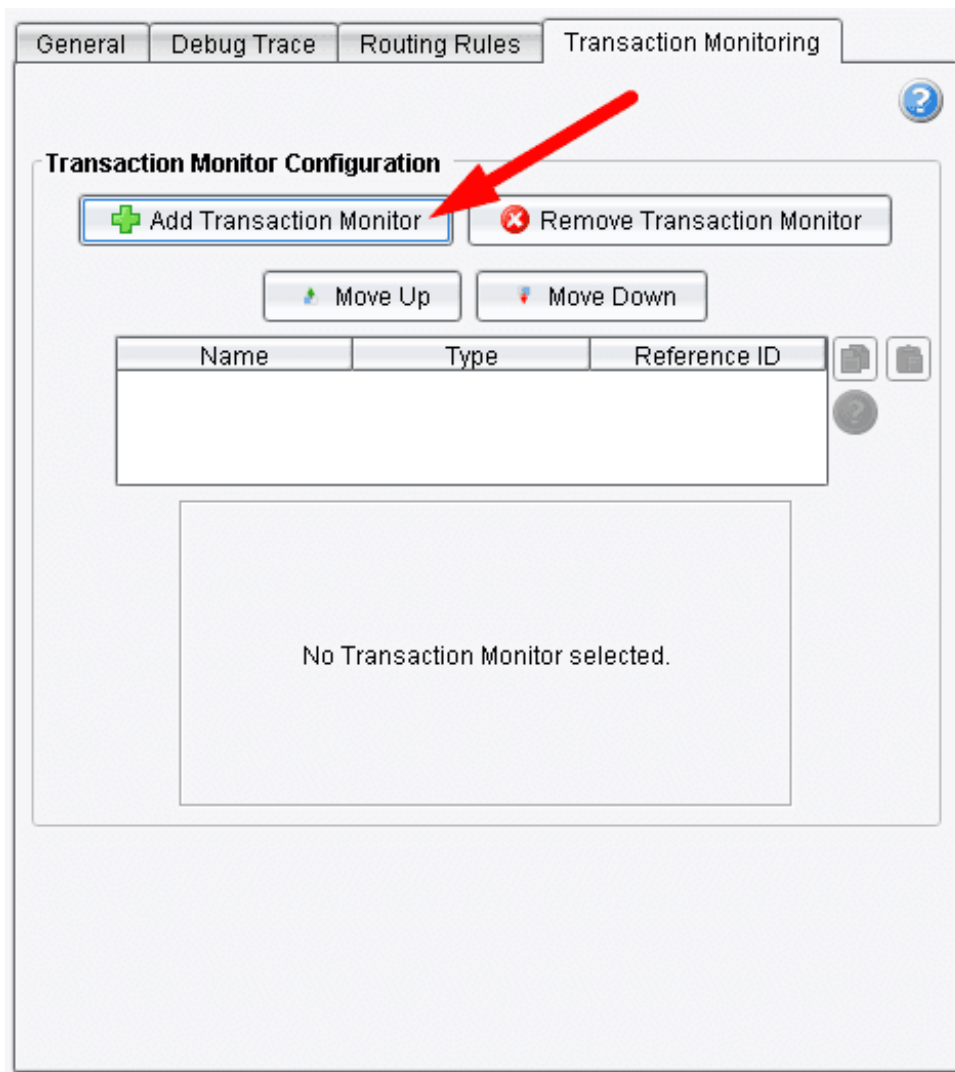
Then, double click to input the condition. You want to say if the Criminal equals **X** you'll wish to create this screening code **Criminal = 'X'**. Enter that condition into the text box and click **Return**.



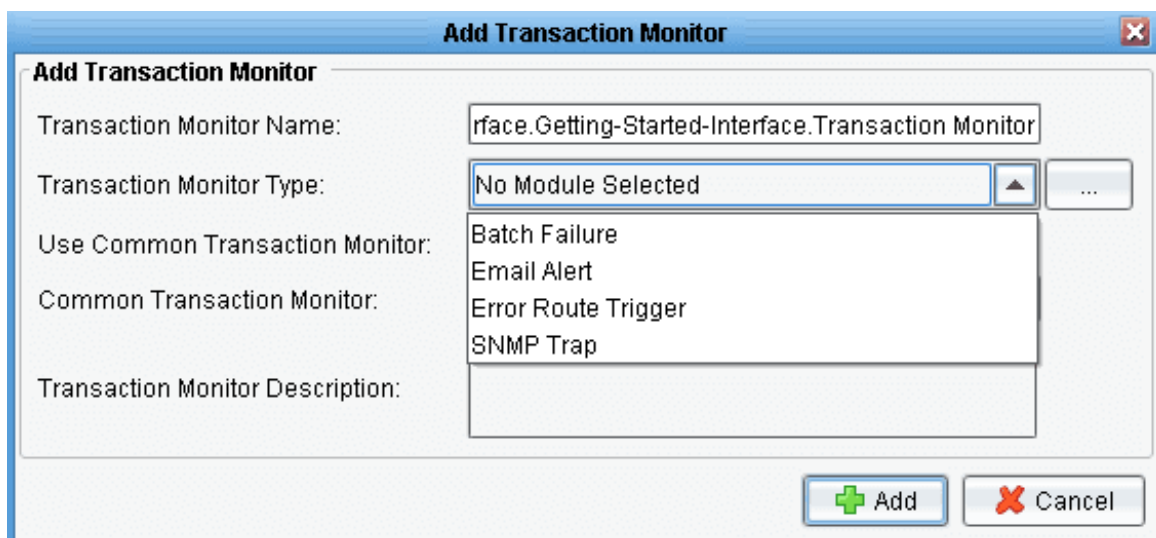
Now, as you did with the for each, you'll want to make this **Screening** a child of that condition. Select the **Screening** node and drop it on the **if**.



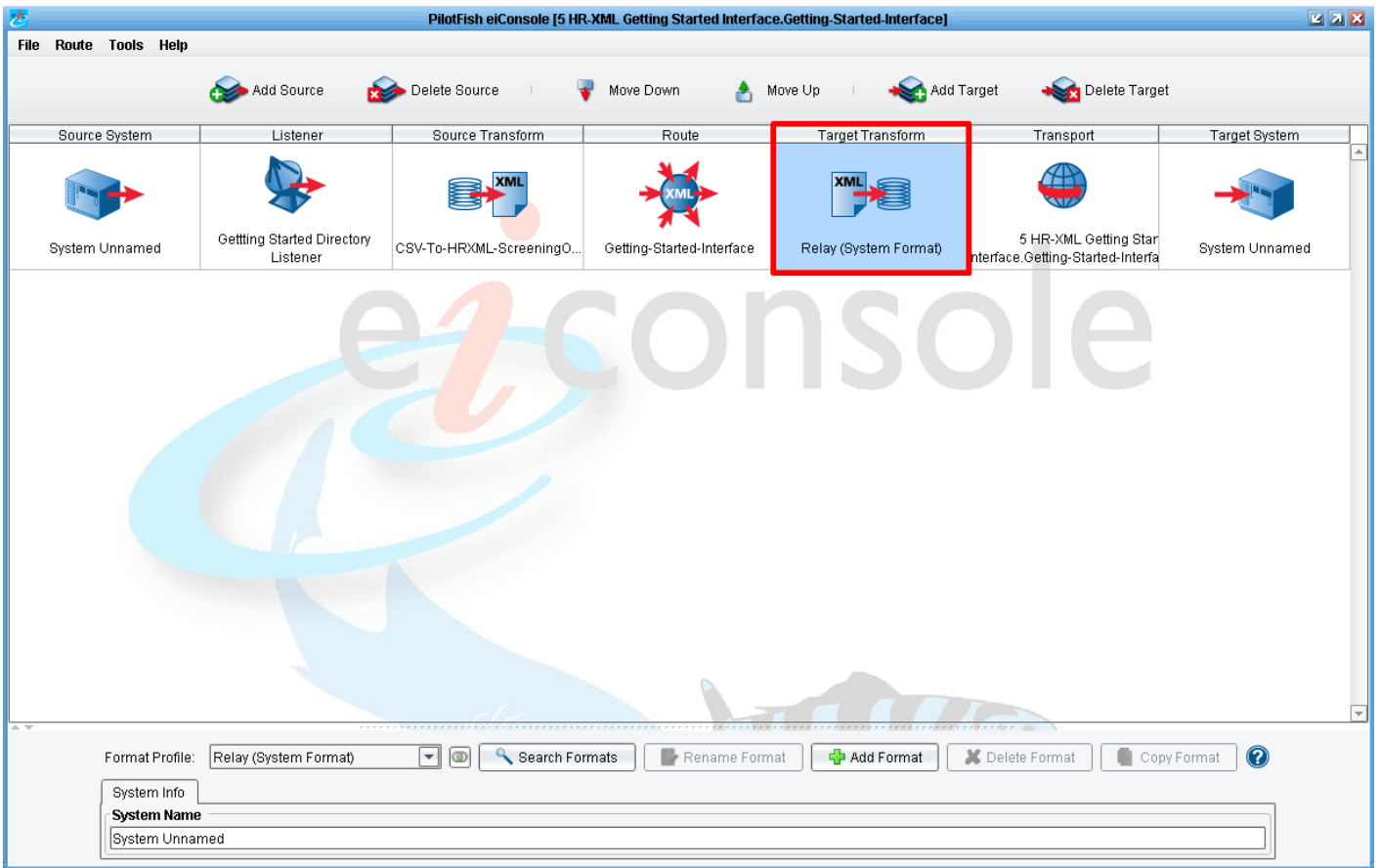
Next, click the **Route** icon in the main route grid. You won't need to configure any **Routing Rules** in this interface, but it's important to know that the **Routing Rules** tab can be used to set up XPath based routing rules to direct the data to one or multiple target systems based on its content.



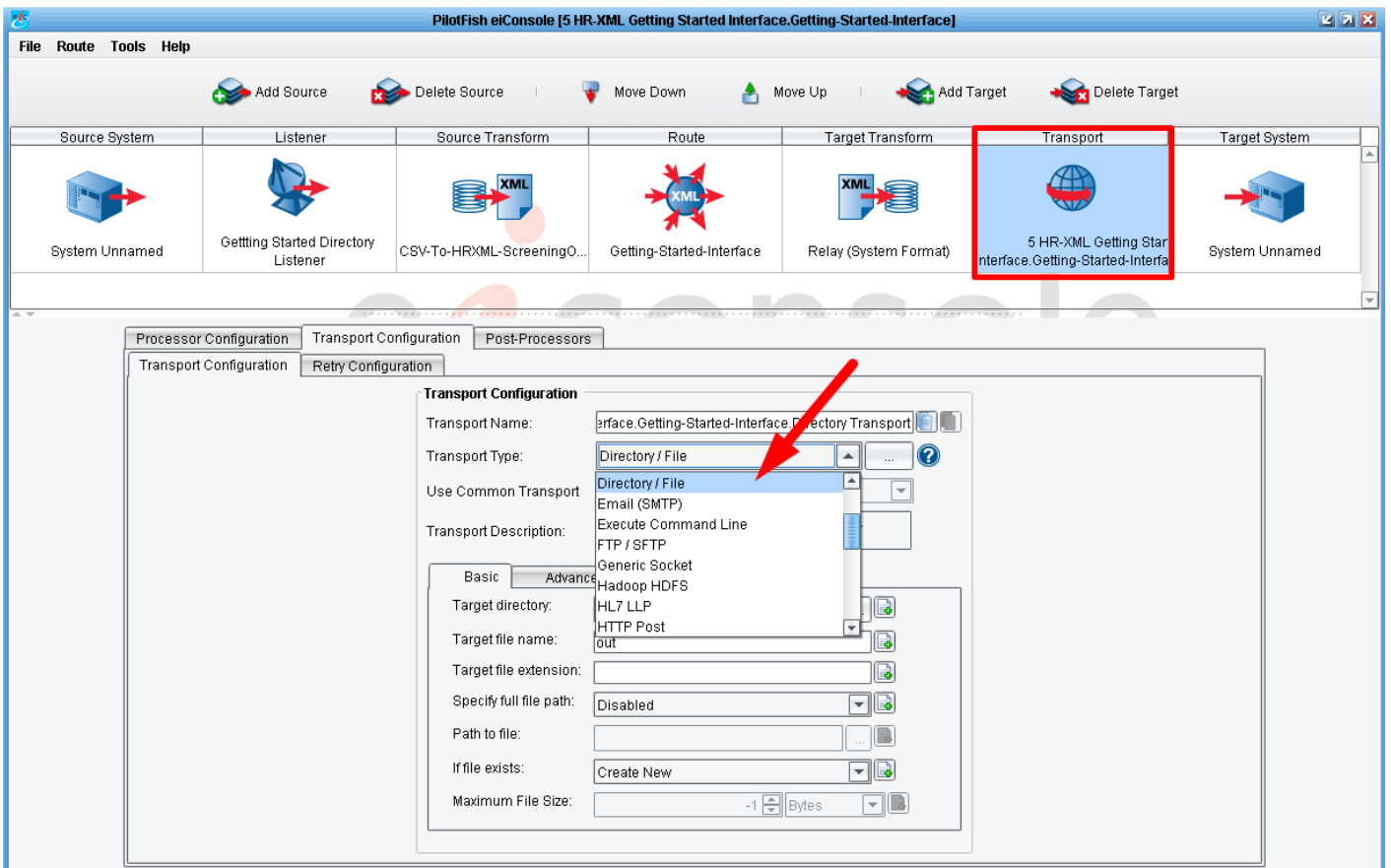
Also configured at this point in the main route grid are **Transaction Monitors**. Transaction Monitors are the eiConsole's tool for exception handling.



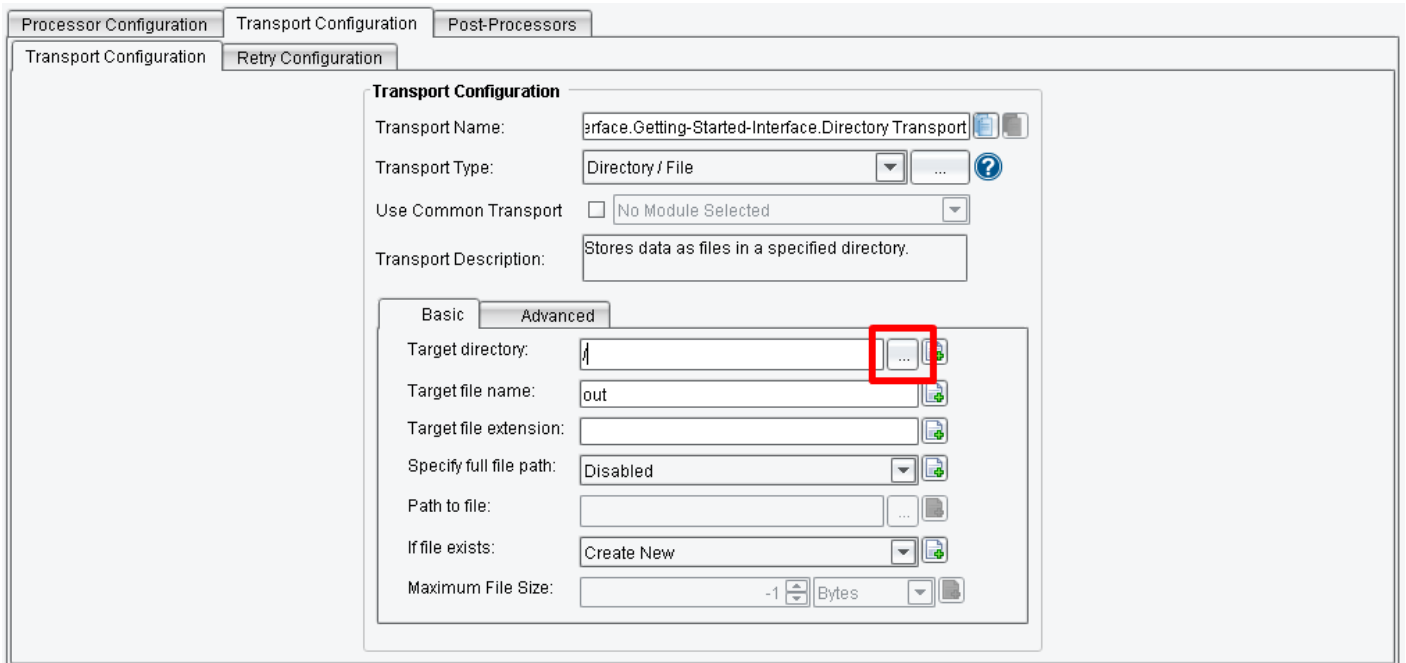
Email, other interfaces, or SNMP traps may be set up so that someone can be proactively notified if something goes wrong in the execution of an interface.



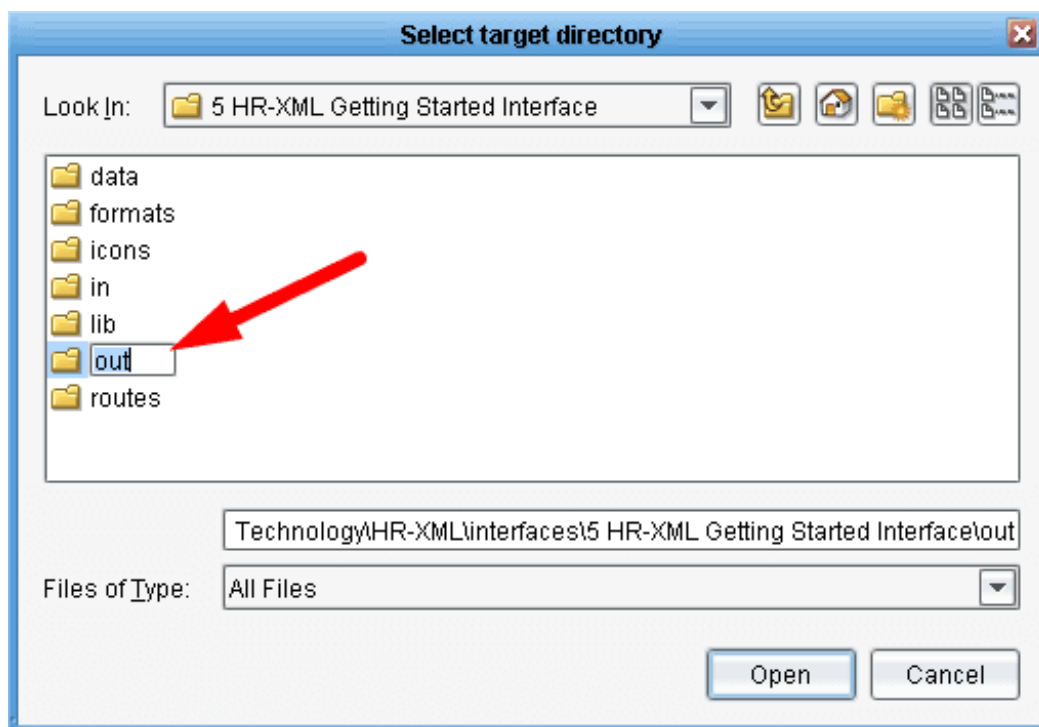
Now click on the **Target Transform** icon. When the format info panel opens, you'll note that the Target Transform is set to **Relay (System Format)**. There is no any transformations on this stage. This is because you'll expect your Target system to directly consume the HR-XML 3.0 BOD that you've created. However, if you need to tweak this, or you need to convert it into a completely different file format, this **Target Transformation** could be used, again leveraging the **Data Mapper**, **File Specification Editor**, and other tools to do further transformation of that data so that it appears exactly how the Target System can consume it.



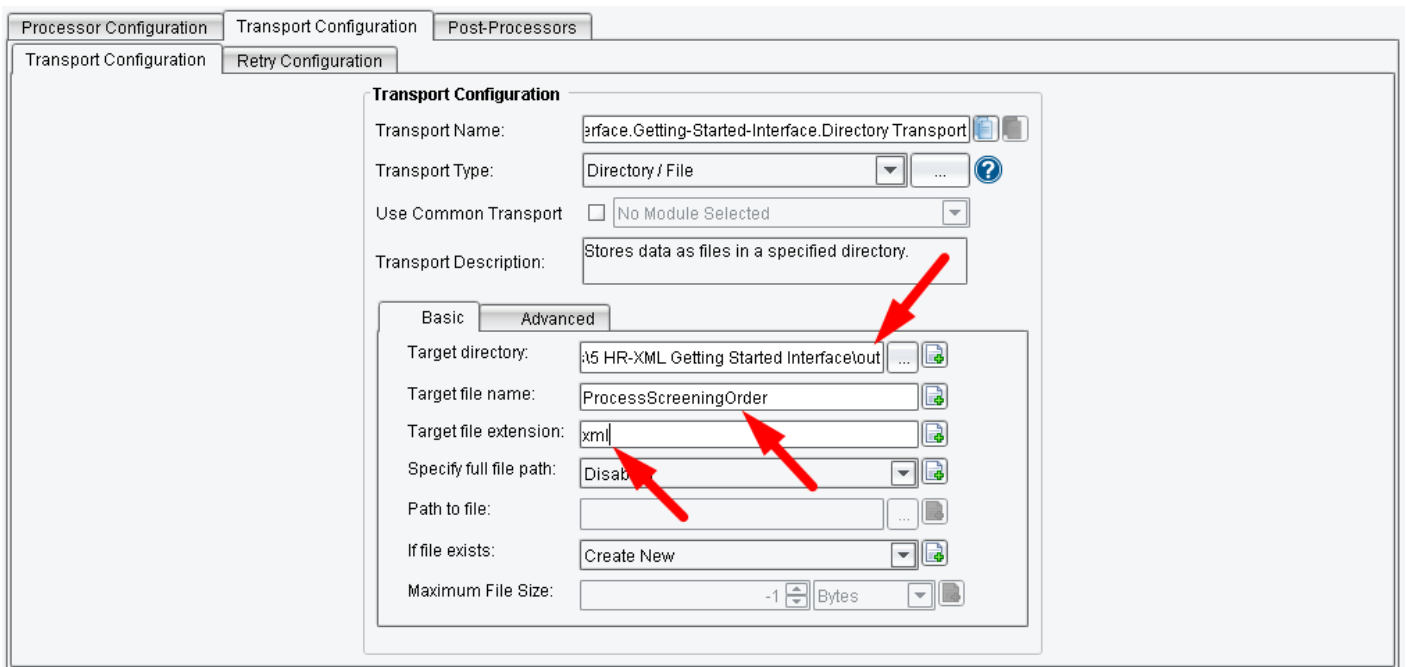
Finally, click on the **Transport** stage. The Transport stage is the mirror image of the Listener. It allows you to configure how the data goes from the eiConsole to the Target system – it handles that connectivity. The various connectivity options can be exposed through expanding the Transport Type dropdown. Here you simply have a **Directory / File** configured and you'll stick with that configuration.



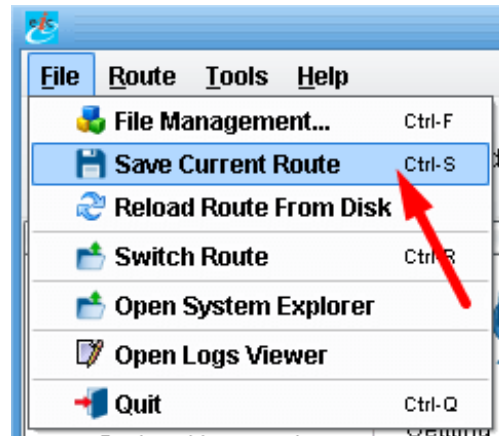
As you did on the Listener side, click the browse button. Click the **X** button next to the Target directory configuration item first, if there is no browse button.



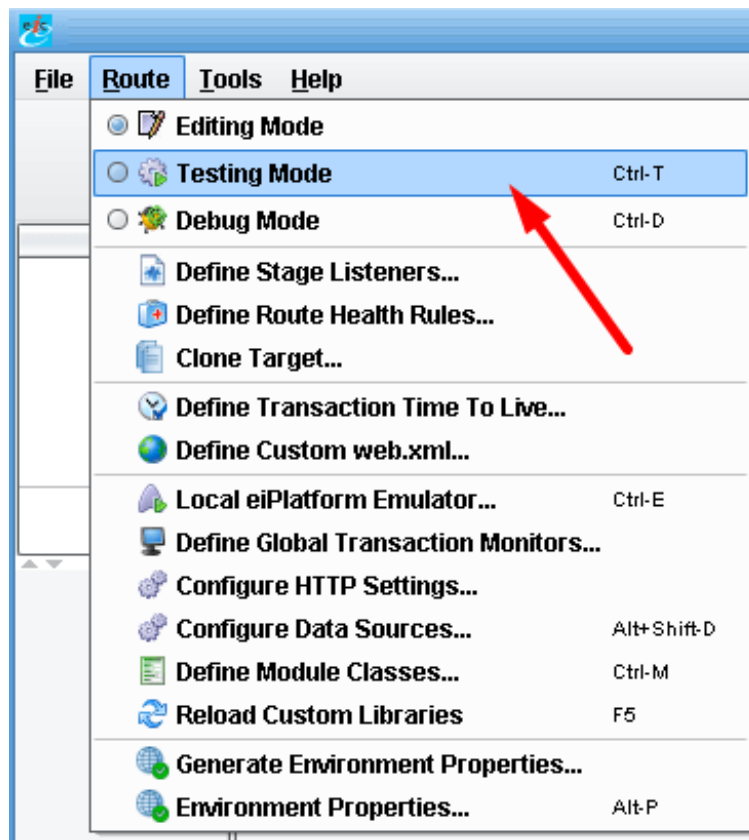
And create a new folder for the output. This time create a folder called **"out"** and click **Open**.



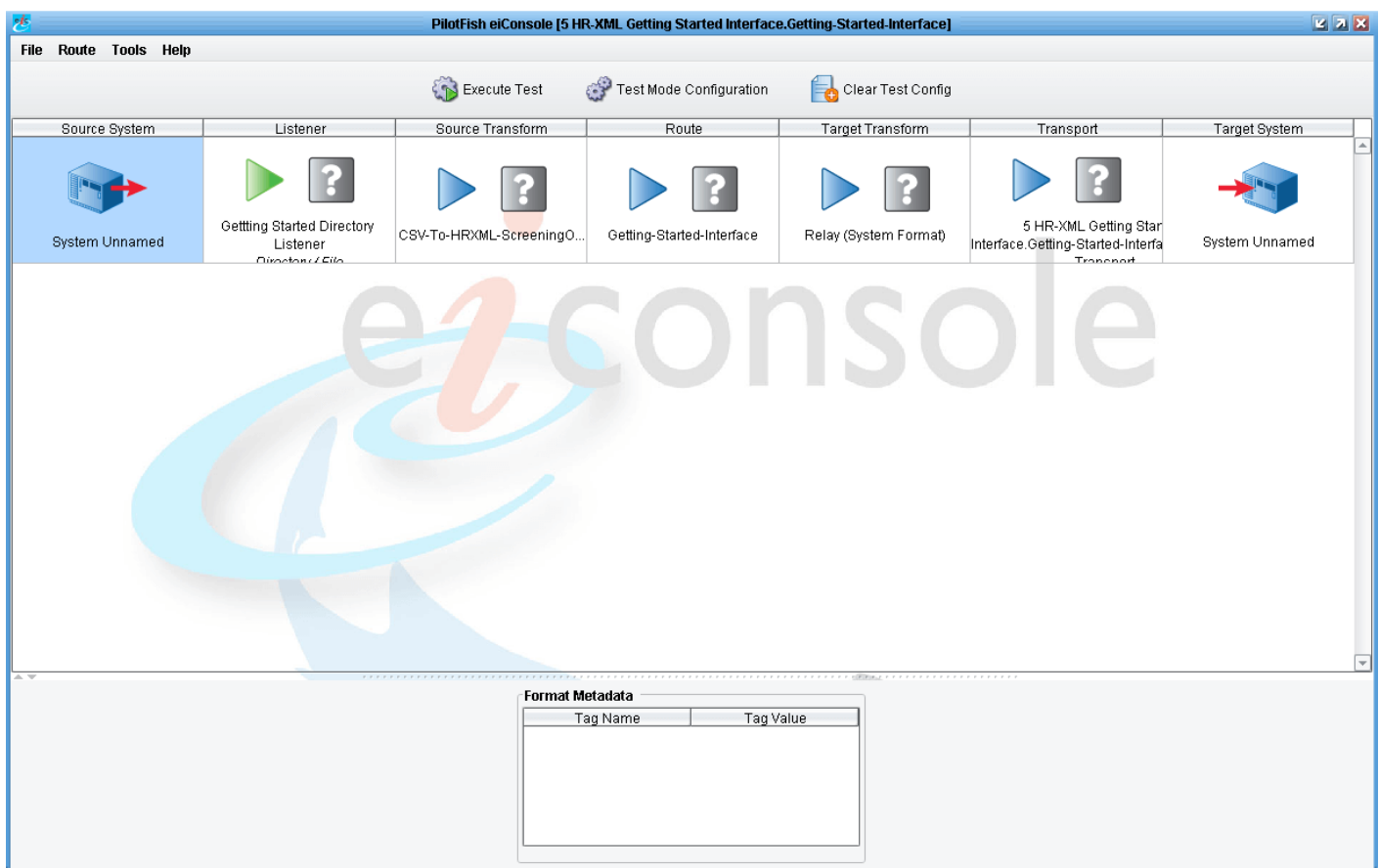
You'll note that you can also change the Target file name or the Target file extension. For example - **ProcessScreeningOrder.xml**.



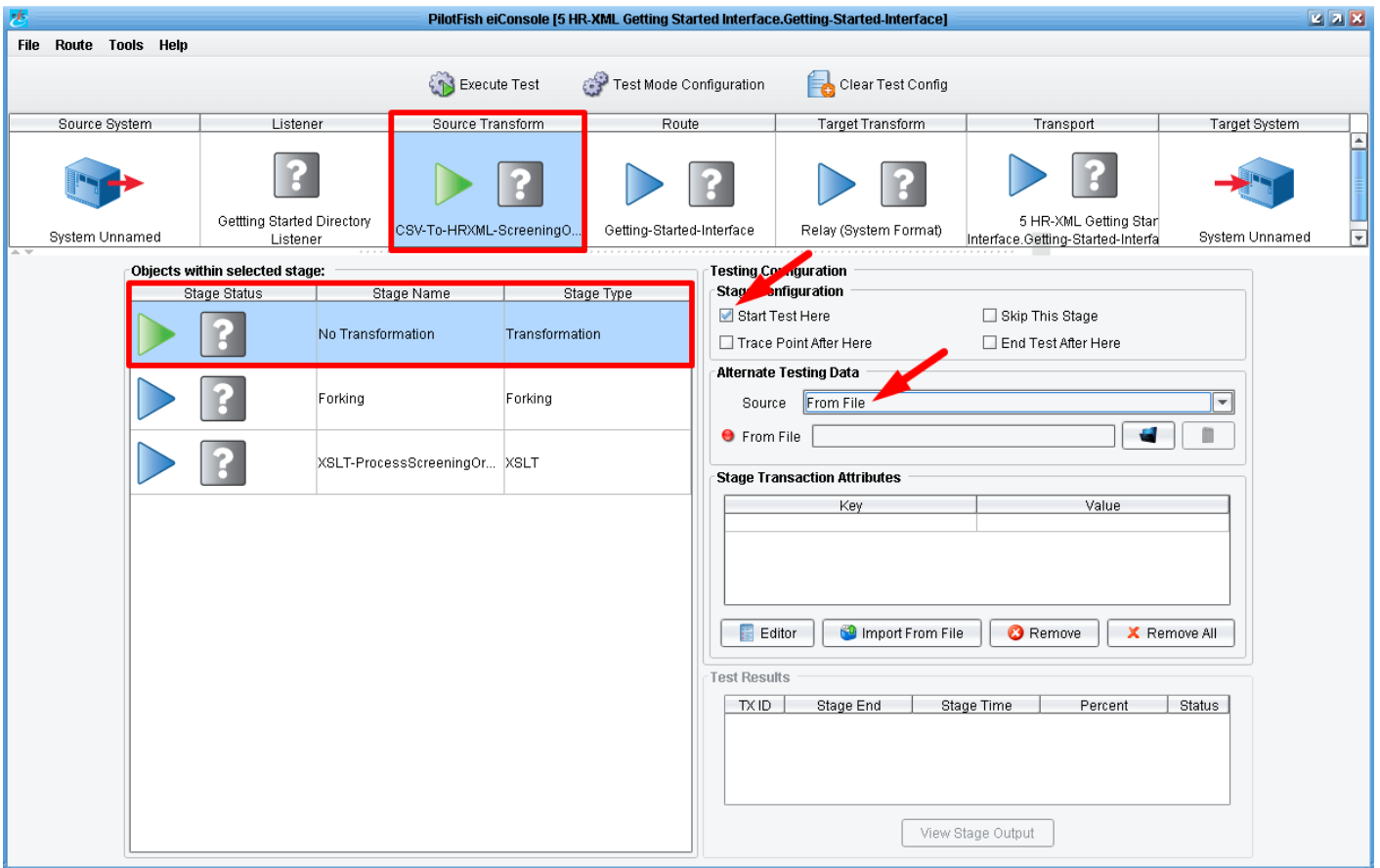
Now that you've configured your topology you can save your interface by choosing File, **Save Current Route**.



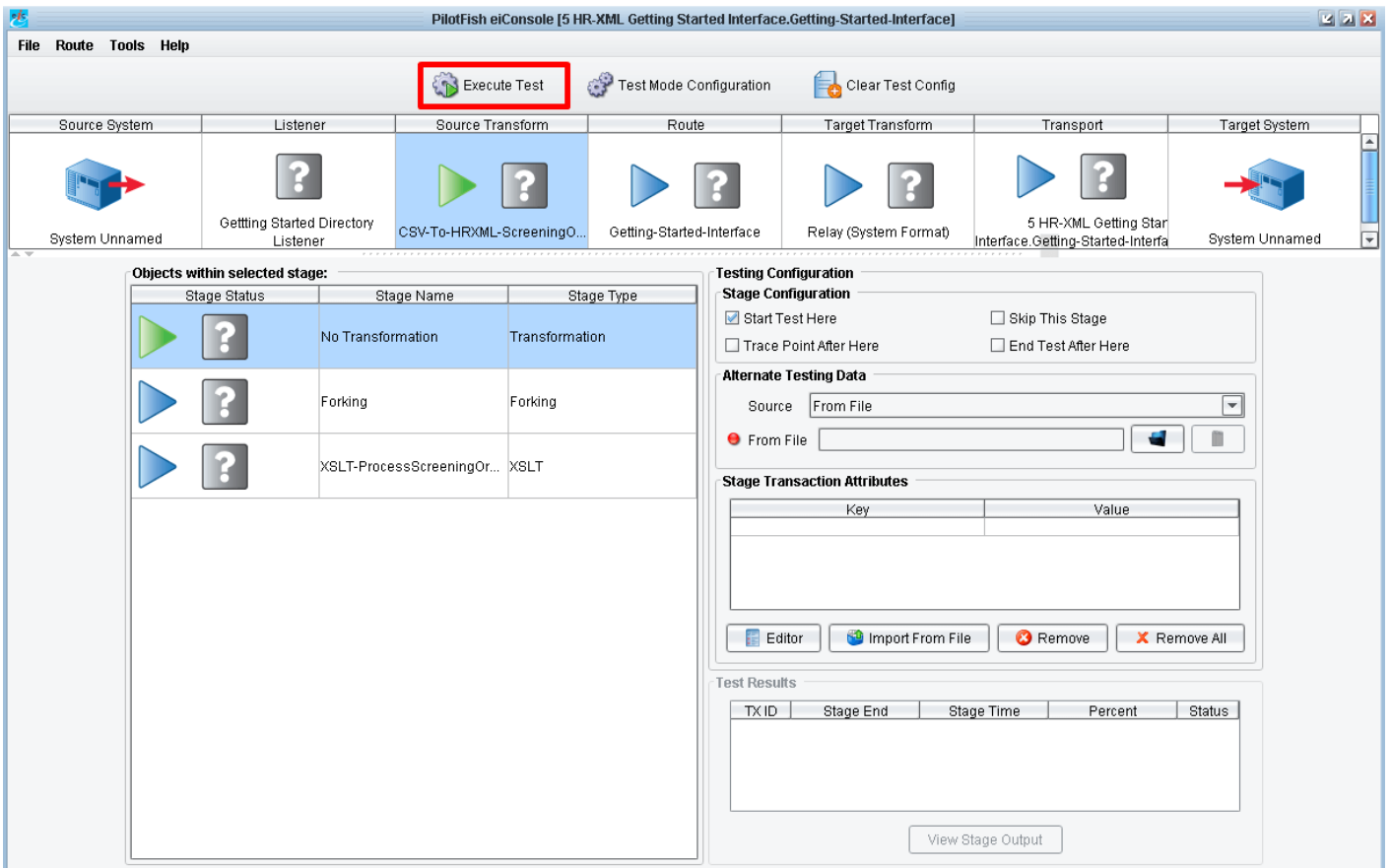
The final step is to test your work. Under the Route menu select **Testing Mode**.



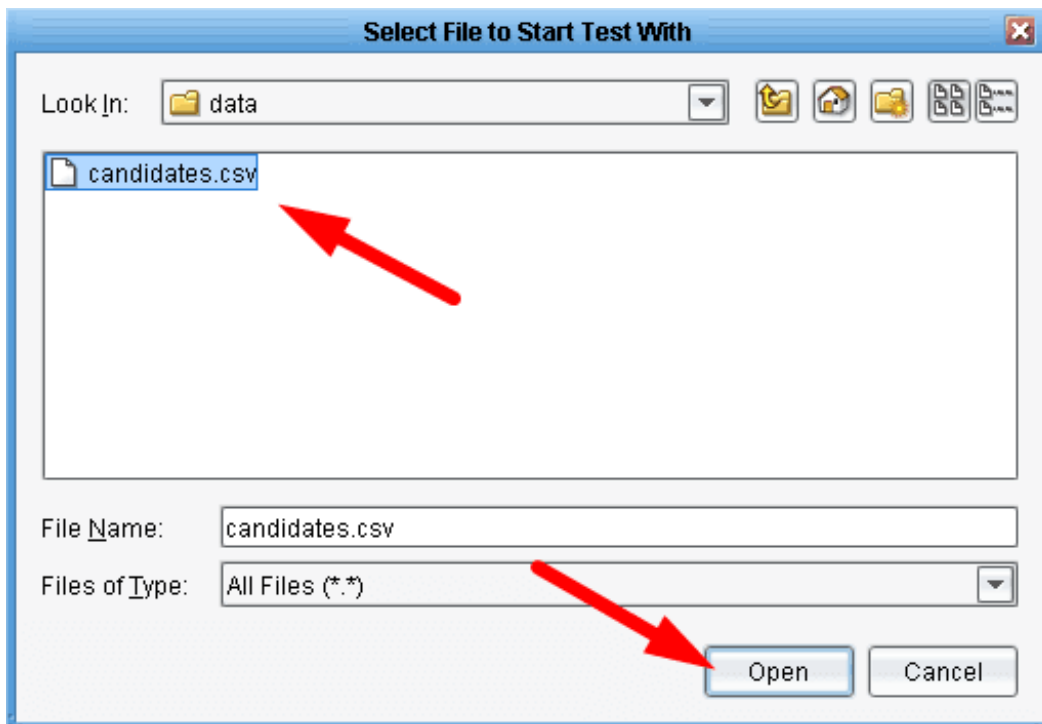
In testing mode you'll note that the icons between the Source System and the Target System now appear as question marks. These indicate stages of a test that you may choose to run or may choose not to run.



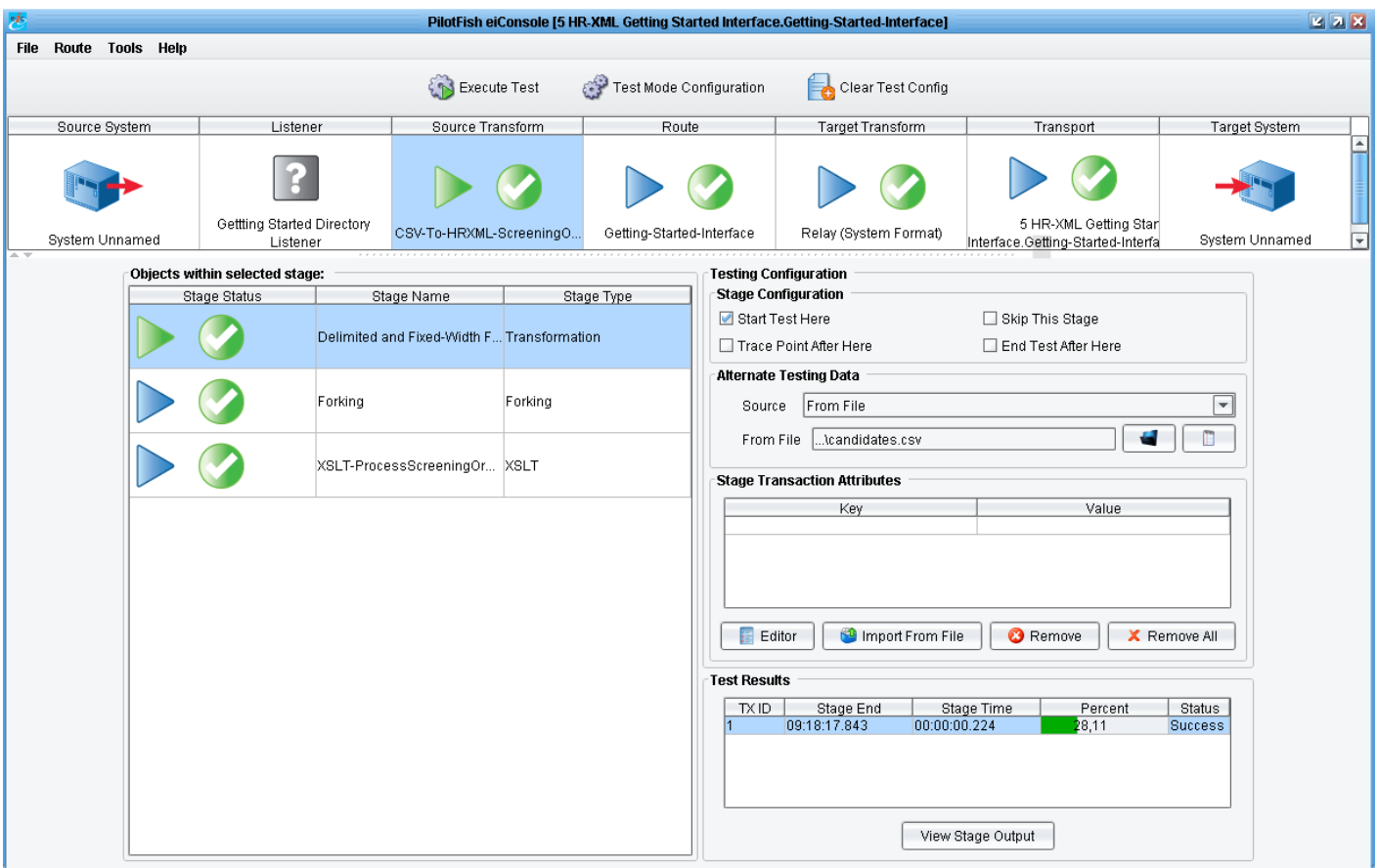
The default configuration will start the test with the Listener and allow it to run all the way through to the Transport. However, let's start with manually specifying the test file and beginning with the Source Transform. To do this, select the **Source Transform** icon in the grid and ensure that the **Delimited and Fixed-Width** file row is selected in the Objects within selected stage area. Then click **Start Test Here** in the Stage Configuration panel and choose **From File** from the **Source** drop-down menu.



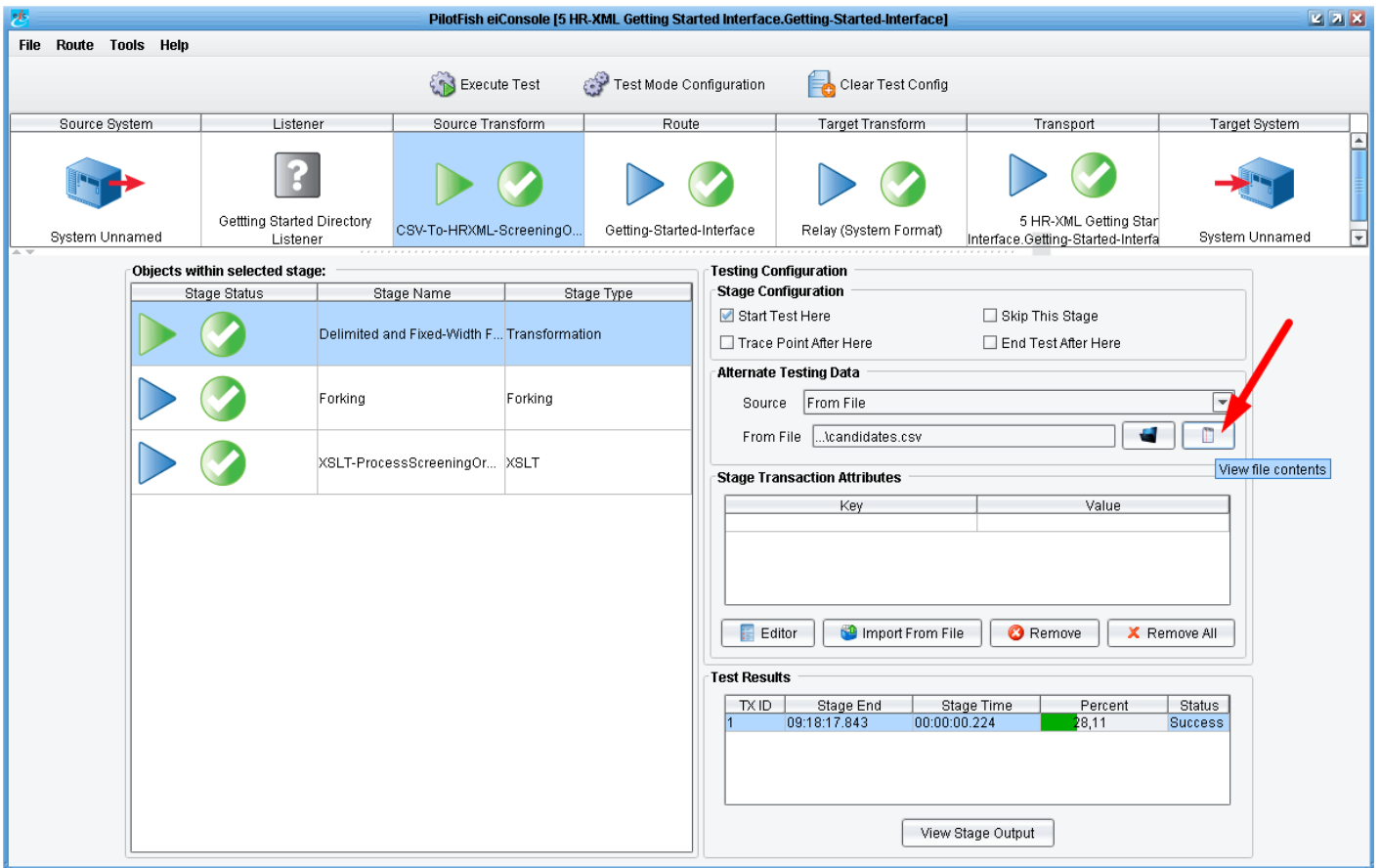
You'll note that the green and blue arrows above shift in order to indicate the path that this test will take. Click the **Execute Test** button.



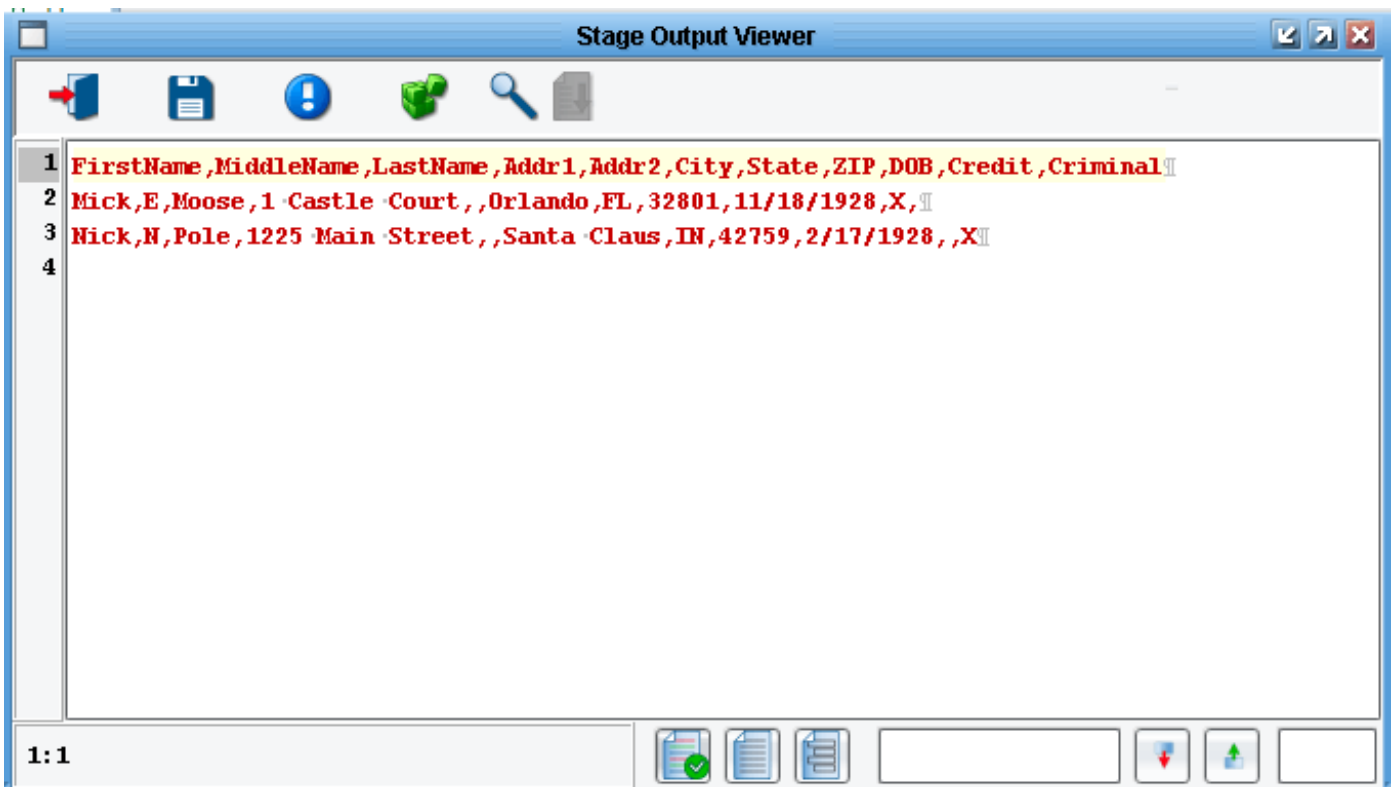
A file dialogue will appear that will allow you to choose the file you want to start your test with. Choose the **candidates.csv** file that you're now familiar with. Again, this is located in the input folder of your tutorial subfolder (`\interfaces\3 HR-XML Templates\data\tutorial\input\`). Click **Open**.



As each stage completes the question mark will either turn into a check, indicating success, or an X indicating failure. Here we see it all completes successfully, green arrows.



You can now take a look at how the data appeared at each point in the process. Clicking the **View** button...



...will again show you your input.

The screenshot displays the PilotFish eiConsole interface for a test configuration titled "5 HR-XML Getting Started Interface.Getting-Started-Interface". The main workflow consists of several stages: Source System (System Unnamed), Listener (Getting Started Directory Listener), Source Transform (CSV-To-HRXML-ScreeningO...), Route (Getting-Started-Interface), Target Transform (Relay (System Format)), Transport (5 HR-XML Getting Star Interface.Getting-Started-Interfa), and Target System (System Unnamed). All stages show a green play button and a checkmark, indicating successful execution.

The "Objects within selected stage" panel shows a table of sub-steps:

Stage Status	Stage Name	Stage Type
	Delimited and Fixed-Width F...	Transformation
	Forking	Forking
	XSLT-ProcessScreeningOr...	XSLT

The "Testing Configuration" panel includes options for "Start Test Here", "Trace Point After Here", "Skip This Stage", and "End Test After Here". The "Alternate Testing Data" section shows the source set to "From File" with the file path "...candidates.csv". The "Stage Transaction Attributes" table is currently empty.

The "Test Results" table shows the following data:

TX ID	Stage End	Stage Time	Percent	Status
1	09:18:17.843	00:00:00.224	28,11	Success

A red arrow points to the "View Stage Output" button located below the Test Results table.

Double clicking on the **Delimited and Fixed-Width file** sub-stage in the **Objects within selected stage** area...

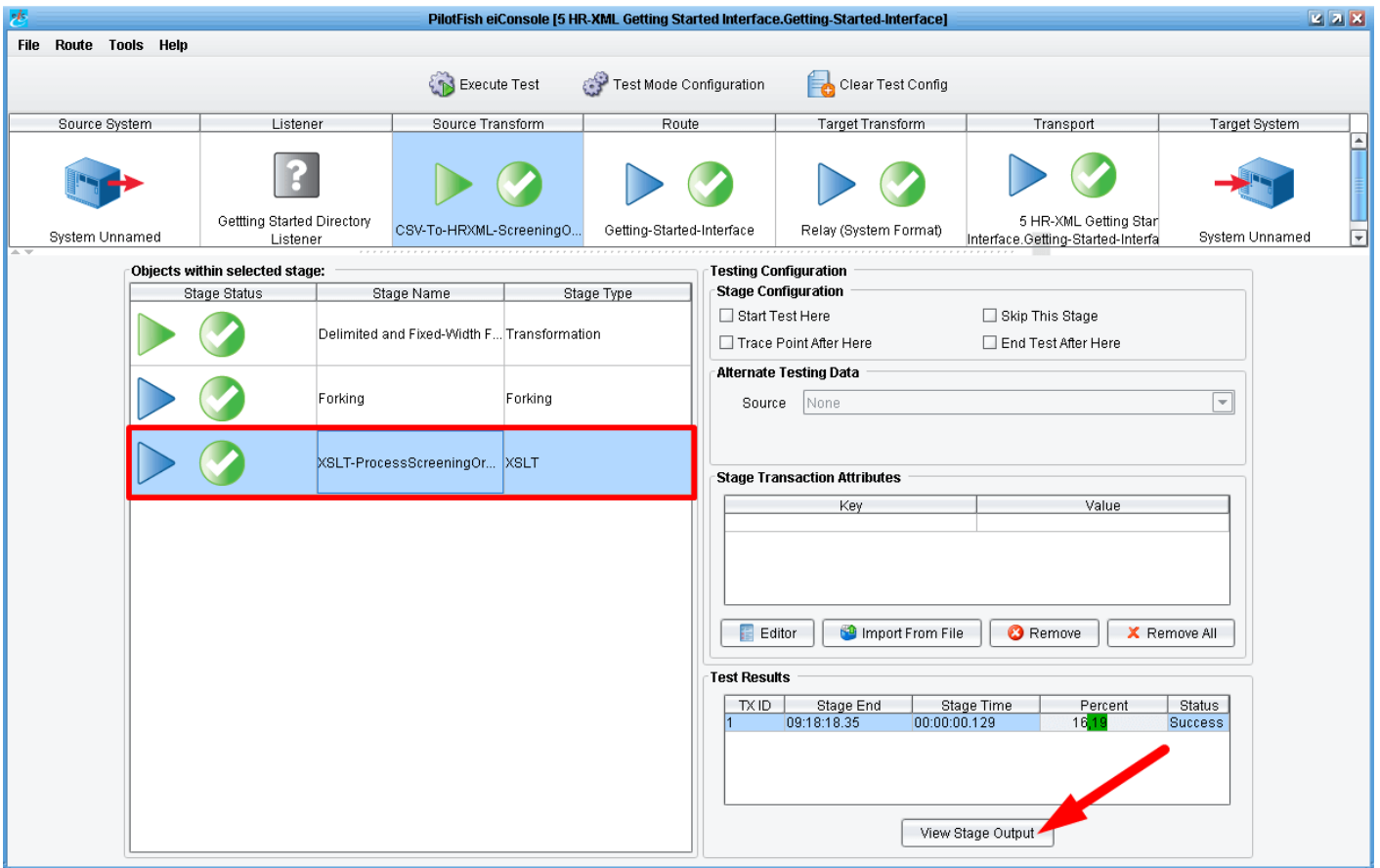


The screenshot shows a window titled "Stage Output Viewer" with a toolbar at the top containing icons for file operations and search. The main area displays XML code with line numbers 1 through 27 on the left. The XML content is as follows:

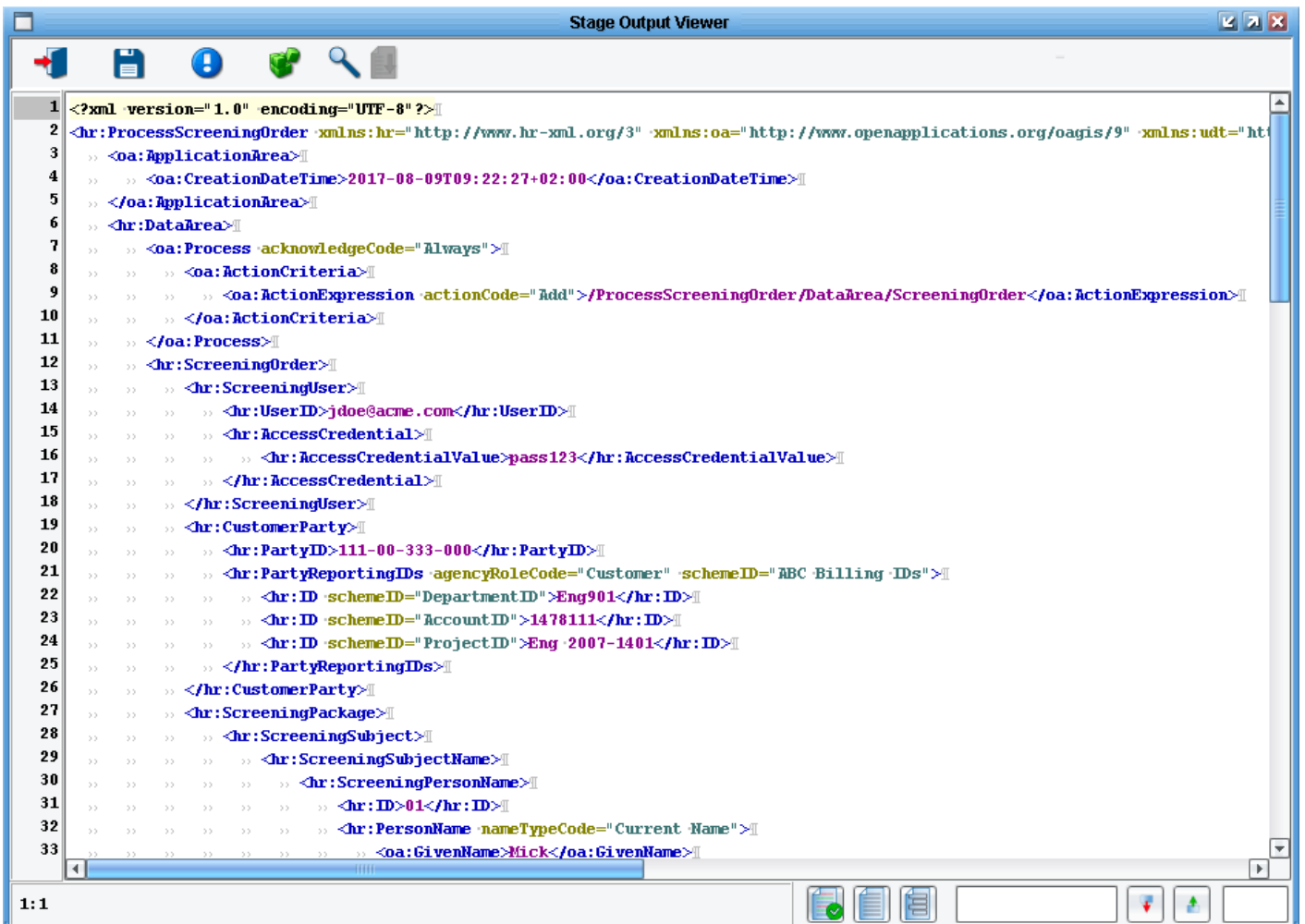
```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <XCSData>
3   >> <Record index="1">
4     >> >> <FirstName index="1">Mick</FirstName>
5     >> >> <MiddleName index="2">E</MiddleName>
6     >> >> <LastName index="3">Moose</LastName>
7     >> >> <Addr1 index="4">1 Castle Court</Addr1>
8     >> >> <City index="6">Orlando</City>
9     >> >> <State index="7">FL</State>
10    >> >> <ZIP index="8">32801</ZIP>
11    >> >> <DOB index="9">11/18/1928</DOB>
12    >> >> <Credit index="10">X</Credit>
13    >> </Record>
14    >> <Record index="2">
15      >> >> <FirstName index="1">Nick</FirstName>
16      >> >> <MiddleName index="2">N</MiddleName>
17      >> >> <LastName index="3">Pole</LastName>
18      >> >> <Addr1 index="4">1225 Main Street</Addr1>
19      >> >> <City index="6">Santa Claus</City>
20      >> >> <State index="7">IN</State>
21      >> >> <ZIP index="8">42759</ZIP>
22      >> >> <DOB index="9">2/17/1928</DOB>
23      >> >> <Criminal index="11">X</Criminal>
24    >> </Record>
25 </XCSData>
26
27
```

At the bottom left, the text "1:1" is displayed. The bottom toolbar contains icons for file operations and a search icon.

This will bring up the XML representation of that flat file.



Next you used XSLT generated in the Data Mapper to convert that data into the HR-XML 3.0 BOD. Double clicking the XSLT stage...



...will show you the generated Process Screening Order.

```

Stage Output Viewer
22 >> >> >> >> <hr:ID schemeID="DepartmentID">Eng901</hr:ID>]]
23 >> >> >> >> <hr:ID schemeID="AccountID">1478111</hr:ID>]]
24 >> >> >> >> <hr:ID schemeID="ProjectID">Eng 2007-1401</hr:ID>]]
25 >> >> >> >> </hr:PartyReportingIDs>]]
26 >> >> >> >> </hr:CustomerParty>]]
27 >> >> >> >> <hr:ScreeningPackage>]]
28 >> >> >> >> <hr:ScreeningSubject>]]
29 >> >> >> >> <hr:ScreeningSubjectName>]]
30 >> >> >> >> <hr:ScreeningPersonName>]]
31 >> >> >> >> <hr:ID>01</hr:ID>]]
32 >> >> >> >> <hr:PersonName nameTypeCode="Current Name">]]
33 >> >> >> >> <oa:GivenName>Mick</oa:GivenName>]]
34 >> >> >> >> <hr:MiddleName>E</hr:MiddleName>]]
35 >> >> >> >> <hr:FamilyName>Moose</hr:FamilyName>]]
36 >> >> >> >> </hr:PersonName>]]
37 >> >> >> >> </hr:ScreeningPersonName>]]
38 >> >> >> >> </hr:ScreeningSubjectName>]]
39 >> >> >> >> <hr:Communication preferredIndicator="true" sequence="1">]]
40 >> >> >> >> <hr:Address>]]
41 >> >> >> >> <oa:LineOne>1 Castle Court</oa:LineOne>]]
42 >> >> >> >> <oa:LineTwo />]]
43 >> >> >> >> <oa:CityName>Orlando</oa:CityName>]]
44 >> >> >> >> <oa:CountrySubDivisionCode>FL</oa:CountrySubDivisionCode>]]
45 >> >> >> >> <hr:CountryCode>US</hr:CountryCode>]]
46 >> >> >> >> <oa:PostalCode>32801</oa:PostalCode>]]
47 >> >> >> >> </hr:Address>]]
48 >> >> >> >> </hr:Communication>]]
49 >> >> >> >> <hr:FreeFormBirthDate>]]
50 >> >> >> >> <hr:FormattedDateTime>1928-11-18</hr:FormattedDateTime>]]
51 >> >> >> >> </hr:FreeFormBirthDate>]]
52 >> >> >> >> </hr:ScreeningSubject>]]
53 >> >> >> >> <hr:Screening>]]
54 >> >> >> >> <hr:PartyReportingIDs agencyRoleCode="Requester" schemeID="ATS Co.">]]

```

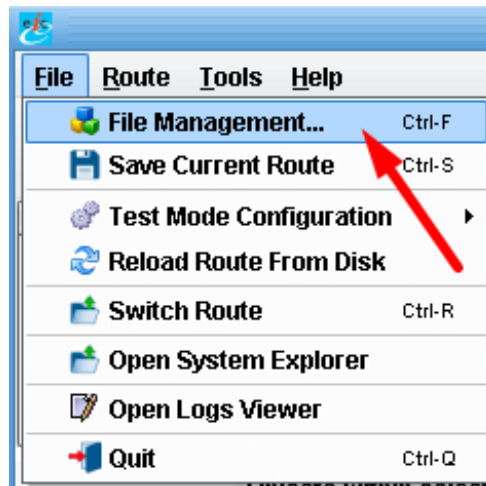
You'll see the values that you mapped dynamically populated into the template.

```

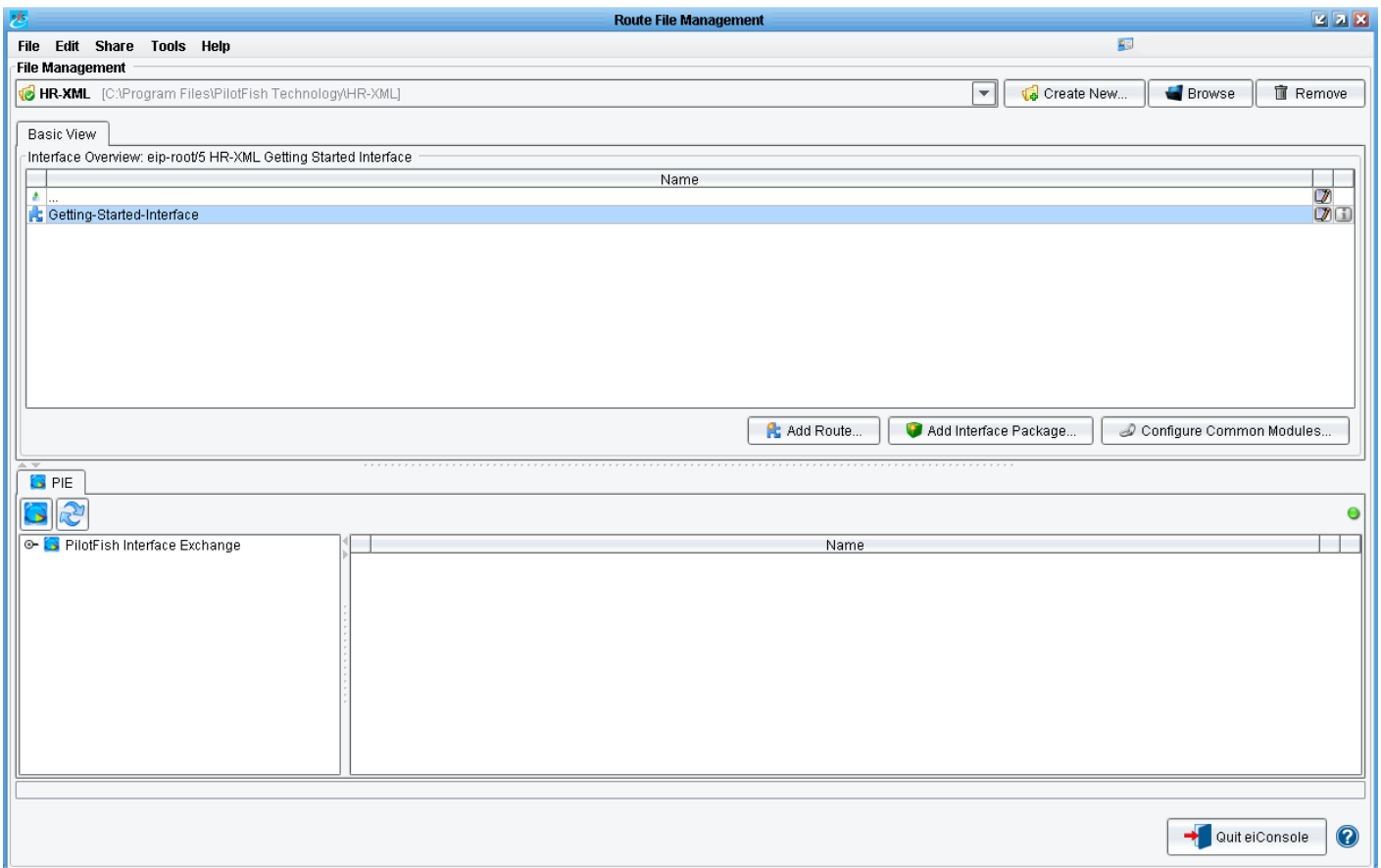
ProcessScreeningOrder.xml
1 <?xml version="1.0" encoding="UTF-8"?><hr:ProcessScreeningOrder xmlns:hr="http://www.hr-xml.org/3" xmlns:udt="
http://www.openapplications.org/oagis/9/unqualifieddatatypes/1.1" xmlns:oa="http://www.openapplications.org/oagis/9"
systemEnvironmentCode="Production" releaseID="3.0" languageCode="en-US"><oa:ApplicationArea><oa:CreationDate>
2016-04-25T15:35:21+02:00</oa:CreationDate></oa:ApplicationArea><hr:DataArea><oa:Process acknowledgeCode="Always"
><oa:ActionCriteria><oa:ActionExpression actionCode="Add"/></oa:ActionCriteria></oa:Process><hr:ScreeningOrder><hr:ScreeningUser><hr:UserID>jdoe@acme.com
</hr:UserID><hr:AccessCredential><hr:AccessCredentialValue>pass123
</hr:AccessCredentialValue></hr:AccessCredential></hr:ScreeningUser><hr:CustomerParty><hr:PartyID>111-00-333-000
</hr:PartyID><hr:PartyReportingIDs schemeID="ABC Billing IDs" agencyRoleCode="Customer"><hr:ID schemeID="DepartmentID">Eng901
</hr:ID><hr:ID schemeID="AccountID">1478111</hr:ID><hr:ID schemeID="ProjectID">Eng 2007-1401
</hr:ID></hr:PartyReportingIDs></hr:CustomerParty><hr:ScreeningPackage><hr:ScreeningSubject><hr:ScreeningSubjectName><hr:ScreeningPerso
nName><hr:ID>01</hr:ID><hr:PersonName nameTypeCode="Current Name"><oa:GivenName>Mick</oa:GivenName><hr:MiddleName>E
</hr:MiddleName><hr:FamilyName>Moose
</hr:FamilyName></hr:PersonName></hr:ScreeningPersonName></hr:ScreeningSubjectName><hr:Communication sequence="1" preferredIndicator=
"true"><hr:Address><oa:LineOne>1 Castle Court</oa:LineOne><oa:LineTwo/><oa:CityName>Orlando</oa:CityName><oa:CountrySubDivisionCode>FL
</oa:CountrySubDivisionCode><hr:CountryCode>US</hr:CountryCode><oa:PostalCode>32801
</oa:PostalCode></hr:Address></hr:Communication><hr:FreeFormBirthDate><hr:FormattedDateTime>1928-11-18
</hr:FormattedDateTime></hr:FreeFormBirthDate></hr:ScreeningSubject><hr:Screening><hr:PartyReportingIDs schemeID="ATS Co."
agencyRoleCode="Requester"><hr:ID schemeID="RequisitionID">998Eng901</hr:ID><hr:ID schemeID="CandidateID">1111478111
</hr:ID></hr:PartyReportingIDs><hr:PartyReportingIDs schemeID="ABC Billing IDs" agencyRoleCode="Customer"><hr:ID schemeID=
"DepartmentID">Eng901</hr:ID><hr:ID schemeID="AccountID">1478111</hr:ID><hr:ID schemeID="ProjectID">Eng 2007-1401
</hr:ID></hr:PartyReportingIDs><hr:ScreeningSearchTypeCode>Credit
</hr:ScreeningSearchTypeCode><hr:SearchCredit><hr:SearchCreditTypeCode>Summary</hr:SearchCreditTypeCode><hr:CreditBureauCode>Experian
</hr:CreditBureauCode><hr:EndUserName>Acme Co. Inc.</hr:EndUserName><hr:ScreeningPermissiblePurposeCode>Employment
</hr:ScreeningPermissiblePurposeCode></hr:SearchCredit></hr:Screening></hr:ScreeningPackage><hr:ScreeningSubject><hr:ScreeningSubjectName><hr:ScreeningPersonName><hr:ID>01</hr:ID><hr:PersonName nameTypeCode="Current Name"><oa:GivenName>Nick
</oa:GivenName><hr:MiddleName>N</hr:MiddleName><hr:FamilyName>Pole
</hr:FamilyName></hr:PersonName></hr:ScreeningPersonName></hr:ScreeningSubjectName><hr:Communication sequence="1" preferredIndicator=
"true"><hr:Address><oa:LineOne>1225 Main Street</oa:LineOne><oa:LineTwo/><oa:CityName>Santa Claus
</oa:CityName><oa:CountrySubDivisionCode>IN</oa:CountrySubDivisionCode><hr:CountryCode>US</hr:CountryCode><oa:PostalCode>42759
</oa:PostalCode></hr:Address></hr:Communication><hr:FreeFormBirthDate><hr:FormattedDateTime>1928-02-17
</hr:FormattedDateTime></hr:FreeFormBirthDate></hr:Screening><hr:PartyReportingIDs schemeID="ATS Co."
agencyRoleCode="Requester"><hr:ID schemeID="RequisitionID">998Eng901</hr:ID><hr:ID schemeID="CandidateID">1111478111
</hr:ID></hr:PartyReportingIDs><hr:PartyReportingIDs schemeID="ABC Billing IDs" agencyRoleCode="Customer"><hr:ID schemeID=
"DepartmentID">Eng901</hr:ID><hr:ID schemeID="AccountID">1478111</hr:ID><hr:ID schemeID="ProjectID">Eng 2007-1401
</hr:ID></hr:PartyReportingIDs><hr:ScreeningSearchTypeCode>Criminal
</hr:ScreeningSearchTypeCode><hr:SearchCriminal><hr:SearchCriminalTypeCode>Statewide
</hr:SearchCriminalTypeCode><hr:SearchPeriod><oa:Duration>P7Y</oa:Duration></hr:SearchPeriod><oa:CityName>Raleigh
</oa:CityName><oa:CountrySubDivisionCode listID="State">NC</oa:CountrySubDivisionCode><oa:CountrySubDivisionCode listID="County">NC

```

The Routing and Target Transformation stages didn't do much work here. And finally you dropped the file in a directory. Navigating to the output folder that you created you should see a file called **ProcessScreeningOrder.xml** containing the same view that you saw as the output of your process.



Congratulations! You've now built your first HR-XML 3.0 interface using the eiConsole for HR-XML. The next step, of course, is to deploy this interface somewhere. To do that, save your interface again. Then choose File, **File Management**.



Your Getting Started interface is now ready for deployment to an eiPlatform server.

Thanks again for downloading the eiConsole for HR-XML. We're absolutely sure you'll find this is the best, fastest, most cost effective way to build these interfaces. If you have any more questions, please visit our website (www.pilotfishtechnology.com) or give us a call (860-632-9900)... we are always happy to offer some free training.