eiConsole for Healthcare – Getting Started Tutorial

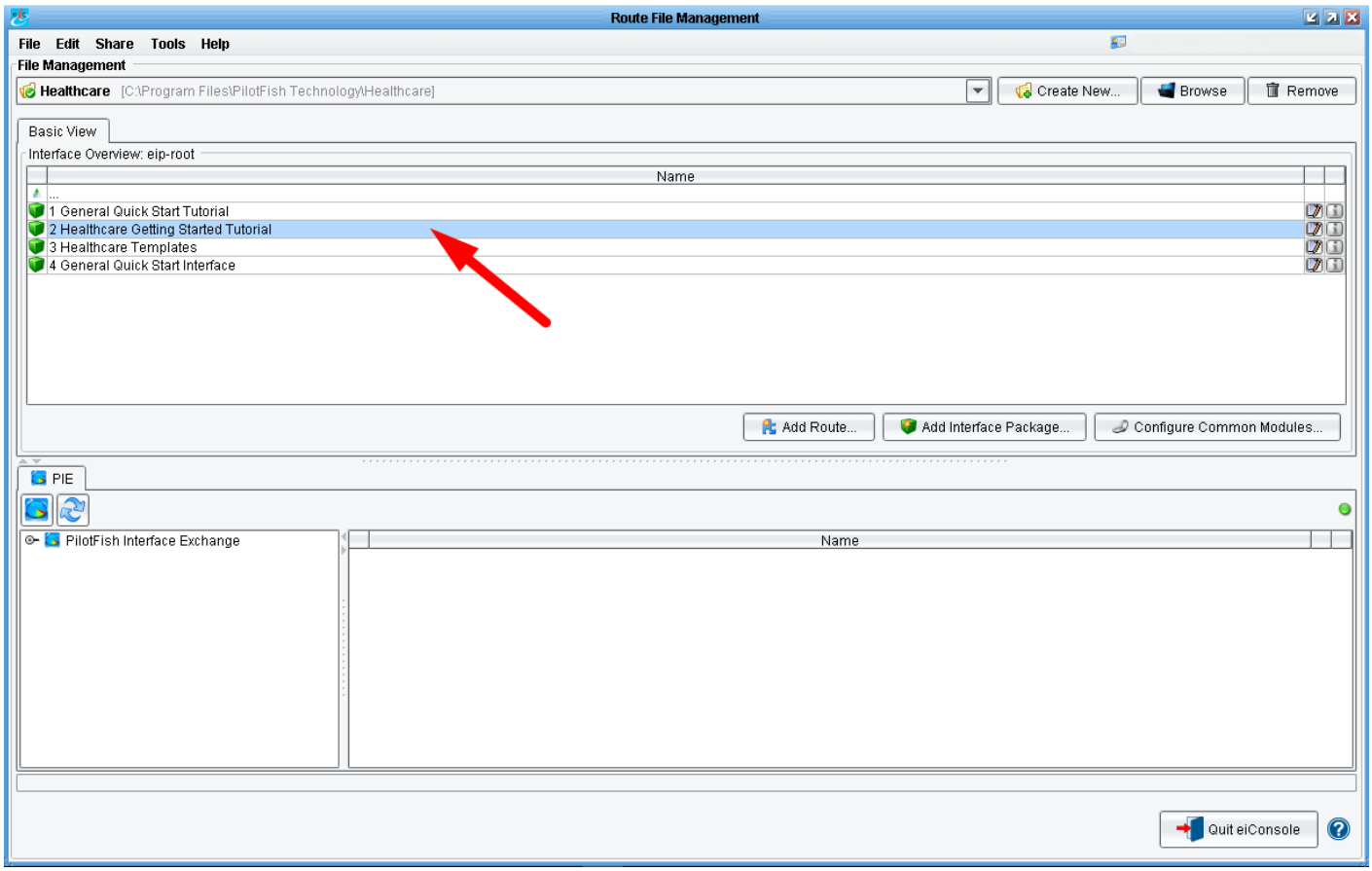
https://cms.pilotfishtechnology.com/eiconsole-for-healthcare-getting-started-tutorial

Welcome to the eiConsole for Healthcare Getting Started Tutorial. In this tutorial you'll build a simple interface that accepts HL7 input and converts it into an XLS spreadsheet. This interface will take the average user 45-60 minutes to complete.

The **General Quick Start Tutorial** and **General Level I Tutorials** (Modules 1-12) are prerequisites before moving on to the healthcare-specific tutorials and documentation. If you have not completed these, please do so now. The Healthcare Getting Started Tutorial builds on concepts learned in the previous tutorials that are not repeated here and are required to complete this tutorial.

If you just downloaded the eiConsole for Healthcare your bundle will have all the files you need. If you have not installed the eiConsole recently, click eiConsole **Update** to download and install the latest slipstream release. Then, visit **eiConsole Bundles** to download the latest industry bundle that includes the sample files that you'll need to do the tutorial. Click here if you need help importing a **Bundle**.

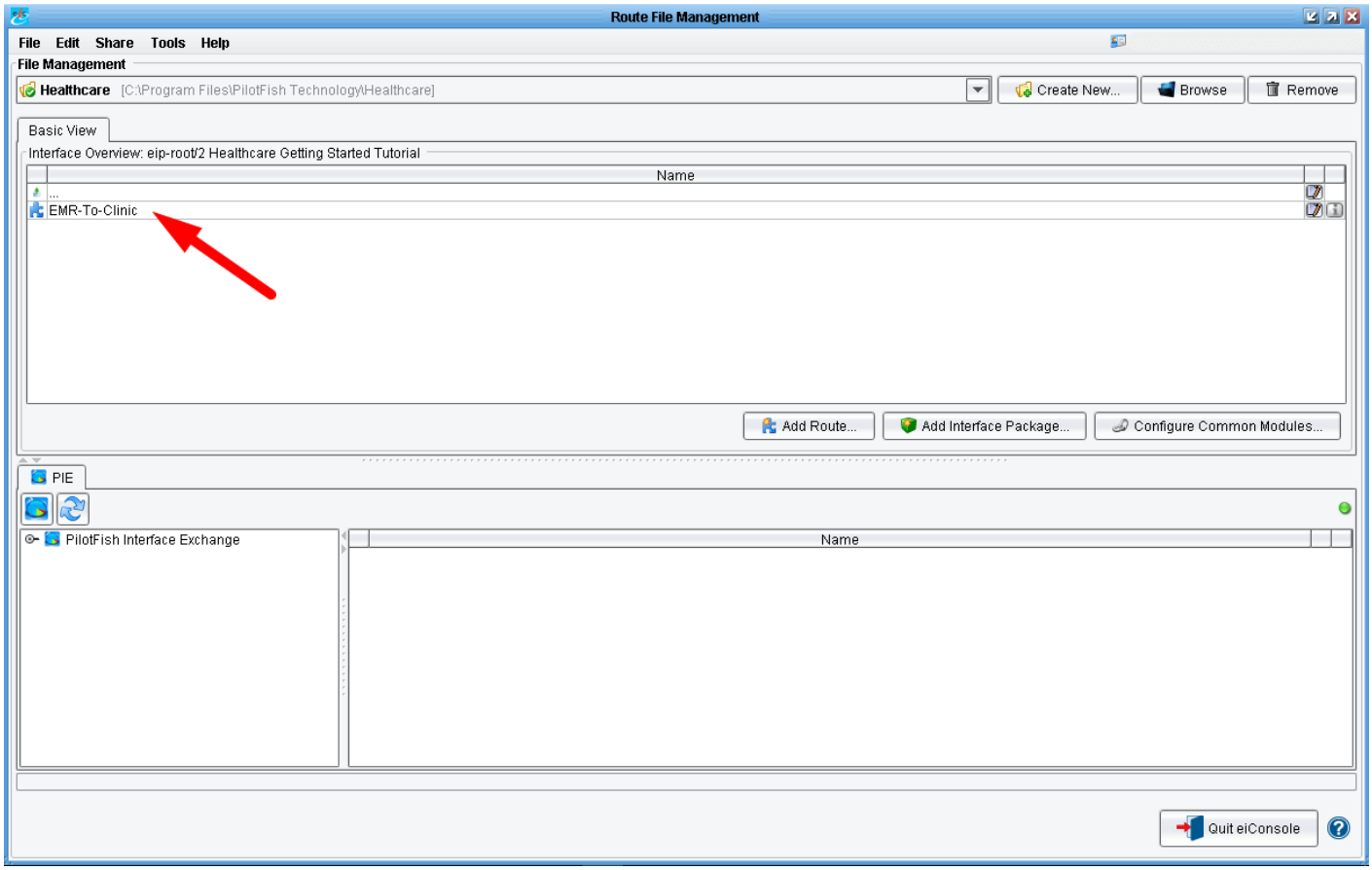
The Route File Management Window



We begin on the **Route File Management** window.

We have 4 eiConsole Packages in our Interface Overview grid, three from the eiConsole for Healthcare bundle and the fourth, the **4 General Quick Start Interface**, your newly created, first interface.

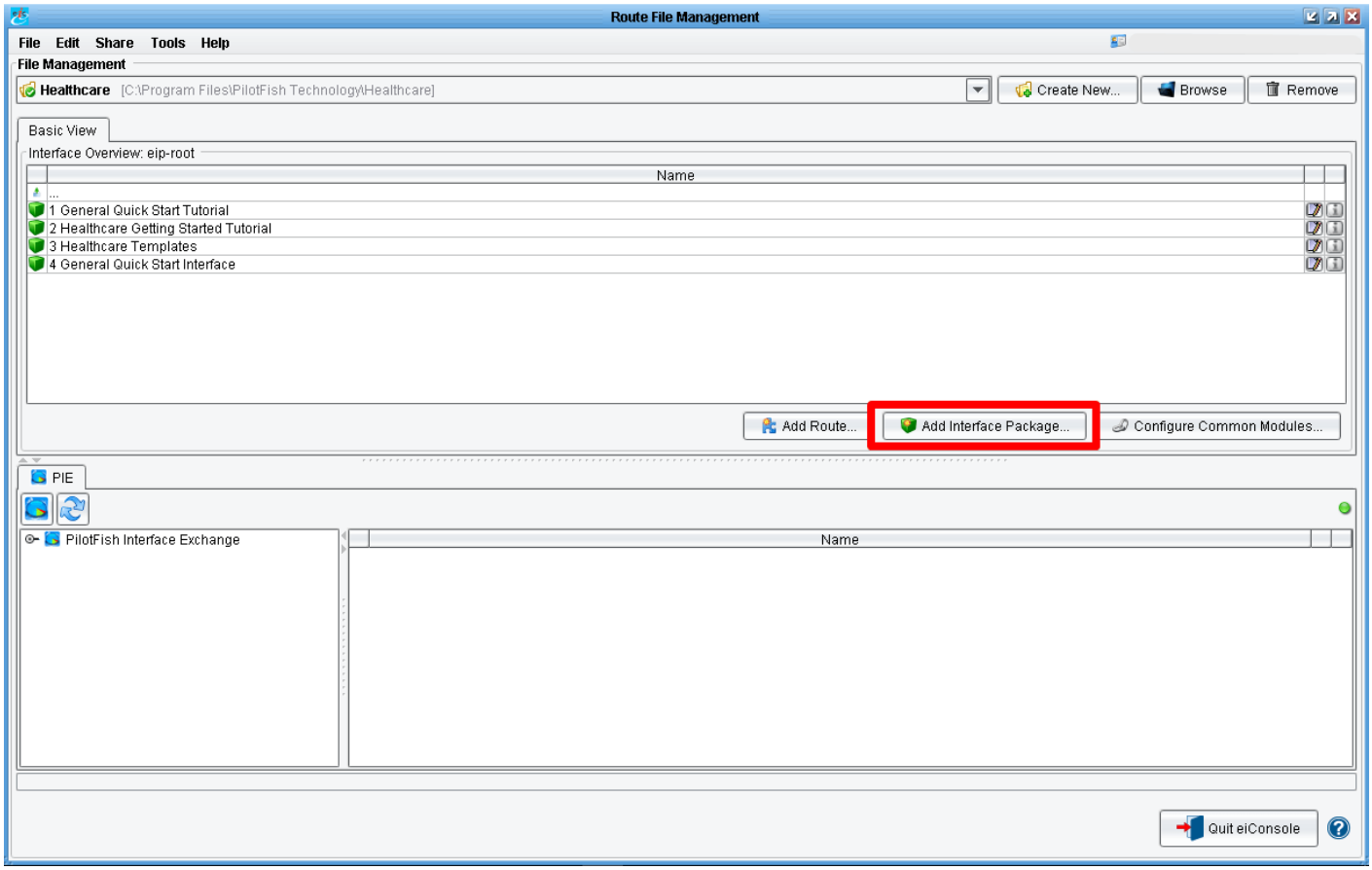
We'll begin our tutorial by double clicking the **2 Healthcare Getting Started Tutorial** Package row.



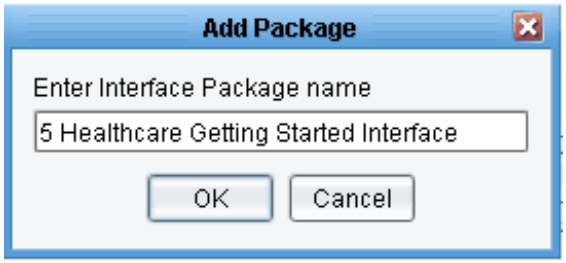
The sample interface, **EMR-To-Clinic**, appears in the **Interface Overview** grid. The blue icon indicates that this is a fully configured interface. It is the same interface you will be configuring from scratch during the course of this tutorial. You will use sample files located in the Working Directory (eip-root) for this interface. Note their location. Feel free to double click the sample interface to browse through the configuration screens. Or if you run into a problem in the tutorial you can always review the sample interface for reference.

Note: in order to successfully run the sample interface in the eiConsole's test mode you will need to set the polling directory for the Listener and Transport stages to point to appropriate folders in the sample tutorial that you downloaded on your computer.

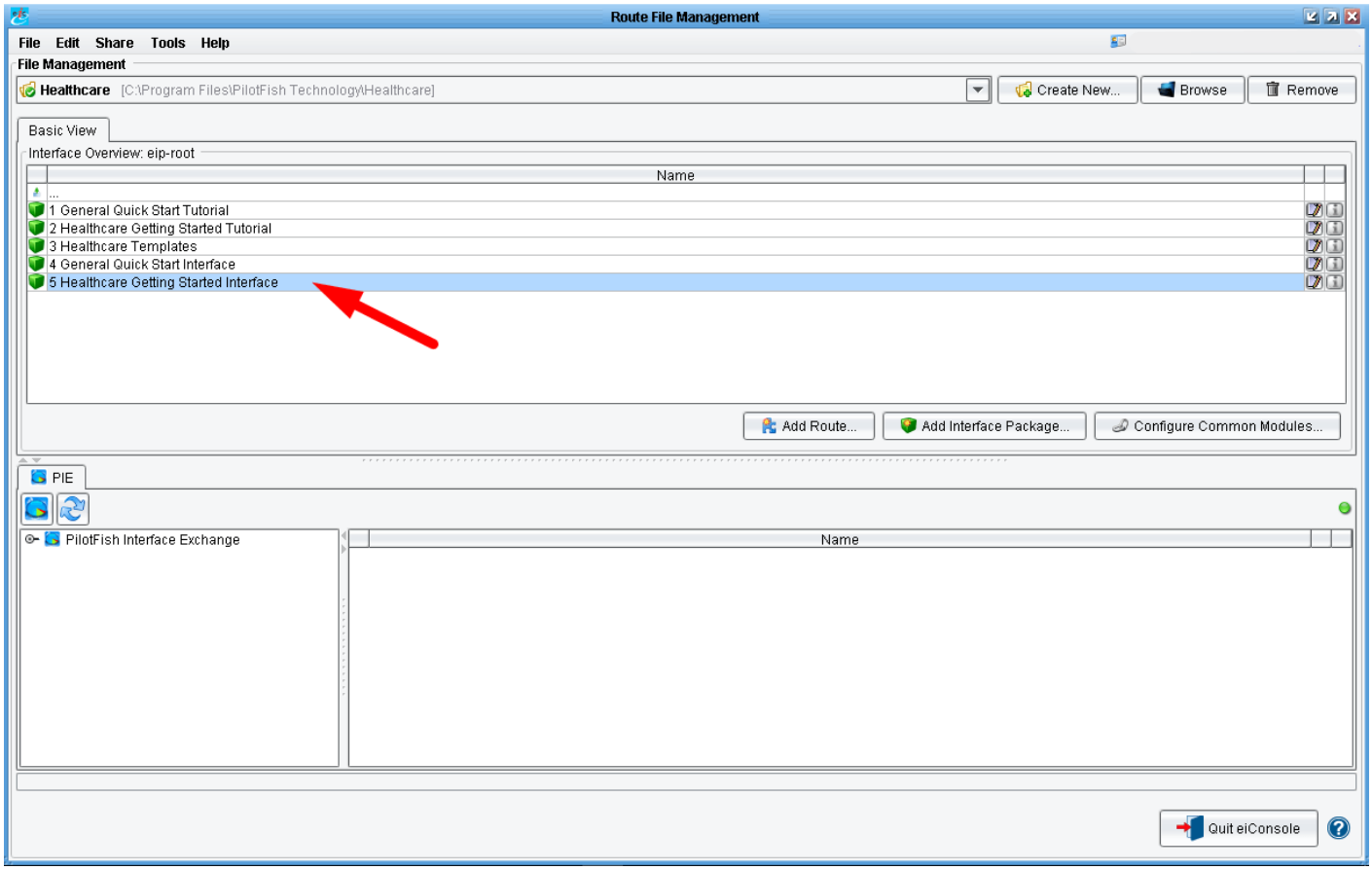
We'll return to the main Working Directory, eip-root. To do so, double click anywhere in the row above **EMR-To-Clinic**.



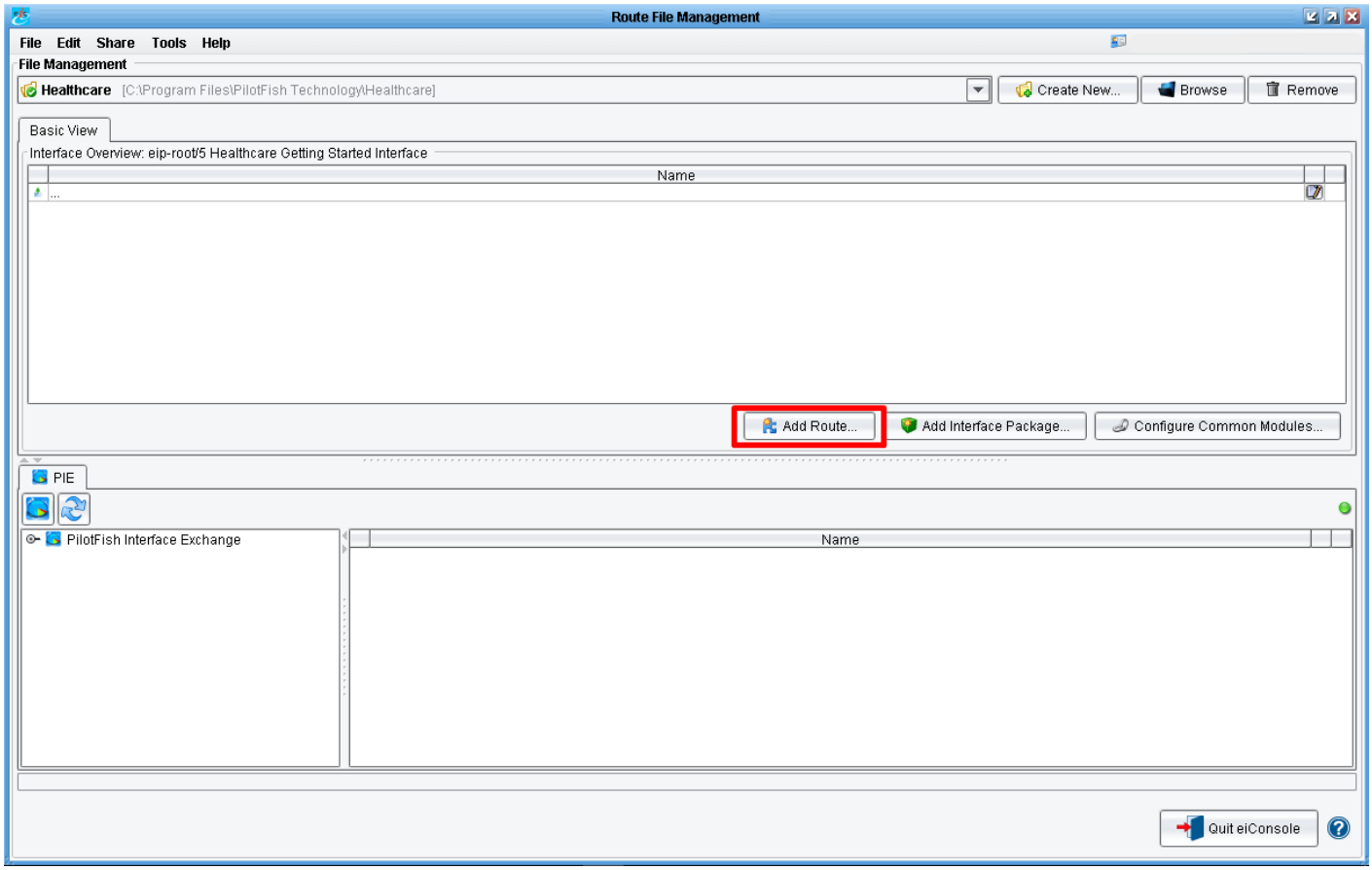
The Route File Management Window opens with the 4 Packages. Next, we will want to create a new package for the interface we will be configuring. Click the **Add** **Interface Package** button**.**



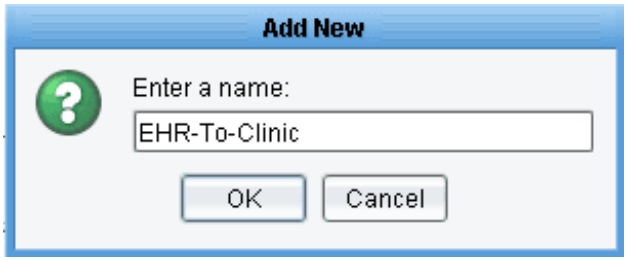
The **Add Interface** dialogue opens**.** We'll name it **"5 Healthcare Getting Started Interface"** for ease of reference. (You can name it anything you'd like.) Click **OK**.



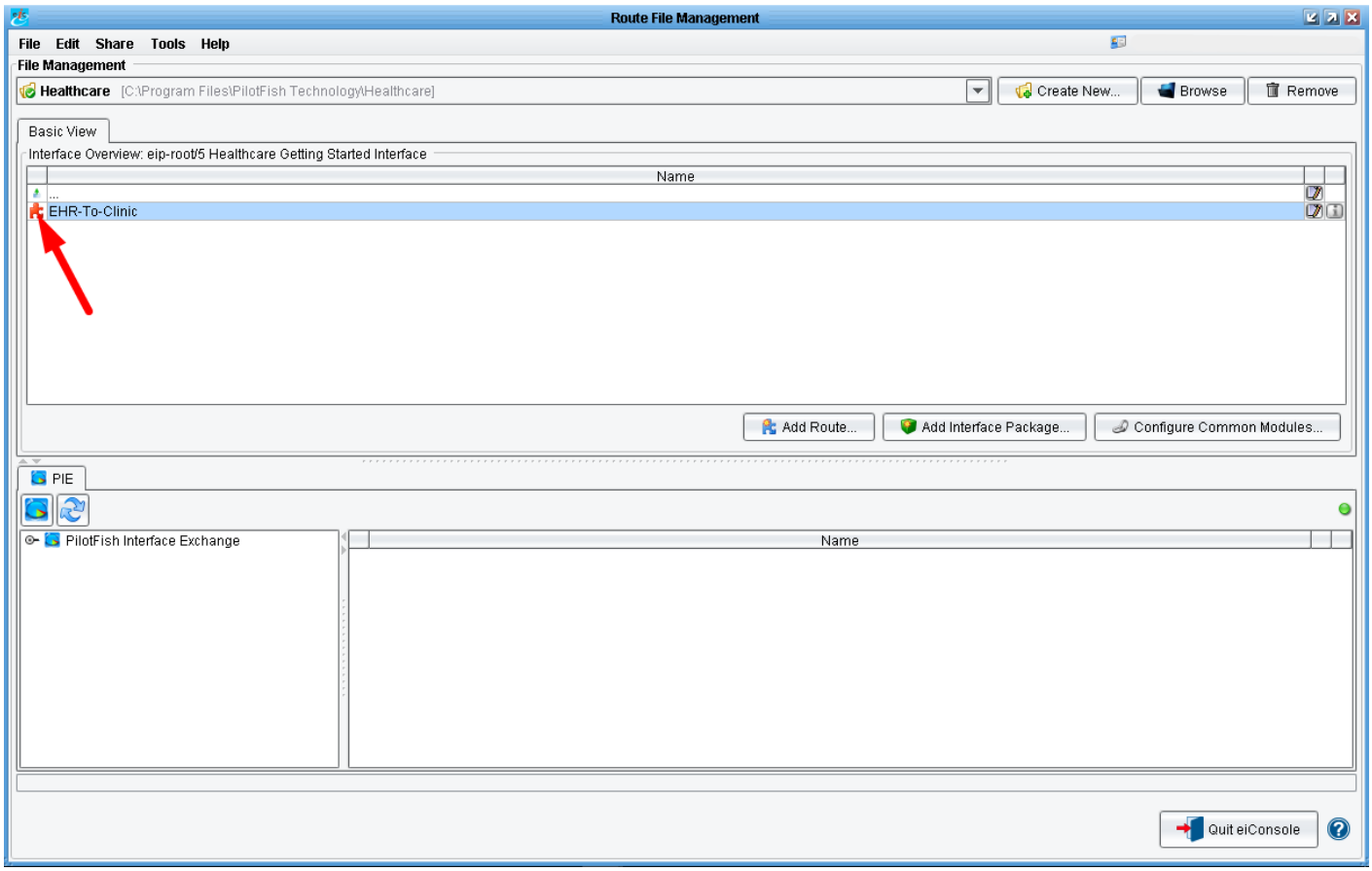
The Route File Management window opens with your new Package. Double click anywhere on the last row to open your new **Interface Package**.



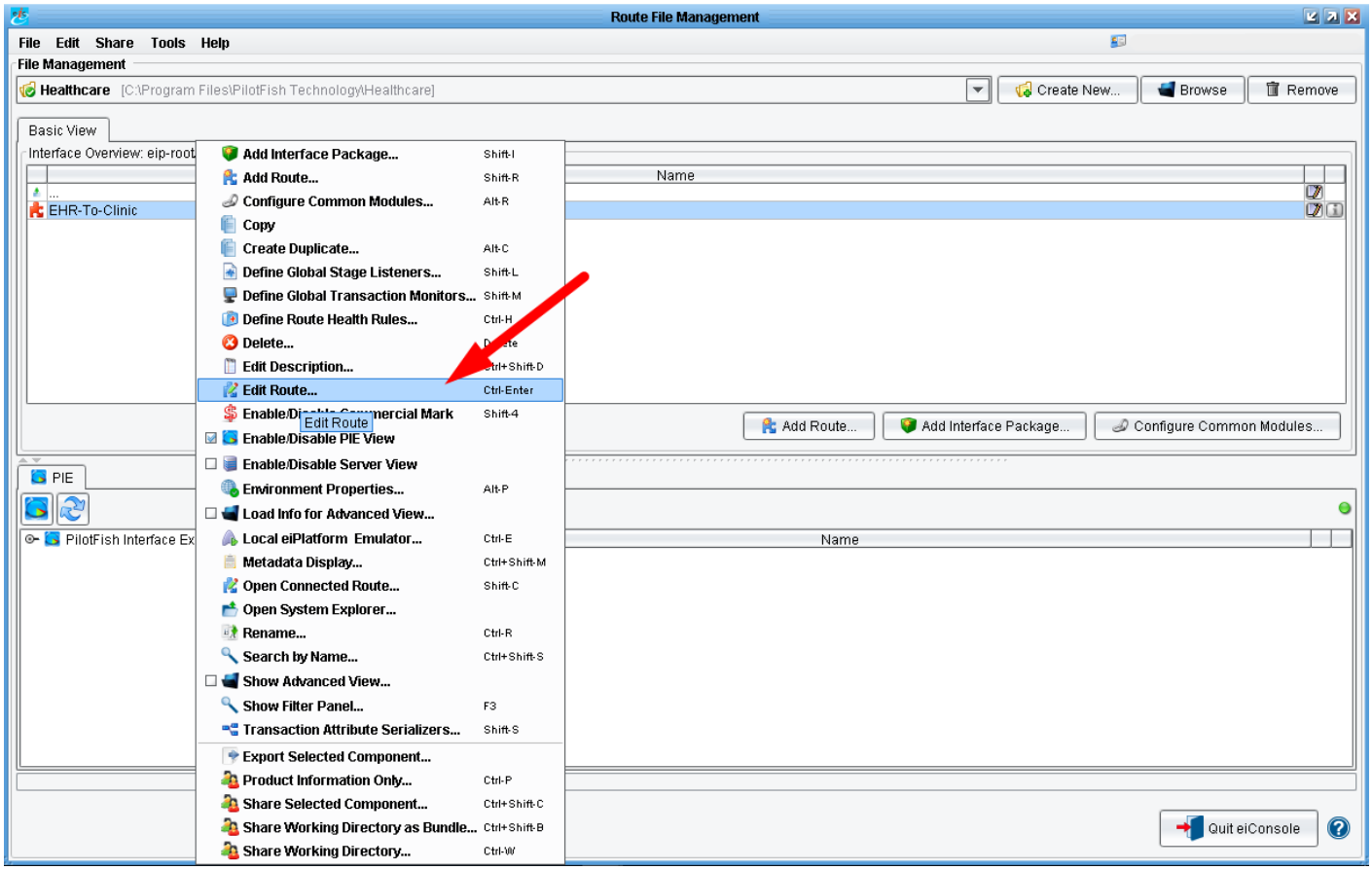
Now that you have your 5 Healthcare Getting Started Interface directory selected, next, we need to add a route. Click the **Add Route** button.



The **Add New** dialogue window will open. To differentiate it from the sample interface EMR-To-Clinic, I'll enter the name **"EHR-To-Clinic"** (you don't need to use hyphens, spaces are OK, too) and click **OK.**



You'll see that a new Route, **"EHR-To-Clinic"**, now appears in the Route File Management grid and it has a red icon next to it (a partially configured interface.) Next, let's look at some options for editing the Route.



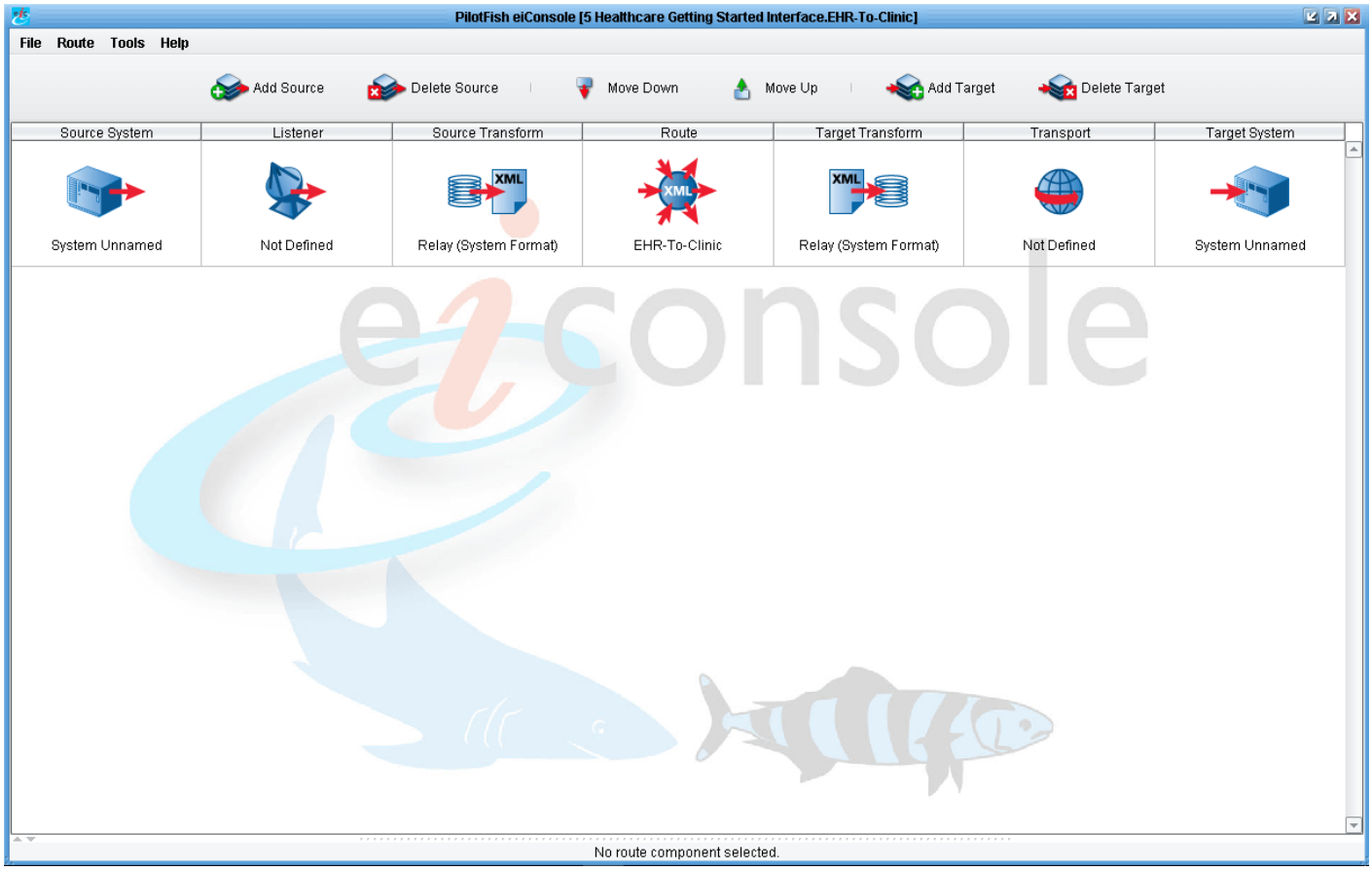
Double click the Route row to open the Main Route grid, or alternatively, you can select the route, right click and then select **Edit Route** from the drop down, and double click.

Note: the available options, for example, if you'd like to copy a route, rename, delete a route, etc.

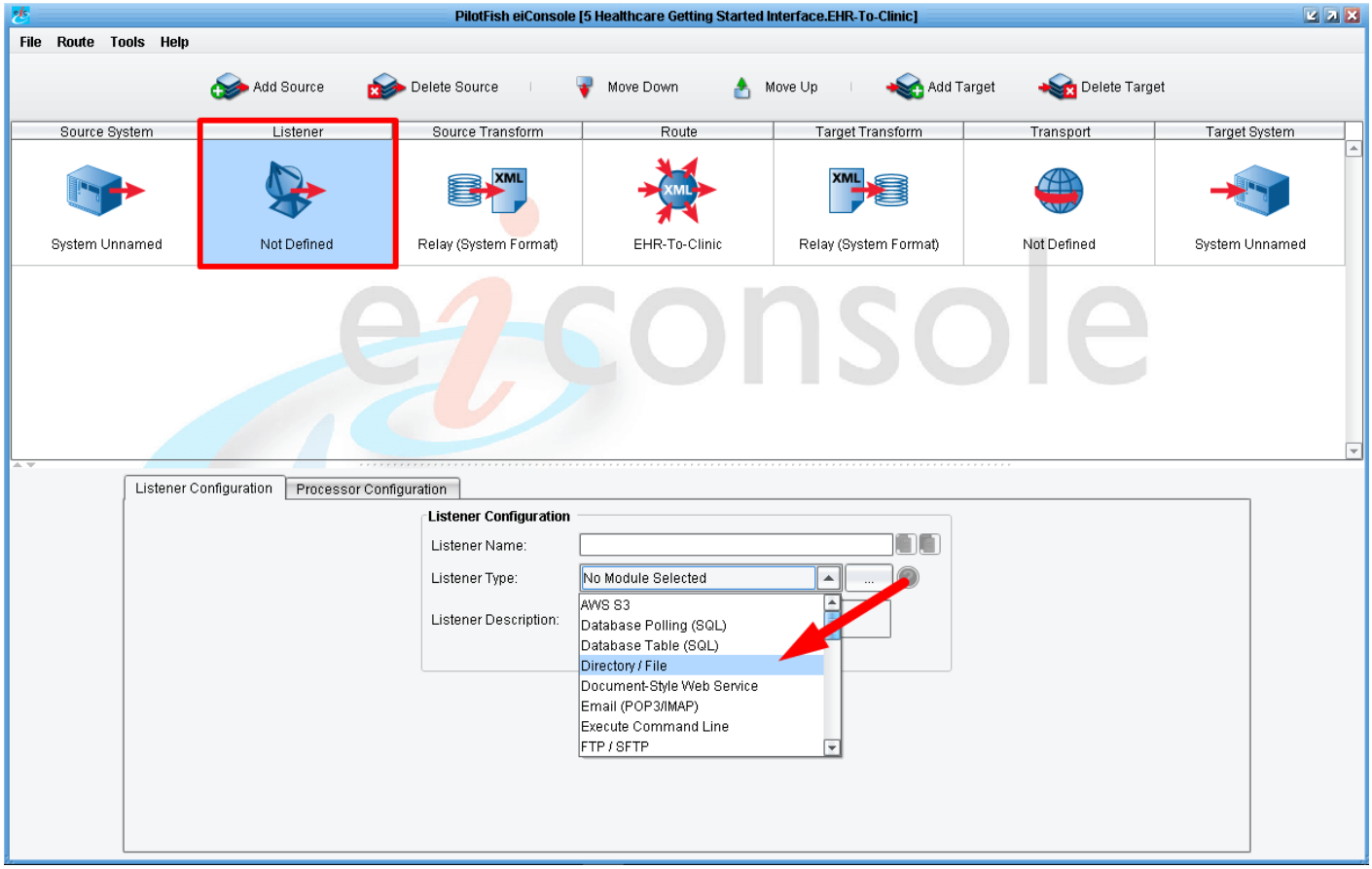
So, open the Main Route grid with whatever method you choose.

The eiConsole's Main Route Grid

The eiConsole's Main Route Grid opens.



Configuring the Listener Stage

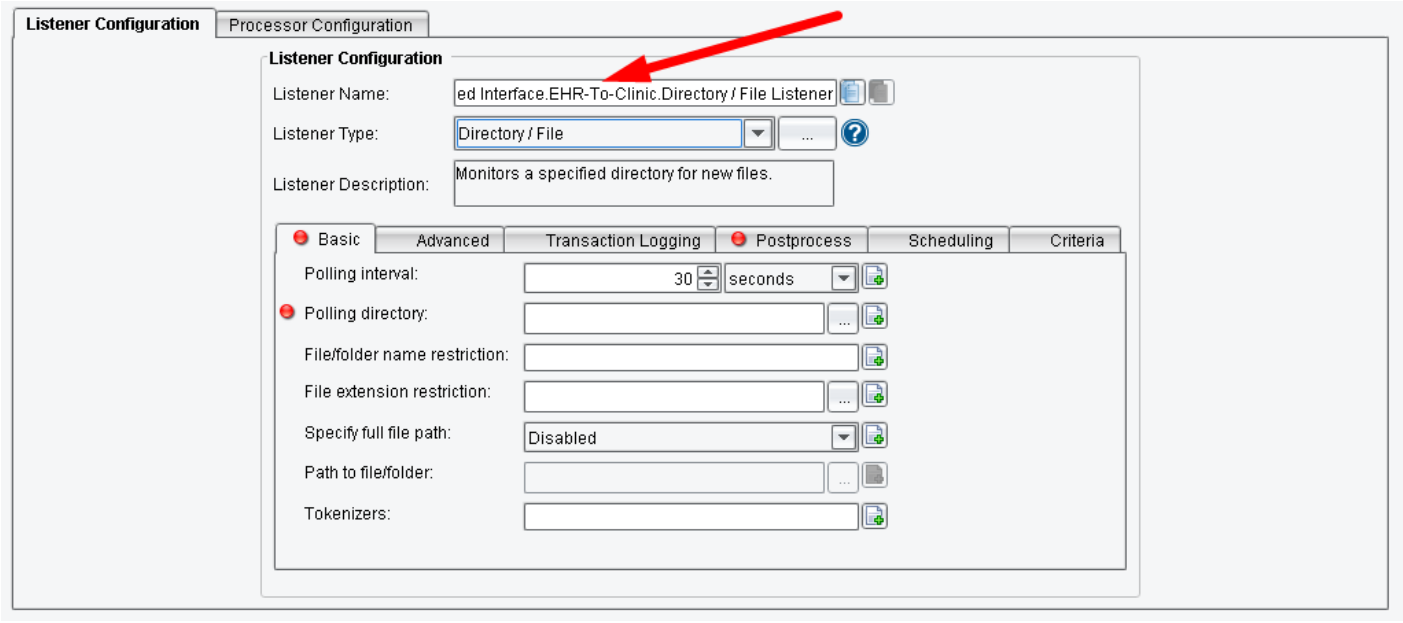


Next, we will configure the Listener. In this interface our Source will be an EHR system producing lab results and sending them out to a Target System, assumed to be an external Clinic or Practice.

We click on the **Listener** stage and then select the Listener Type from the drop down. For simplicity's sake in this tutorial, we'll just go from Directory to Directory, but any number of Listener types are available (scroll through to review the list) and they're all configured through the same pattern. Select the **Directory /**

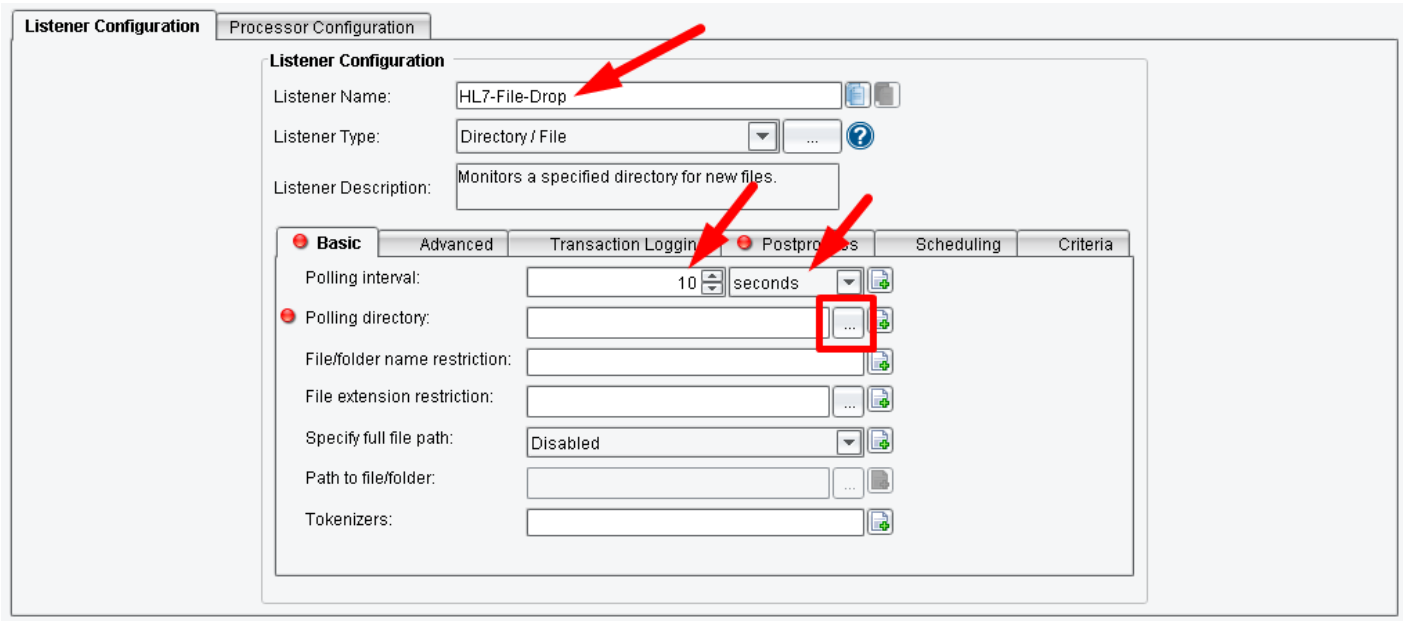
File listener.

Note: when you click on any of the stages specific configuration panels for that stage appear in the bottom half of the window.



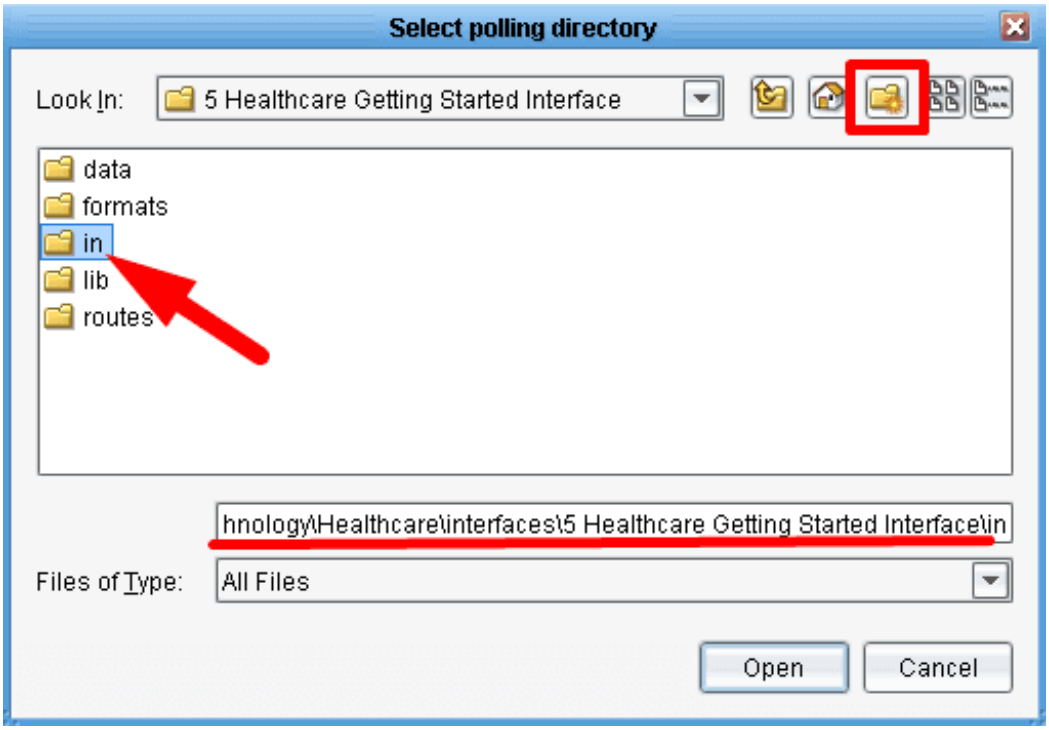
The Listener Configuration panel for the Directory listener opens at the bottom of the window. The red dots indicate the required fields. We'll need to provide the Listener with a name, and fill in the other configuration requirements.

Note: when you select a Listener Type, by default, the eiConsole pre-fills in the Listener Name. We'll want to change it to something more specific to our interface, though.



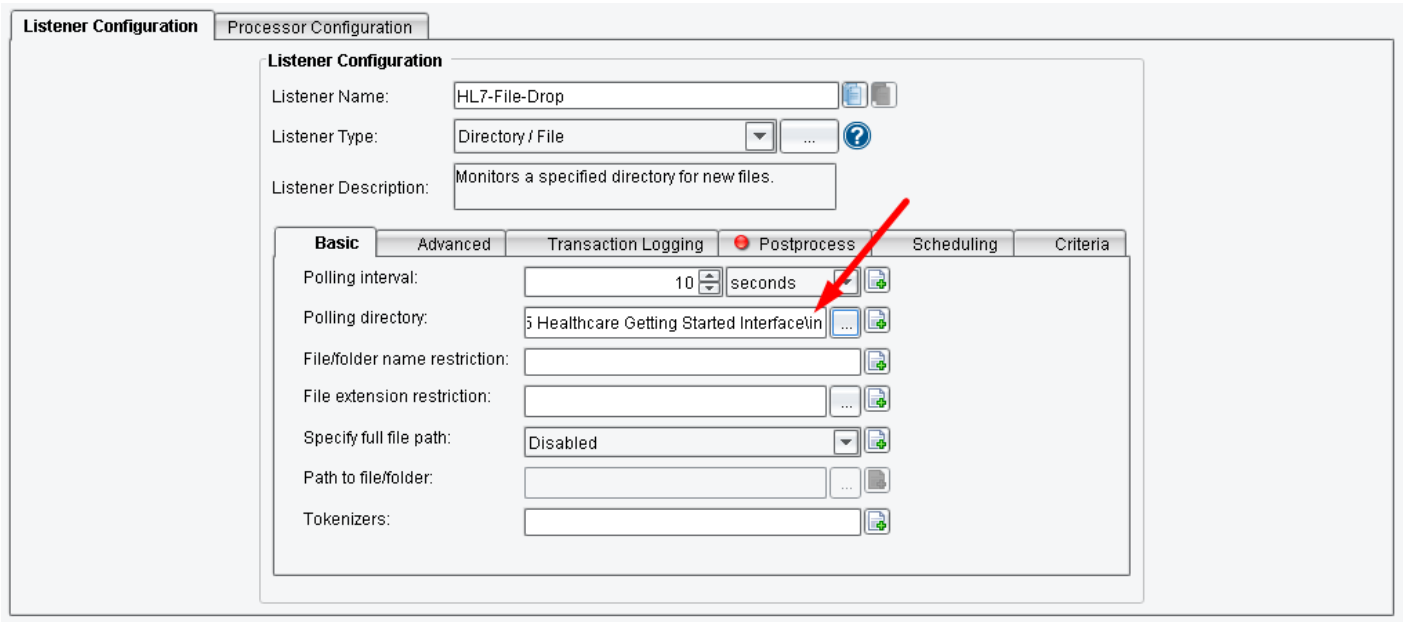
Let's call this Listener the **"HL7-File-Drop"**. Type in the new Listener Name. We'll set the Polling interval to **"10"** seconds, meaning we'll poll the directory that we choose every 10 seconds for new files. *(You can select from any number of polling interval options using the drop down menu.)*

Next, click **Ellipsis** button to select your polling directory.

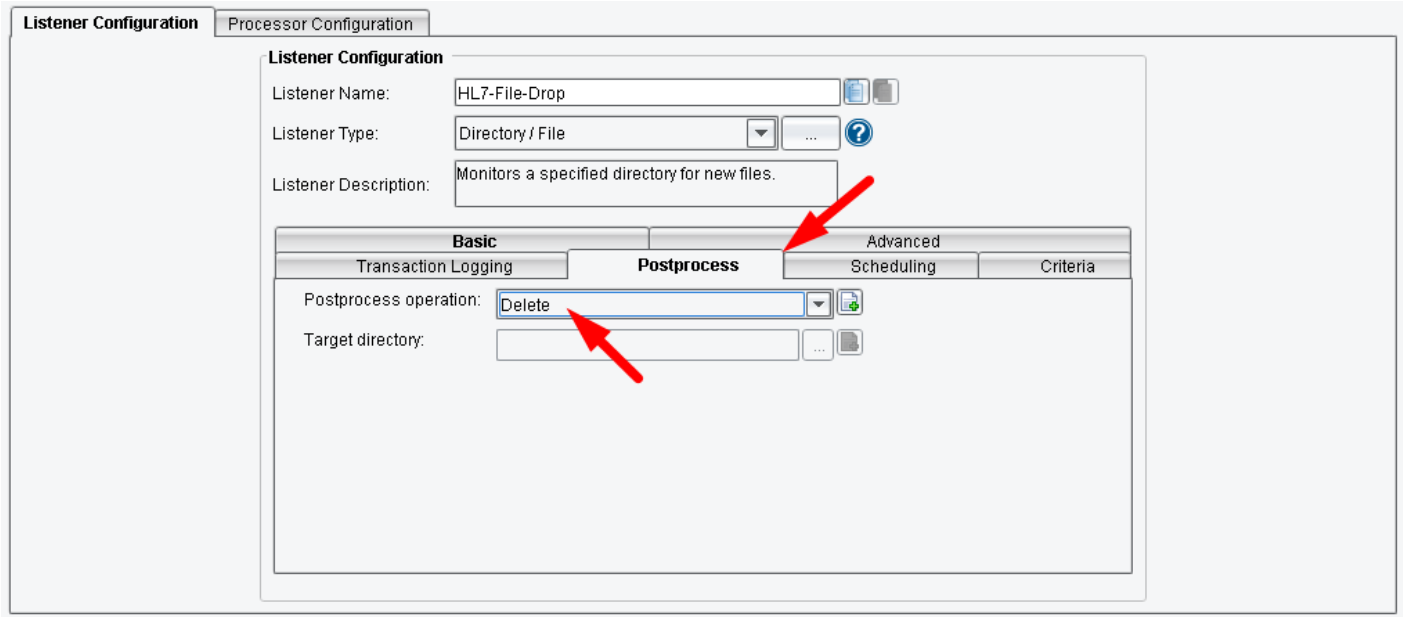


Note: any location on your computer can be selected as a polling directory.

For consistency in this tutorial select /Healthcare/interfaces/**5 Healthcare Getting Started Interface**. Click the **New Folder** button and name the folder **"in"**. Click return/enter. Then click **Open**.



This will set the Polling Directory to the **"in"** folder.

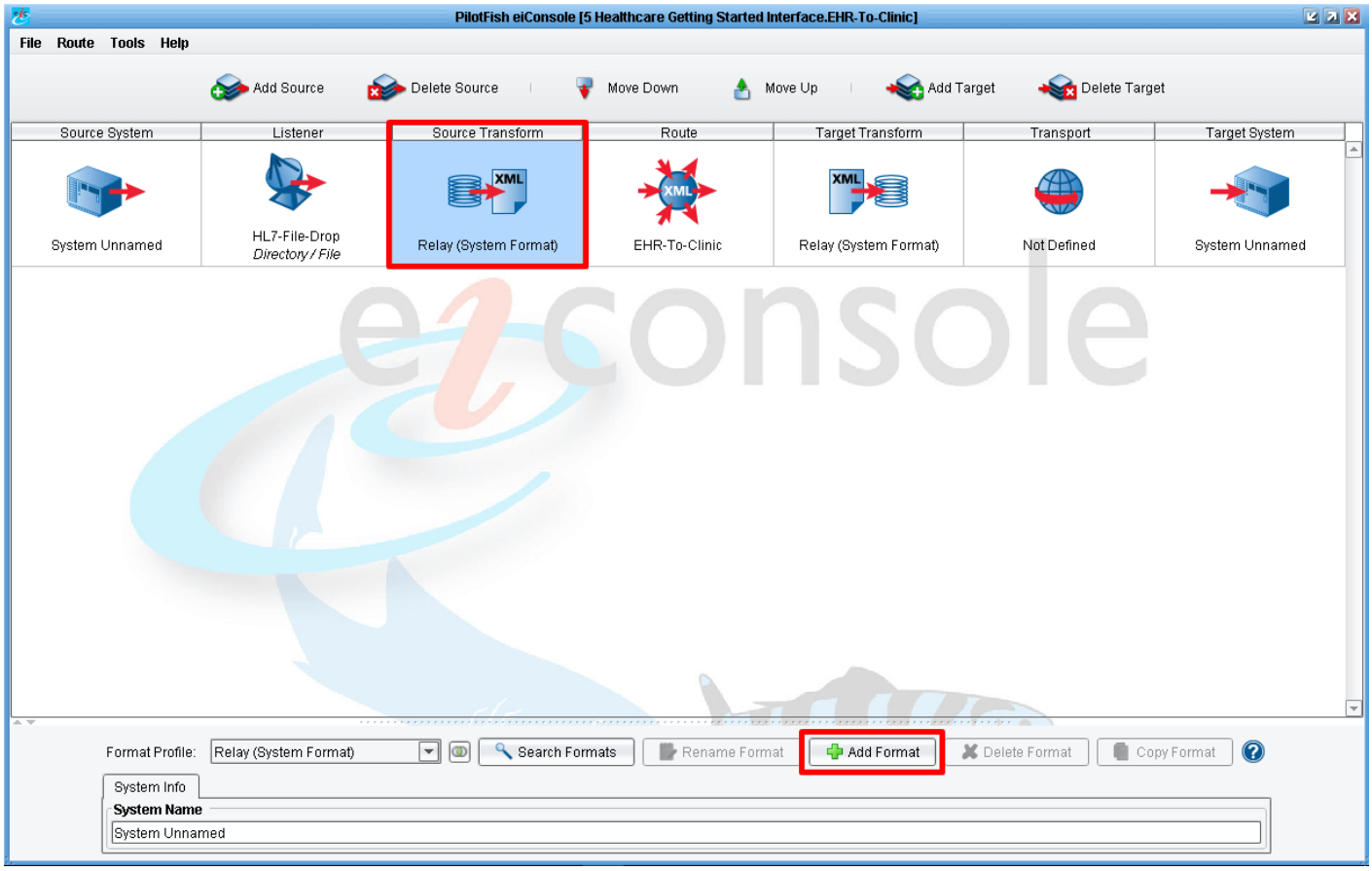


Set the Postprocess operation.

Note: as you fill in the configuration items marked with the red dots, all of the red dots are gone, indicating that all the required fields have been filled out.

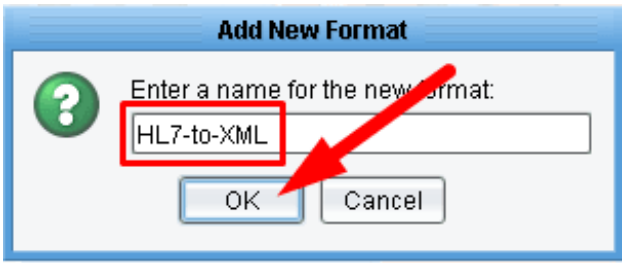
With the Listener stage configured we'll move on to the next stage, the Source Transform.

Configuring the Source Transform Stage



Next, click the **Source Transform** stage.

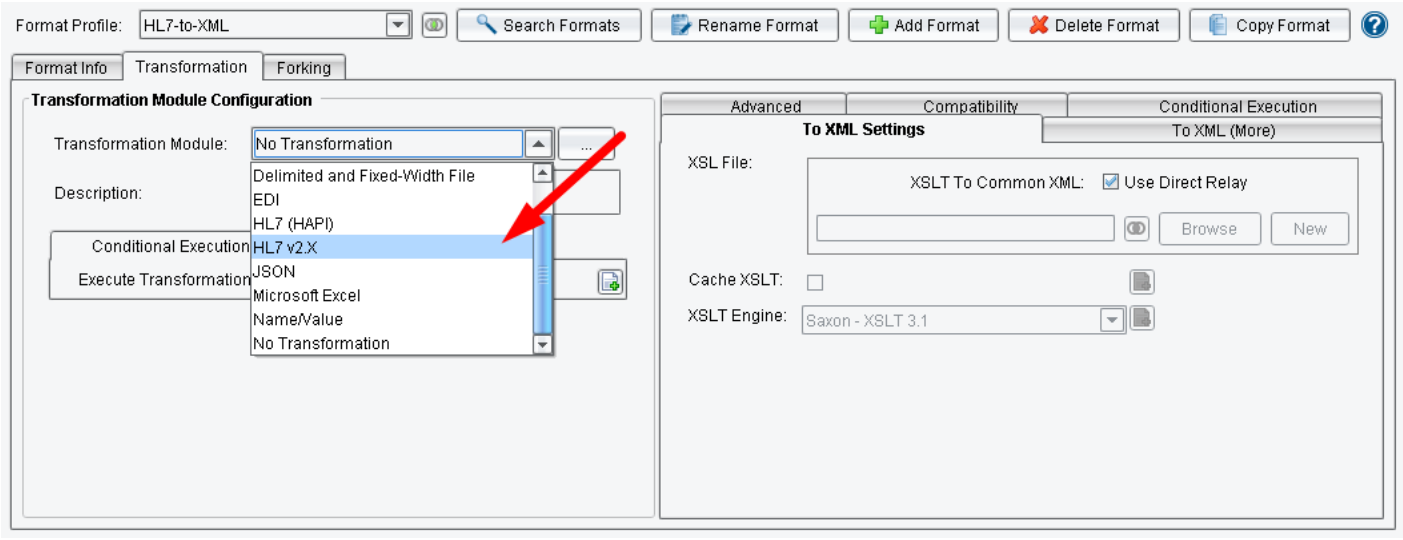
Here we'll want to add a Format or Data Transformation that will convert an incoming HL7 message to XML. Click the **Add Format** button.



This opens the Add New Format dialog window.

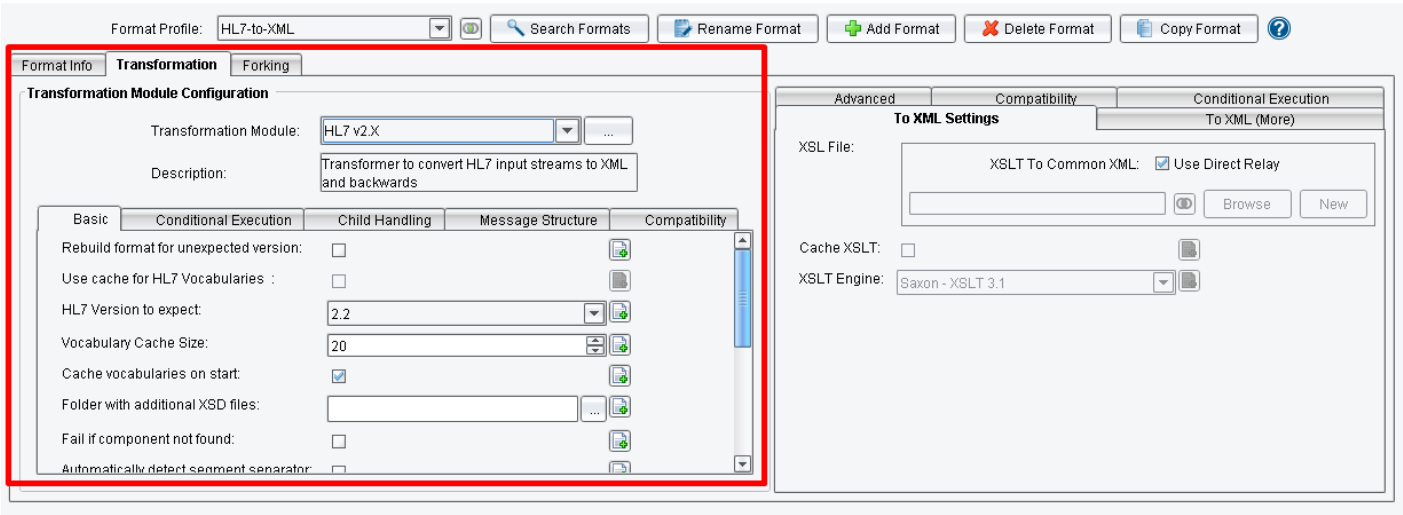
This format will translate the inbound HL7 into an XML representation, so we'll call it **"HL7-to-XML"**. Enter the name and click **OK**.

Note: as you might expect, when you create a new format the format gets added to your Working Directory formats folder.

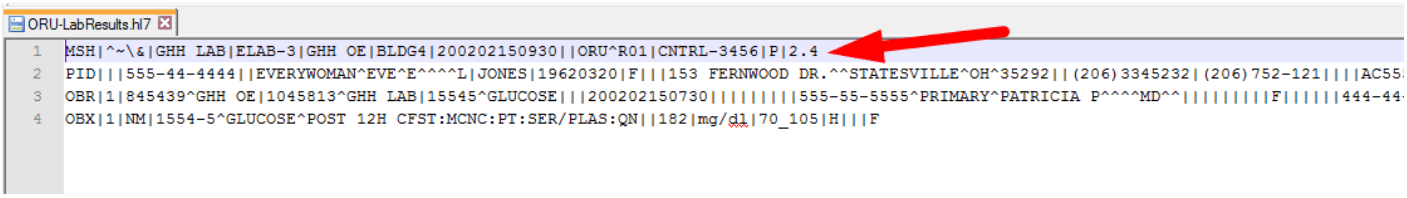


The Transformation Configuration panel will appear, including both the Transformation Module and XSLT Configuration. Transformation Modules are used to parse data from non-XML formats into an XML representation. Whereas XSLT, and the eiConsole's Data Mapper, are used for the logical mapping of that format onto another.

In the Transformation Module drop down choose the **HL7 v2.X** transformer. This indicates that we expect HL7 2.x input that we'll parse to XML.

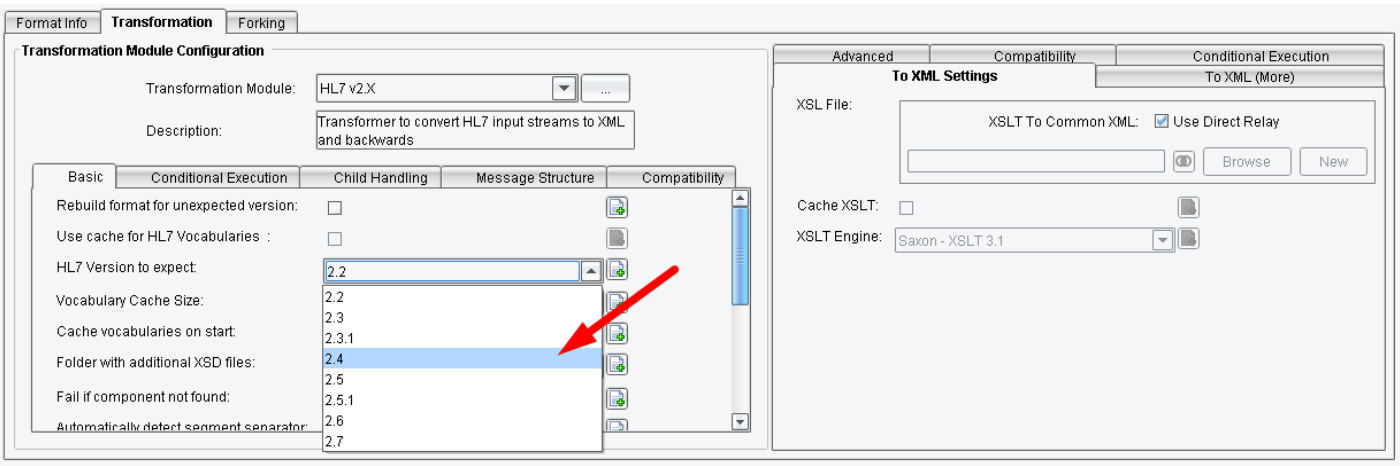


Once we have selected the HL7 2.x Transformer, a new configuration panel opens. But before we configure it we have another step.



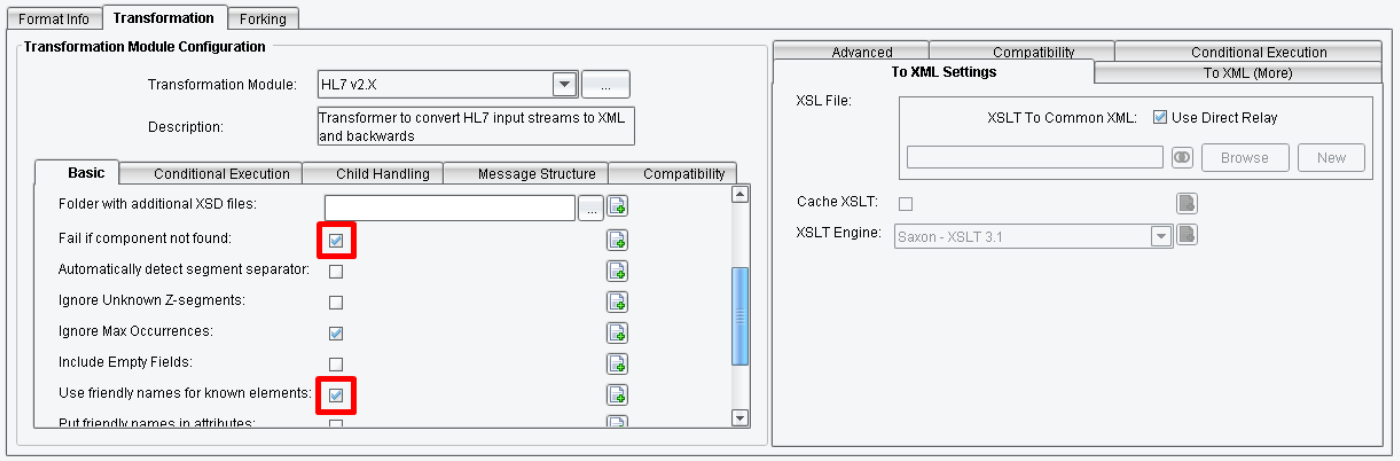
Let's take a look in the data folder **(in the your current {working directory}/interfaces/2 Healthcare Getting Started Tutorial/data.)** You'll see the **ORU-LabResults.hl7 file**. Open it. This is the input we'll expect.

Here you can see a rather typical HL7 transaction. And as you can see the transaction version is **2.4**.



So, in the **HL7 Version to expect** field, select **2.4** from the drop down.

Note: that even if the incoming data is not HL7 2.4, our lenient HL7 parser will do its best to parse the data into a usable XML representation.



There are a few other options we'll want to check off. We'll select the **Fail if component not found** and the **Use friendly names for known elements** check boxes.

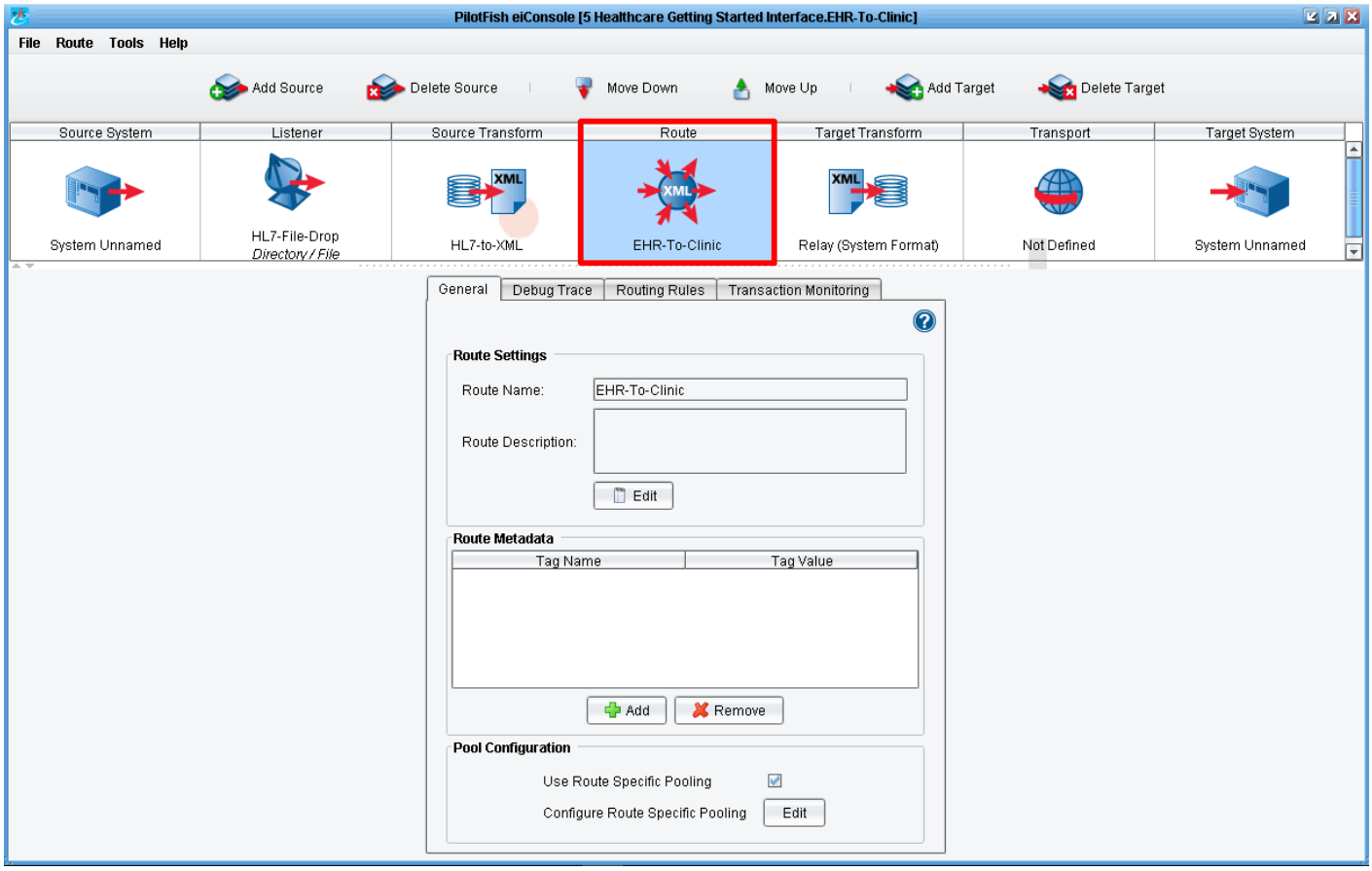
Note: the eiConsole for Healthcare's Friendly Name option replaces cryptic HL7 names with simple, understandable synonyms derived from field descriptions in the appropriate version of the HL7 vocabulary. This is a very useful feature, especially if you are new to HL7 and not that fluent in the HL7 syntax.

We'll want to make sure the **HL7default namespace in output** box is unchecked.

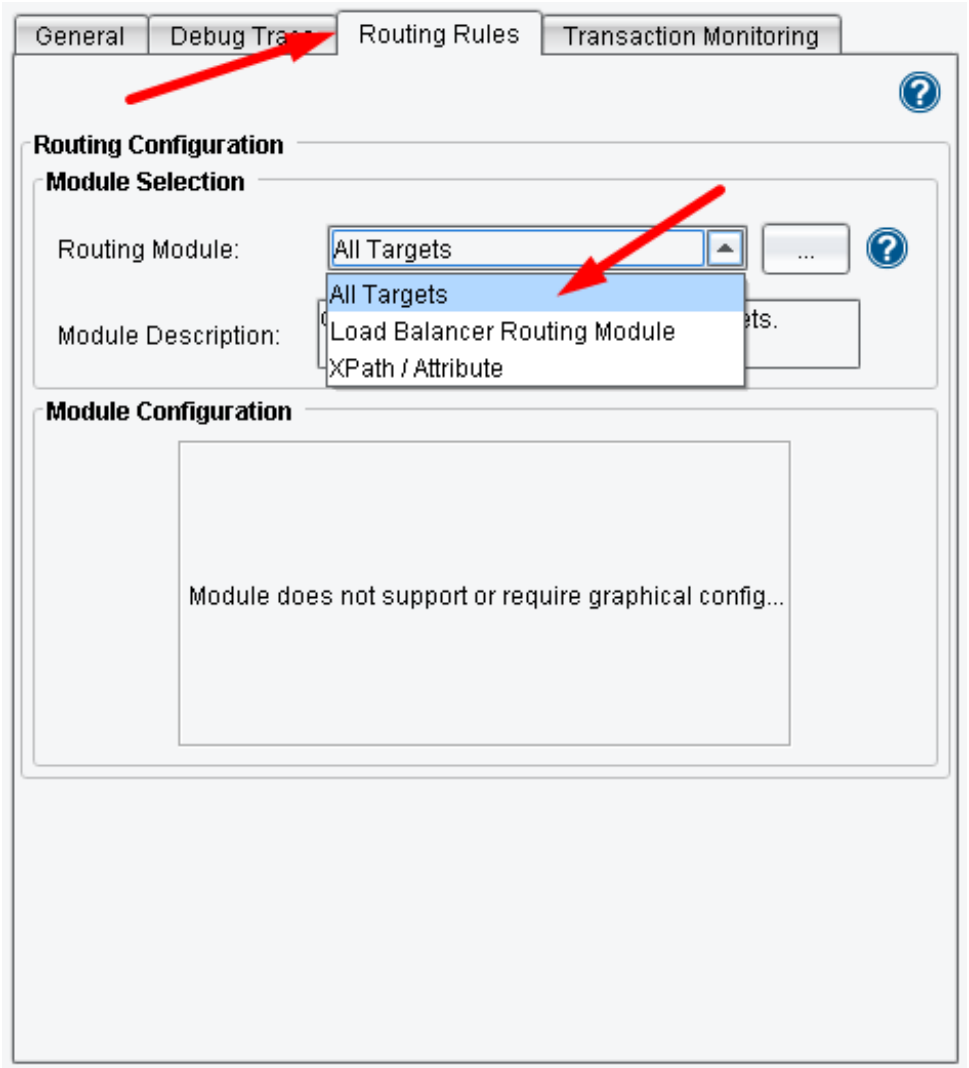
And we'll also leave unchecked the **Put Friendly Names in Attributes** (this allows you to have friendly names associated with elements but not in the element name) and **Use HL7 default namespace in output.**

Note: the eiConsole can take any HL7 message, including ones that aren't quite compliant with the standard, and convert it into an XML representation.

The Route Stage

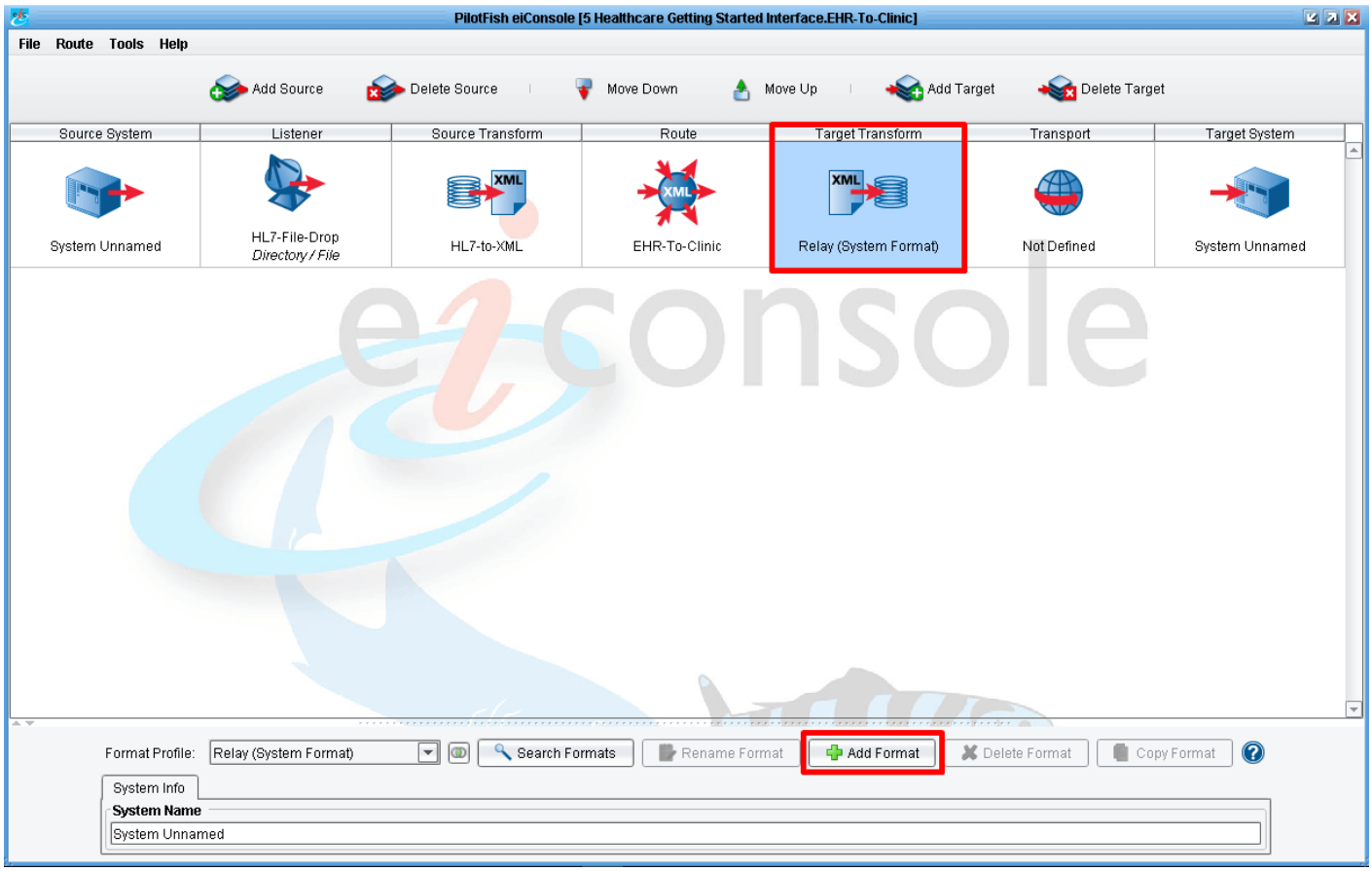


Next, click the **Route** stage. Once again when you do, you'll see a configuration panel open in the bottom half of the screen.

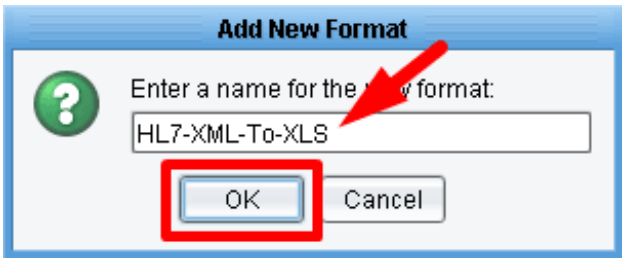


Then select the **Routing Rules** tab. Click the drop down. The Routing Module lets you select from All Targets and XPath / Attribute. But in our example, we have only one Target system, and we can leave the default setting of **All Targets** selected.

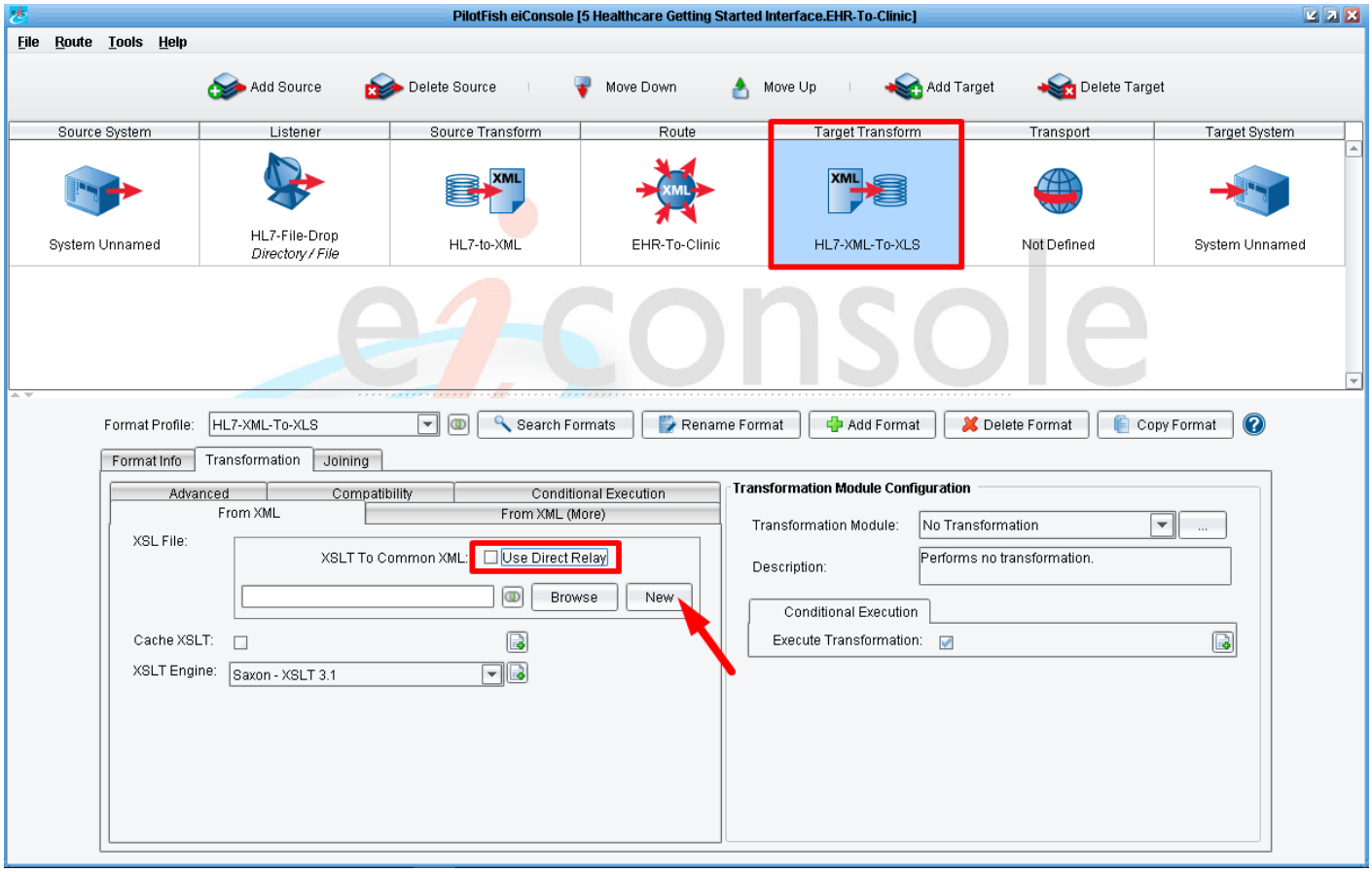
Configuring the Target Transform Stage



Select the **Target Transform** stage. Here we'll want to map our new XML representation of HL7 onto an Excel spreadsheet. Click the **Add Format** button.



In the **Add New Format** dialog, enter **"HL7-XML-To-XLS"** and click **OK**.



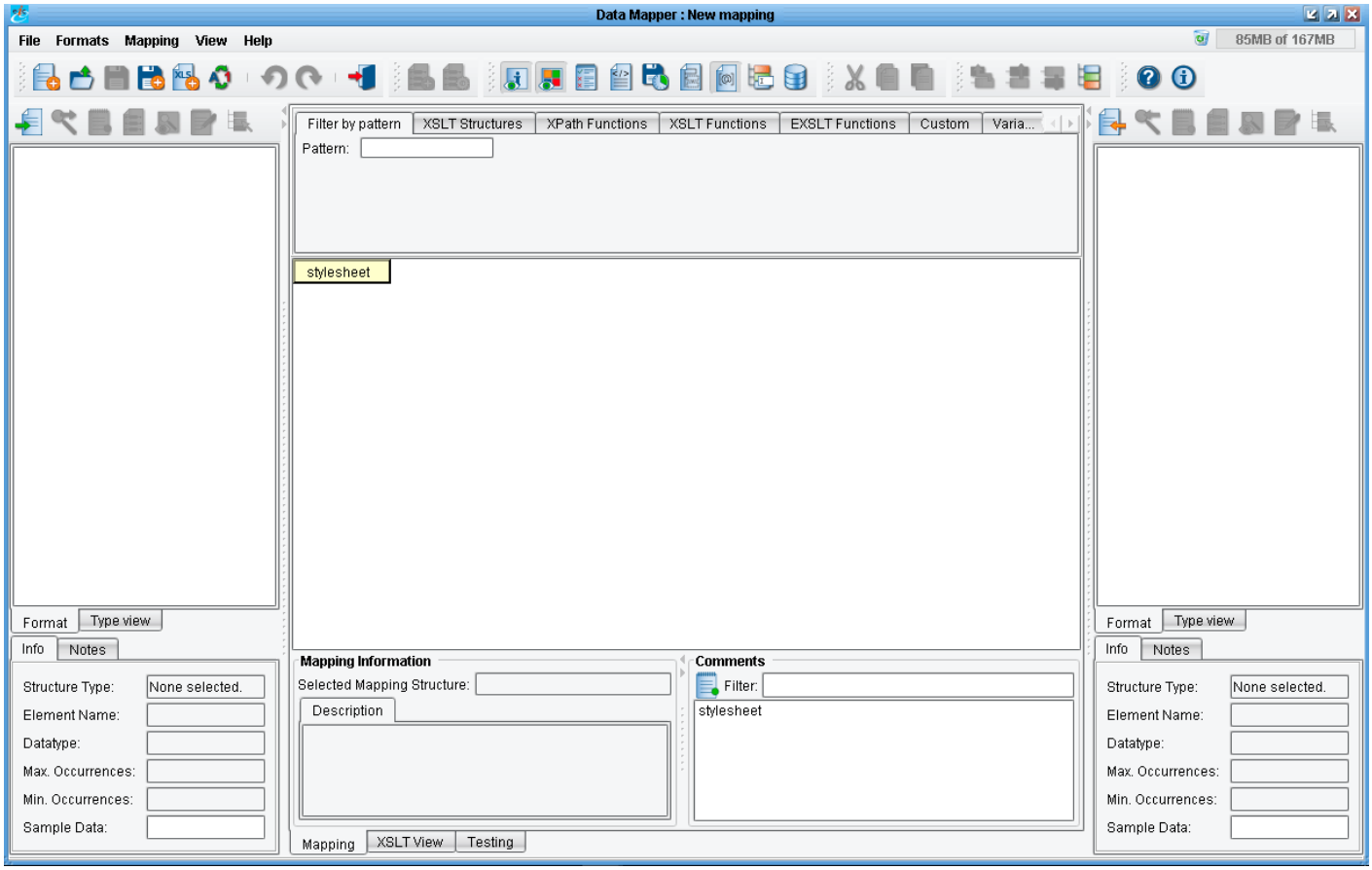
When you click the Target Transform stage again you'll see again that a new configuration panel appears in the bottom half of the window. We have a Transformation Configuration tab, but in this case the XSLT Configuration, the logical mapping piece is first, followed by a Transformation Module, that we'll use to bind the data to the appropriate output format.

The XSLT Configuration panel on the right hand side is used when we want to apply a logical mapping to the XML that's an output of the first part. We'll be using XSLT later, so we'll leave this alone for now and we'll leave the default **Xalan Interpreted** drop down menu item as is.

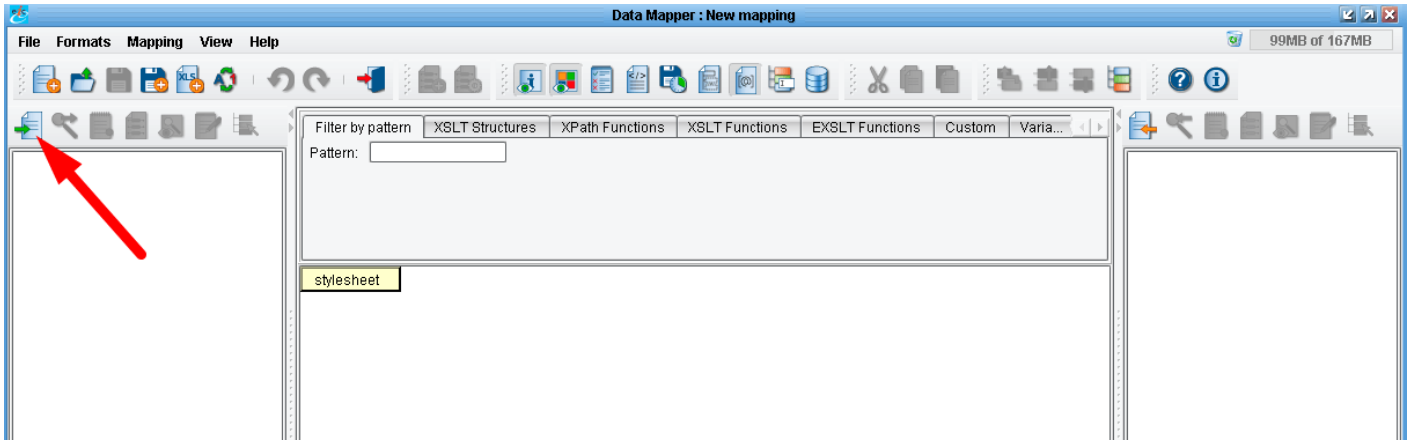
First, we'll do our logical mapping.

Make sure the **Use Direct Relay** box, in the XLST configuration panel, is unchecked and make sure the Transformation Module is set to **No Transformation**. Then click the **New** button.

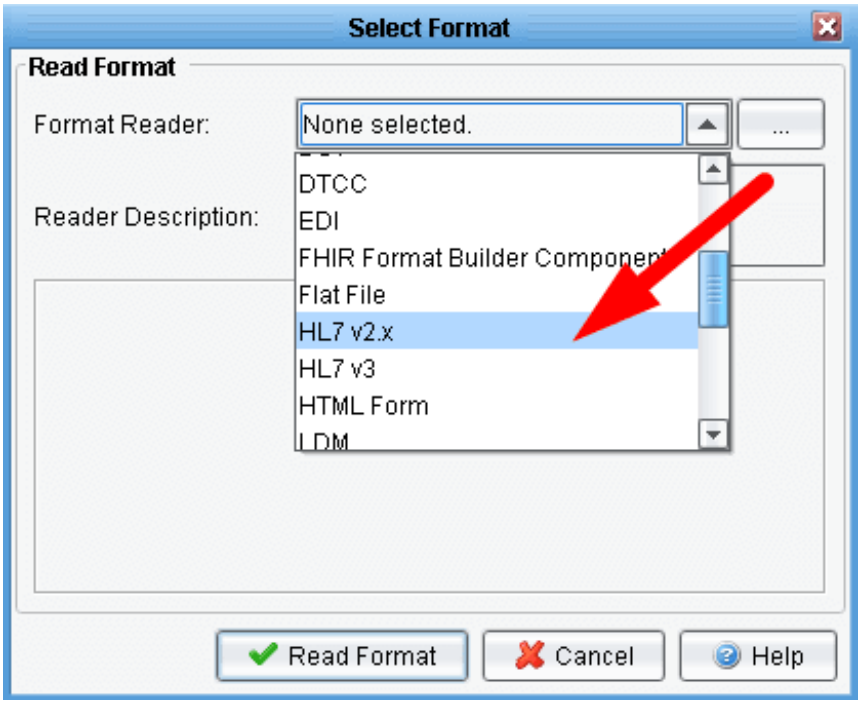
The Source Transform Stage - the Data Mapper



This launches the eiConsole's Data Mapper. The Data Mapper is where we perform logical mapping between any two data formats. It has three panels, a panel on the left for our Source format, a panel on the right for our Target format, and a panel in the middle which will contain the logical mapping between the two. To start, we'll need to load our Source format and Target format, and then create the relationships between the two.

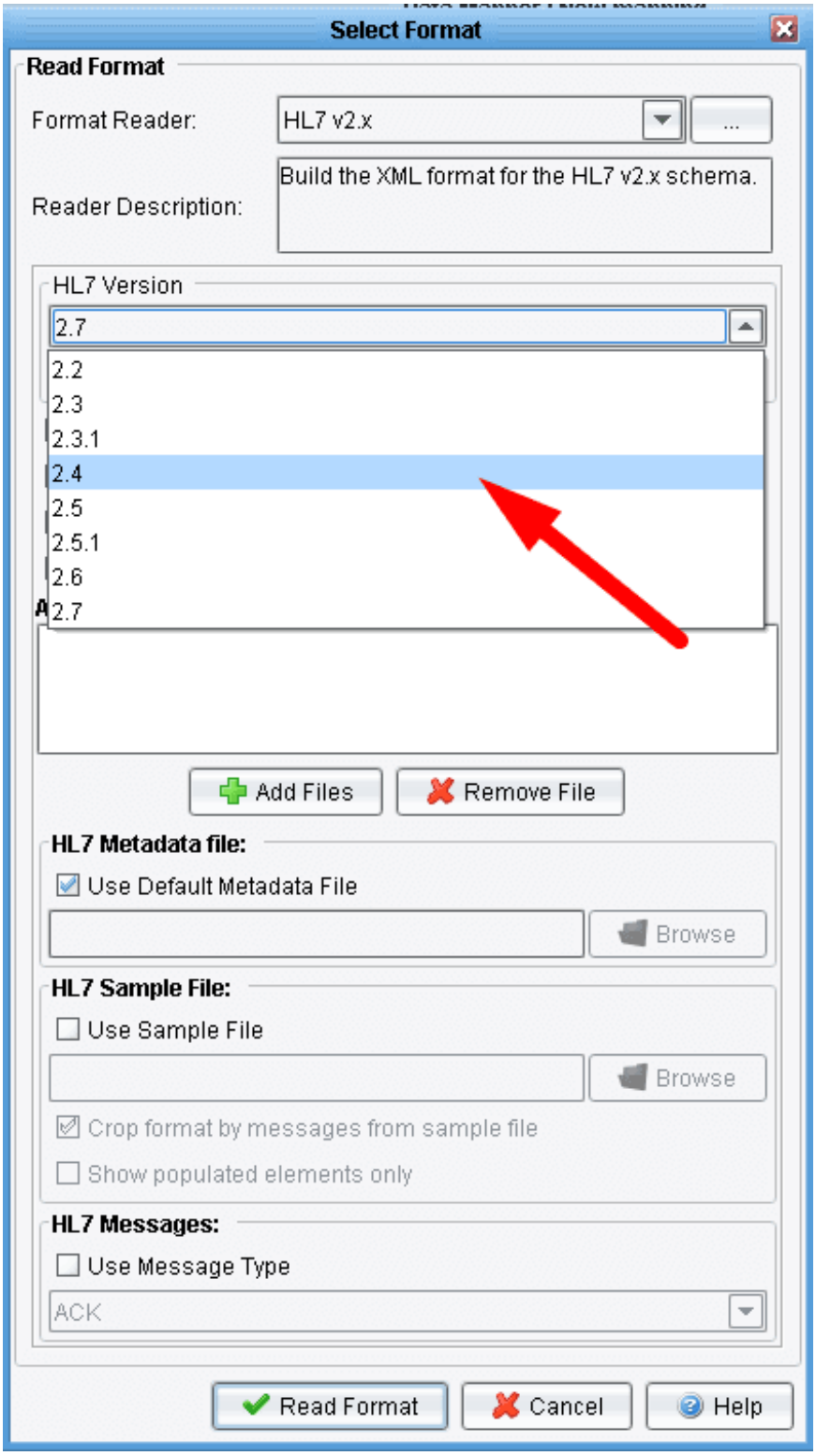


First, we'll load the Metadata for our Source and Target. To do this, click the **Open source format** button.

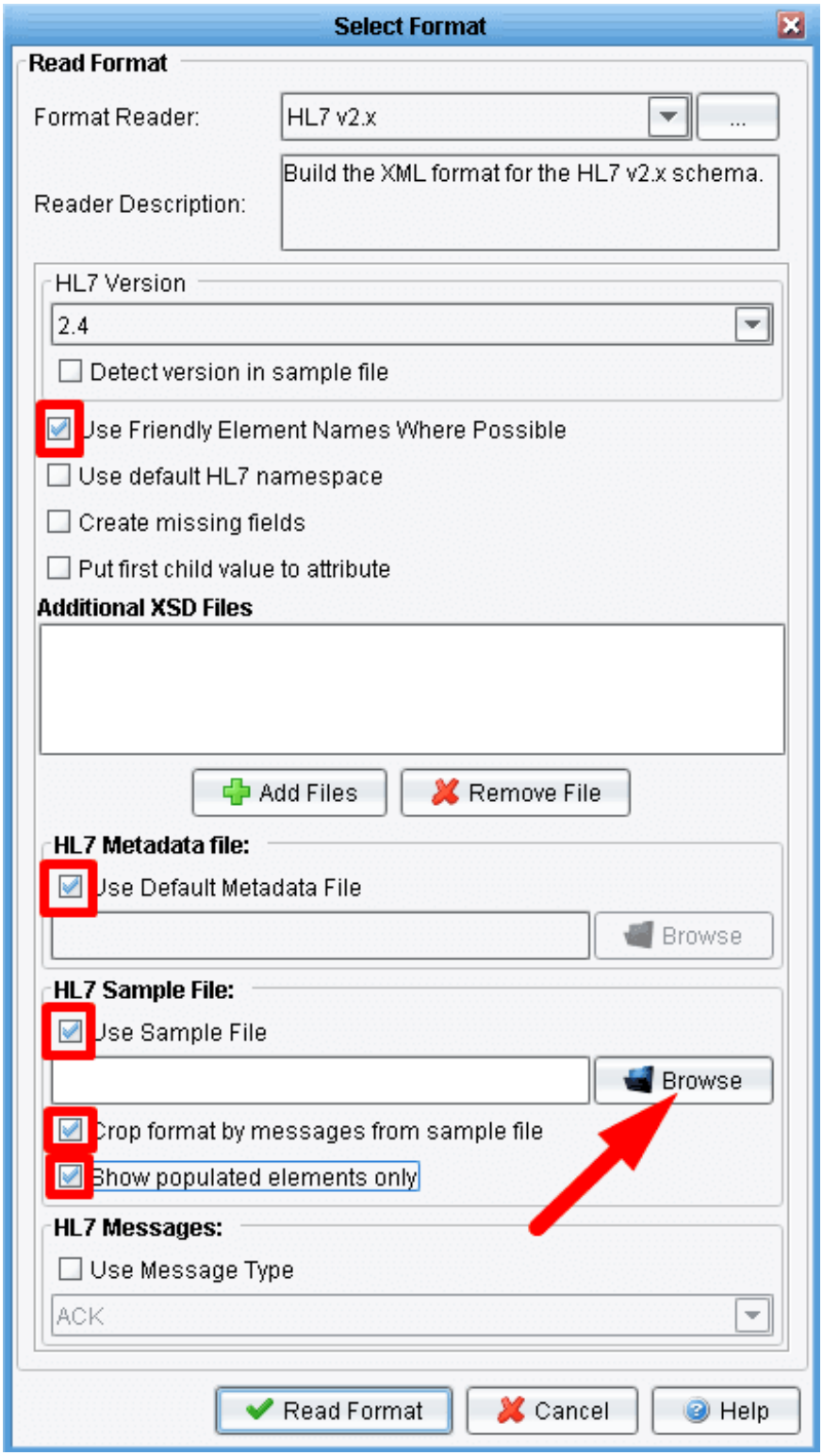


When the Select Format dialogue appears, scroll down and select the **HL7 v2.x** format reader from the drop down.

Note: the options here for directly reading in SQXML, HL7 3.x and DICOM.



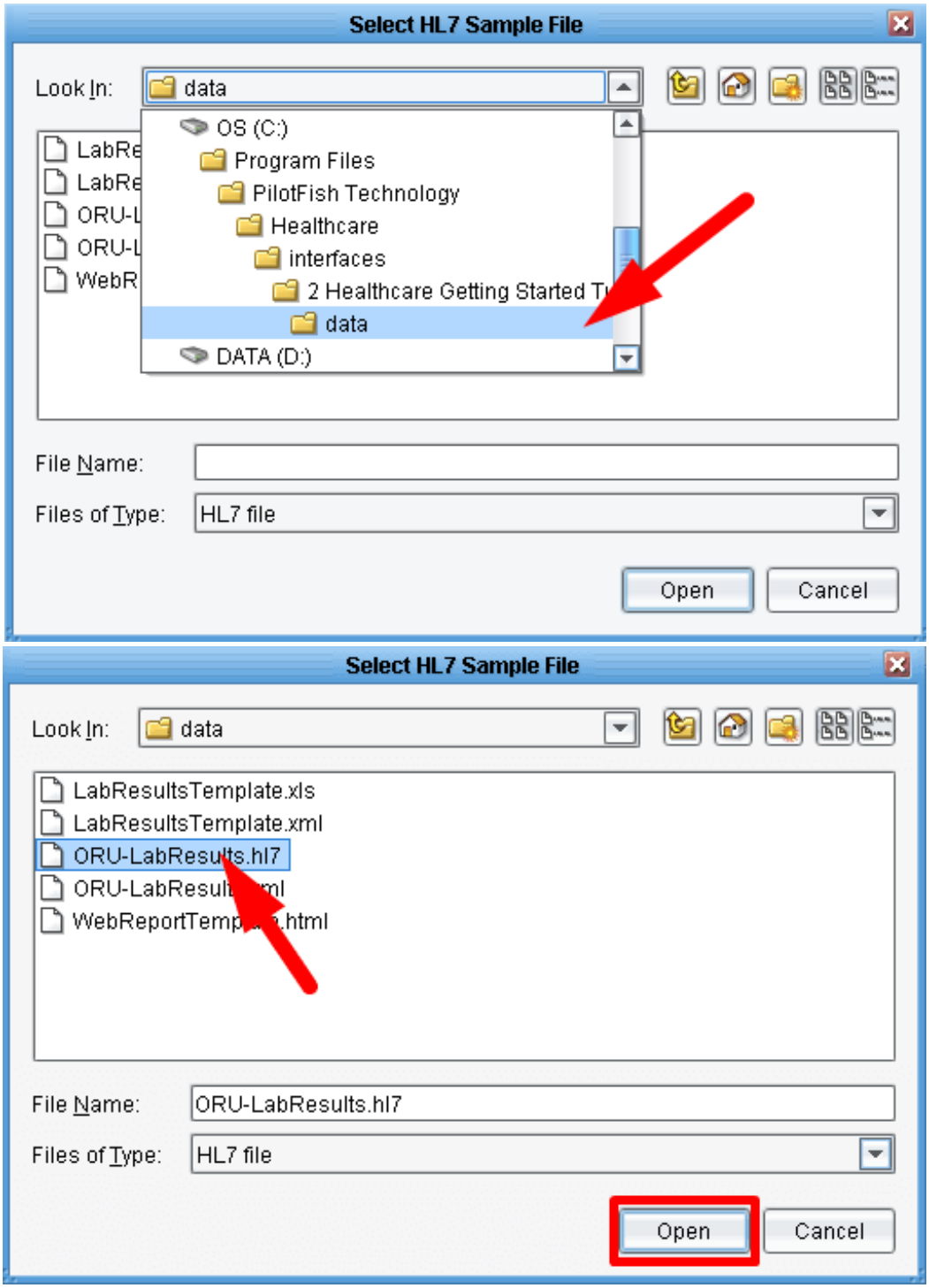
When the **Select Format** dialog window appears choose HL7 Version **2.4**, from the drop down.



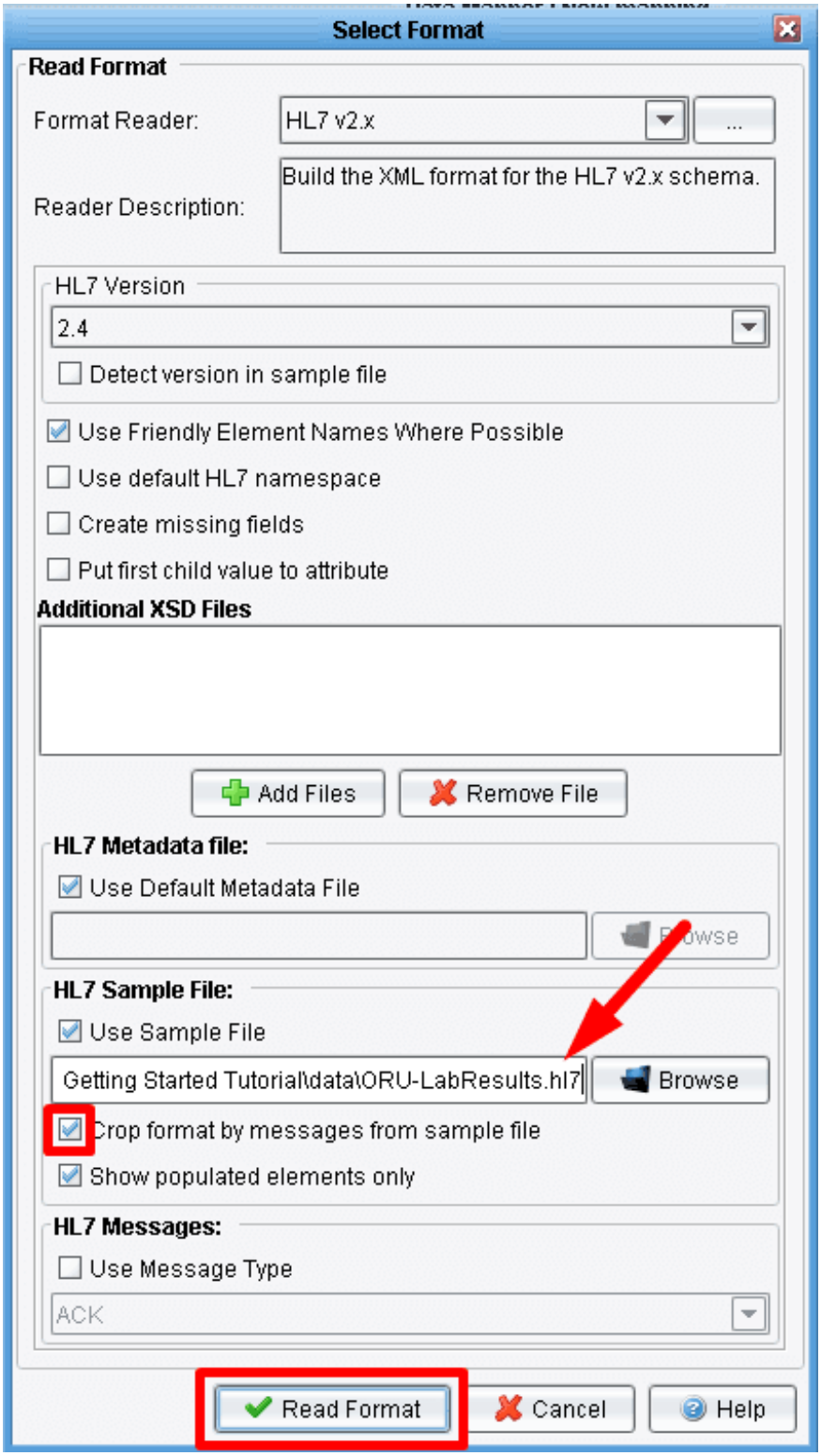
Check the boxes for **Use Friendly Element Names Where Possible**, **Use Default Metadata File, Use Sample File**, **Crop format by messages from sample file** and **Show populated elements only**.

Leave unchecked **Use Default HL7 namespace** and **Use Message Type**.

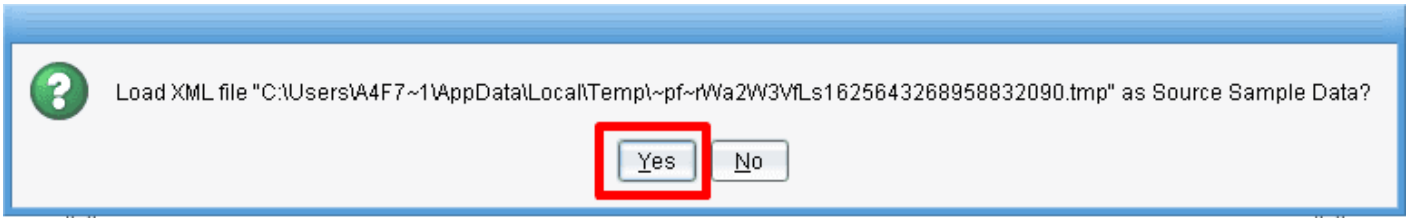
Then click the Use Sample File **Browse** button.



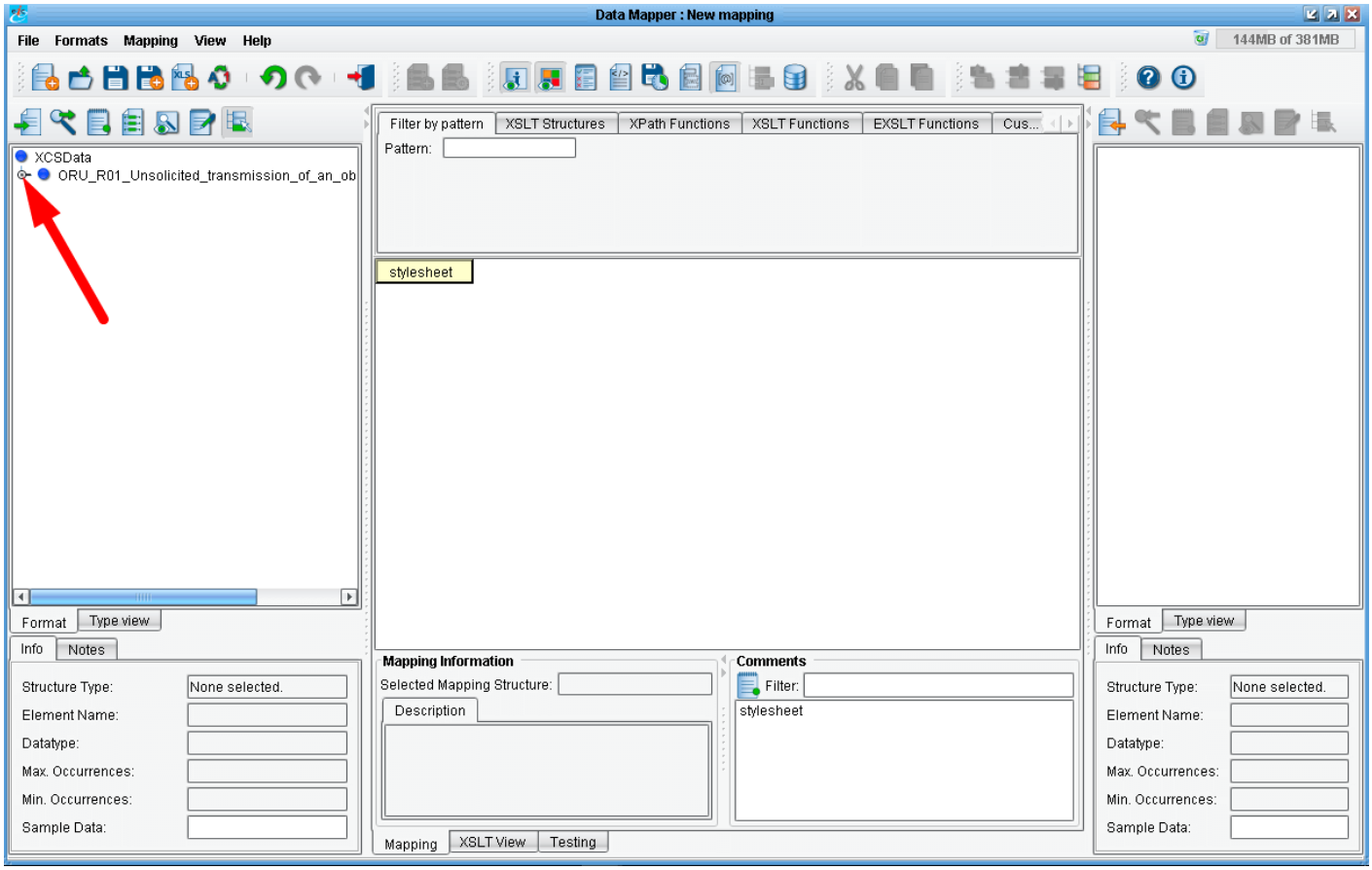
When the window opens, navigate to your distribution folder, and the **data** folder, then select the HL7 sample **ORU-LabResults.hl7** and click **Open.**



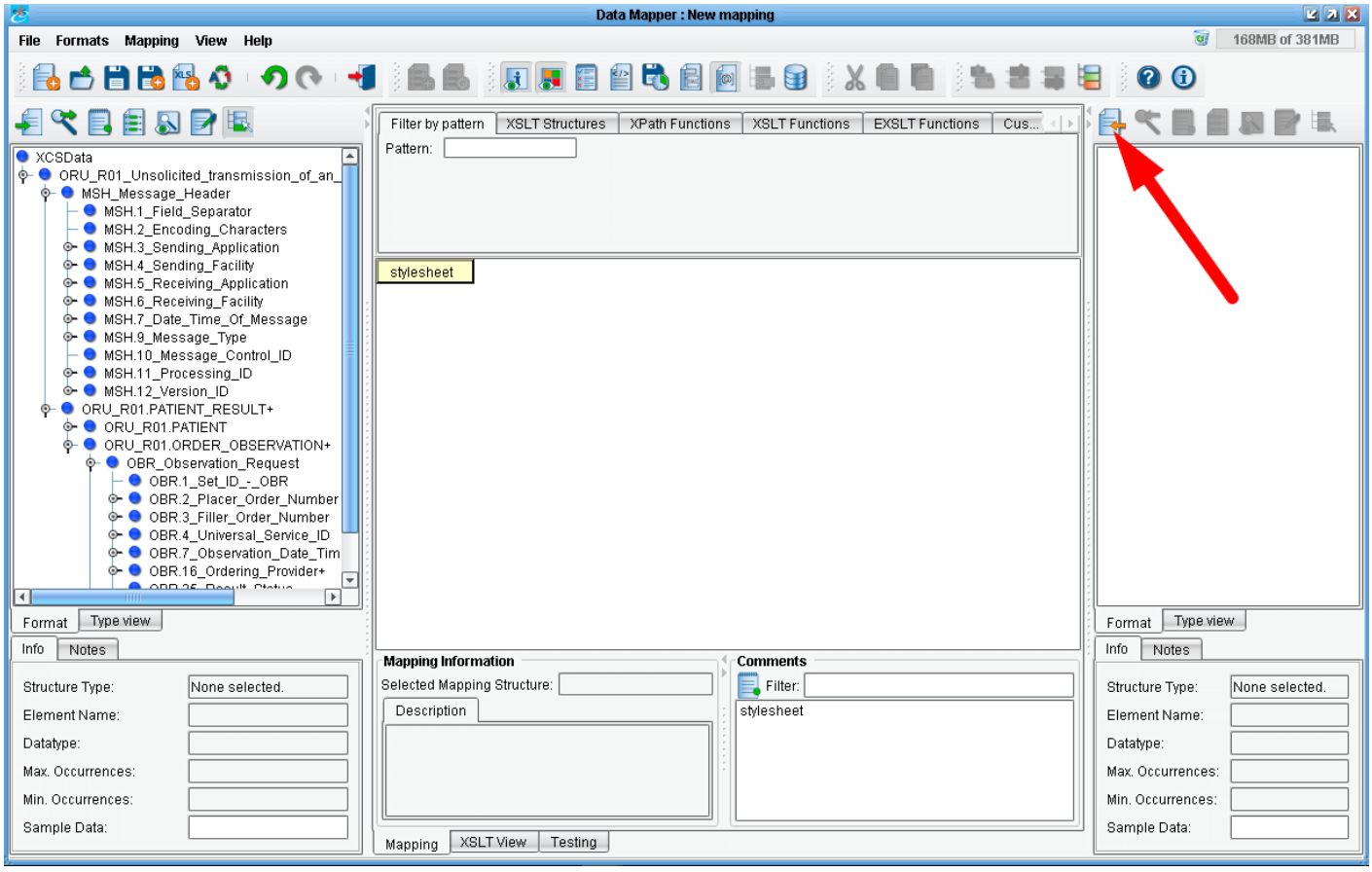
The ORU file now appears in the Sample File window. (Make sure you checked the **Crop format by messages from sample file** box.) Click **Read Format.**



You'll be prompted to Load an XML file as Source Sample Data and click **Yes**.

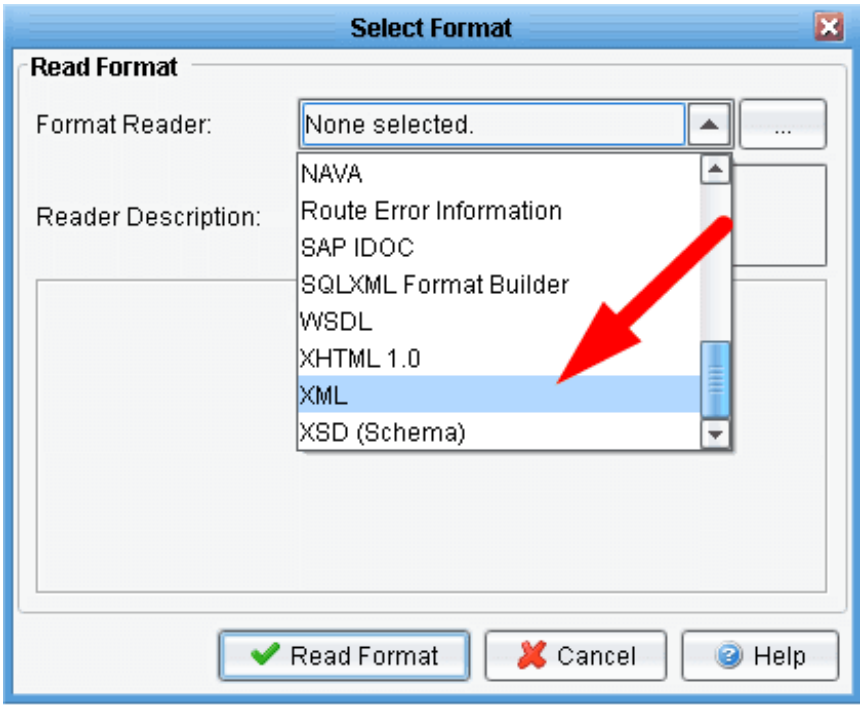


It will take a few seconds to process, then you'll see the full HL7 2.4 data dictionary appear in the left hand panel of the data mapper. Double click on the nodes to fully expand the tree.

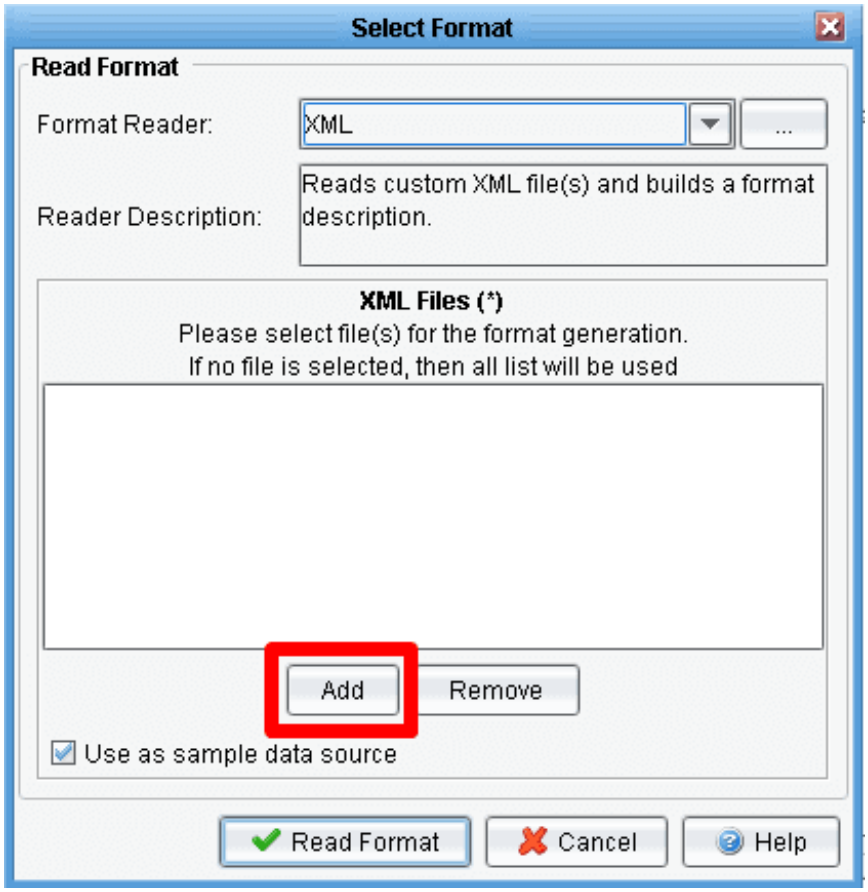


Now, we know that we're going to be using the ORU message, so we can scroll down to that portion of the tree and double click the node to expand it. Also double click the **MSH_Message_Header** node to expand that as well. Your panel should now resemble that shown above.

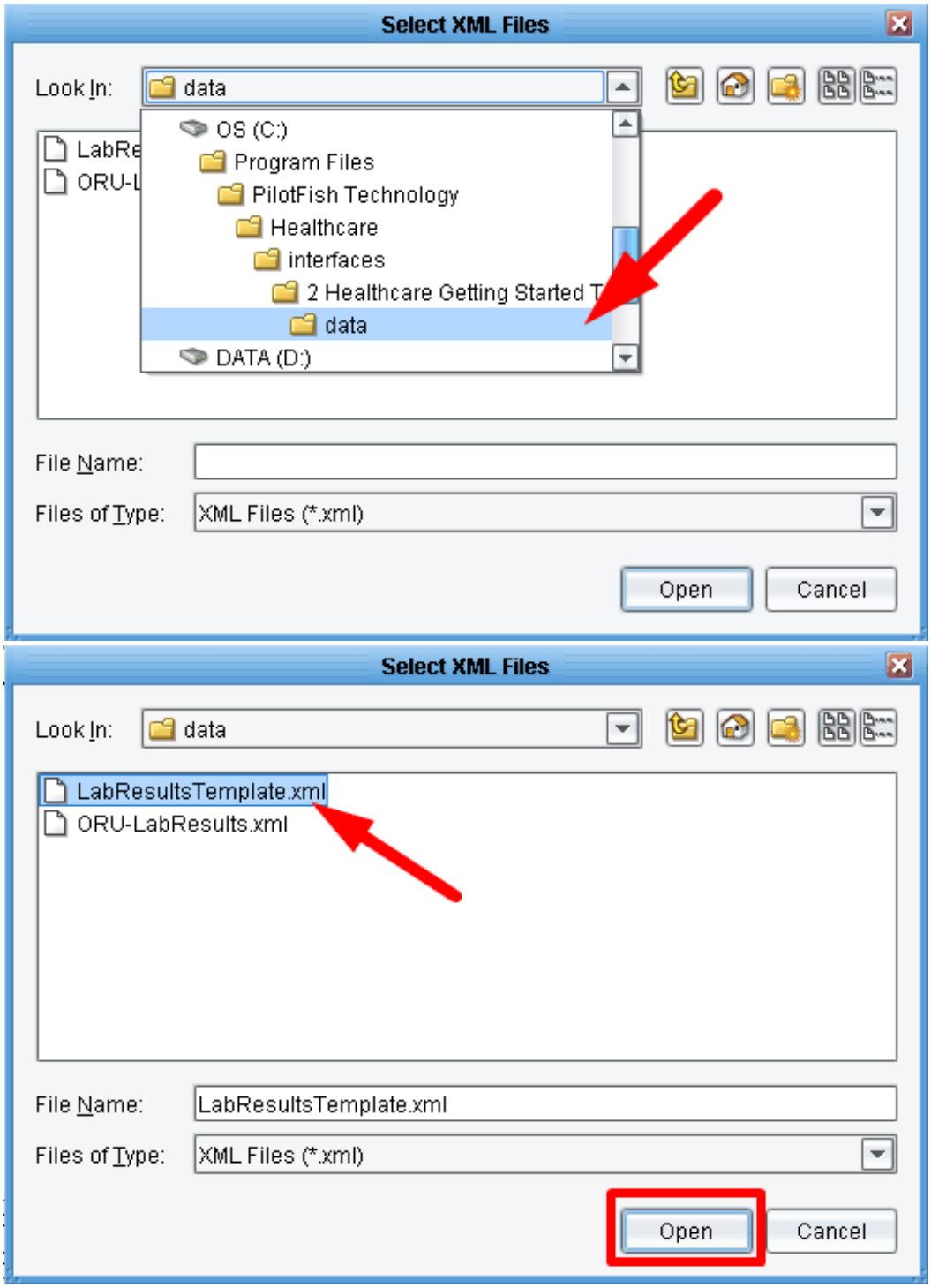
Next, we will want to load in our TargetFormat. Click the **Open Target Format** icon.



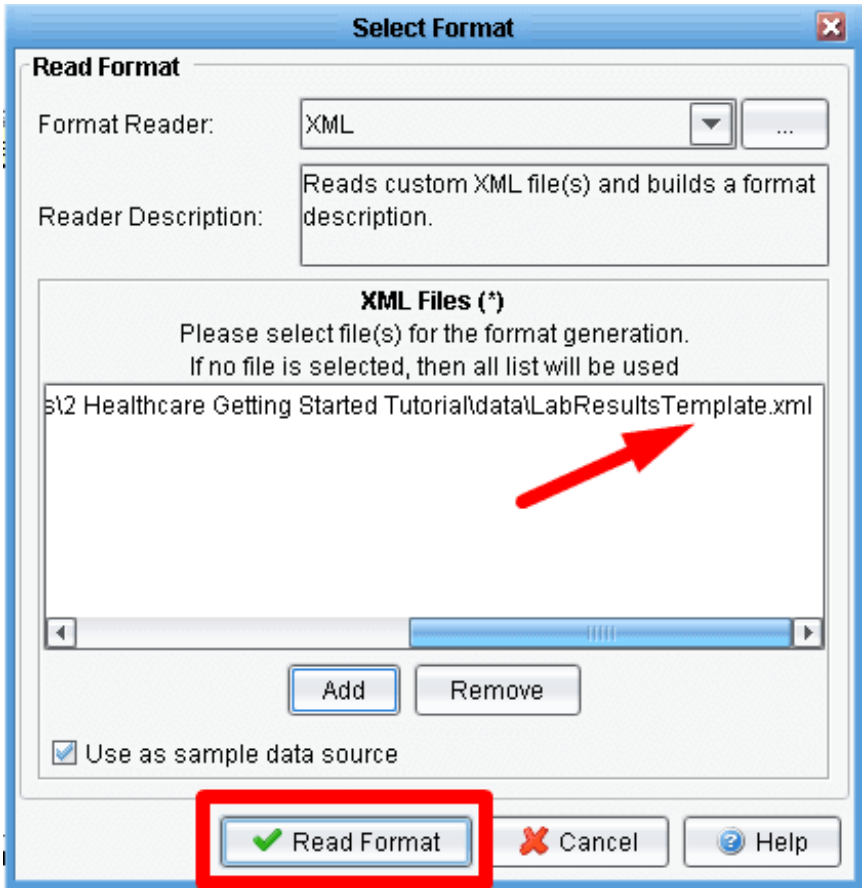
Select the **XML** format reader from the drop down.



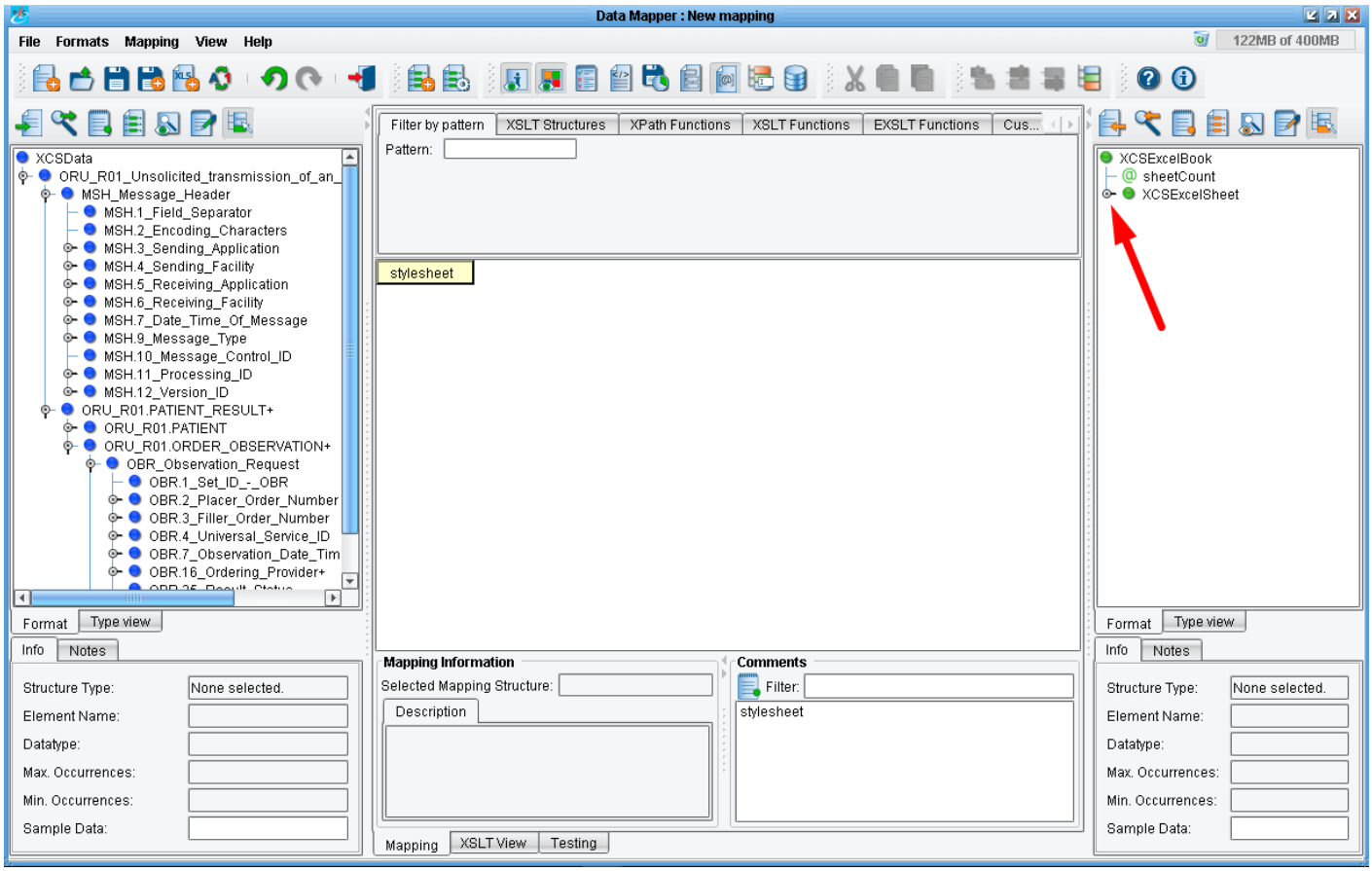
Next, click the **Add** button.



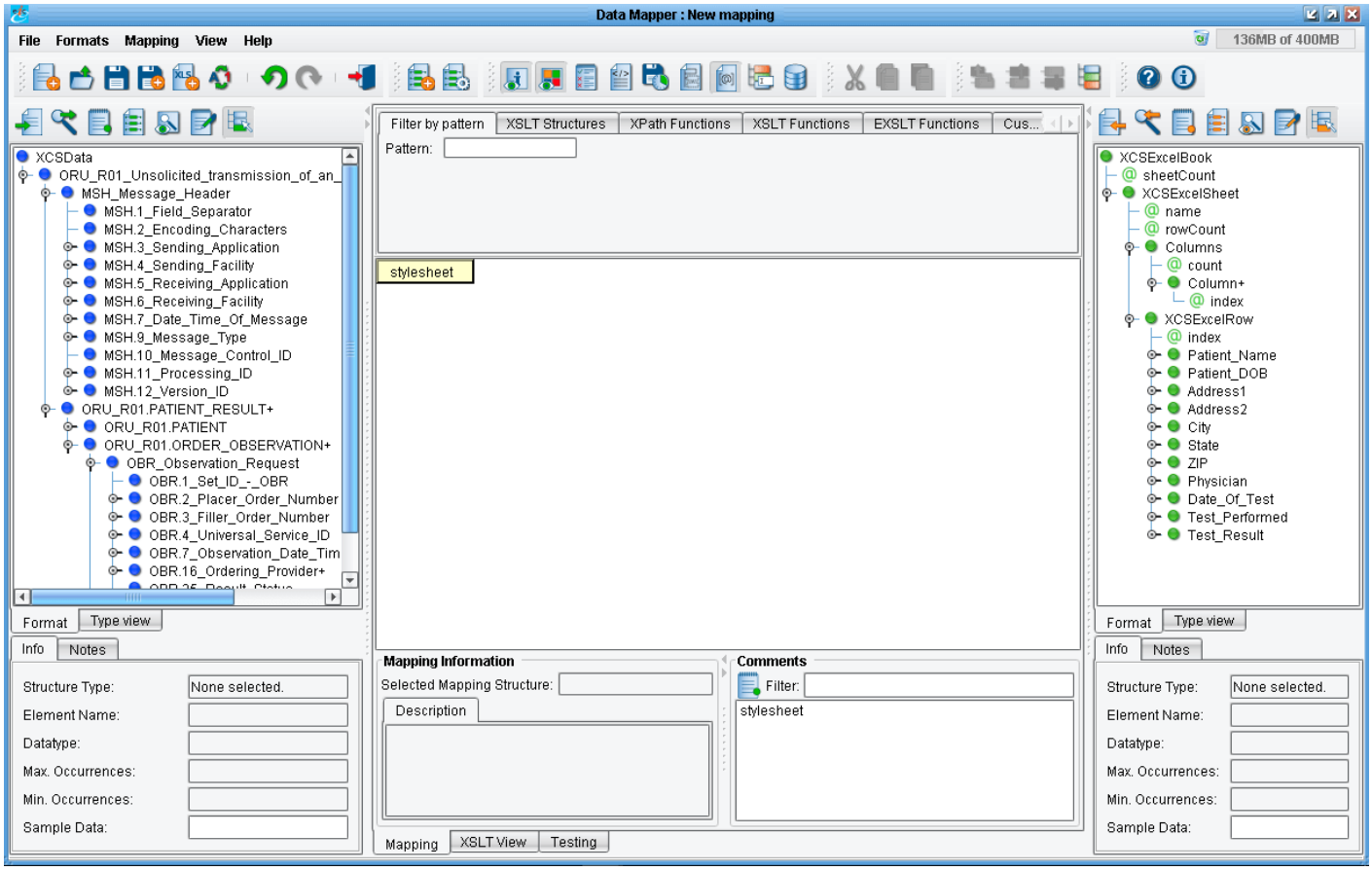
Navigate to your distribution folder, select the data folder and then **LabResultsTemplate.xml** and click **Open**.



When the **Select Format** dialog opens, click **Read Format**.

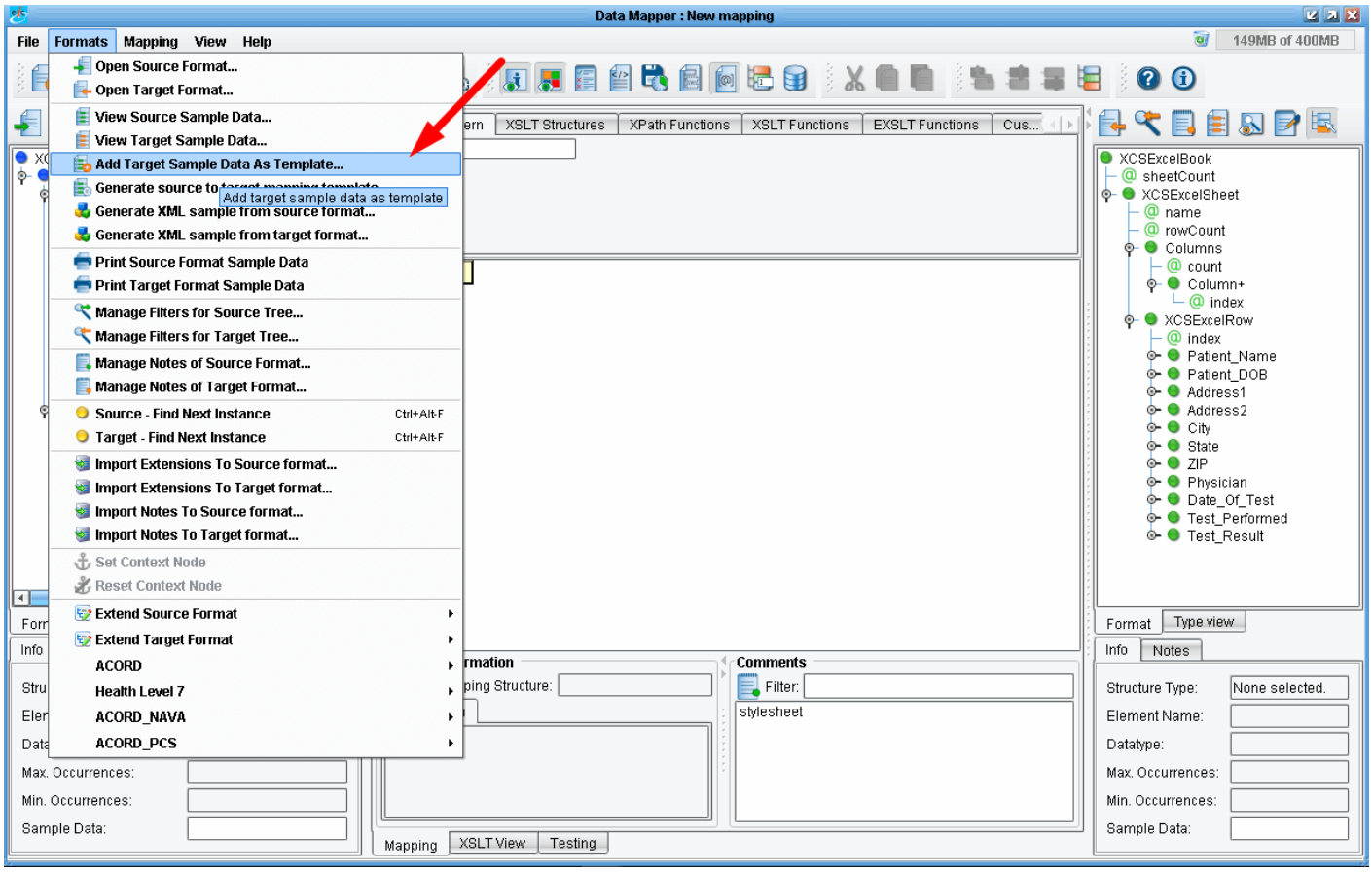


Your TargetFormat opens. Now we will want to expand the tree. Click the nodes.

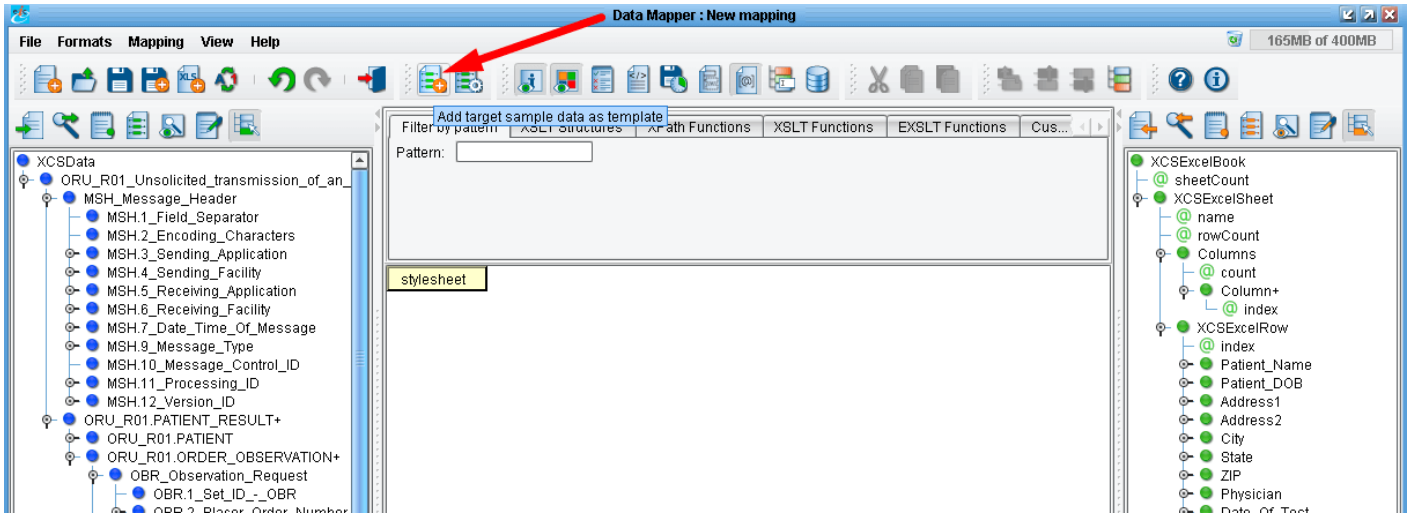


Expanding the tree you'll see that we have a simple XML representation of a single Excel spreadsheet with rows representing basic patient information, physician information and test results.

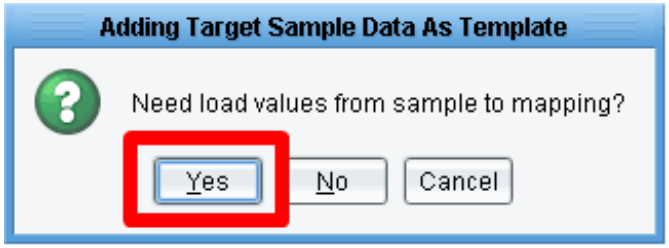
Now we need to create a mapping between our HL7 format and our spreadsheet. The easiest way to begin a mapping is to choose Formats.



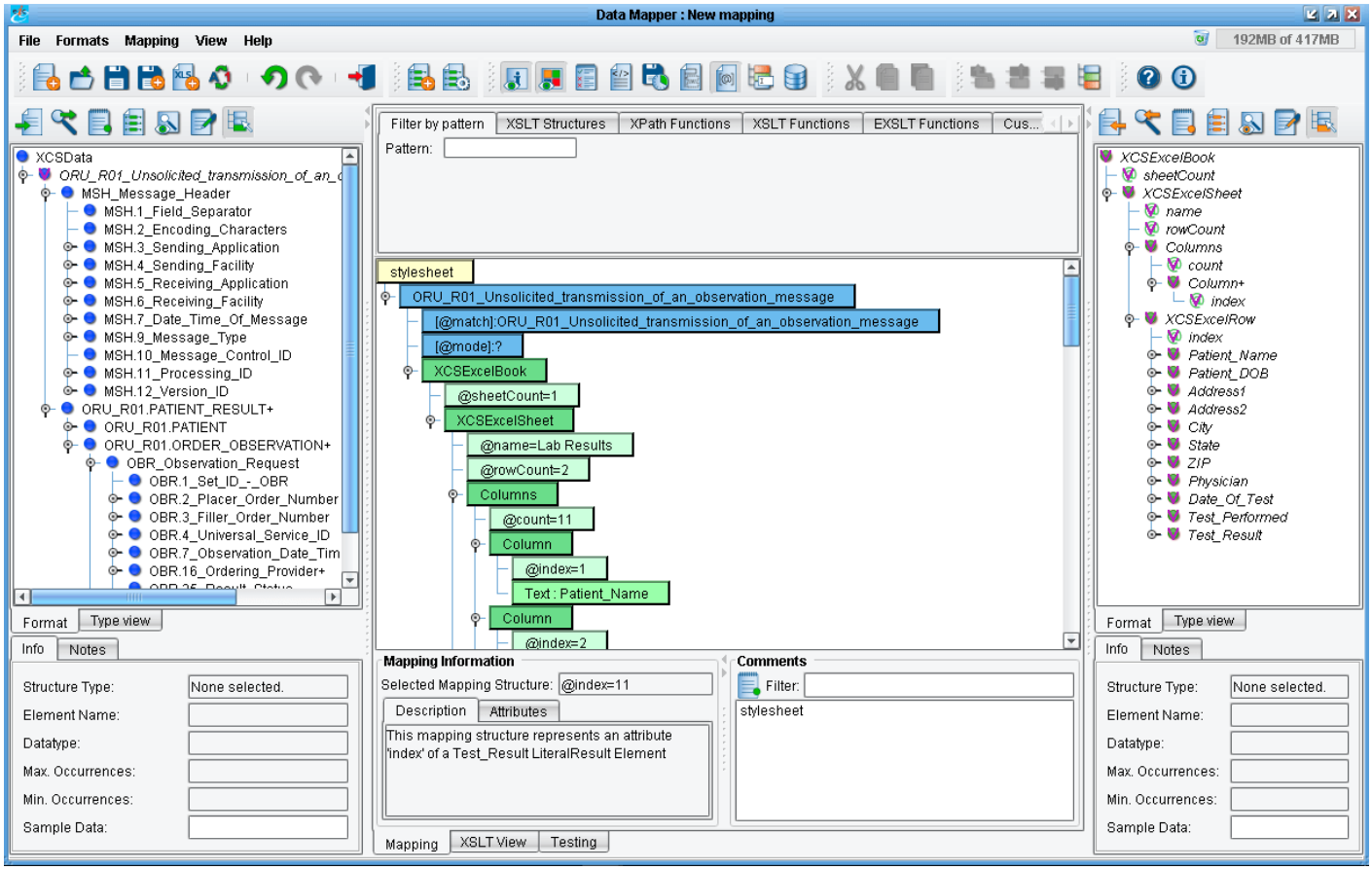
From the **Formats** menu select **Add Target Sample Data As Template.** (While you are at it, browse the drop down to view all the other options through this menu.)



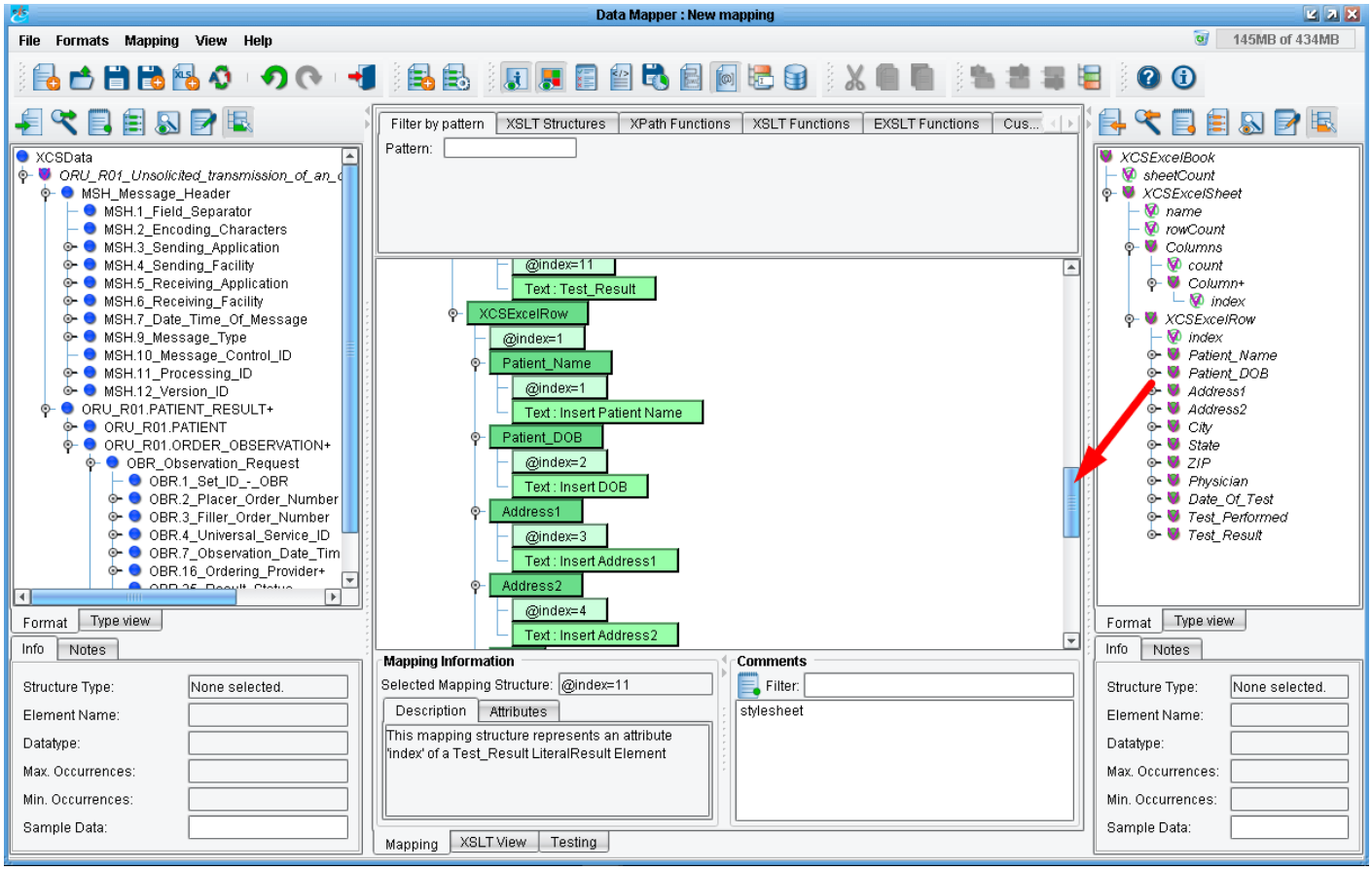
Or you can use **Add Target Sample Data As Template** button on the main toolbar.



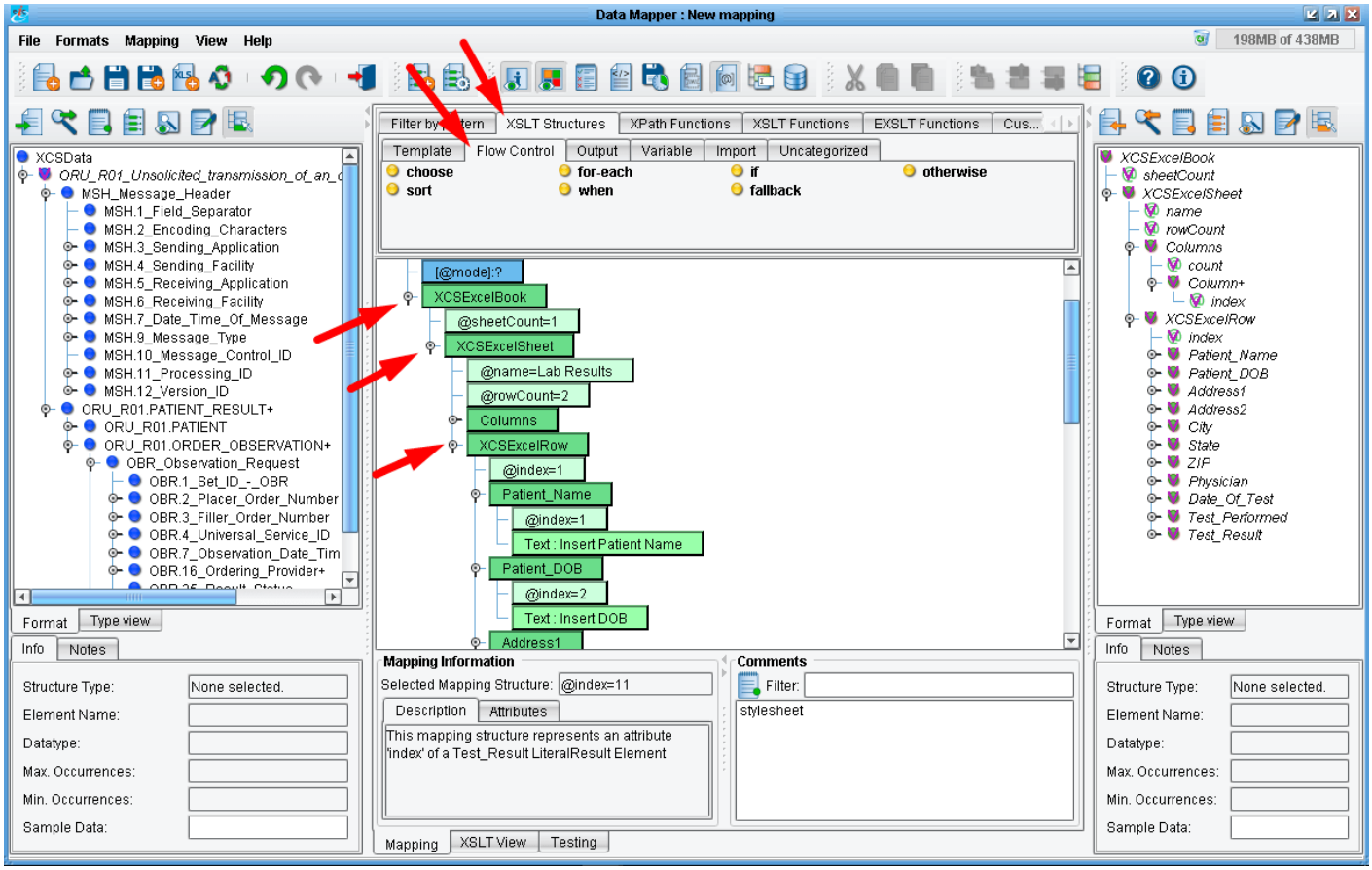
When prompted to add sample data to the mapping, click **Yes.** The sample data will load quickly.



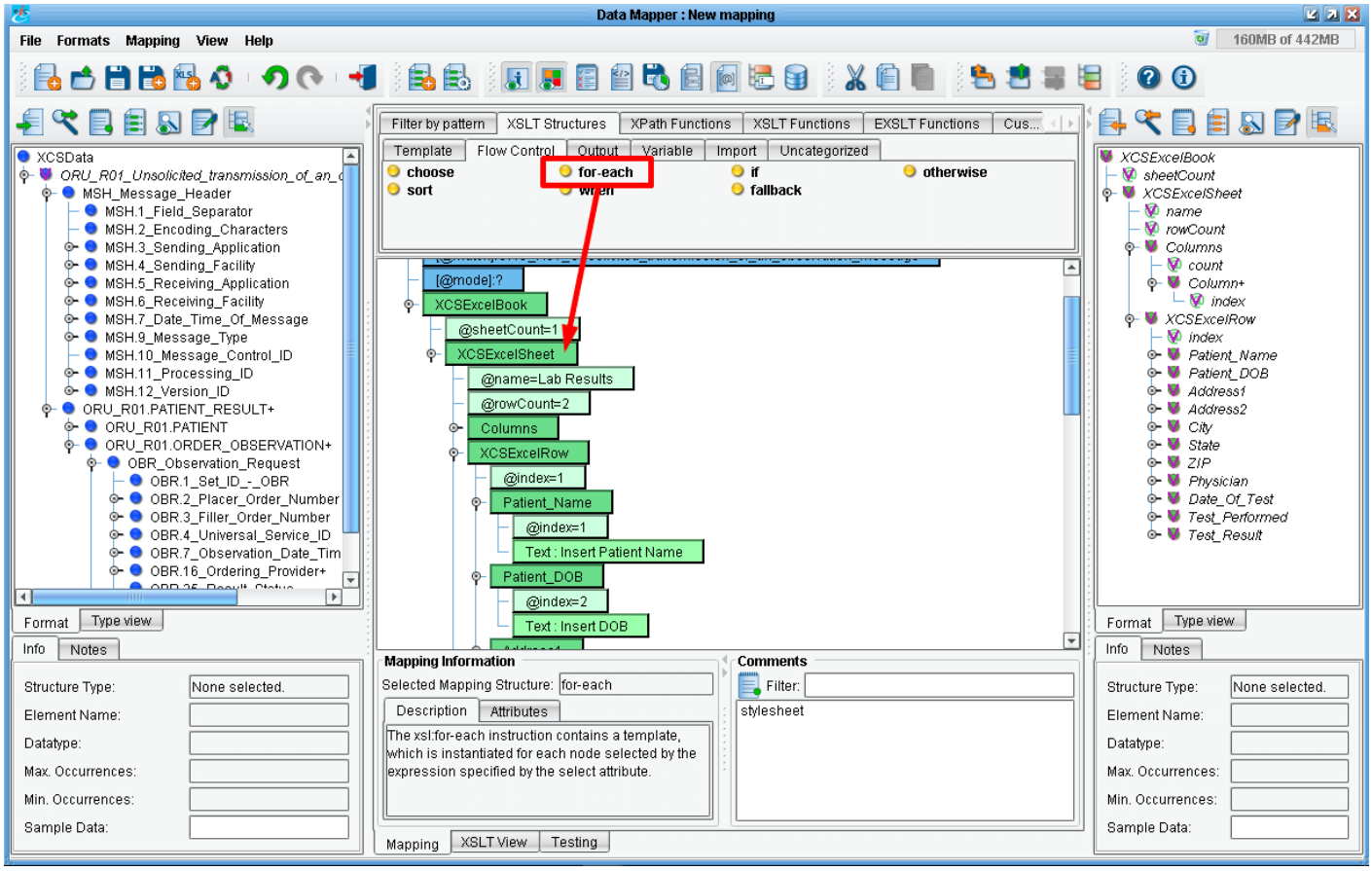
You'll see the center populated with a large number of green nodes.



These green nodes provide the structure of the file that we'll want to create. Now, our job is to map values from the Source file onto this template. Use the scroll bar (right) to review the mapping.



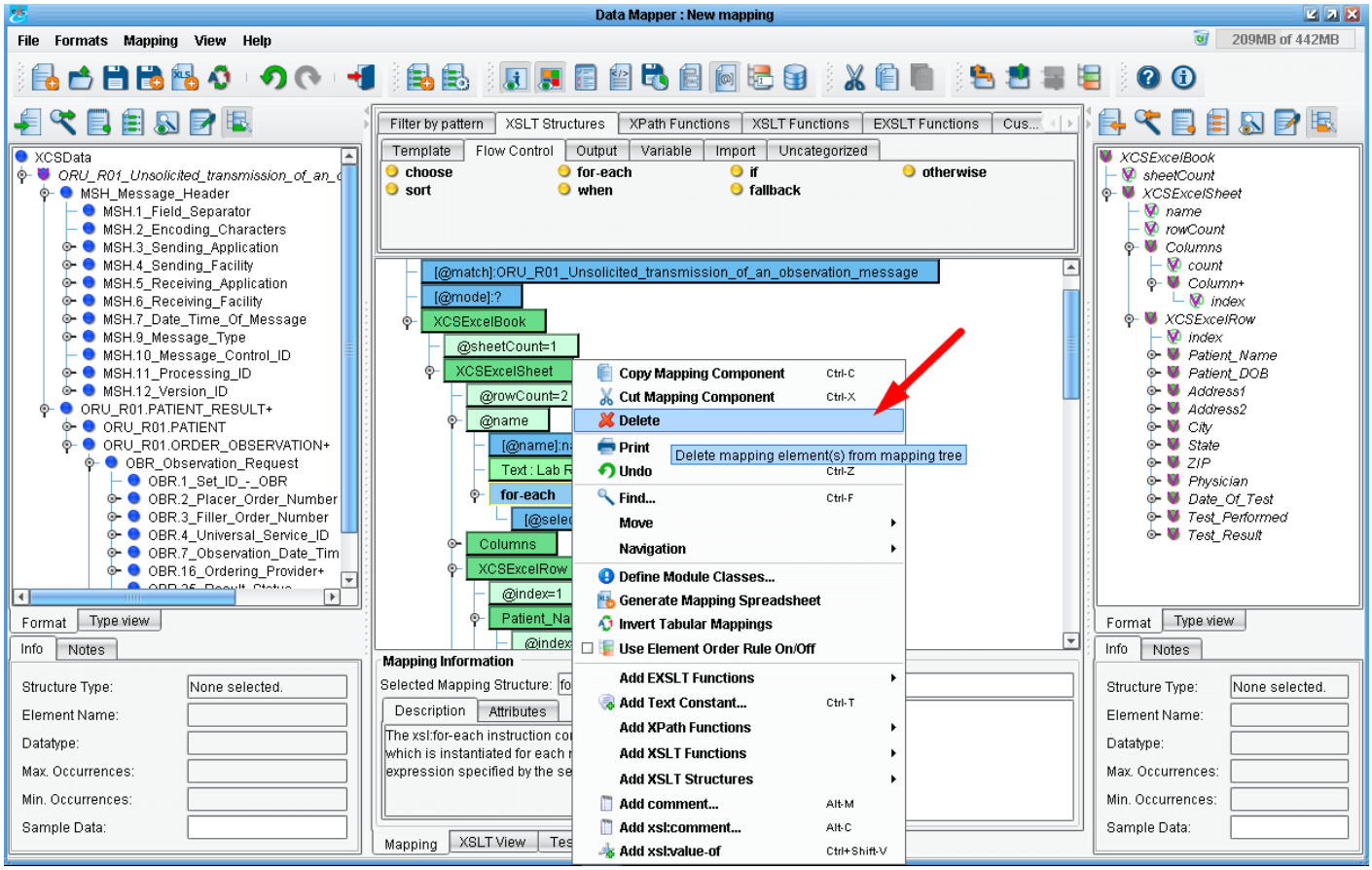
Scroll back to the top of the mapping. Make sure your **XCSExcelBook**, **XCSExcelSheet** nodes are expanded and that you can see the **XCSExcelRow** element (you'll want to be sure that the **Columns** element is NOT expanded). Select the **Flow Control** sub-tab underneath the XSLT Structures tab.



We'll want to create an Excel row for each patient result in our HL7 file (**ORU_R01.PATIENT_RESULT+**).

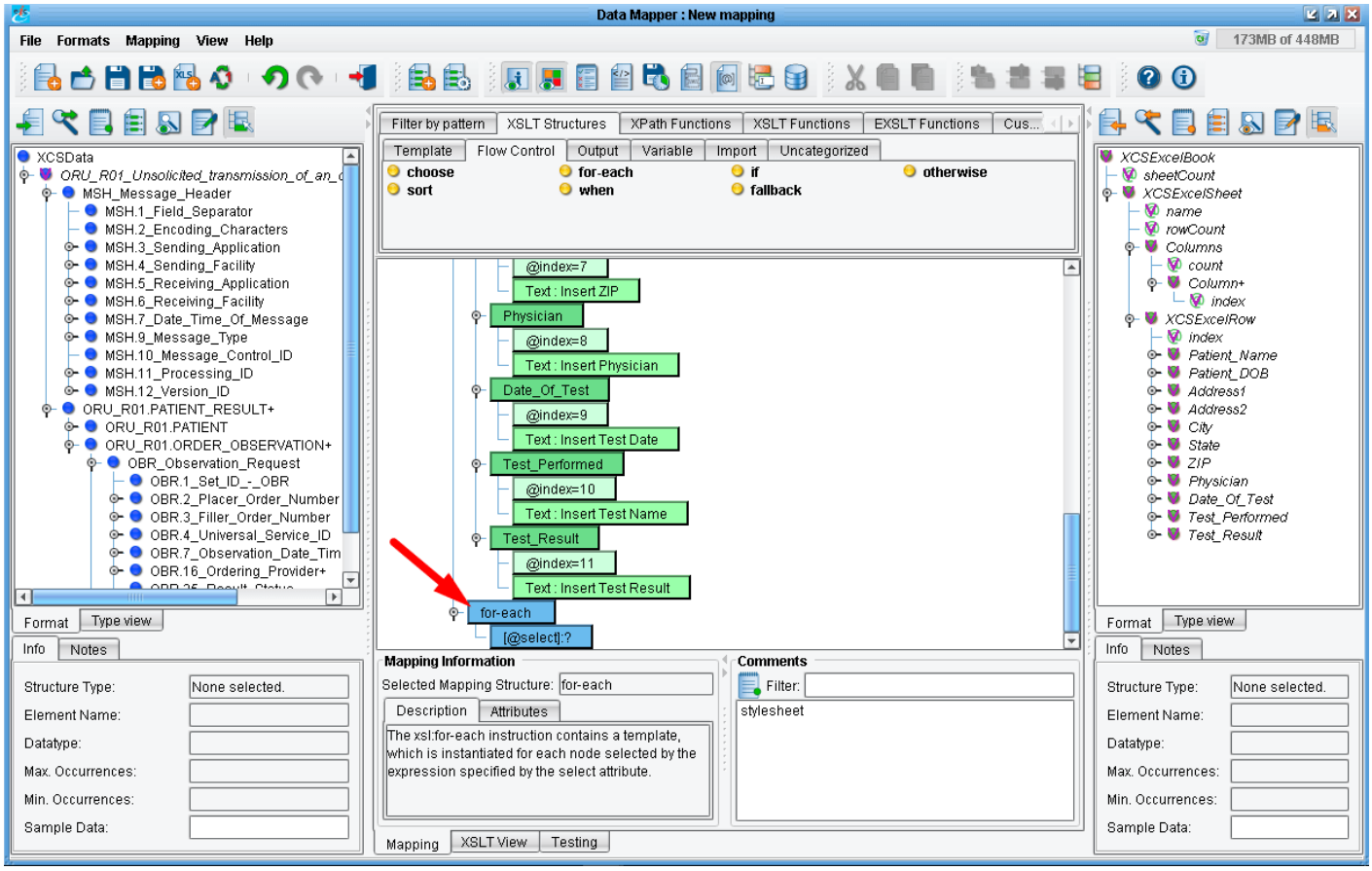
To accomplish this, click and drag the yellow **"for-each"** node on top of the **XCSExcelSheet** element, but don't release your mouse until you are almost at the bottom of the element.

TIP: You'll see a gray bar and yellow highlight appear, which tells you when to release your mouse on the node.

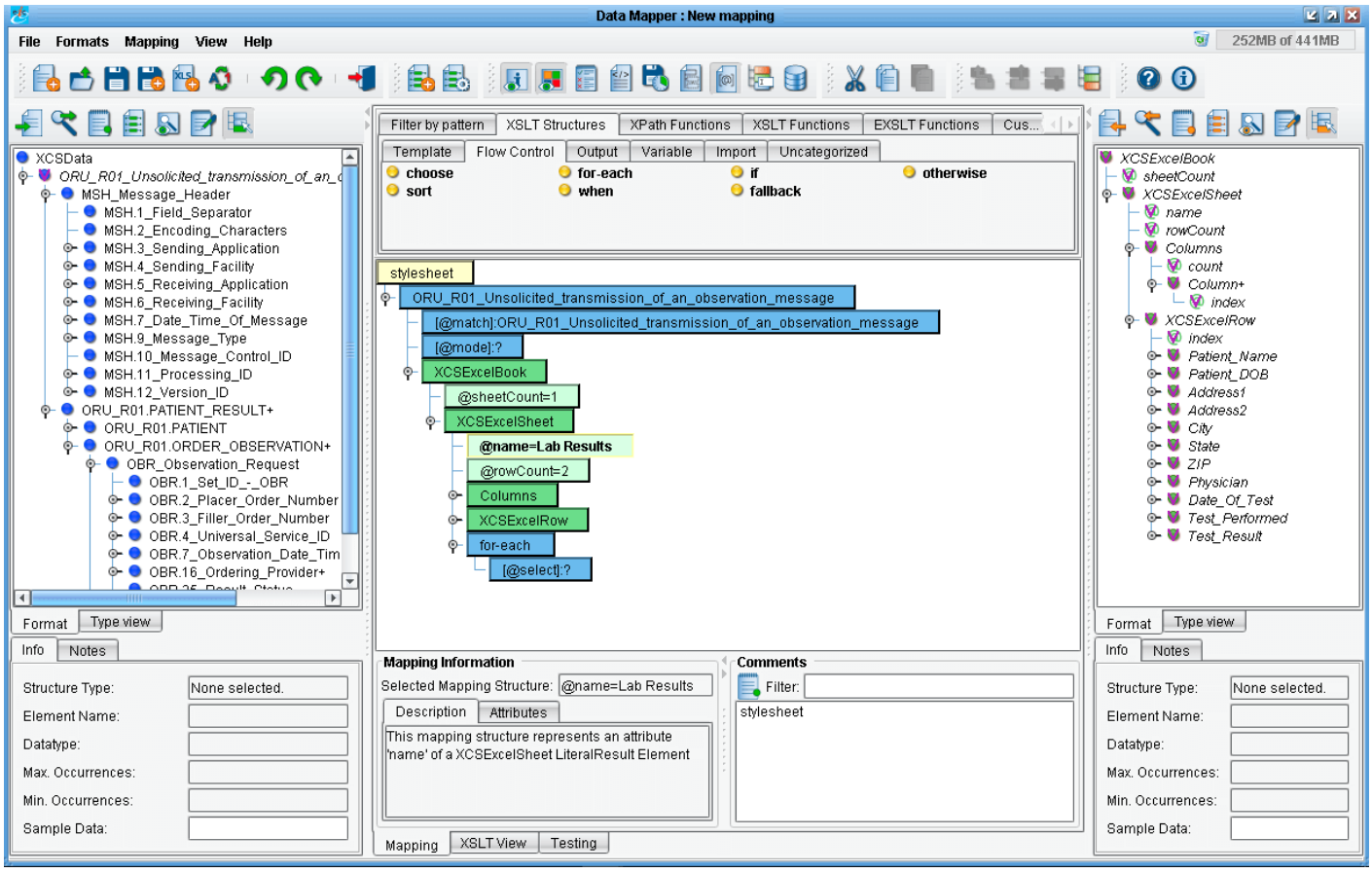


Now if your mapping looks like that above, that is, the **for-each** node appears directly under the **Text : Lab**

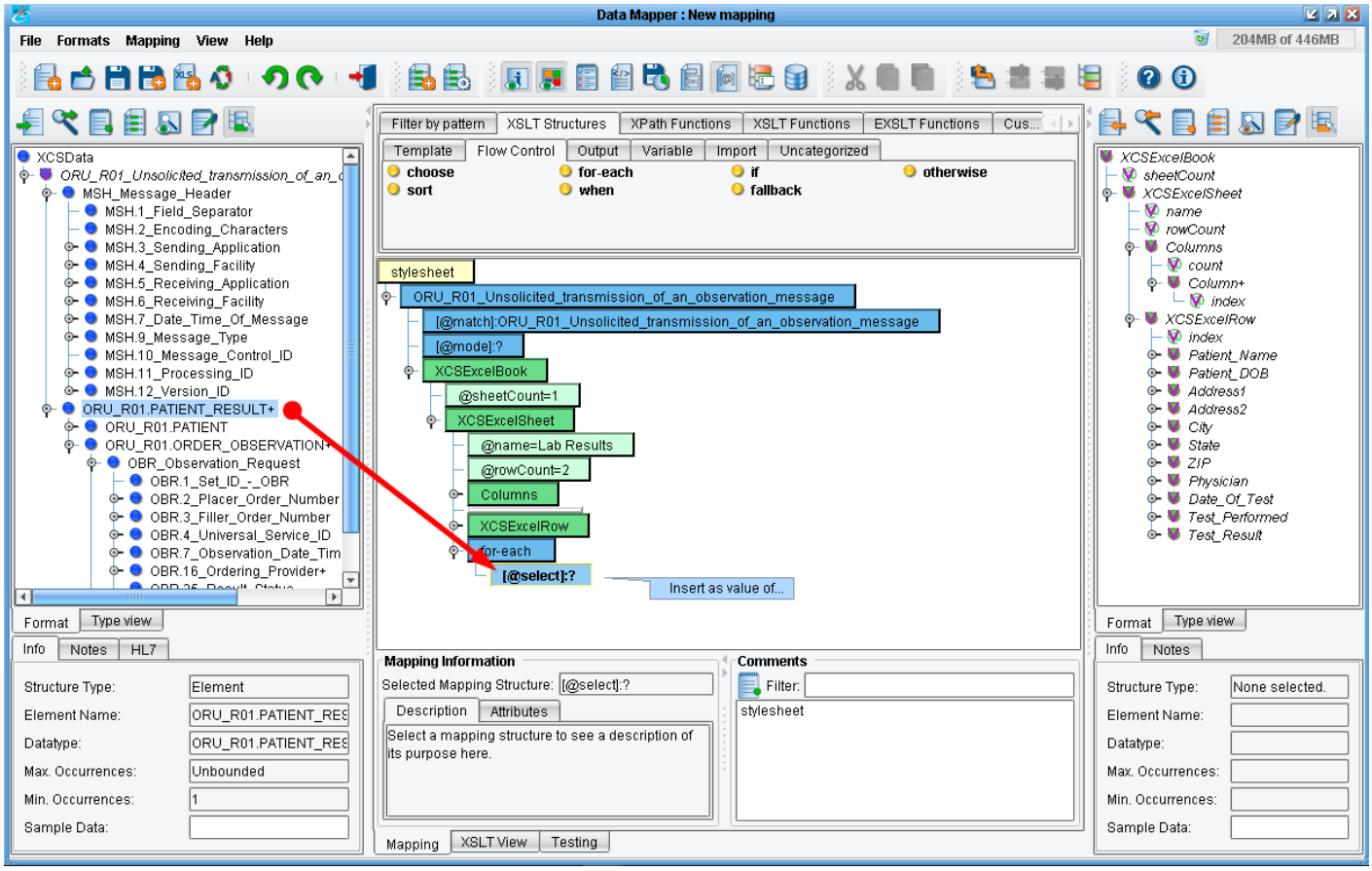
Results node (rather than all the way at the bottom), simply select the node, right click to open the drop down and select **Delete.** Then repeat previous step again. This time make sure you drag & drop the **for-each** onto the bottom portion of the **XCSExcelSheet** element.



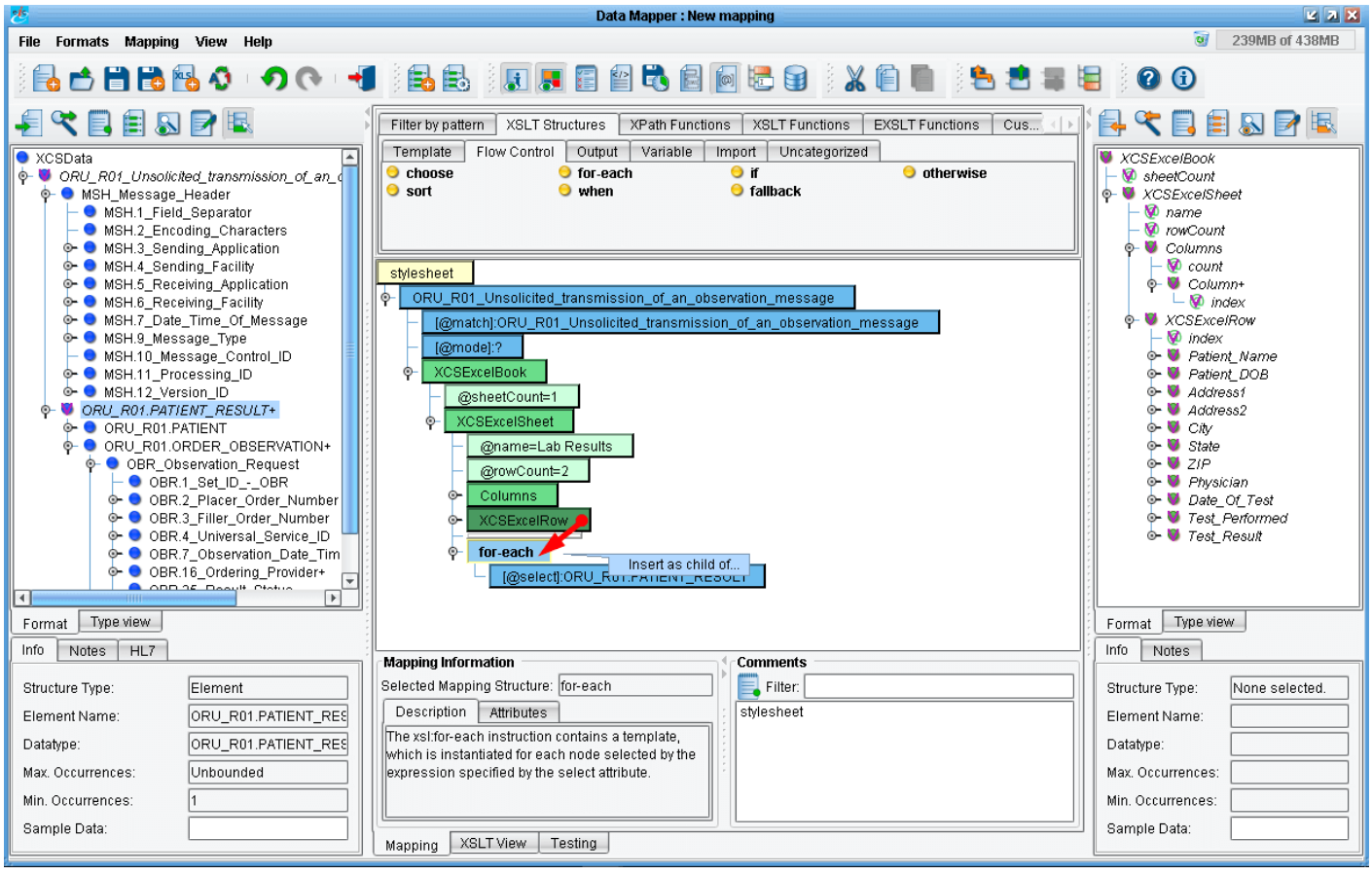
The **for-each** node will appear at the bottom of your mapping. You'll need to scroll down to view this.

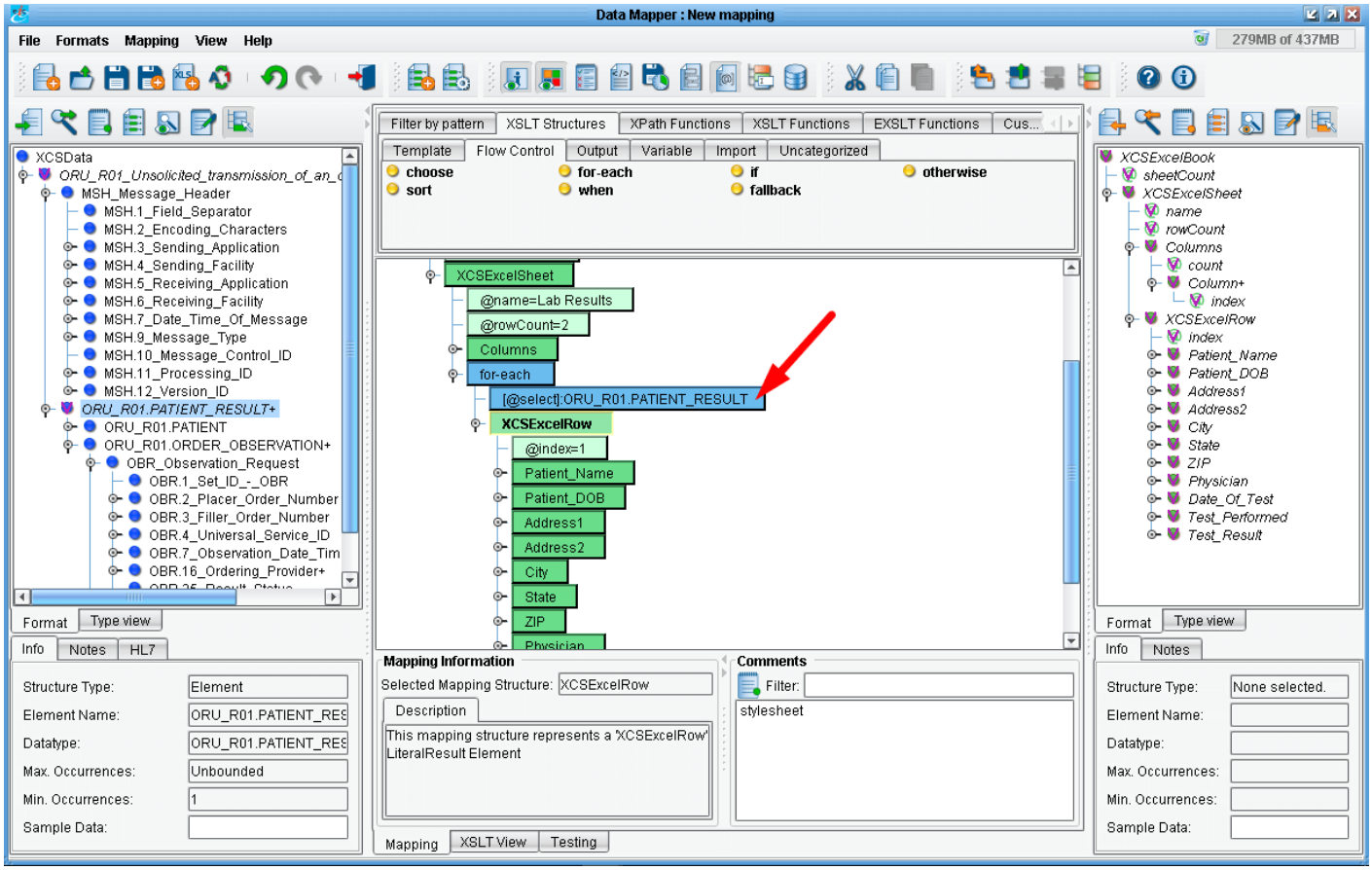


Now scroll up and collapse the **XCSExcelRow** element. This will allow you to see both the **for-each** and the **XCSExcelRow**.

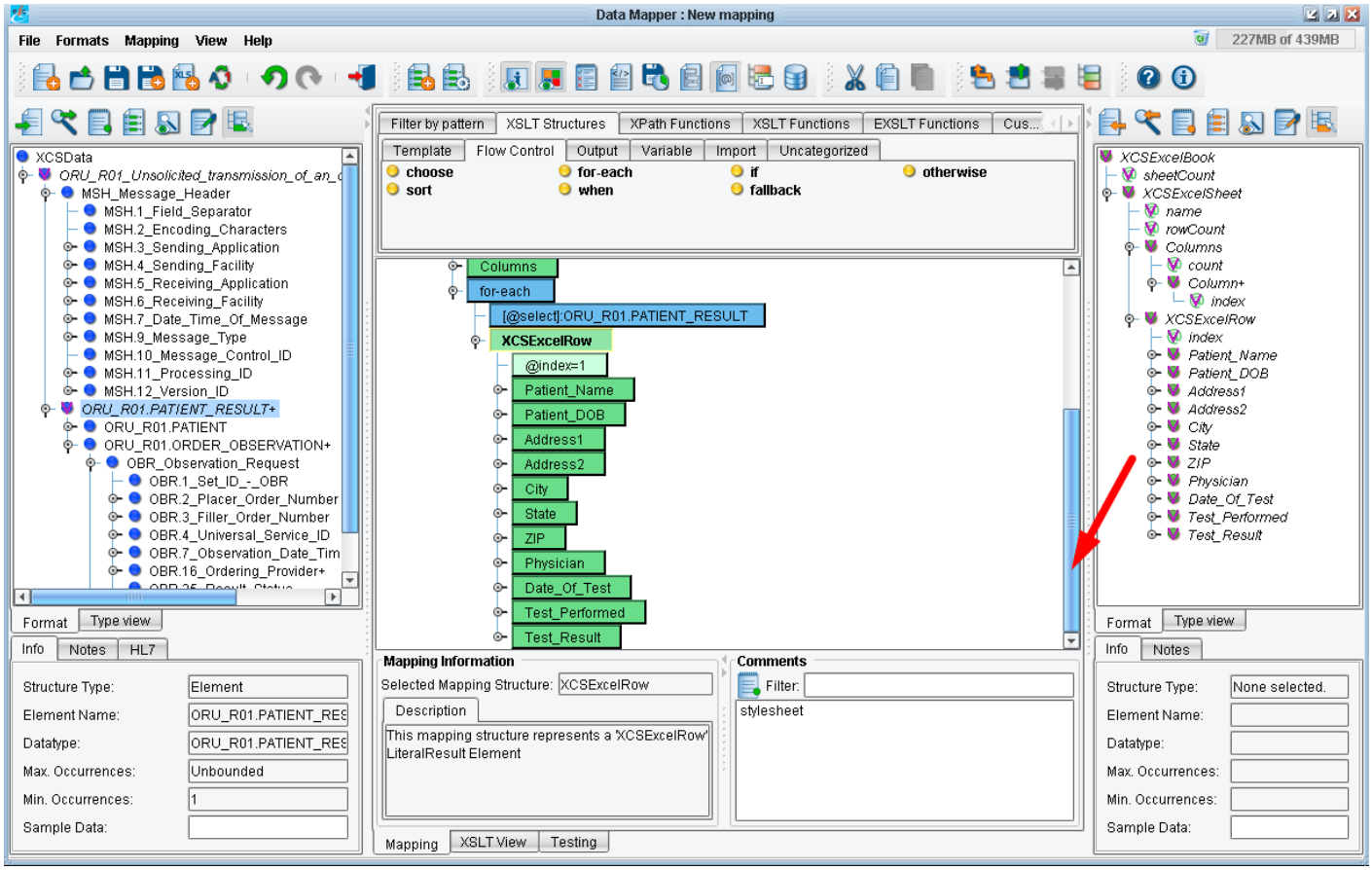


Since we want to generate a row for each instance of a patient result, we'll now drag the **ORU_R01.PATIENT_RESULT+** element onto the select node of the **for-each** (watch for the yellow highlight) and then release the mouse.

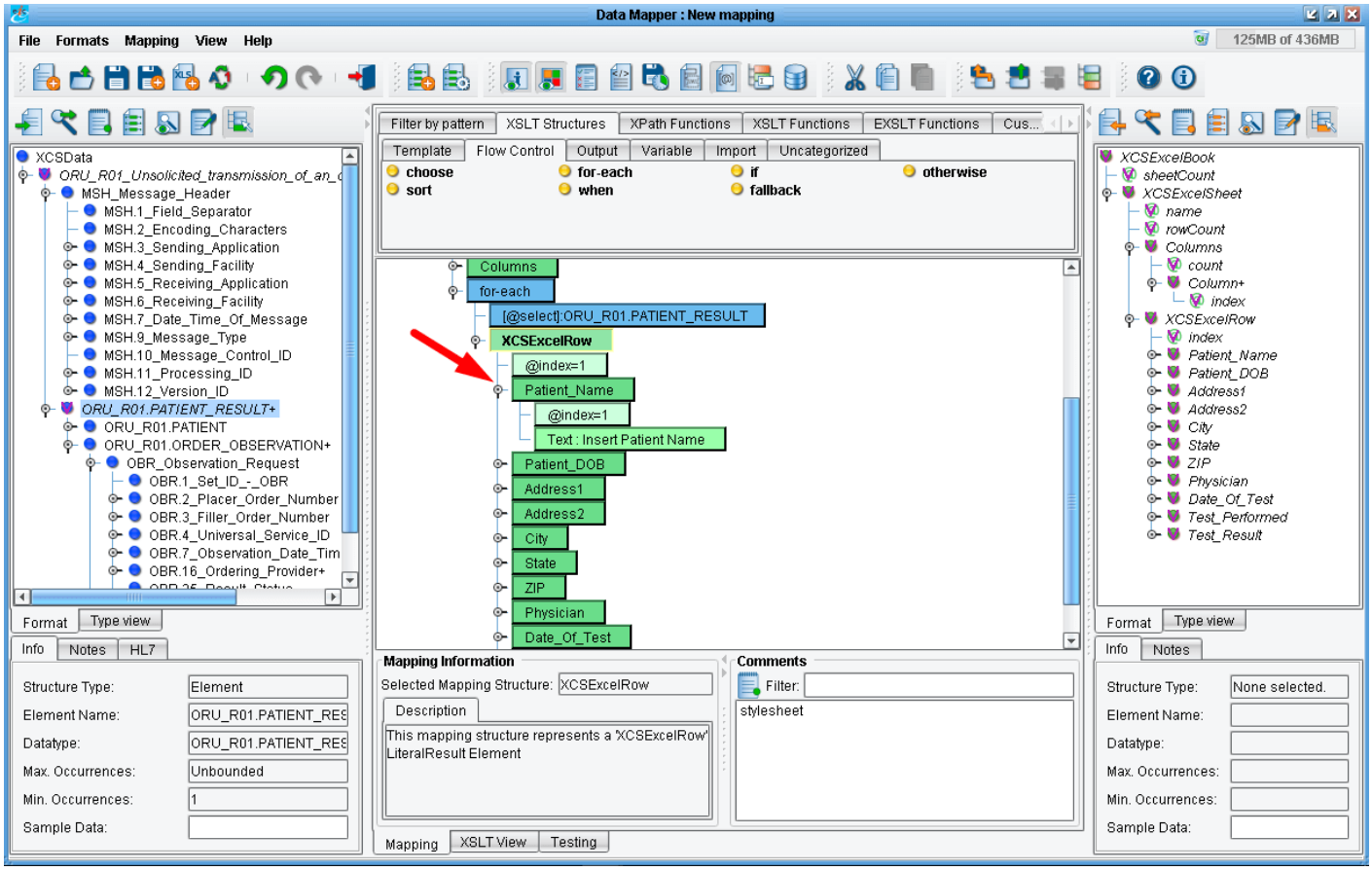




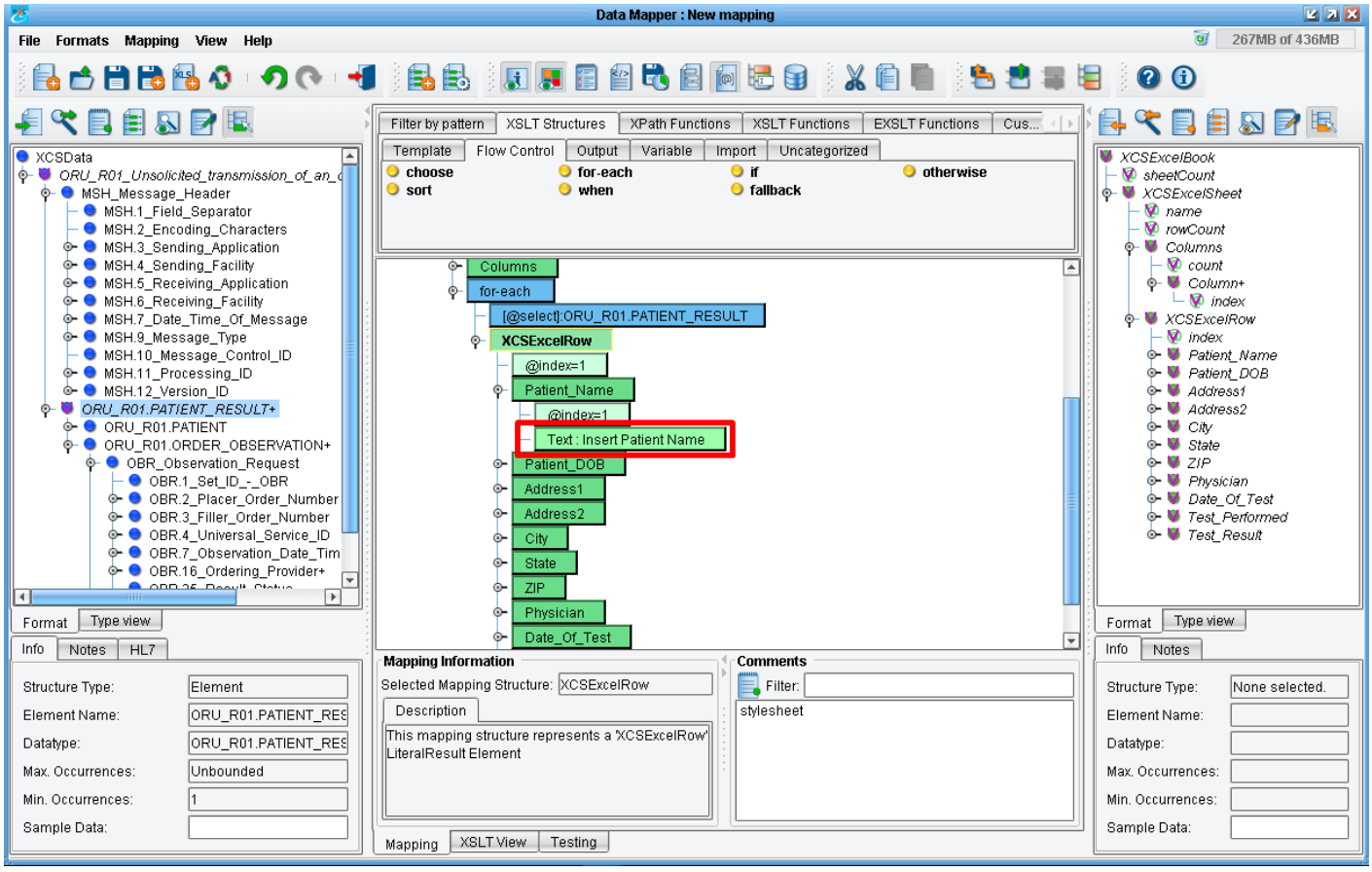
Now, we'll want to make the row a child of the for-each node. Left click the **XCSExcelRow** element (the text will get bolder as you do this) and drop it on top of the **for-each**.



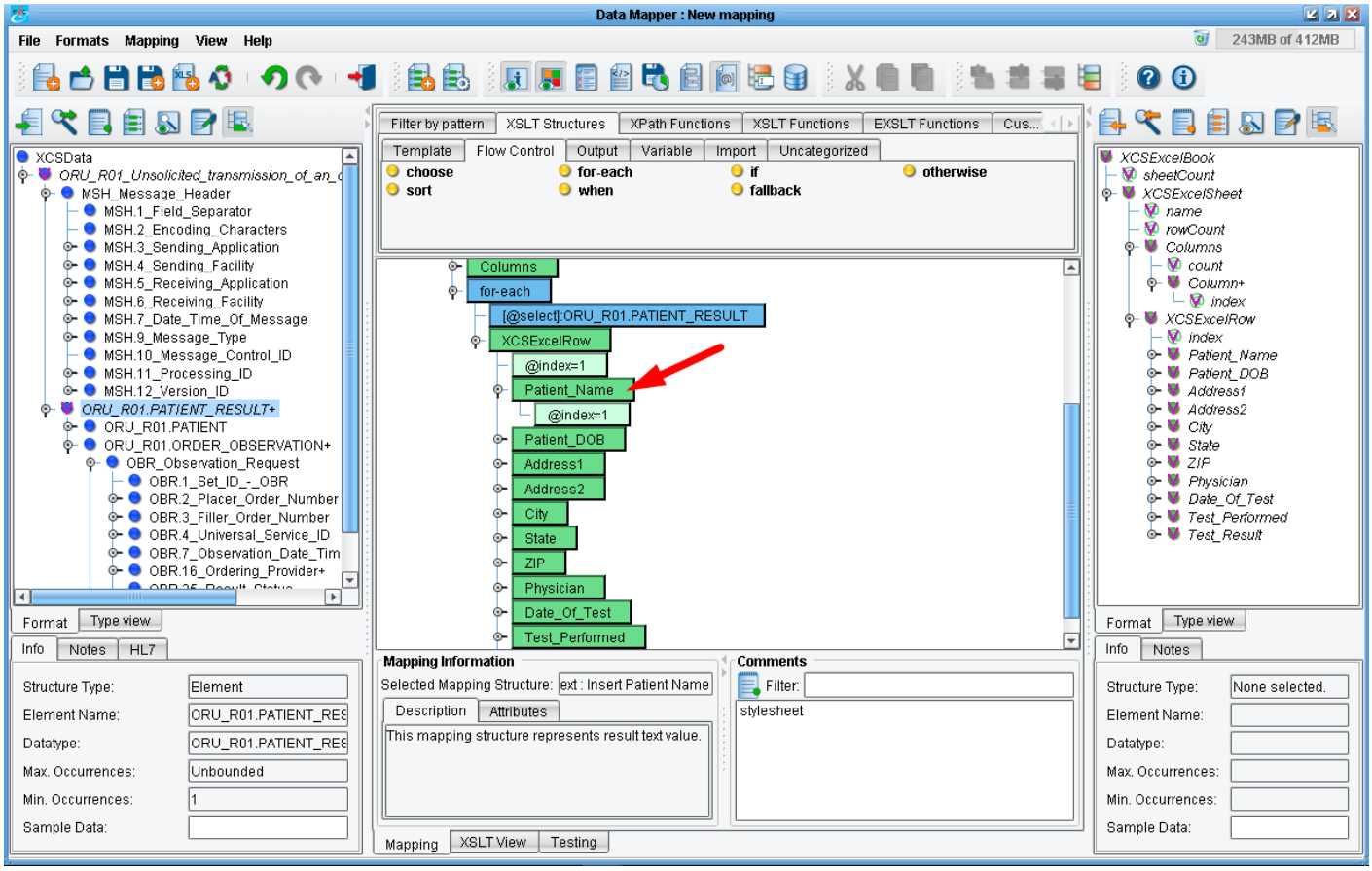
Scroll down and you'll see the system will generate a row for each patient result.



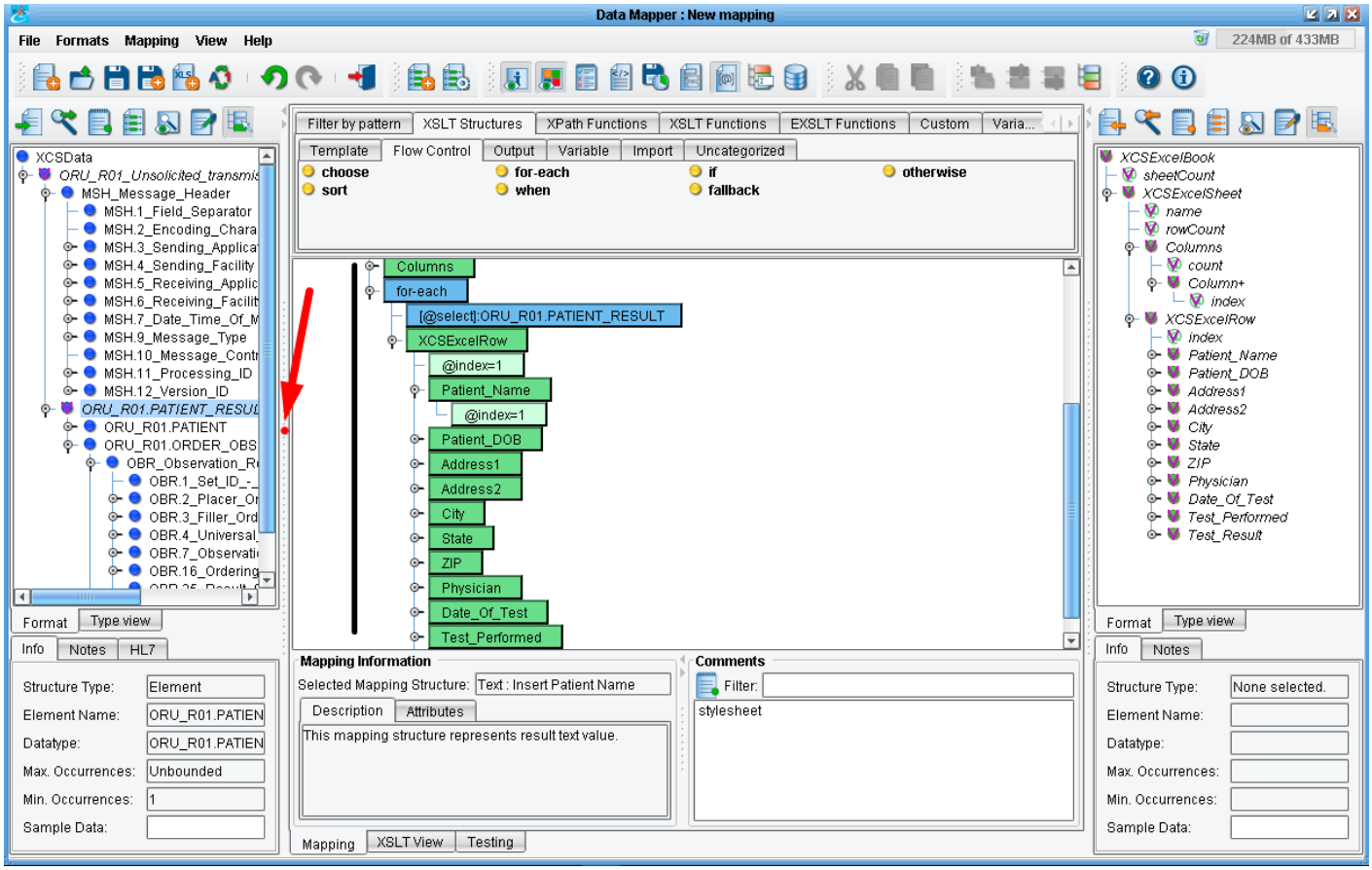
Let's go ahead and drag and drop to populate a few additional pieces of information. Expand the **Patient_Name** node.



You'll see a Text node called: **Insert Patient Name**.

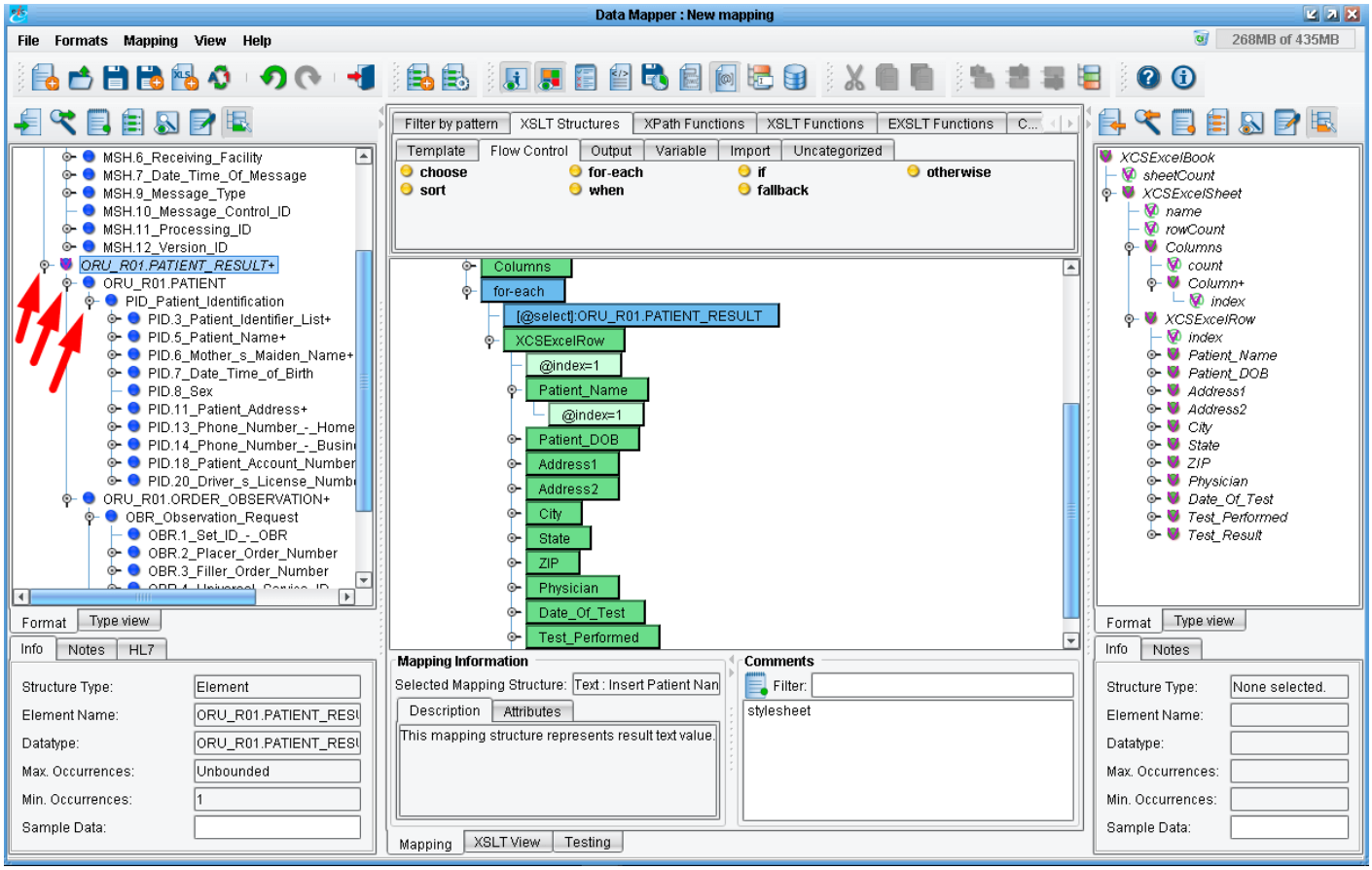


Select **Insert Patient Name,** right click, and select **Delete** from the drop down. Your mapping should now look like the above.

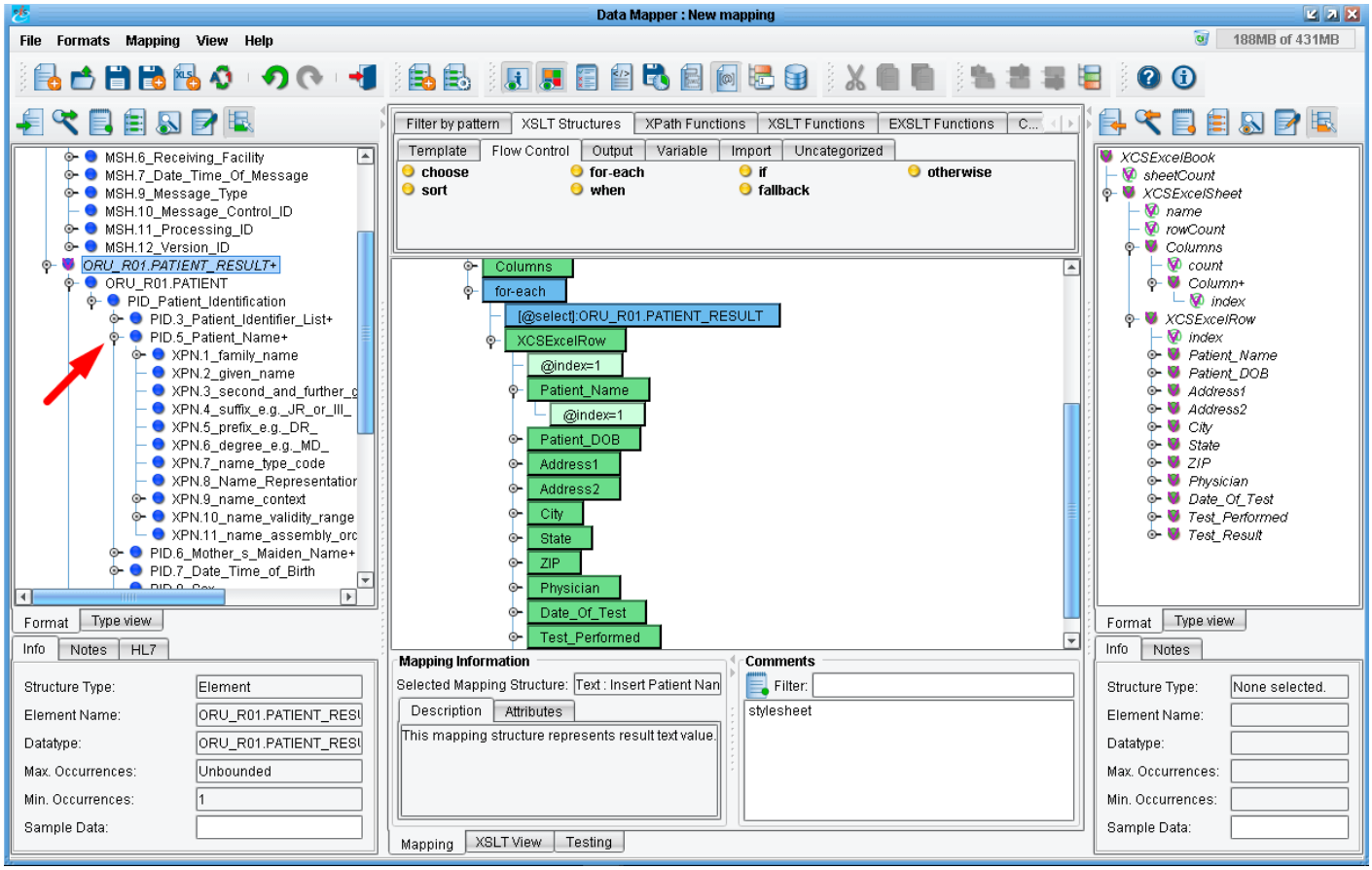


Next, we are going expand the nodes in our left column. So, we will want to expand the column to get a better view without needing to scroll.

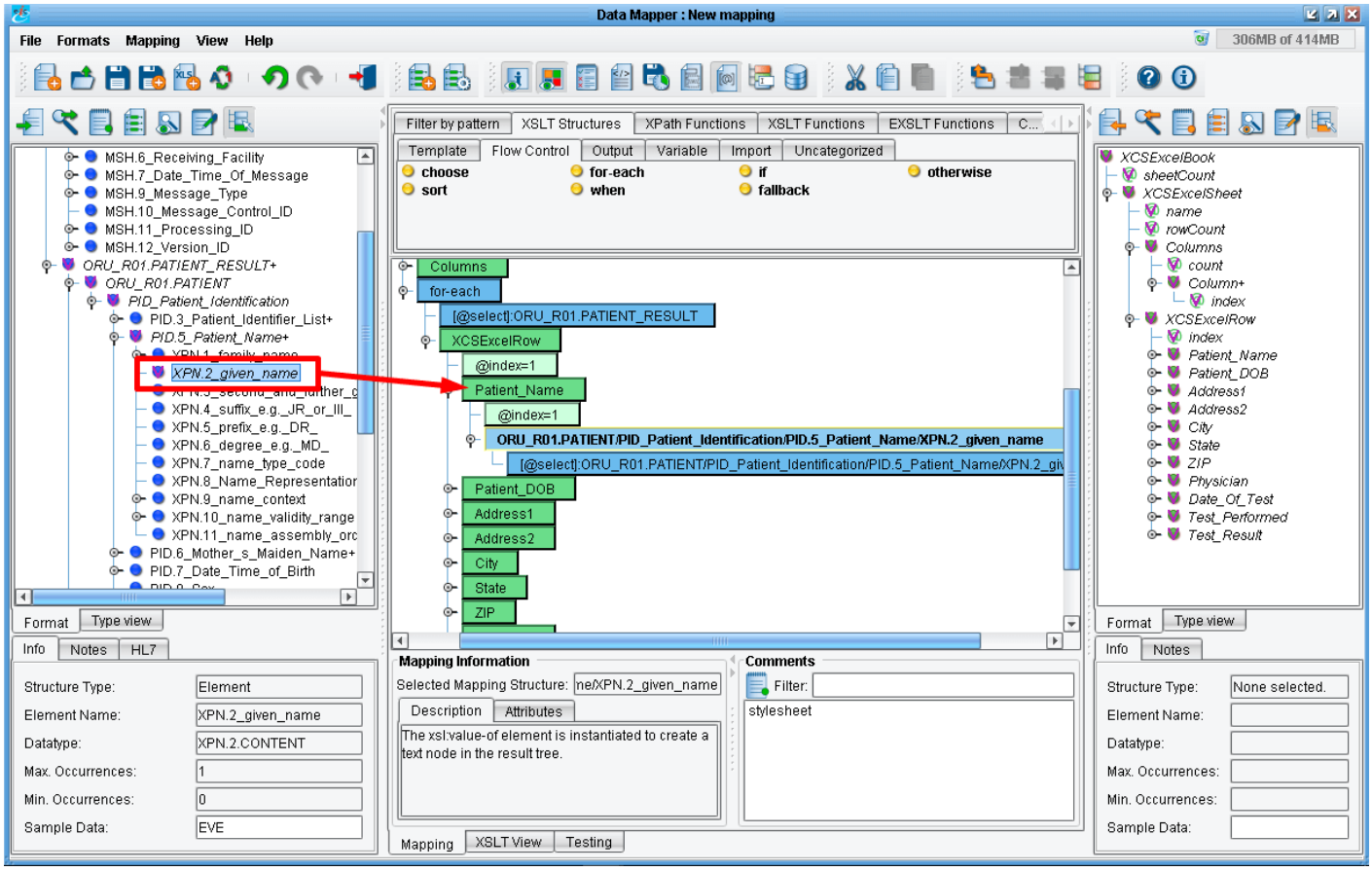
TIP: You can widen your columns by clicking and dragging the vertical column bars, until you see the black line, as above. Release your mouse when you have reached the desired column width.



After you have widened the column, next, expand the **ORU_RO1.PATIENT_RESULT+** Node, **ORU_RO1.PATIENT** node, and **PID_Patient_Identification** node to reveal the patient name.

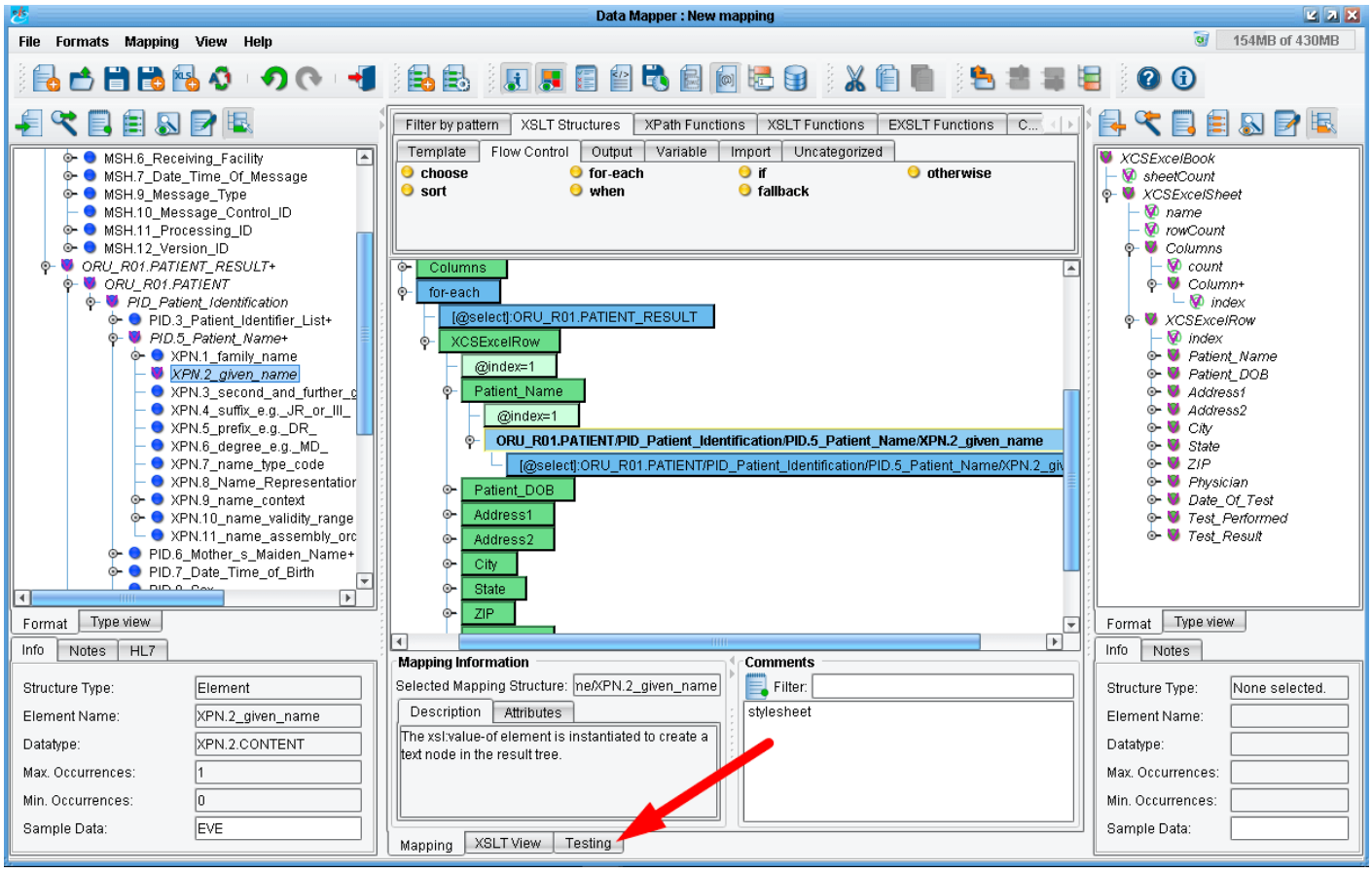


Then, expand the **PID.5_Patient_Name+** where you'll see the **XPN.1_family_name, XPN.2_given_name,** etc.

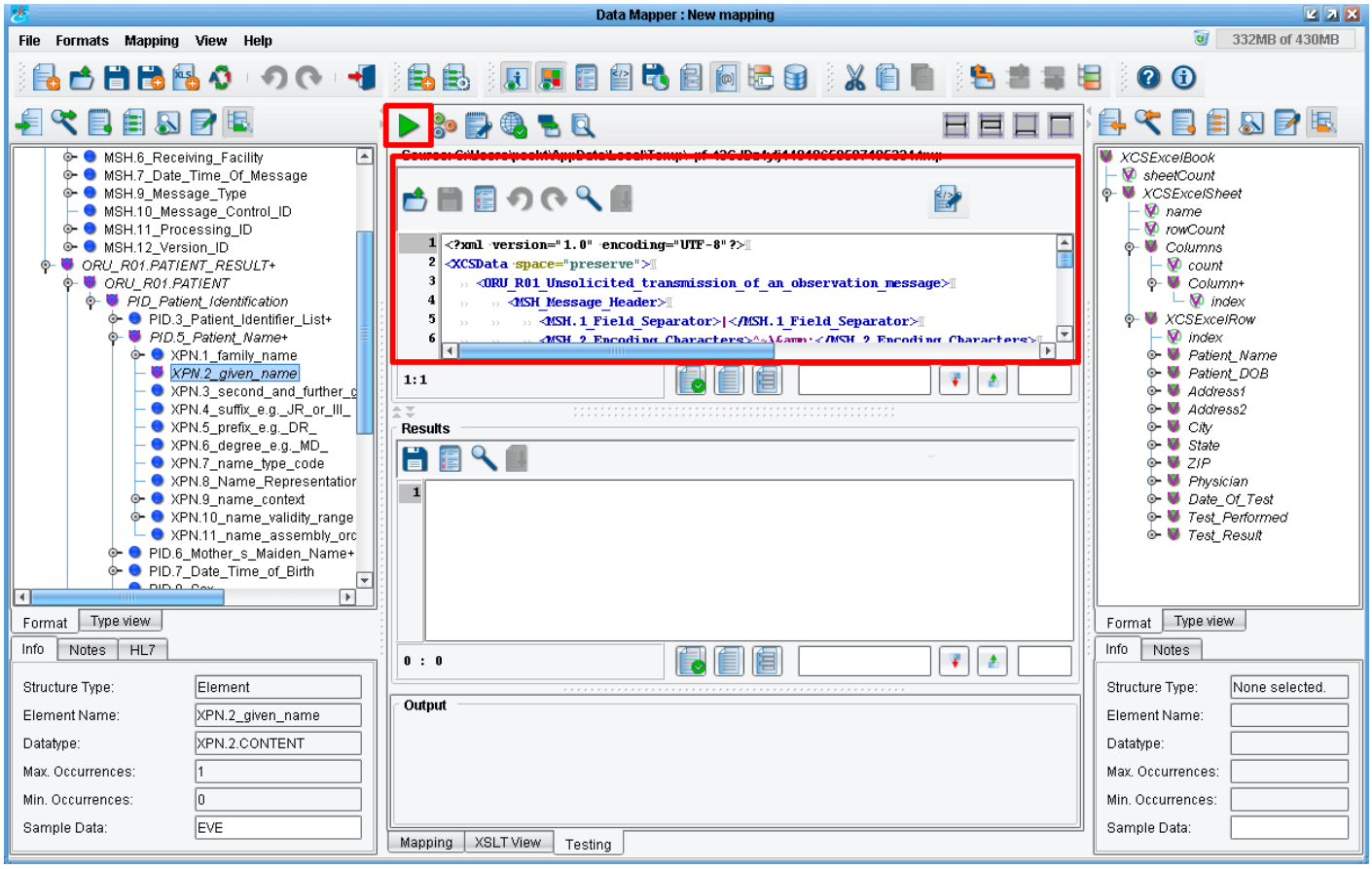


Drag the **XPN.2_given_name** onto the **Patient_Name** node in the center panel. Now, your mapping should look like that above.

A blue node with the path to the **Patient_Name** should appear underneath the **Patient_Name** node in the center panel.

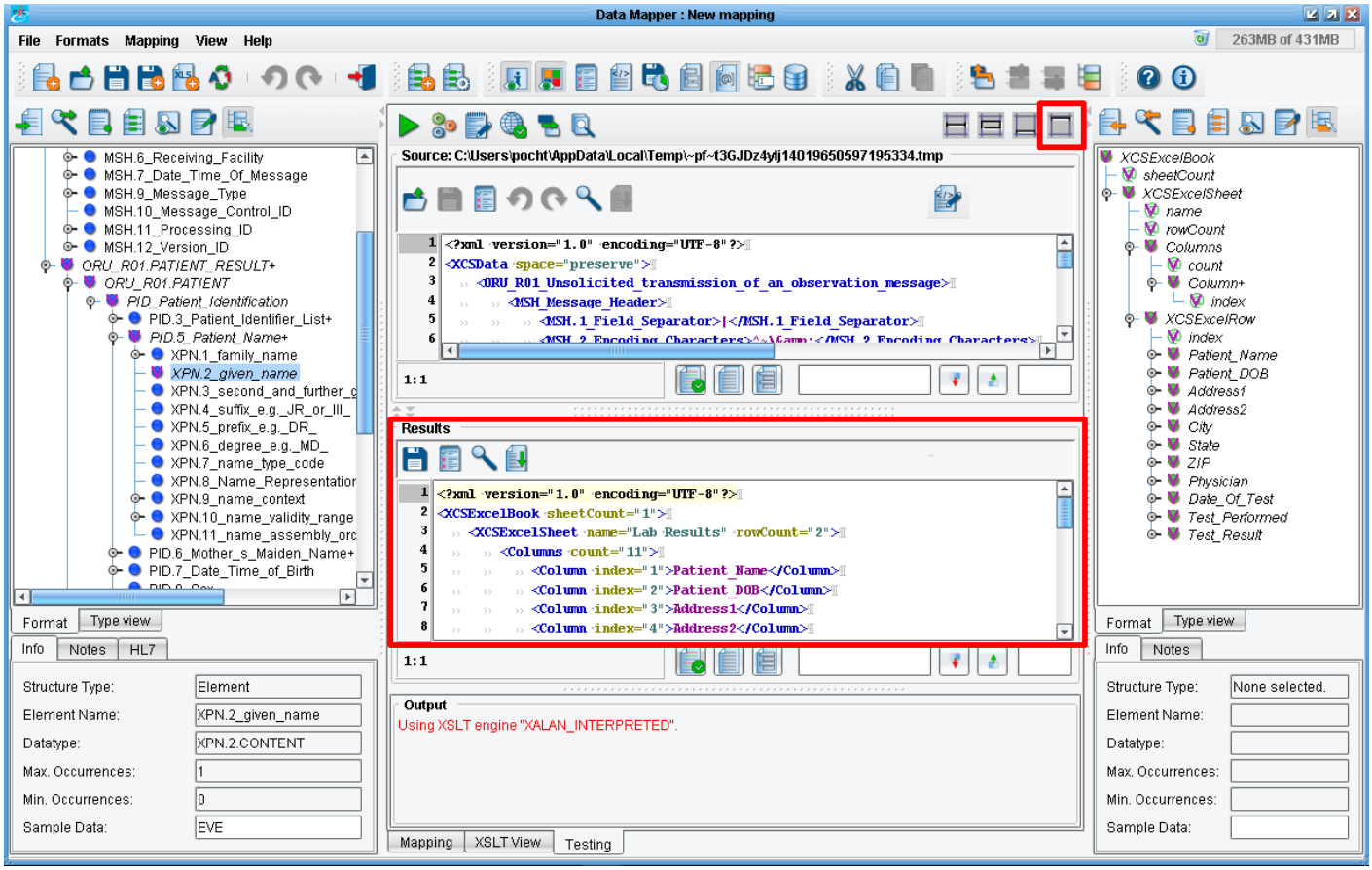


To see if this works click on the **Testing** tab at the bottom of the screen.

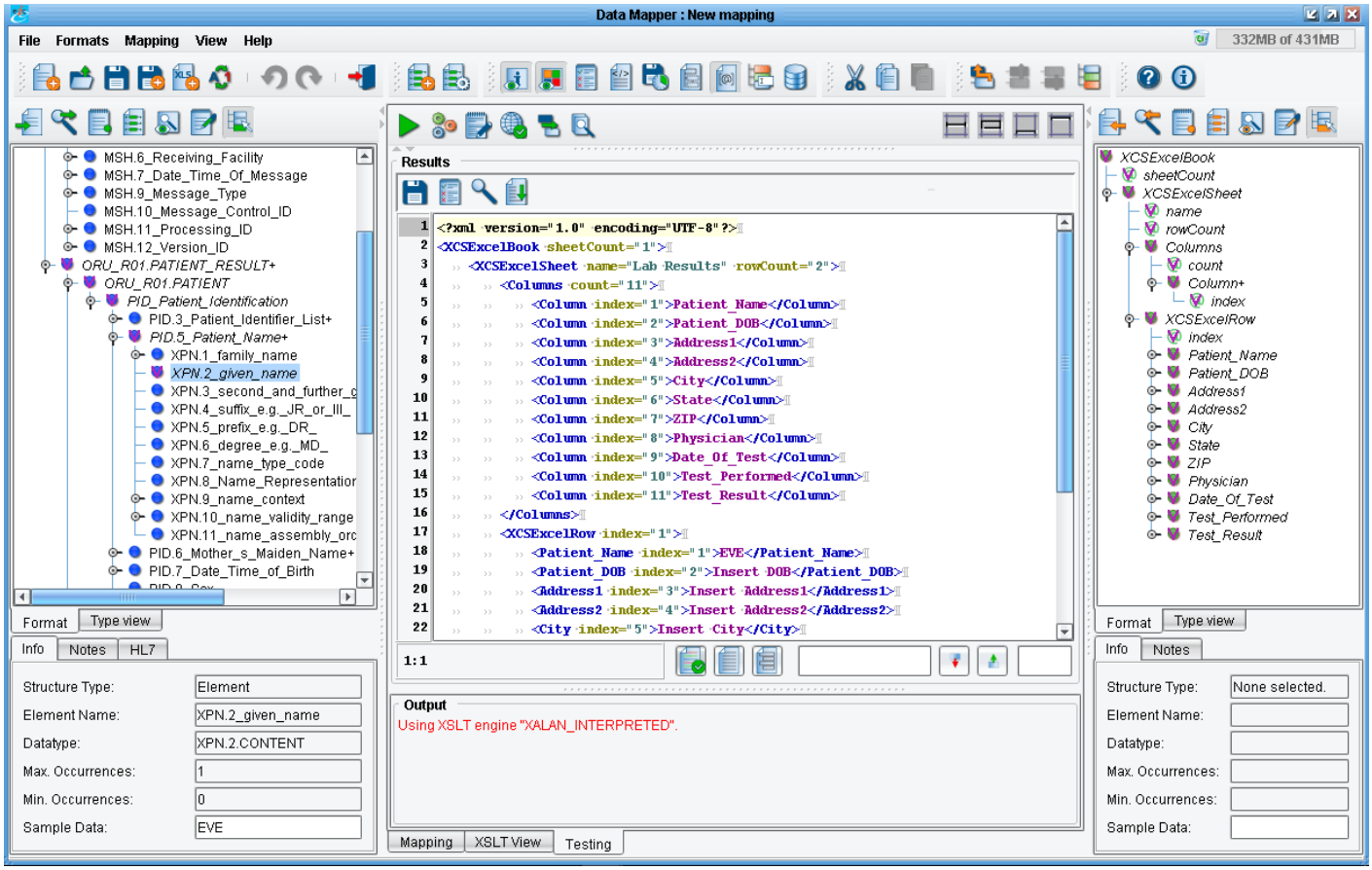


The Testing panel opens.

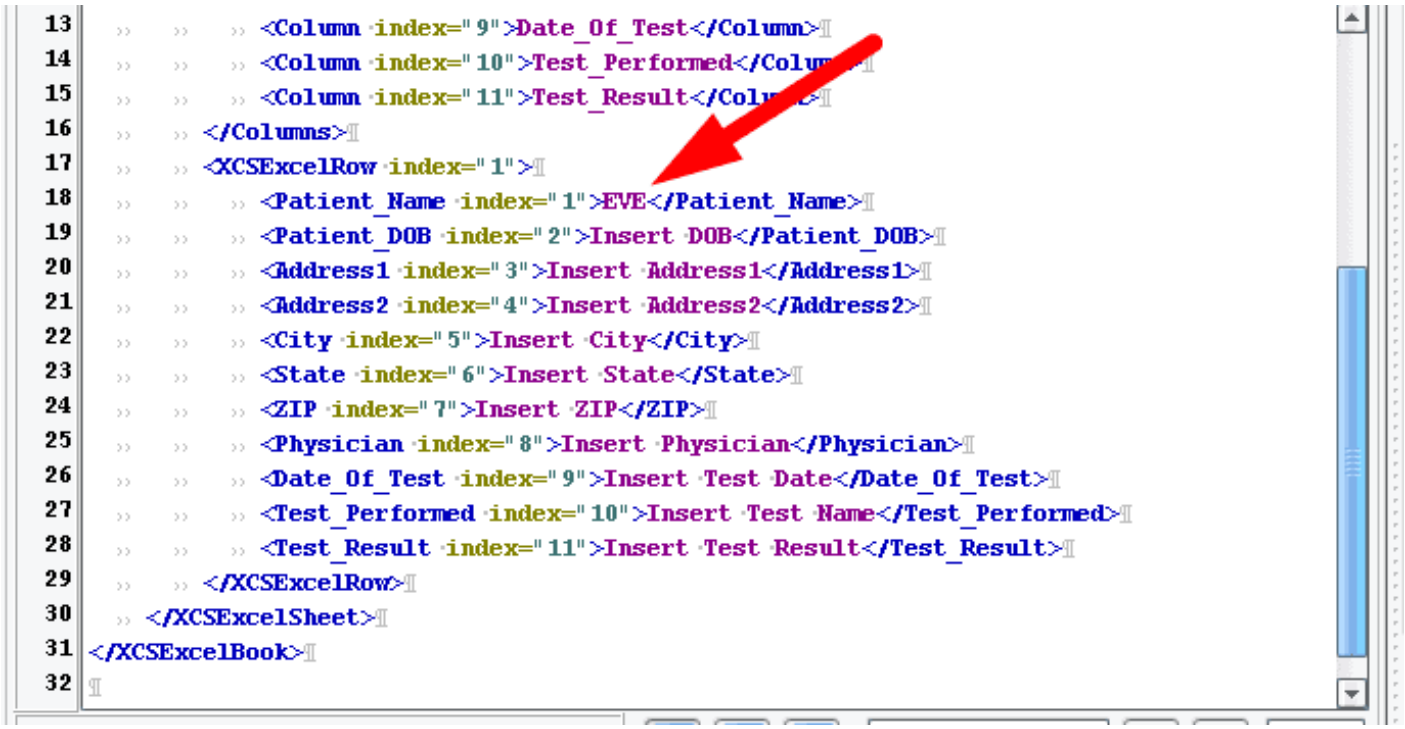
You'll see that the text area for the Source data has been pre-populated with an XML representation of our HL7 message. Click the **Gear** icon to run our transformation.



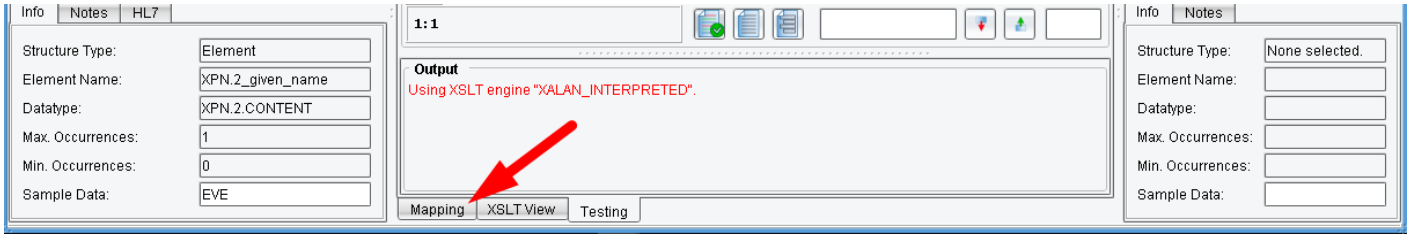
The results of the transformation will appear in the panel below. Clicking the **View Results** icon in the upper right will make this easier to read.



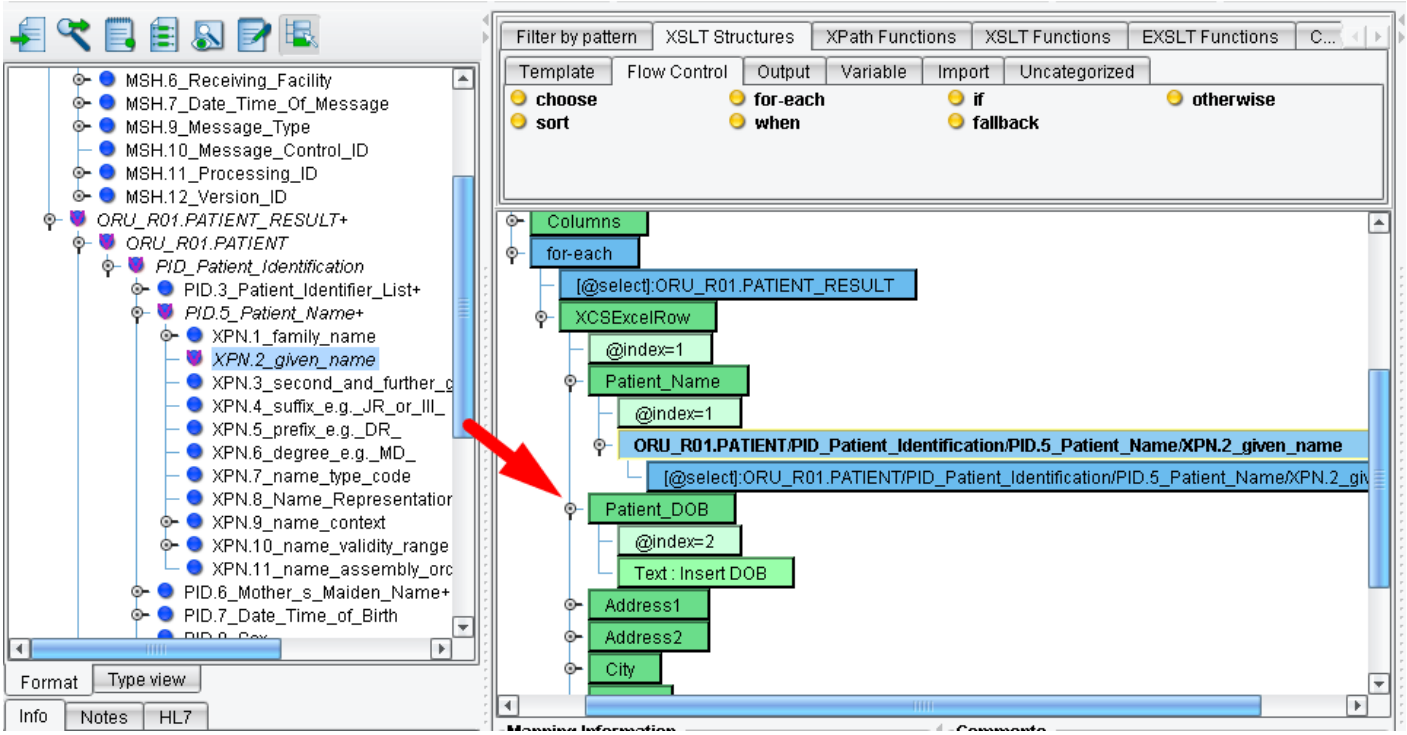
Here we see the panel opens up and indeed we get a better view.



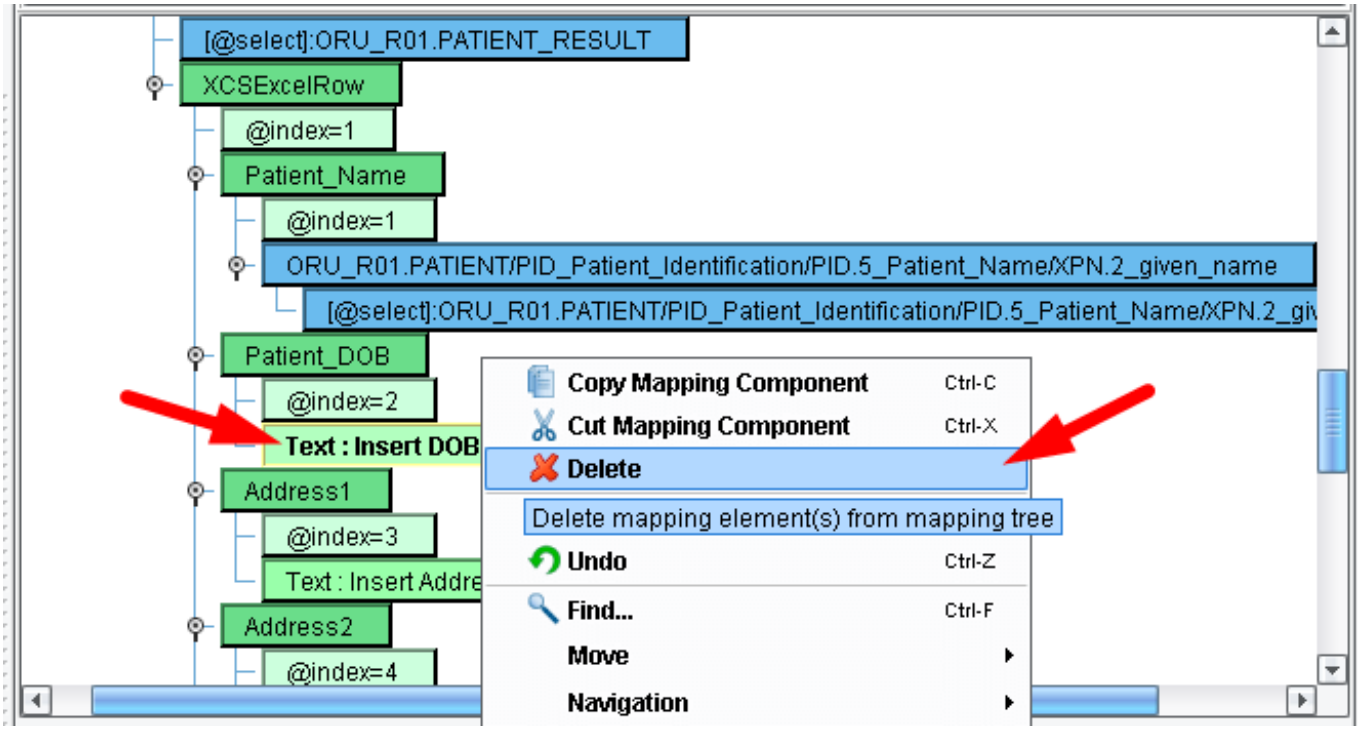
Scrolling down to line 18, we can see that the **Patient_Name** has been populated with the value **"EVE"**, which was in fact the patient's first name or given name from the HL7 message.



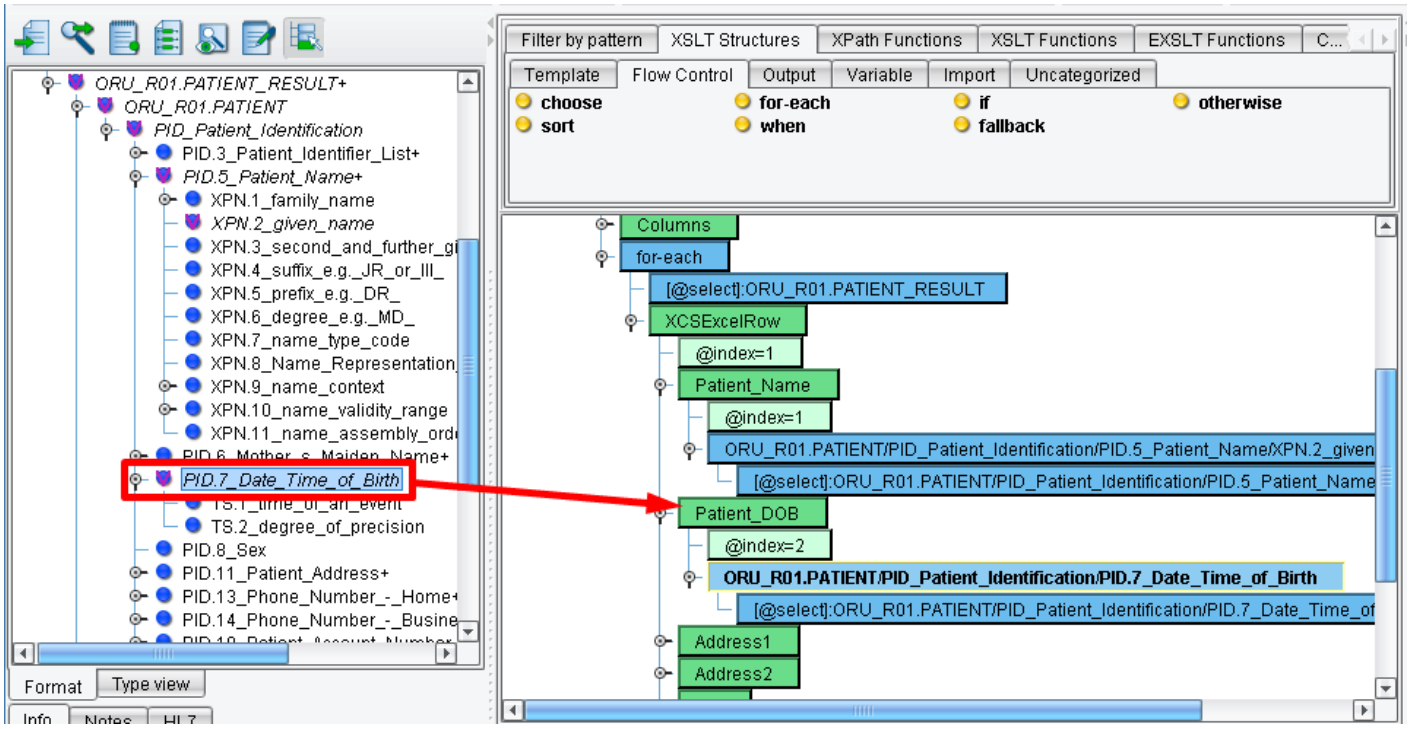
We can return back to the graphical mapping view by clicking the **Mapping** tab.



Let's map a few more fields. Expand the **Patient_DOB** node.

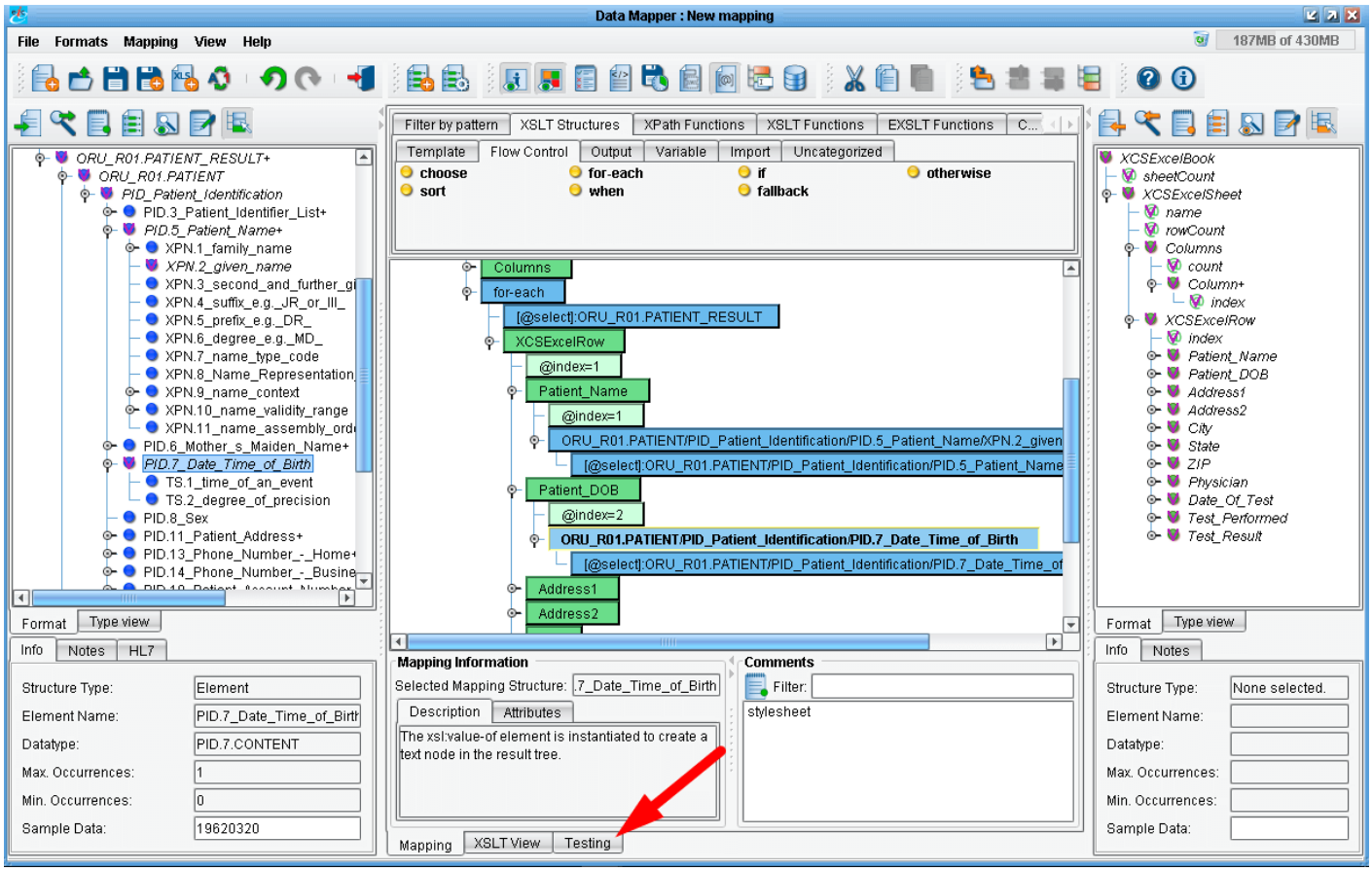


Left click the node **Text: Insert DOB ,** when the text gets bolder indicating it is selected, right click, and choose **Delete** from the context menu.

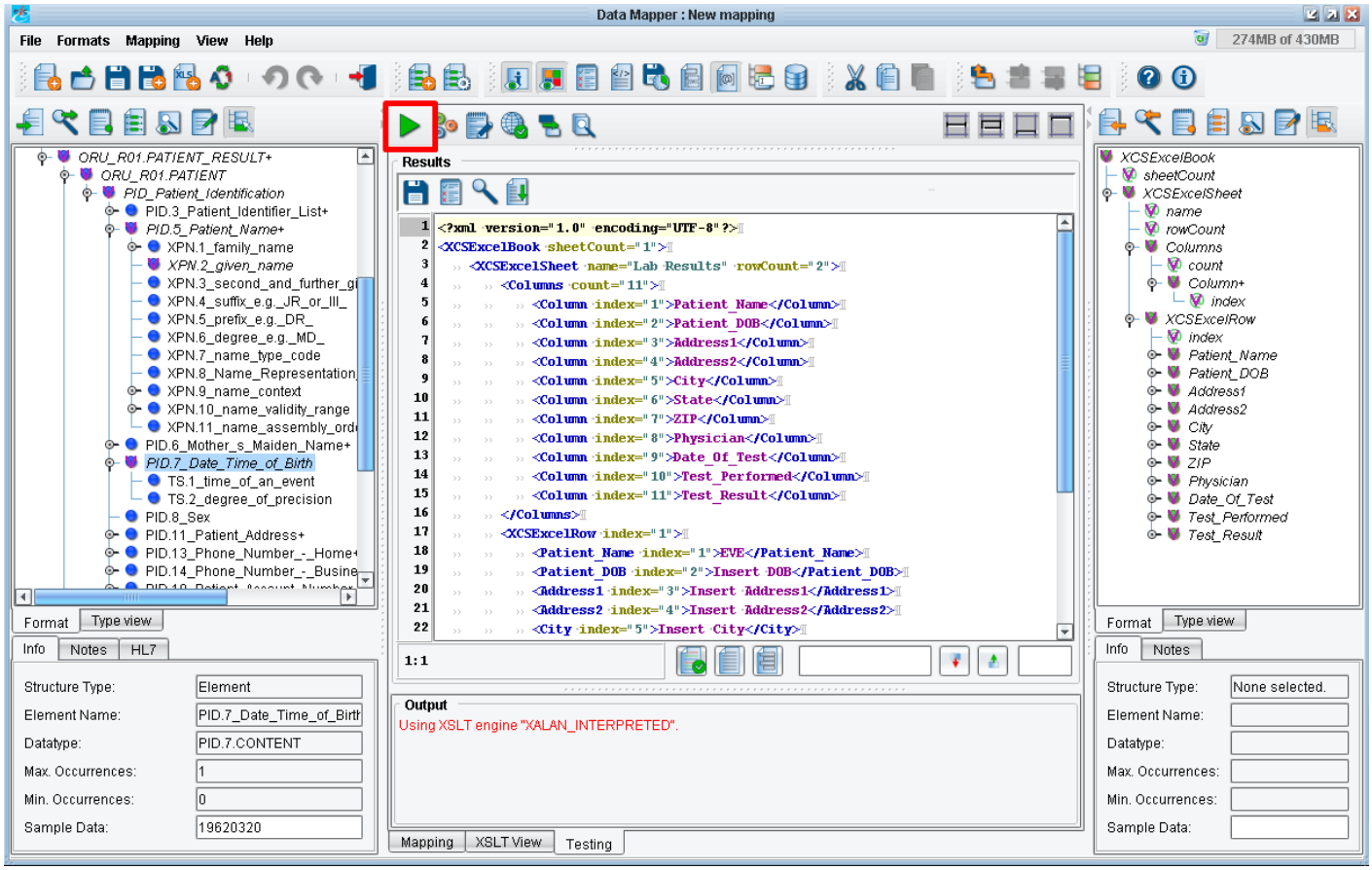


Scroll down and expand the **PID.7_Dat_Time_of_Birth** node to get a better view. This time, let's drag the **PID.7_Date_Time_of_Birth** onto the **Patient_DOB** node.

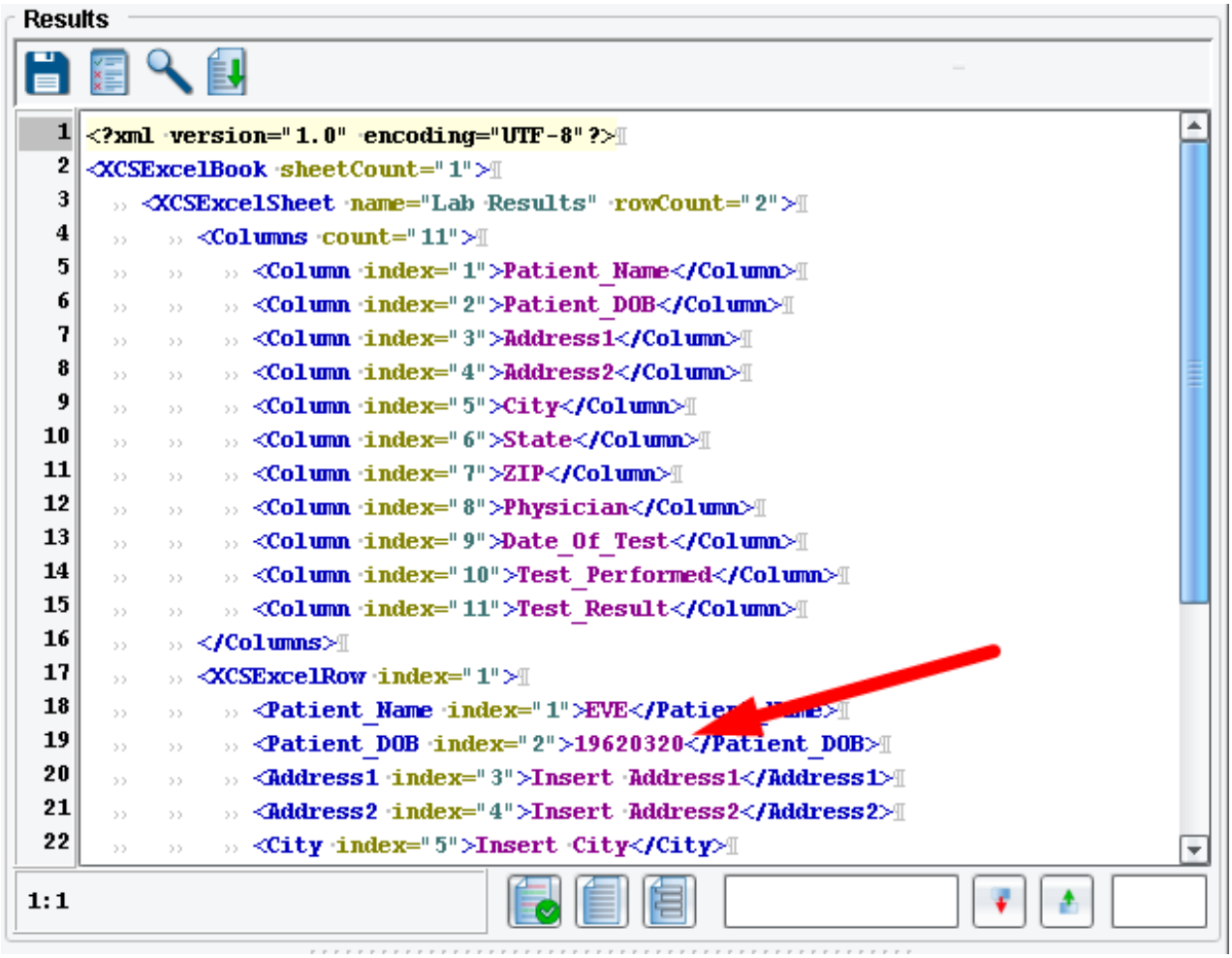
Once we've done that, drag & drop the **PID.7_Date_Time_of_Birth** onto the **Patient_DOB** node. Your mapping should now match that above.



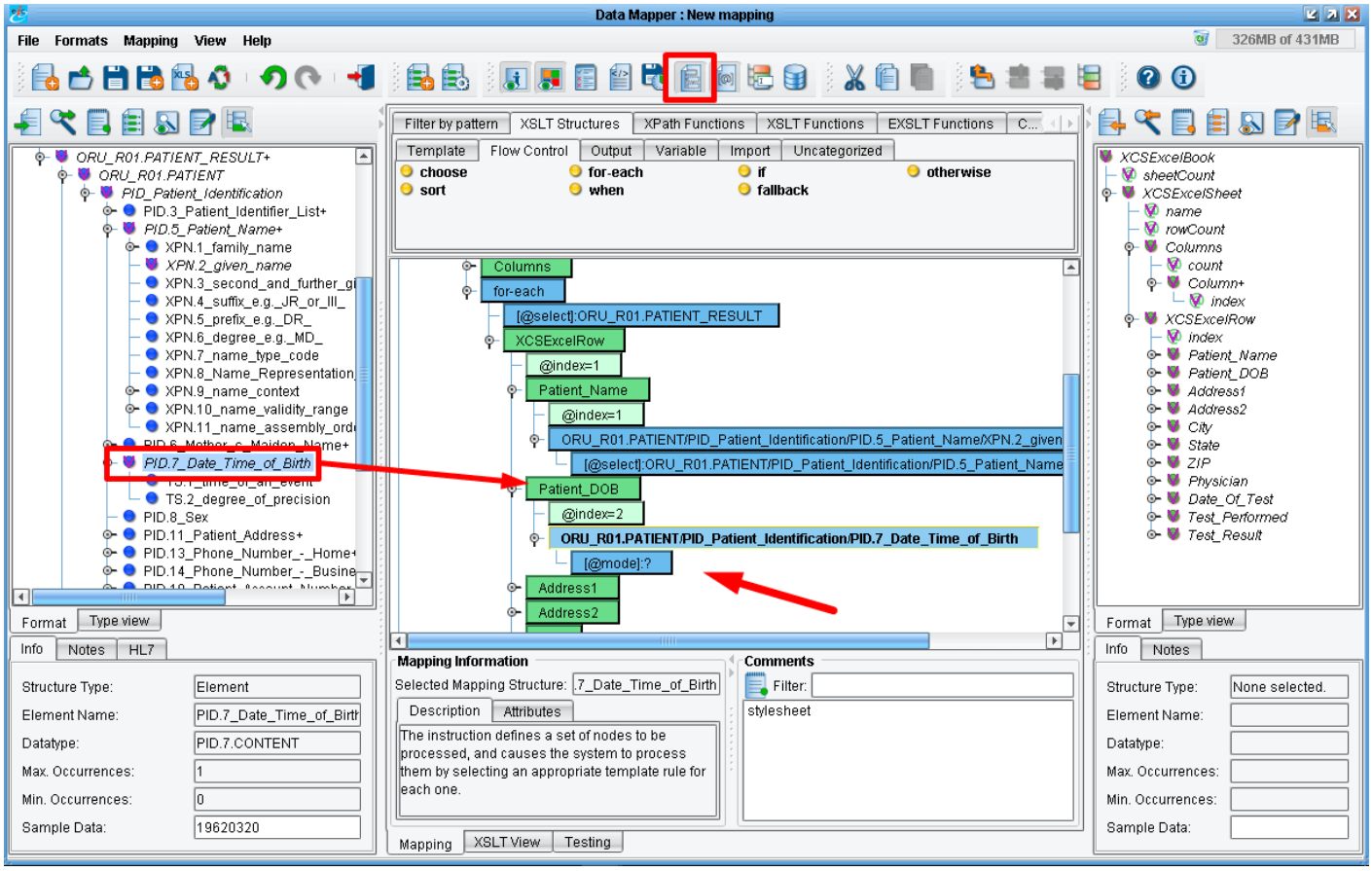
Again, you can click on the **Testing** tab.



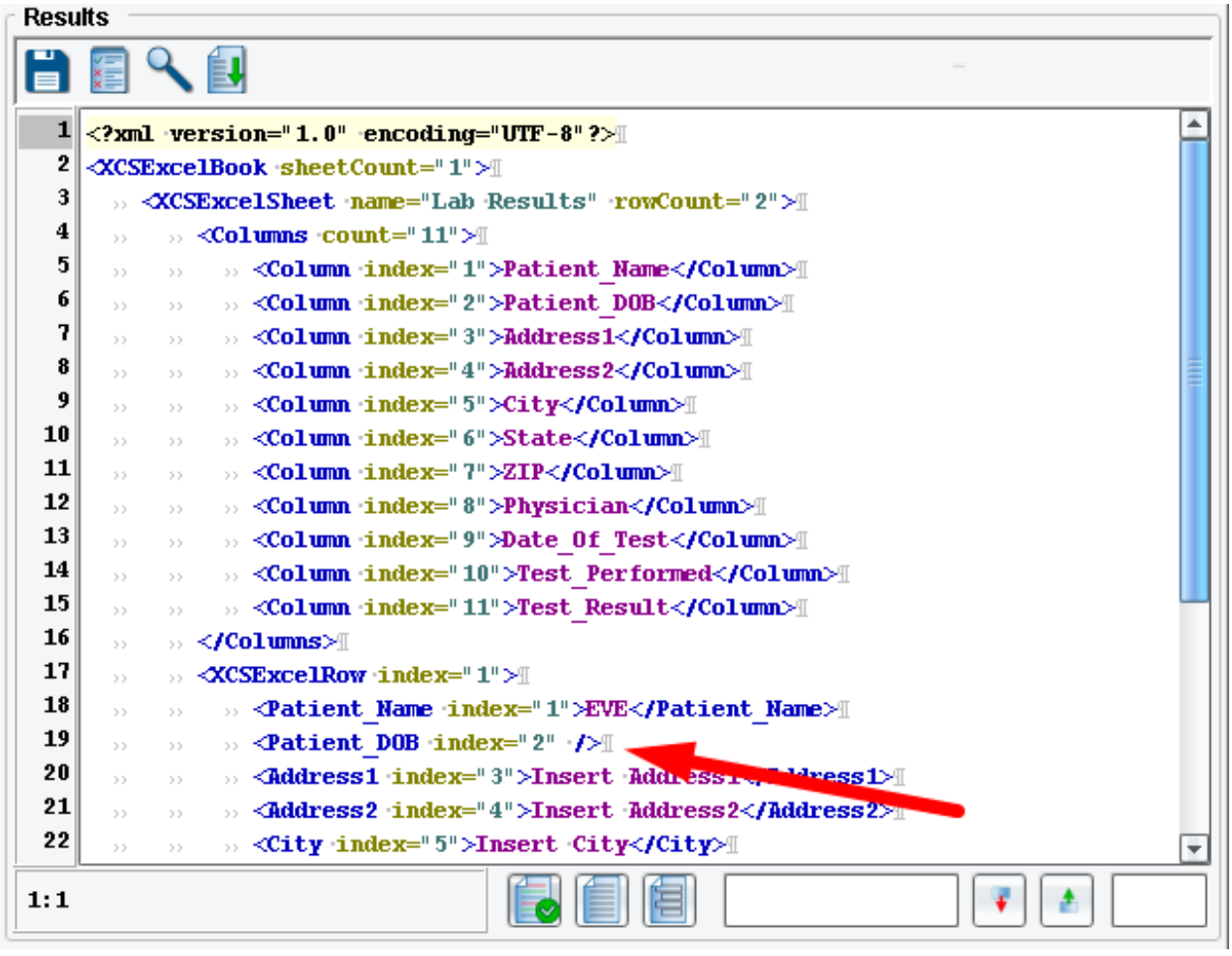
Once the panel opens click the **Gear** icon.



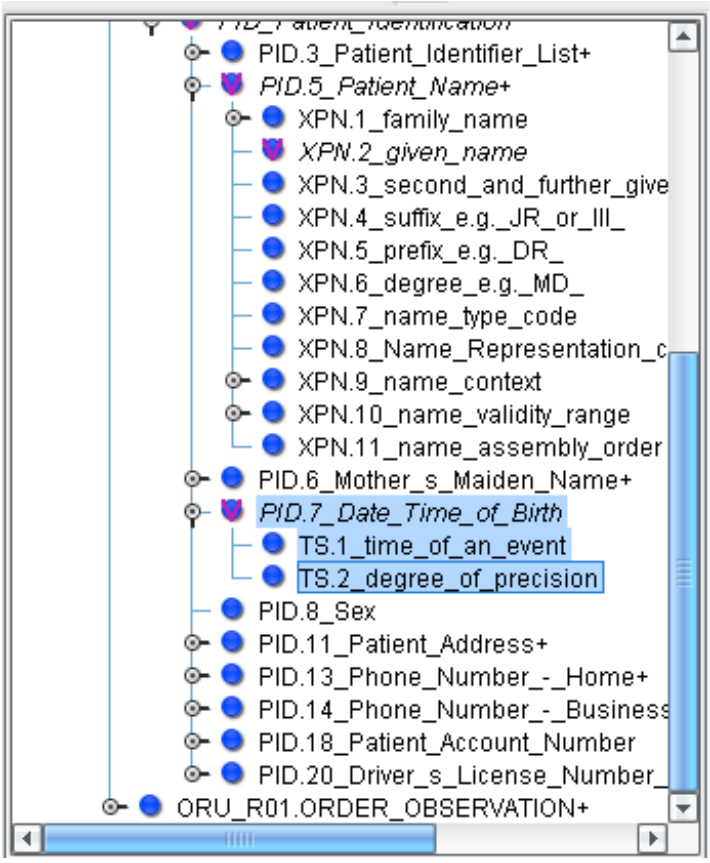
The **Patient_DOB** element appears as expected.



But note! In the toolbar at the top of the mapper you can see the **tooltip** icon indicating that it "enables or disables apply template for complex types". If it's selected and the value (**PID.7_Date_Time_of_Birth**) is draged, your mapping would look like above.

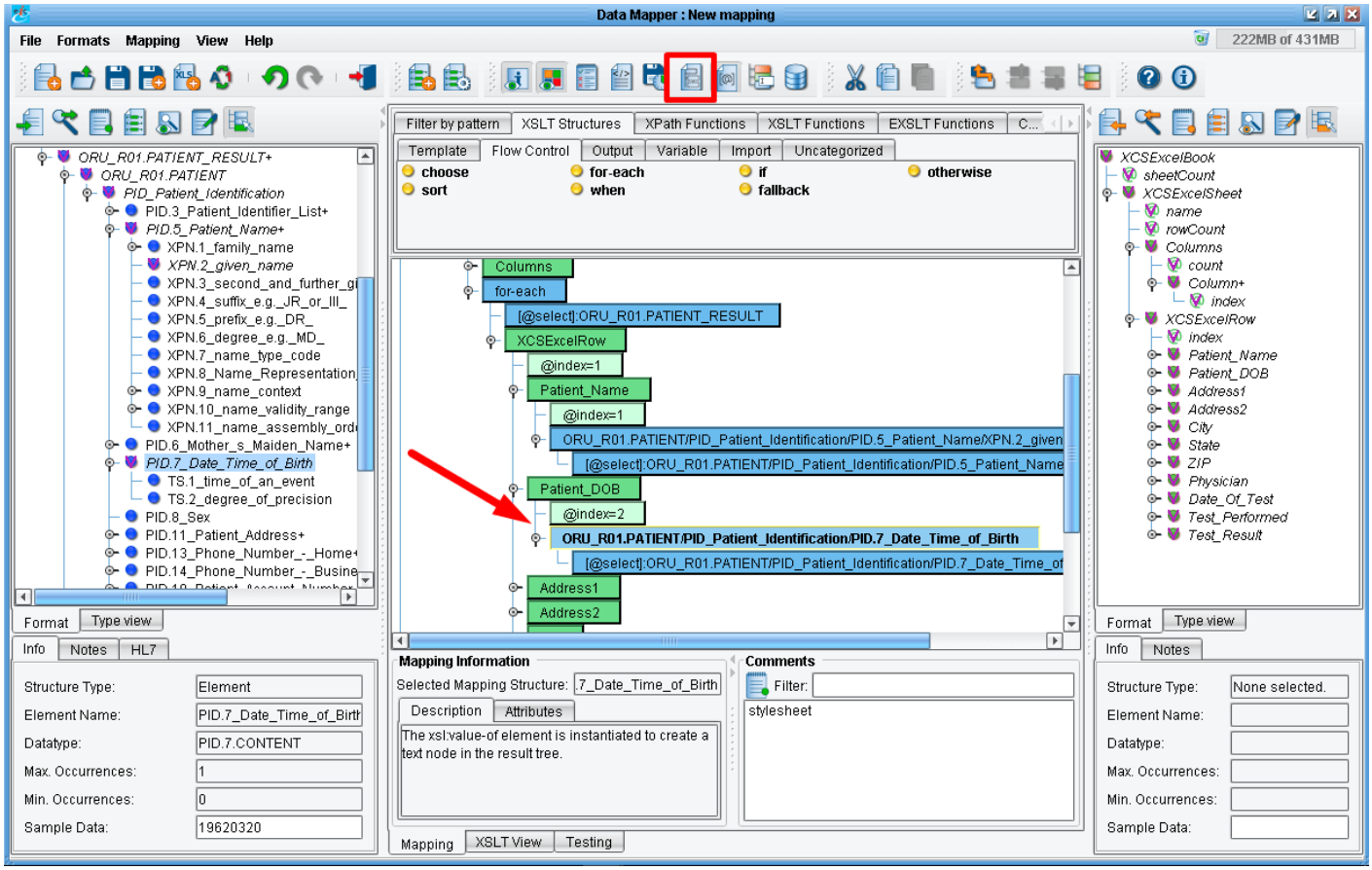


In this case, after running the test you'd see that the patient date of birth was not populated.



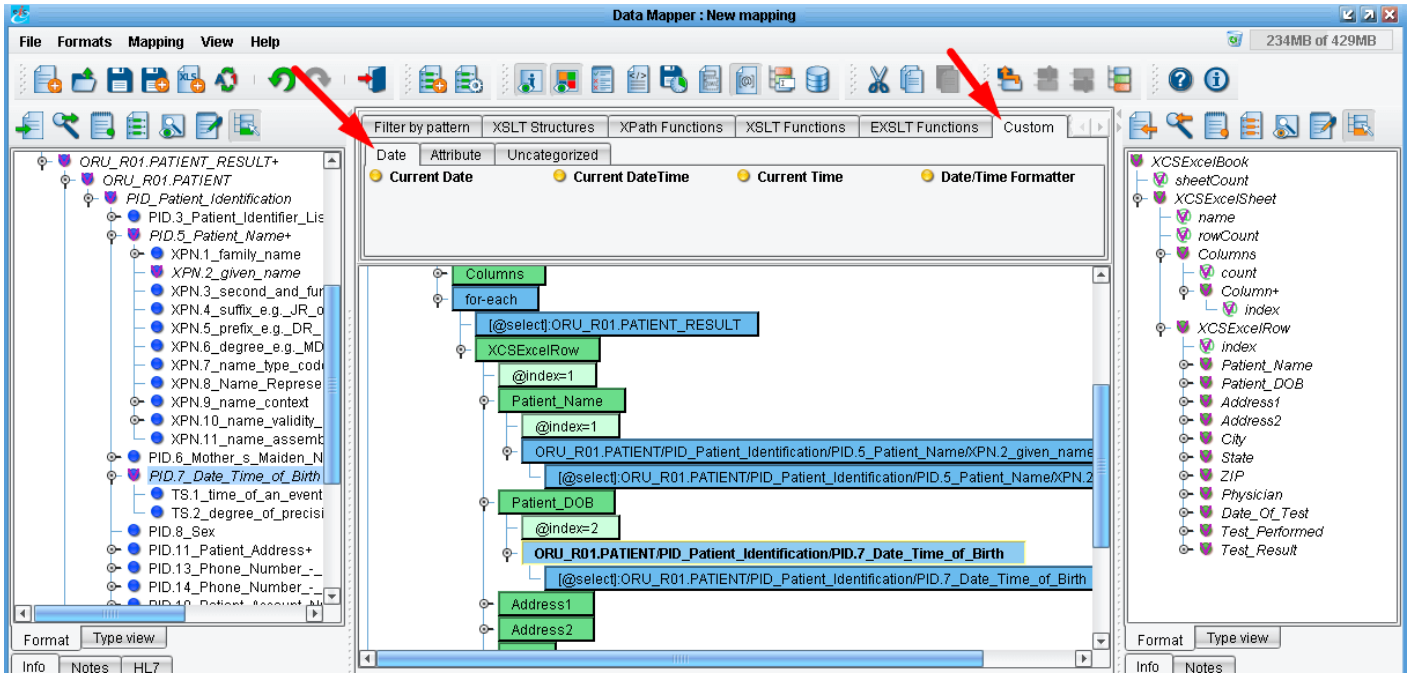
Let's take a look at what happened.

PID.7_Date_Time_of_Birth is a complex node that has 2 children underneath it. If the **Enable or disable apply template for complex types** button is selected, the data mapping expects one-to-one mappings to be handled with leaf nodes of the tree. We'll want to disable this behavior so that we can take the full value of the **Date_Time_of_Birth element** and use it to populate **Patient_DOB**. So check the condition of the **tooltip** icon.

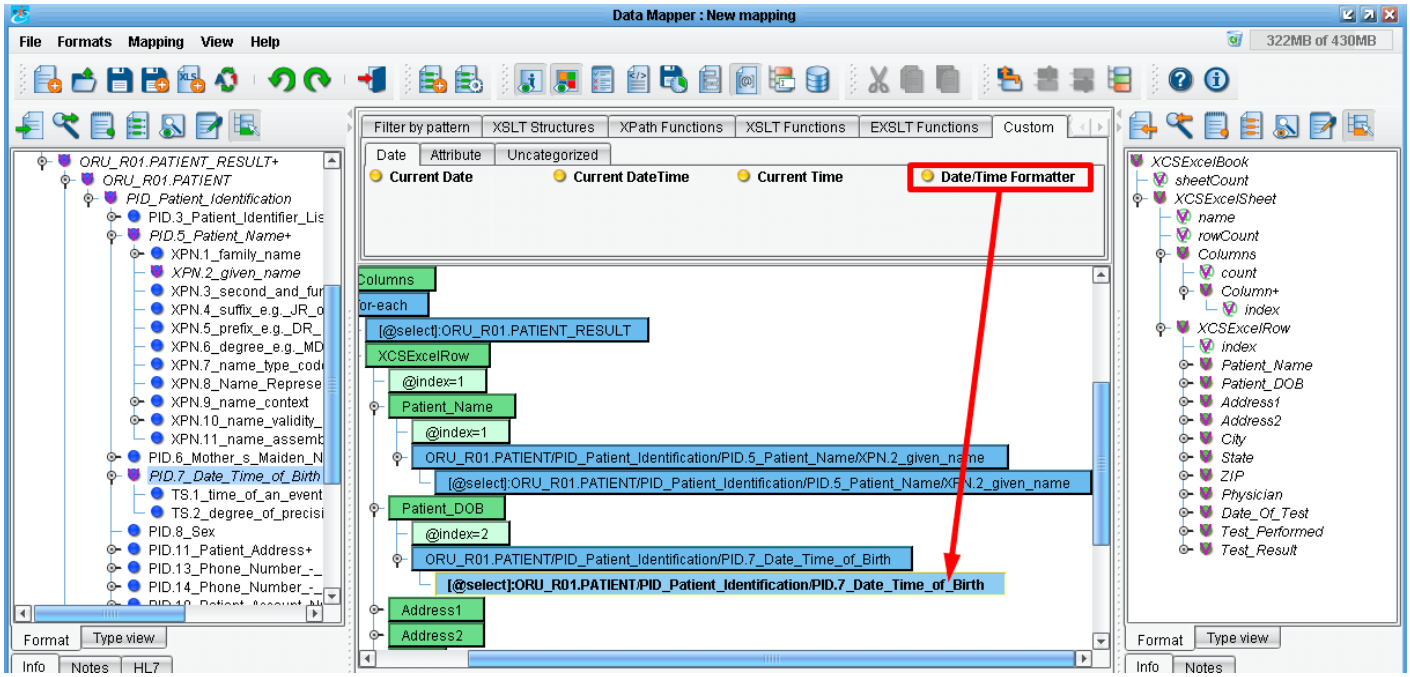


Now the **tooltip** icon is deselected and our mapping looks like above.

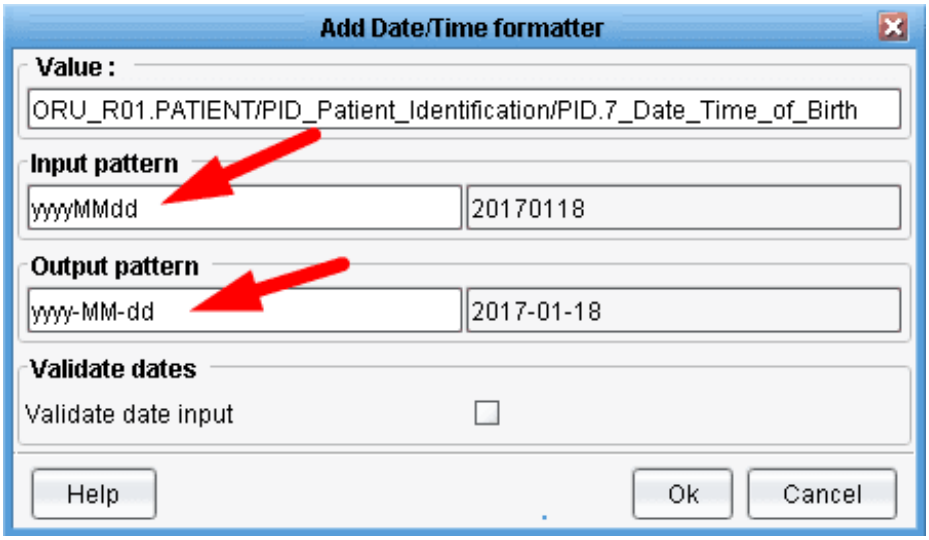
Let's says we want to add some formatting to the Patient_DOB. Click the **Mapping** tab to go back to the Data Mapper.



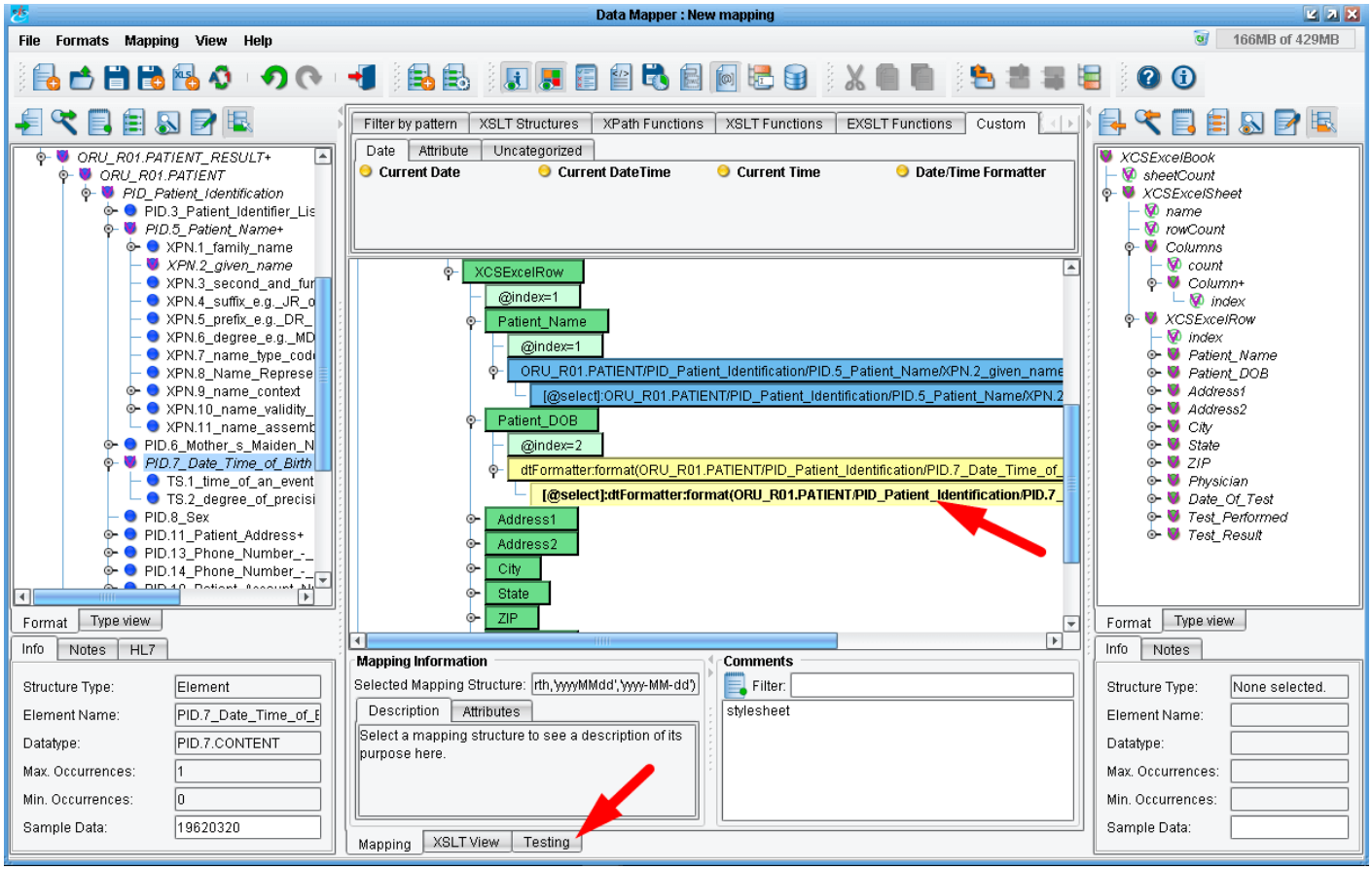
Now, navigate to the **Custom** tab, you will need to scroll to the right. And then select the **Date** sub-tab.



Again, scroll over so you can choose the **Date/Time Formatter** and drag it onto the newly created blue node for the **Date_Time_of_Birth (the second blue node).** Here watch for a yellow bar above the node to make sure you are mapping to the right location. Once it's dropped in the correct position it will launch the **Add Date/Time formatter** dialogue.

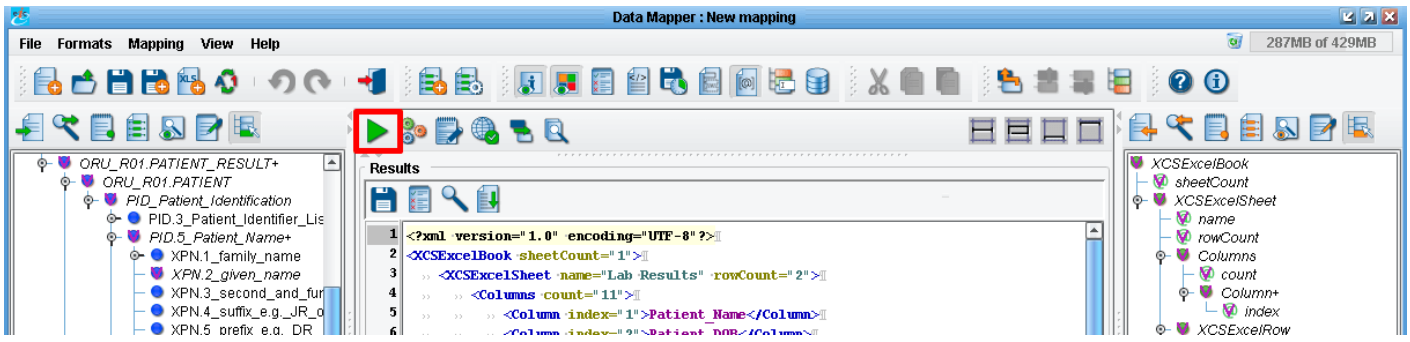


The formatter dialogue opens. The **Add Date/Time formatter** dialogue is where we can add our expected input pattern **"yyyyMMdd"** and our desired output pattern **"yyyy-MM-dd"**. Enter the text as shown, then click **OK**.

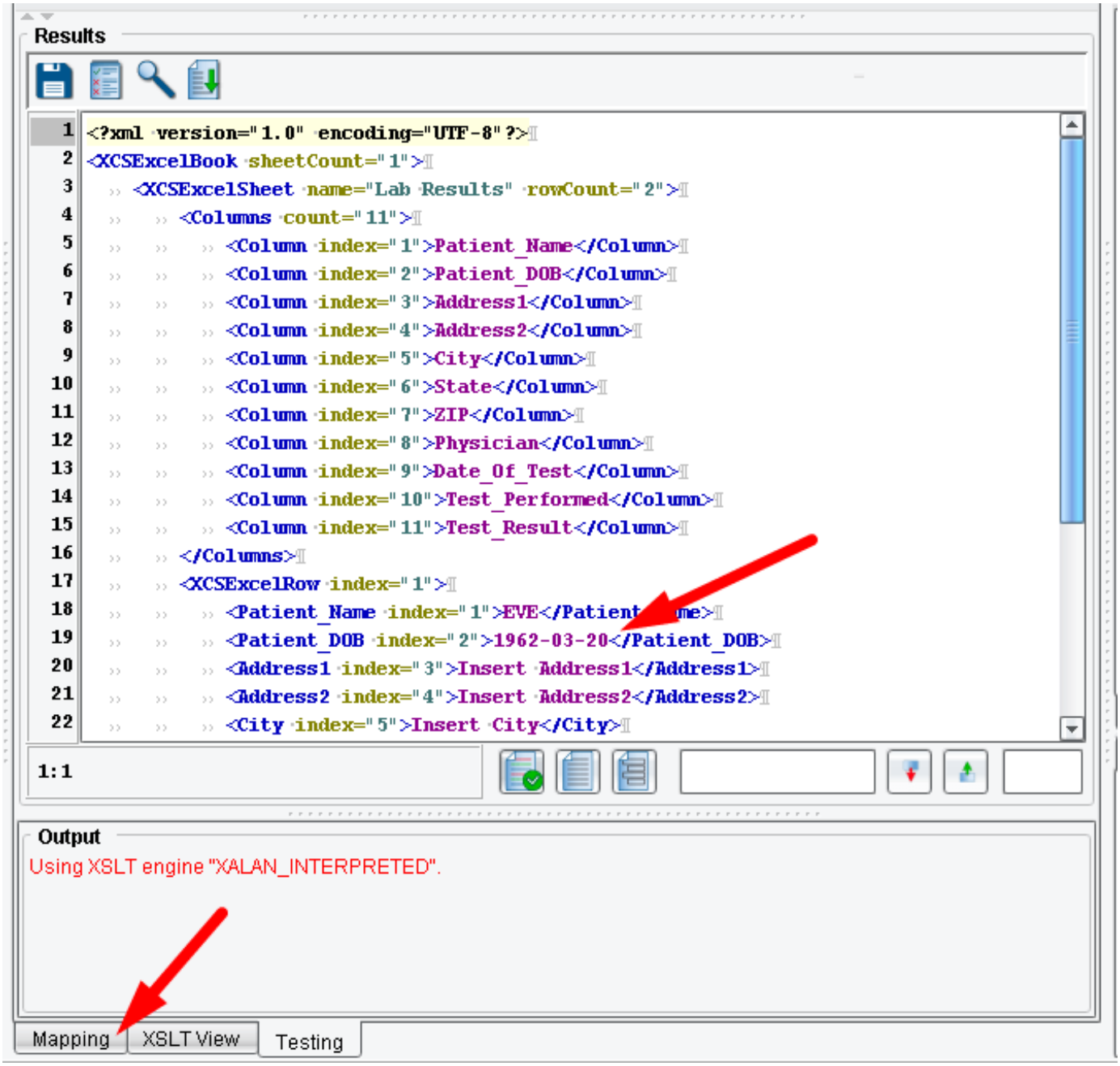


You'll see that the blue node is replaced by a yellow node indicating that we're doing some special work with that value.

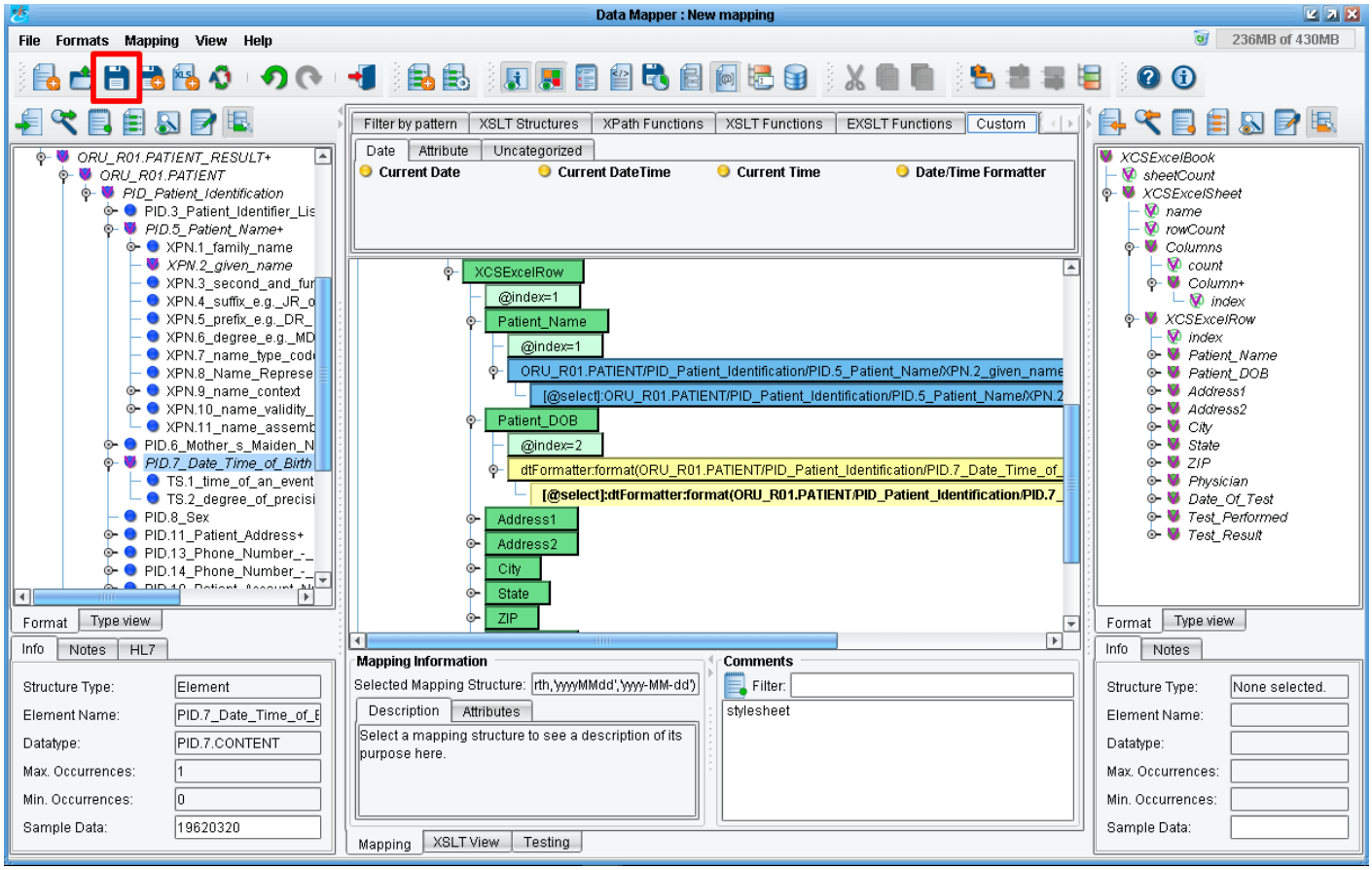
One more time we'll click on the **Testing** tab.



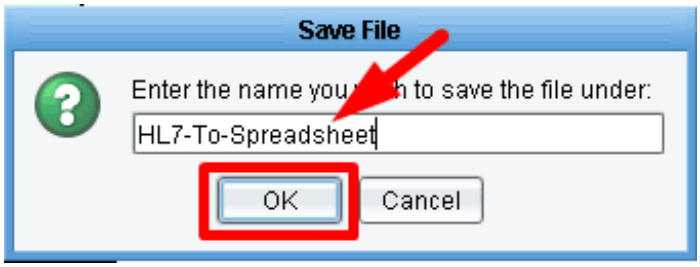
And then click the **Gear** icon to run the test again.



You'll see that we've formatted the Patient_DOB, **1962-03-20**. Next, click the Mapping tab to return to the Data Mapper.

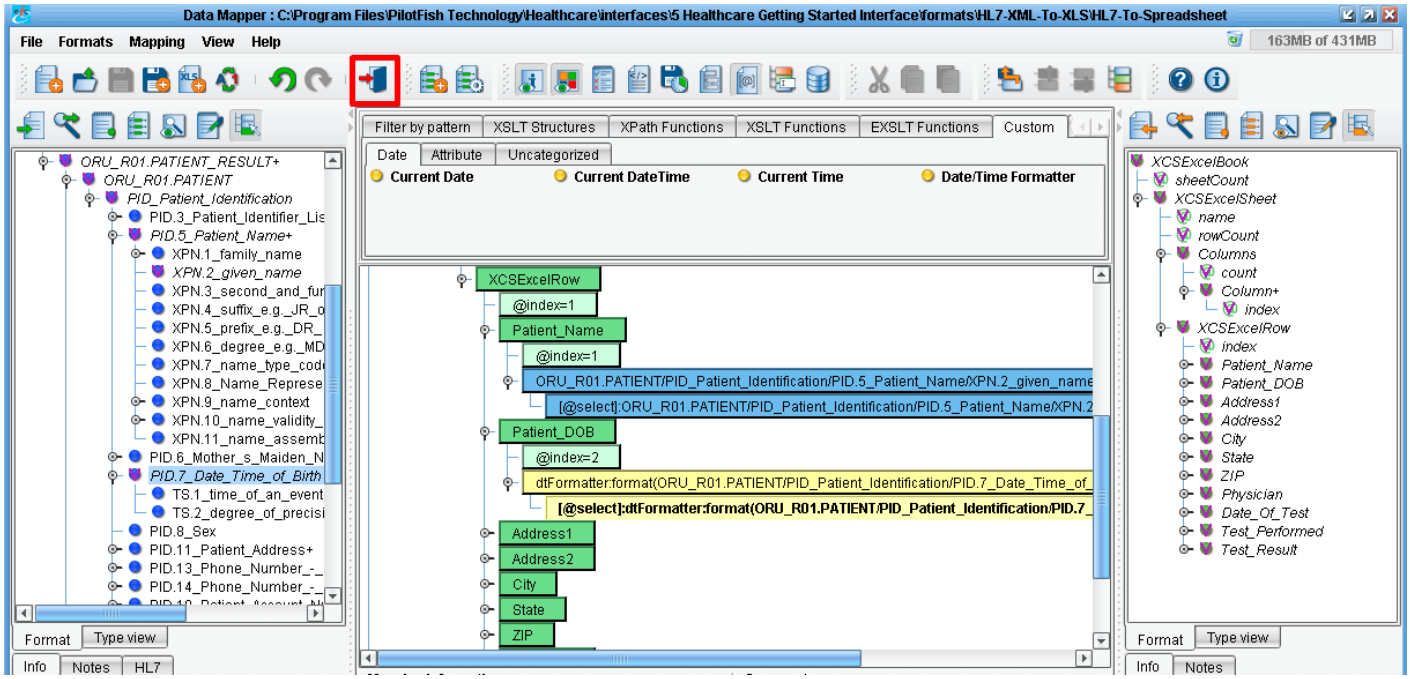


Back at the Data Mapper, feel free to continue to map additional fields as you see fit. When you're done, click the **Save current mapping** icon.

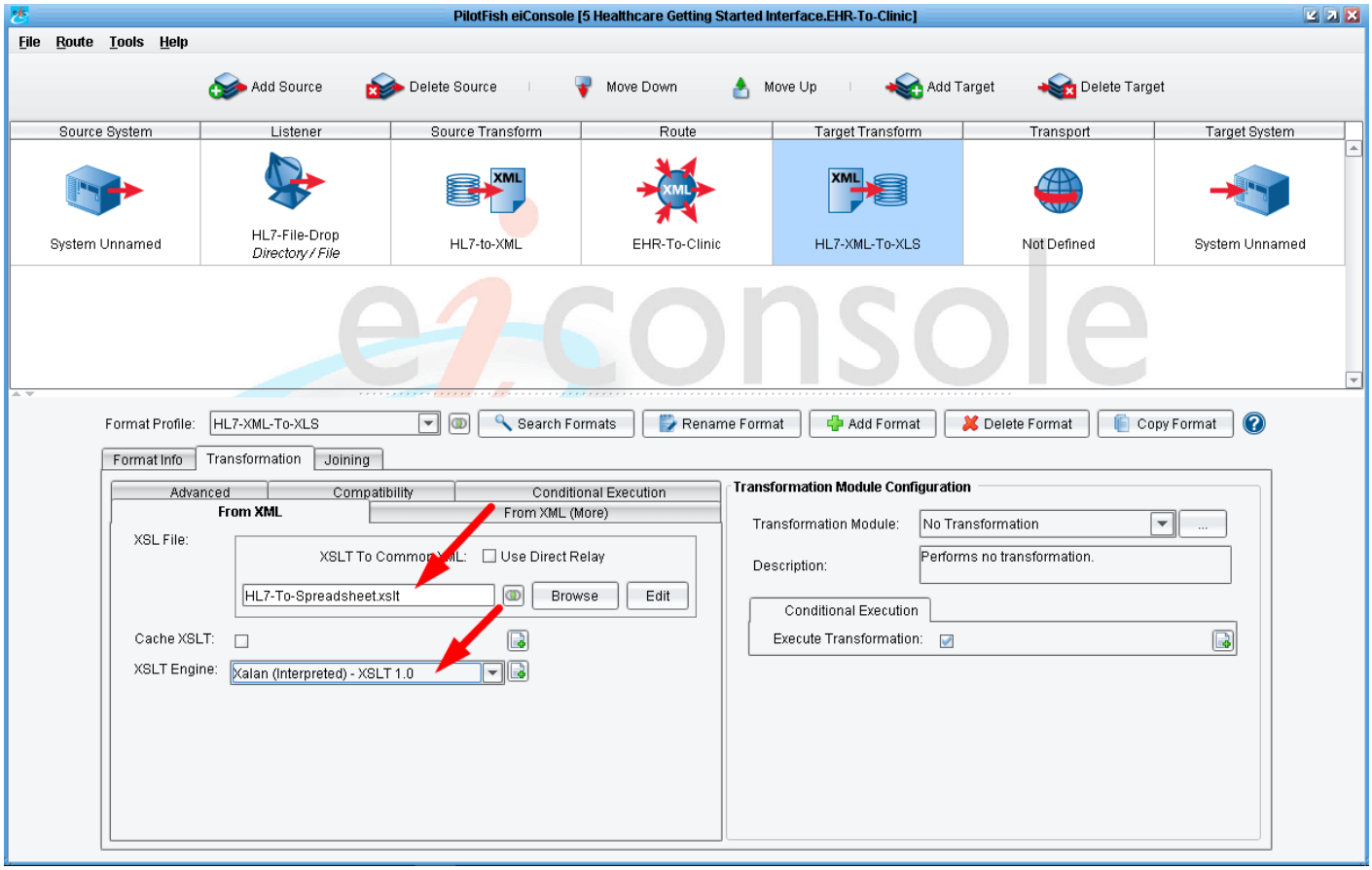


Once the **Save File** dialog opens give the mapping a name. We'll call this **"HL7-To-Spreadsheet"** and click **OK**.

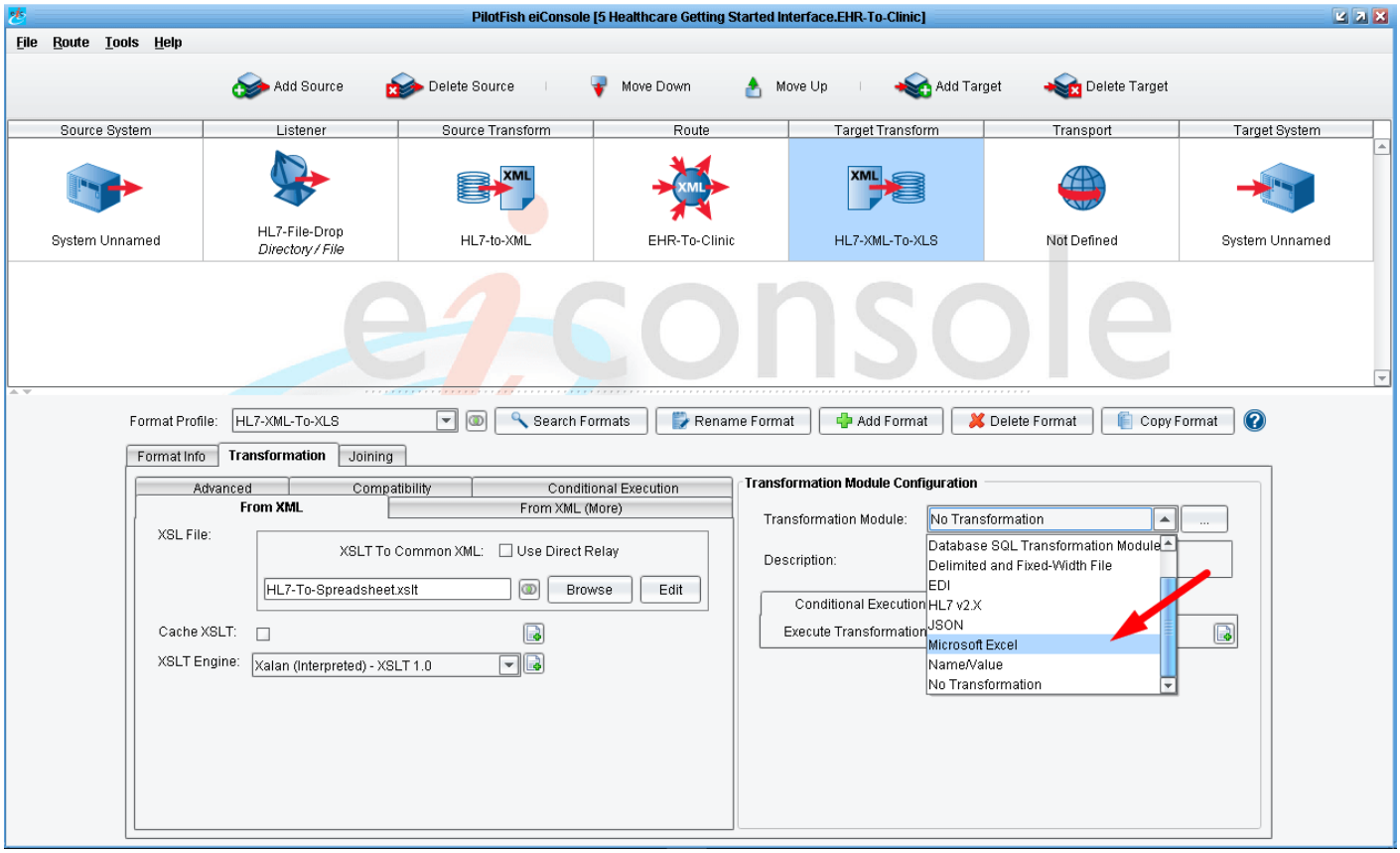
Note: these get stored in your Working Directory "formats" folder.



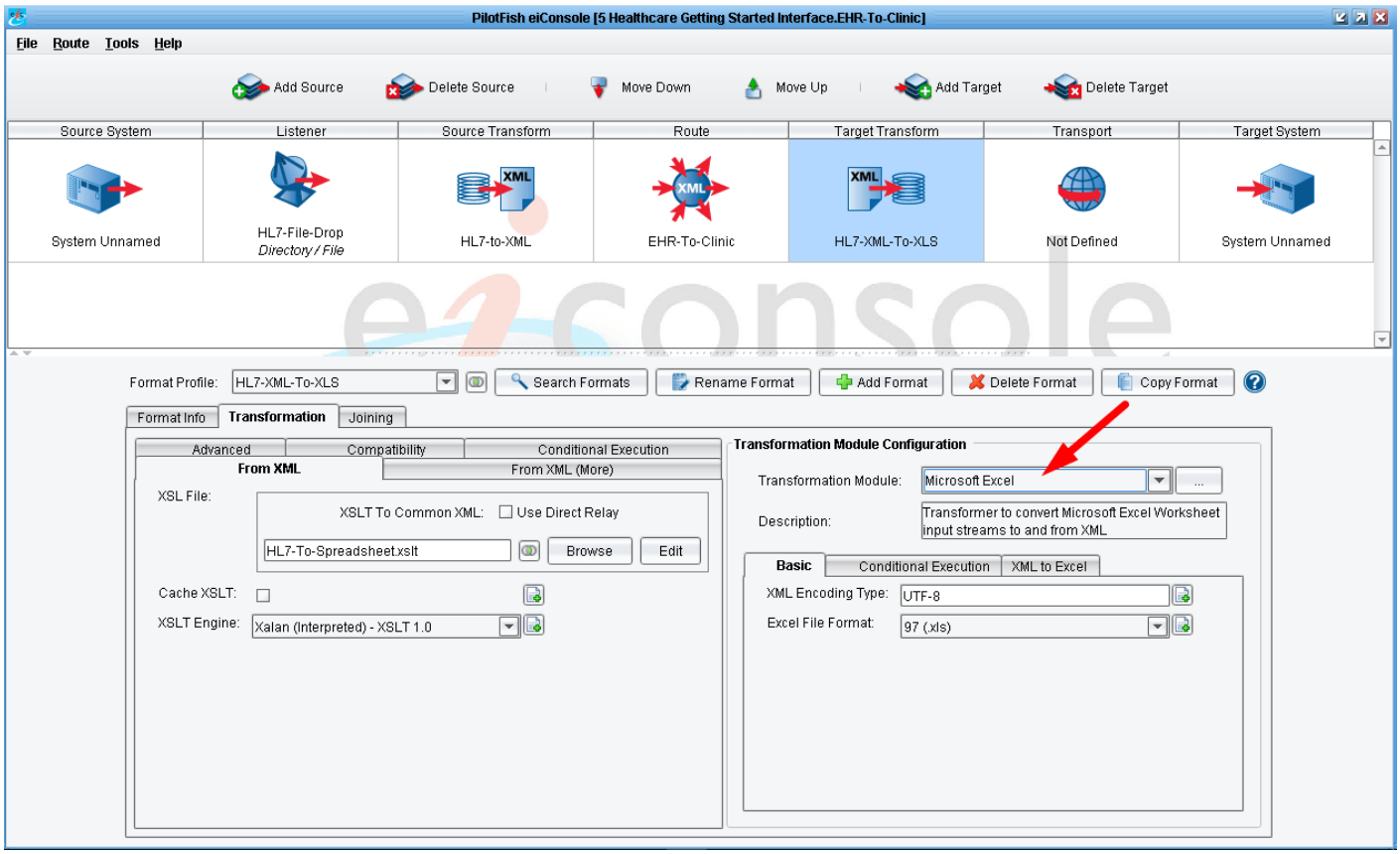
Next, click the **Return to Console** icon.



When the Main Route Grid opens you'll see the name of the newly created HL7 file appears in the XSLT Configuration area. Make sure that the **Xalan interpreted** drop down menu item is selected.

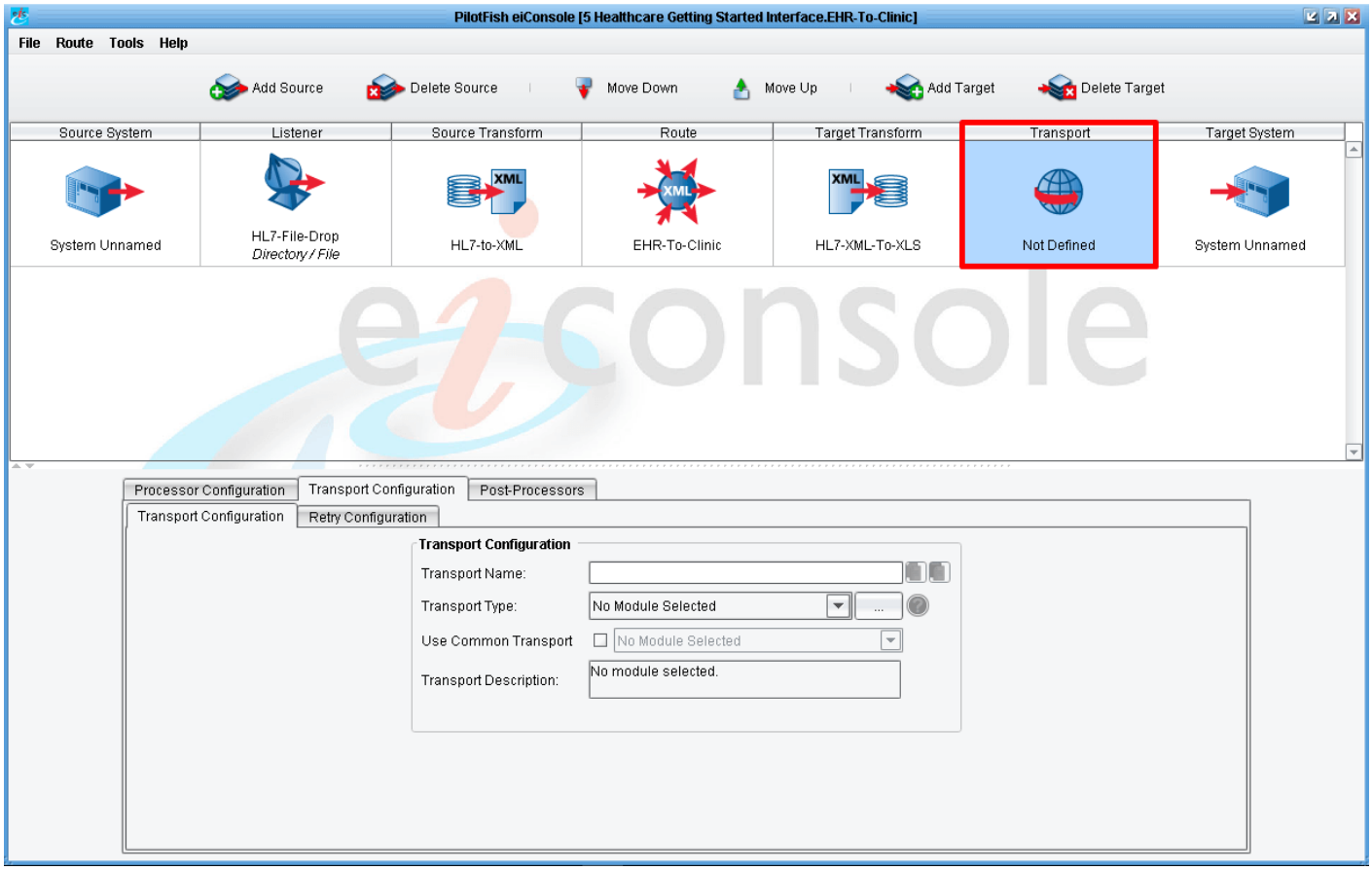


Now, the output of our data mapping will be an XML representation of an Excel spreadsheet, but we need to actually convert that into the XLS binary format. To do this, we'll choose the **Microsoft Excel** transformer from the **Transformation Module** drop down.

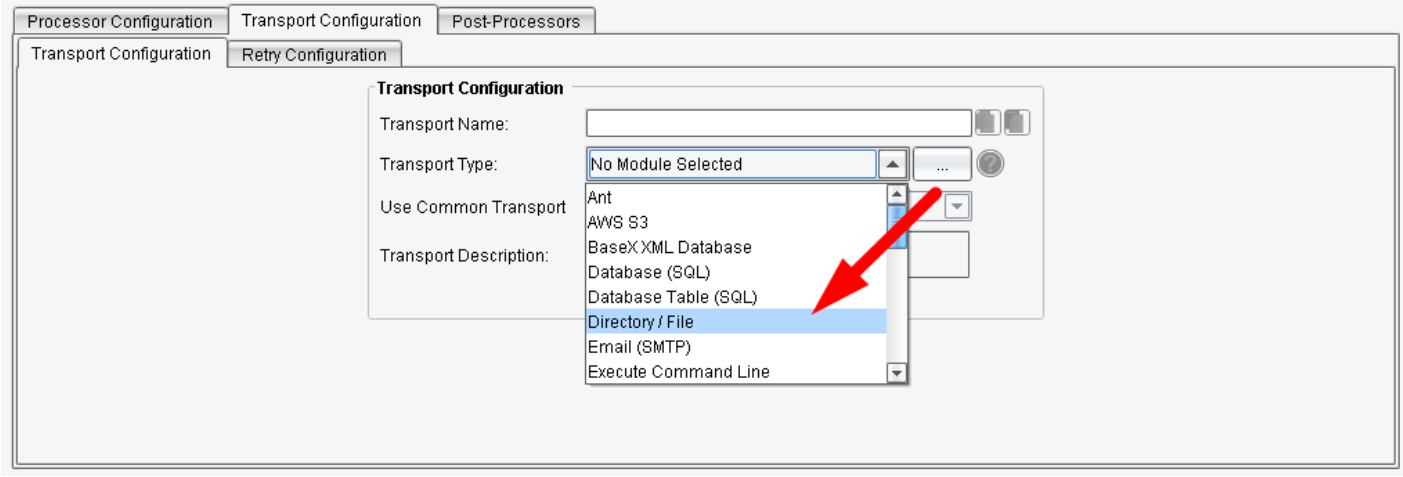


Select this, no further configuration is required.

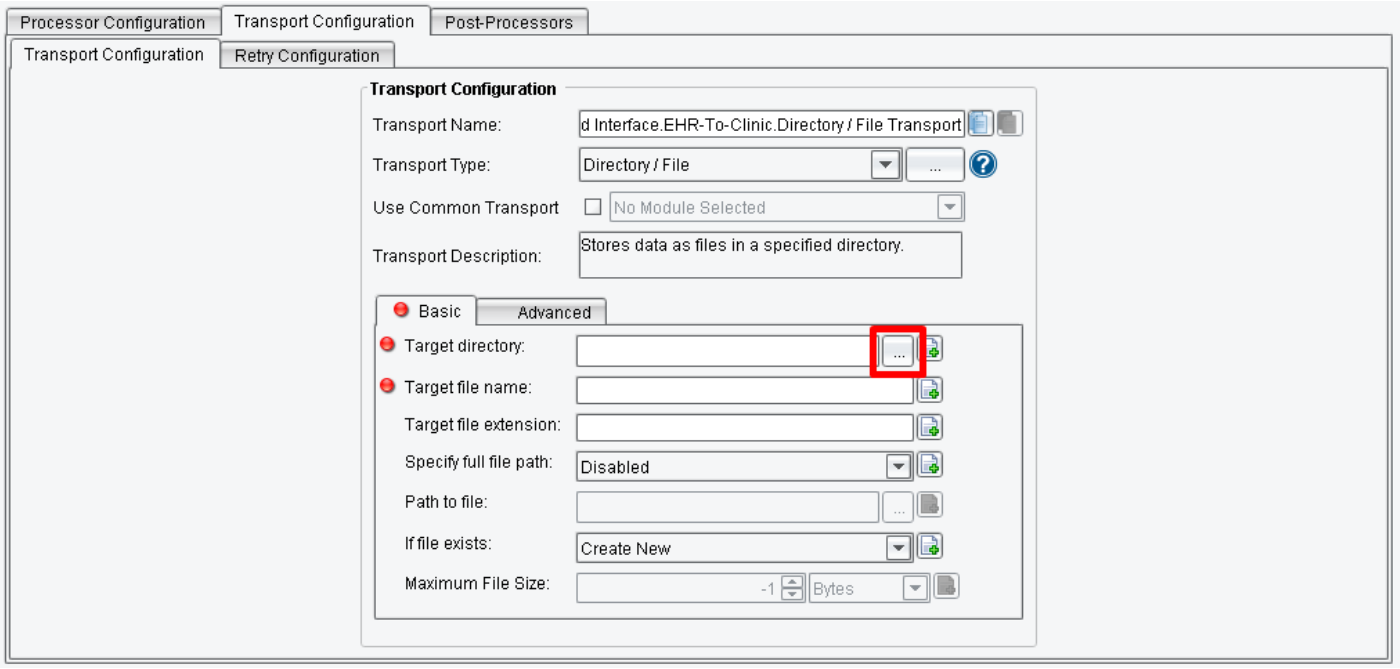
The Transport Stage



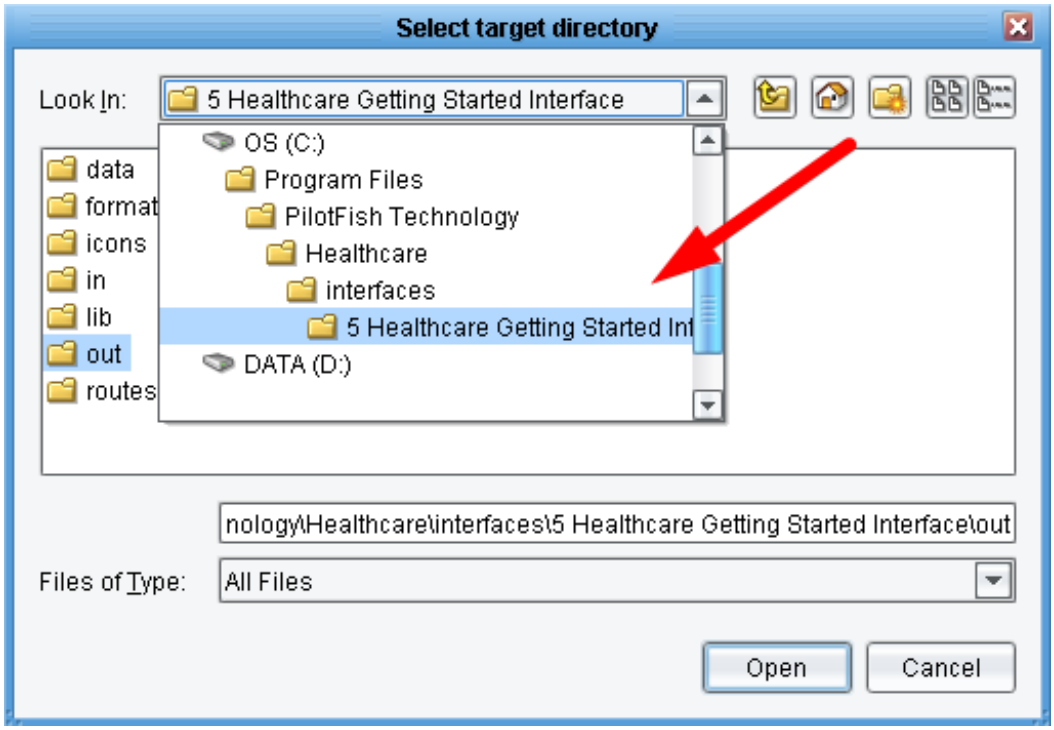
Next, click on the Transport stage. Here we'll configure how we'd like to Transport the data to our Target system. In this case we're just going to be dropping a file into a directory.

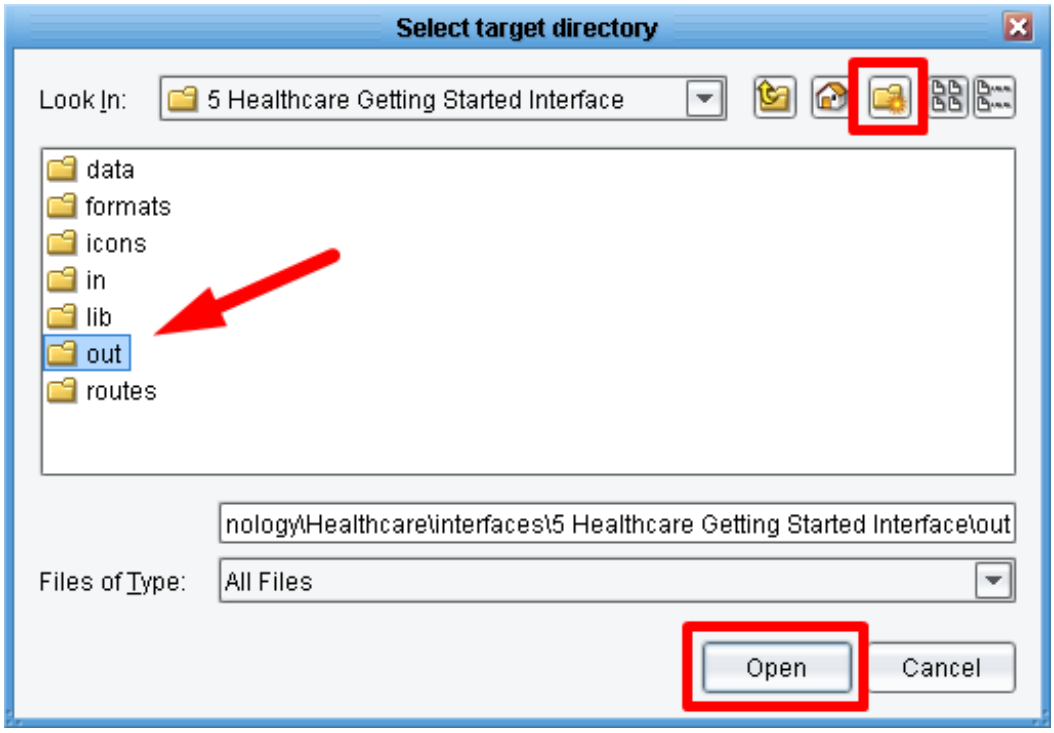


From the **Transport Type** drop down choose **Directory / File**.

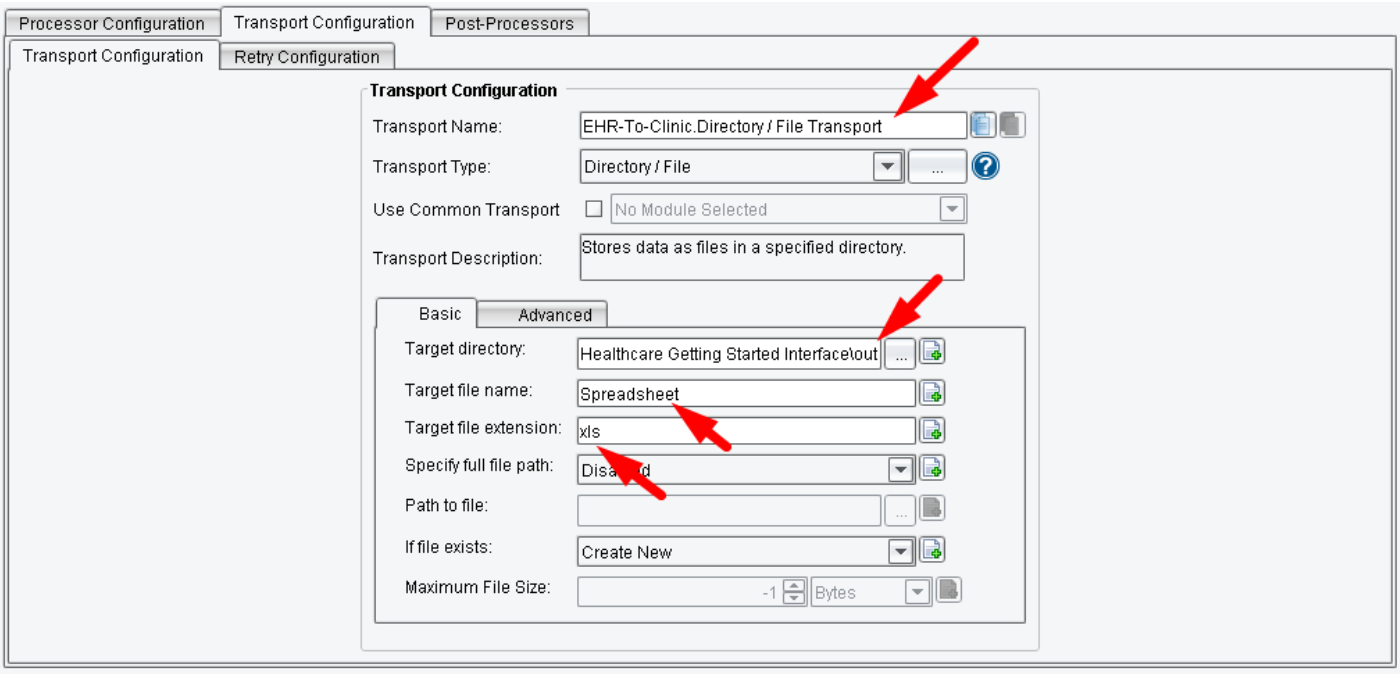


This opens the **Transport Configuration** panel. Once it opens we'll fill in the required information, but first, click the **Ellipsis** button next to the Target directory.





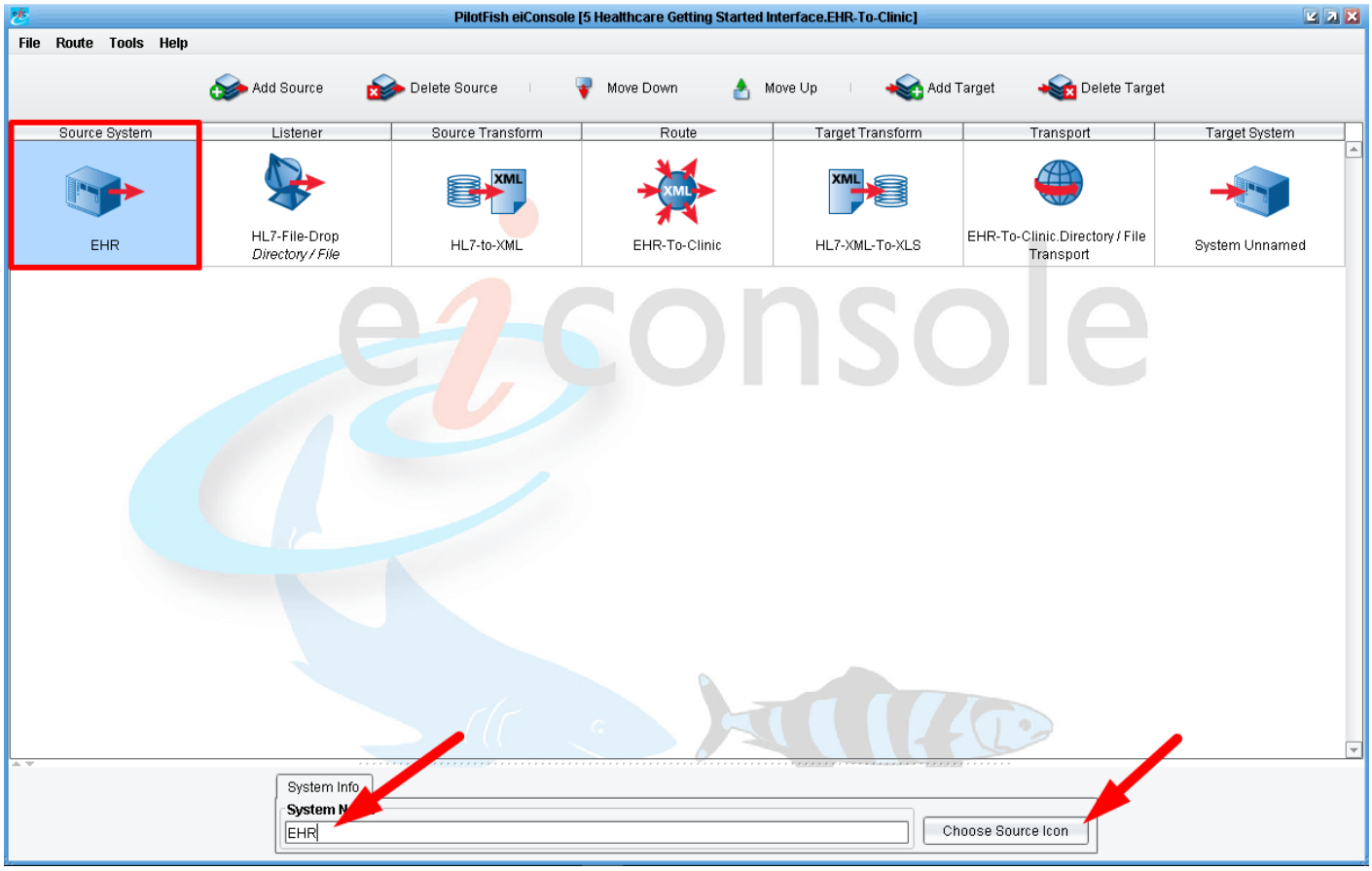
For consistency in this tutorial select **{working directory}/interfaces/5 Healthcare Getting Started Interface**. Click the **New Folder** button and create a new folder naming it **"out"**. Select it and click **Open**.



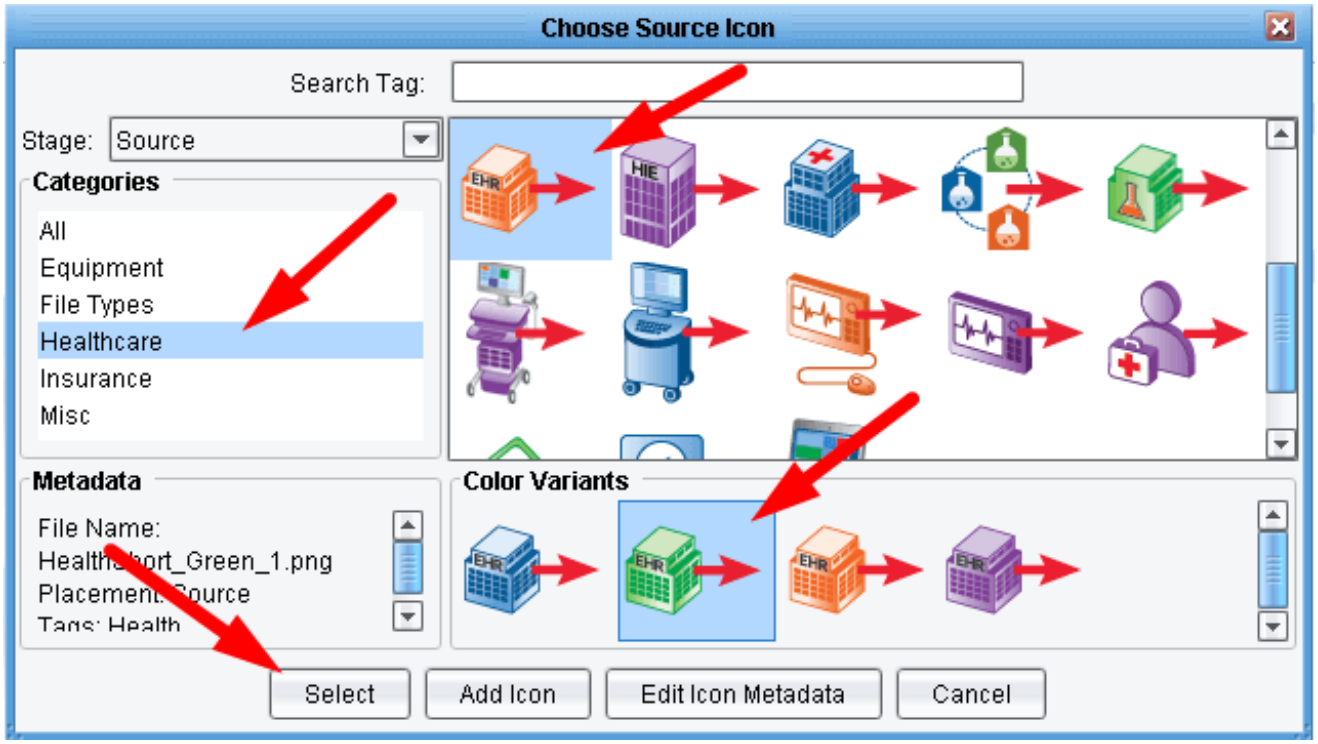
The path to your **Target directory** now appears in the field **Target file name**, let's just call it "**Spreadsheet"**. Enter that in the **Target file name** configuration item field.

We also need to enter a **Target file extension**. Let's enter **"xls,"** since this is a spreadsheet.

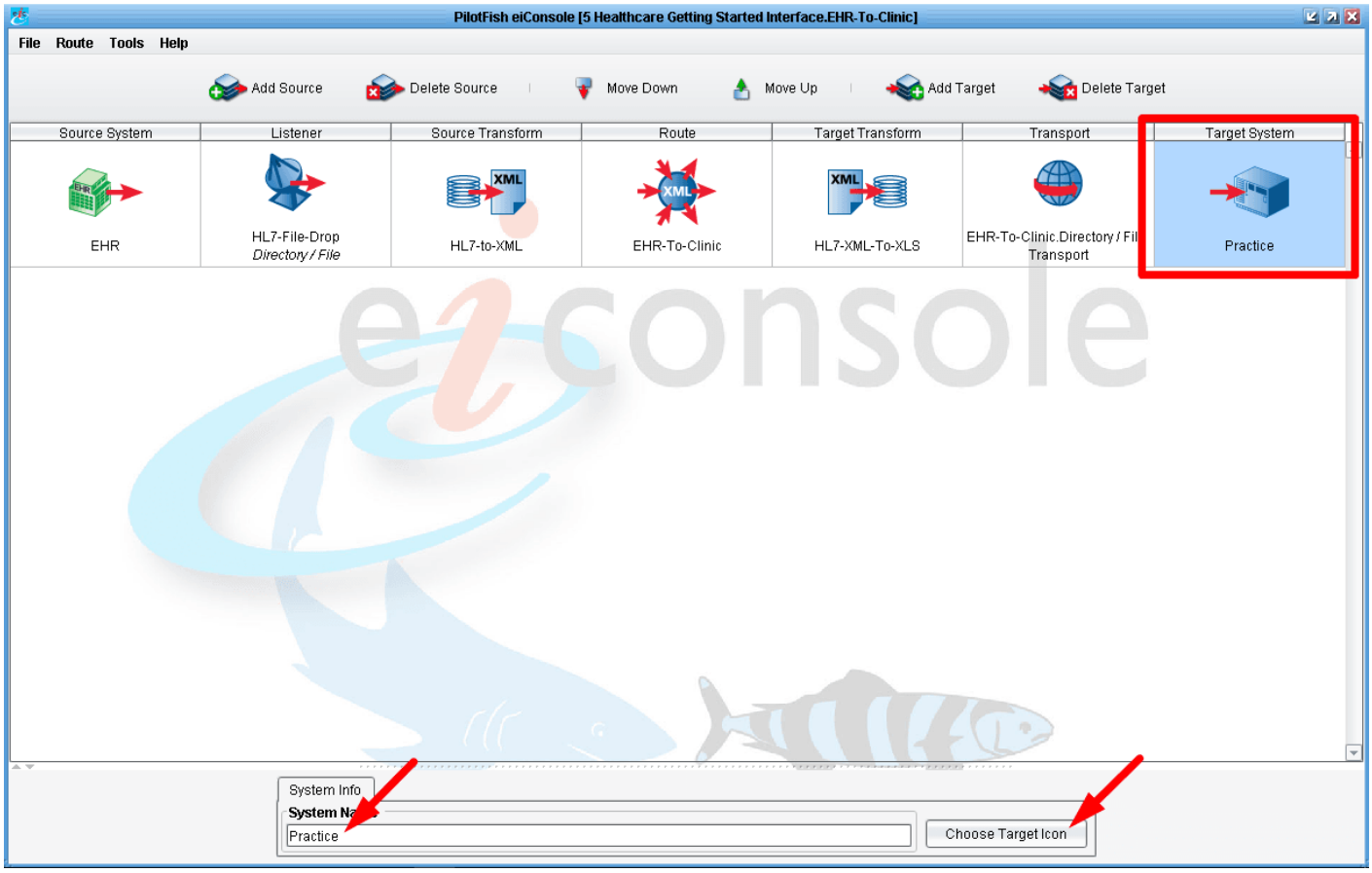
Adding Source and Target Names



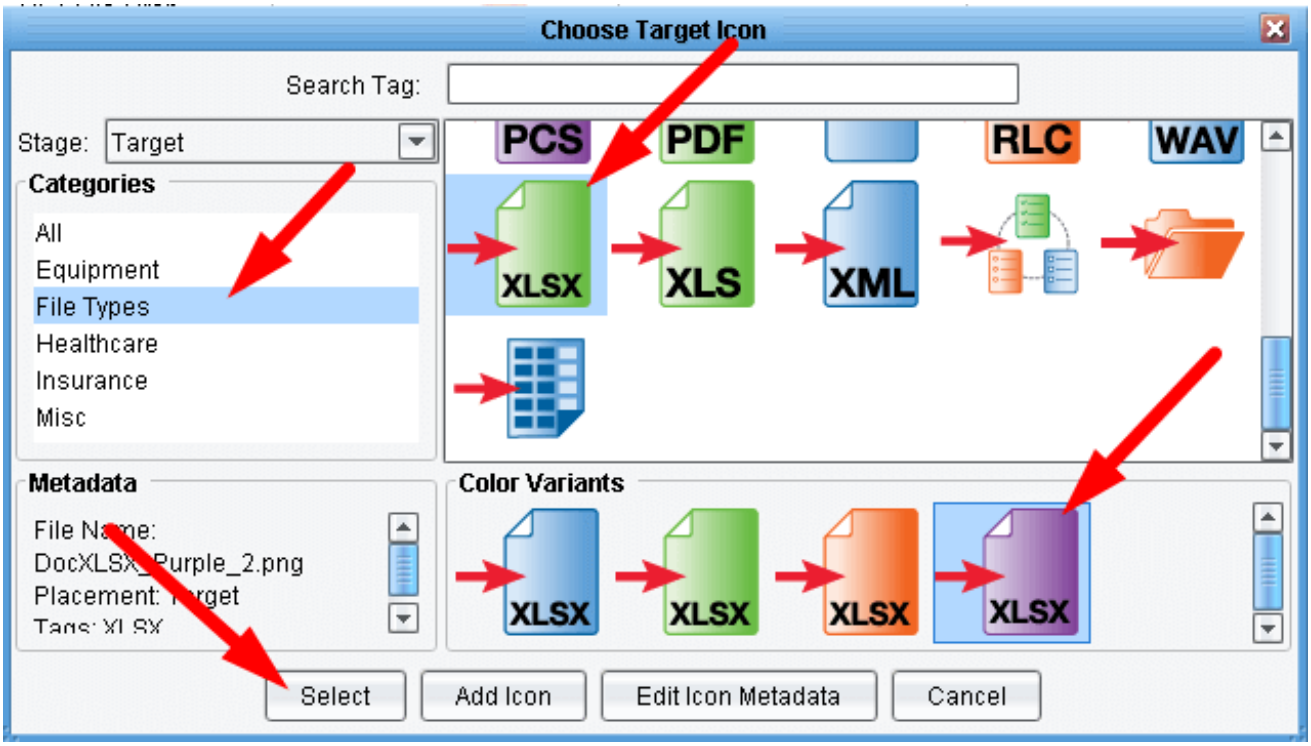
Now, we can go back and add the names of our Source and Target system. Click on the **Source** System icon and in the System Name configuration area call this **EHR**. Click on the **Choose Source Icon** button.



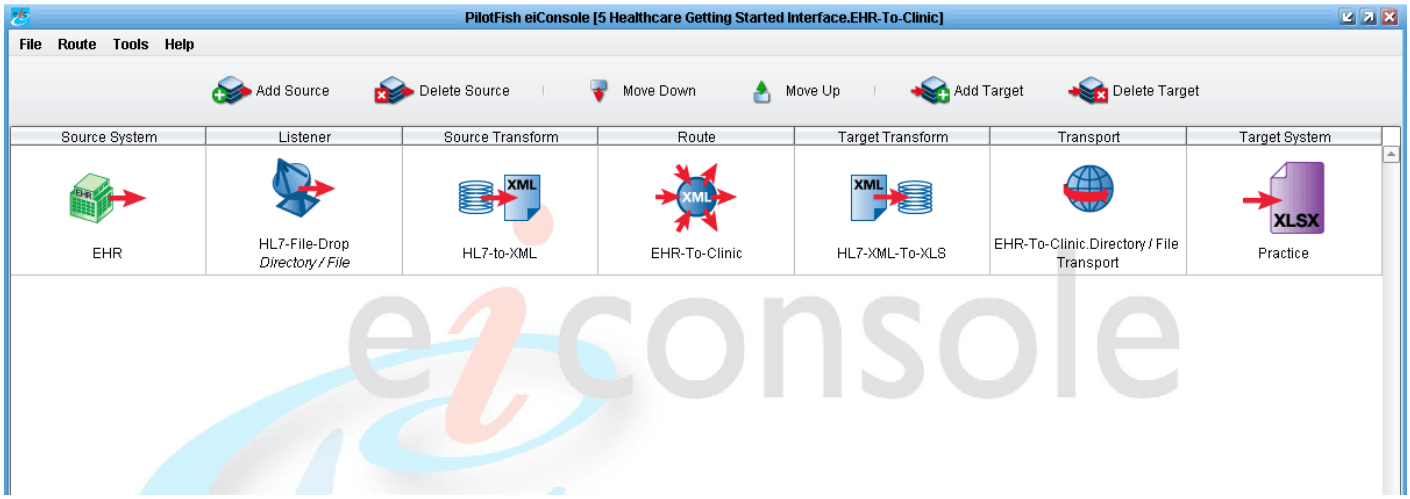
Select the **Healthcare** category, choose any of the **EHR** icons and click **Select**.



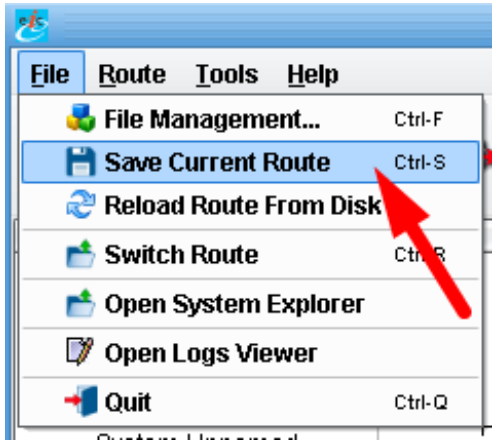
Now click on the **Target** system. We'll name it **Practice**. And again click on the **Choose Target Icon** button.



Select the **File Types** category, choose any of the **XLSX** icons and click on **Select**.

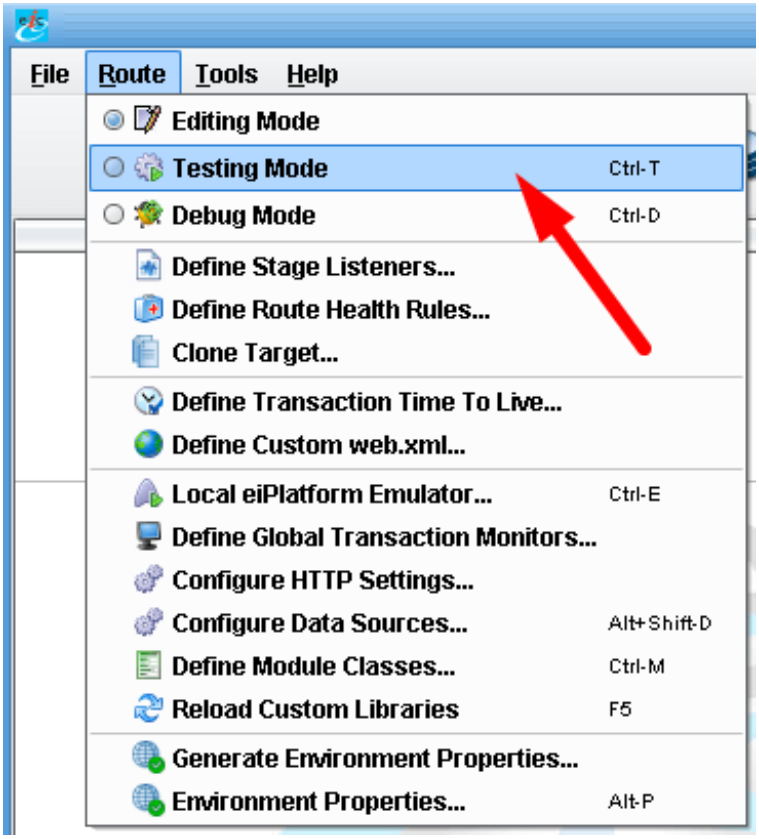


Now the main grid of the eiConsole should look like the one shown above.

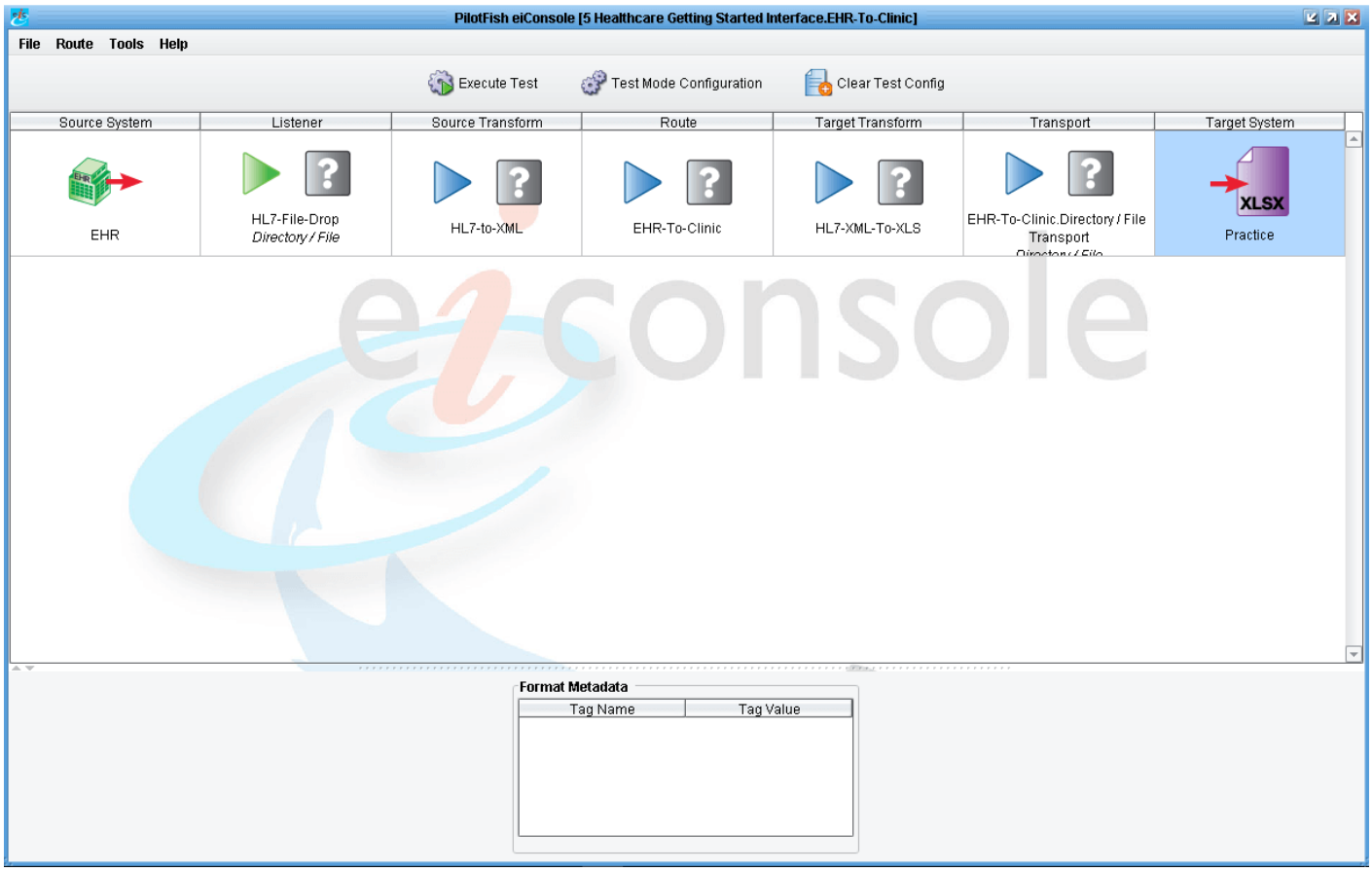


We now have a completely configured interface. In this interface an HL7 system produces HL7 data, the HL7 file gets picked up by a Listener, that HL7 file is converted to XML. It is routed to 1 defined Target system, which converts the HL7 XML to an Excel spreadsheet format, and then drops the results in a directory.

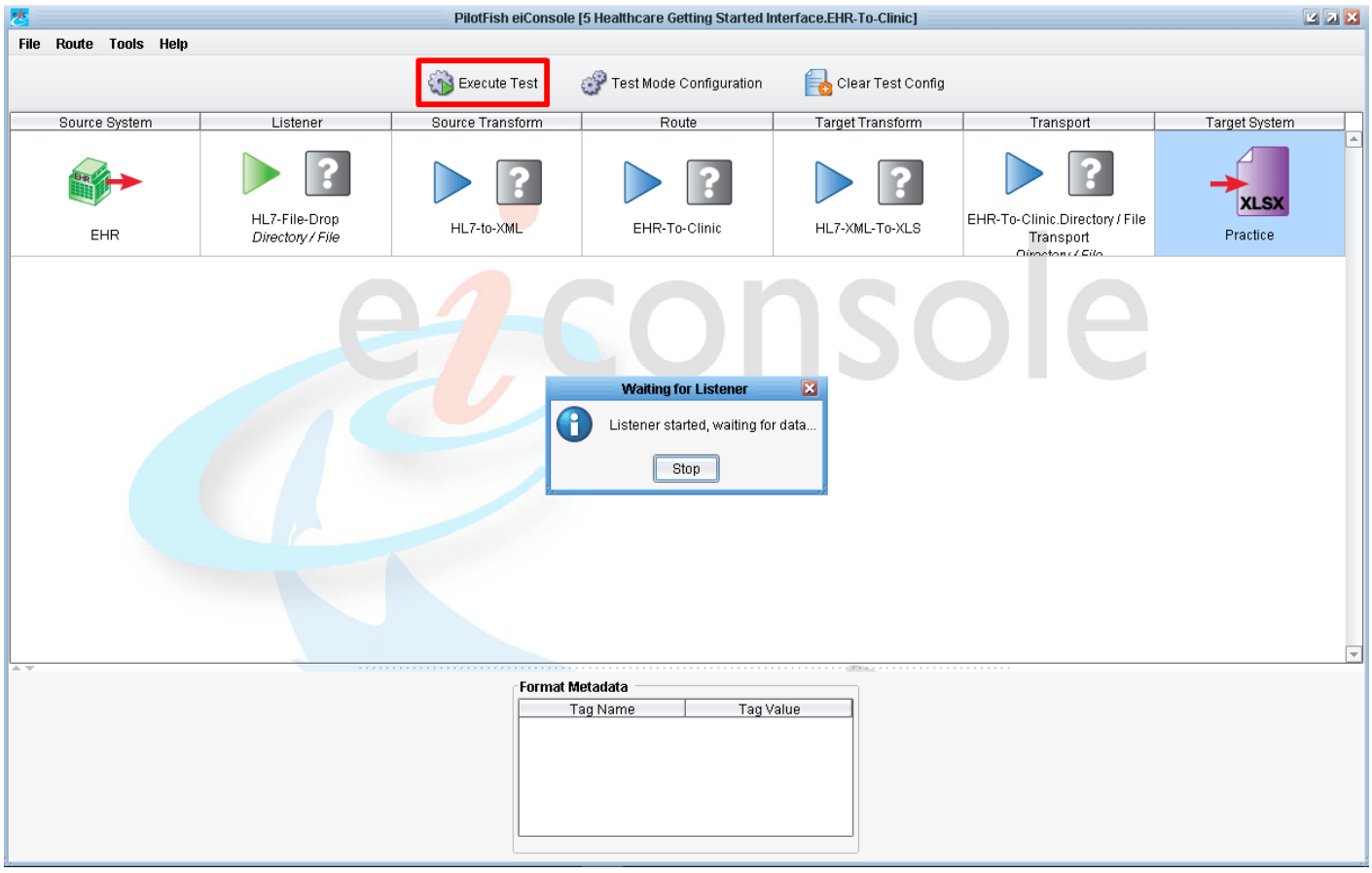
Now that we've configured it, let's see if it works. From the **File** menu, select **Save Current Route**.



Then, in the Route menu, select **Testing Mode**.



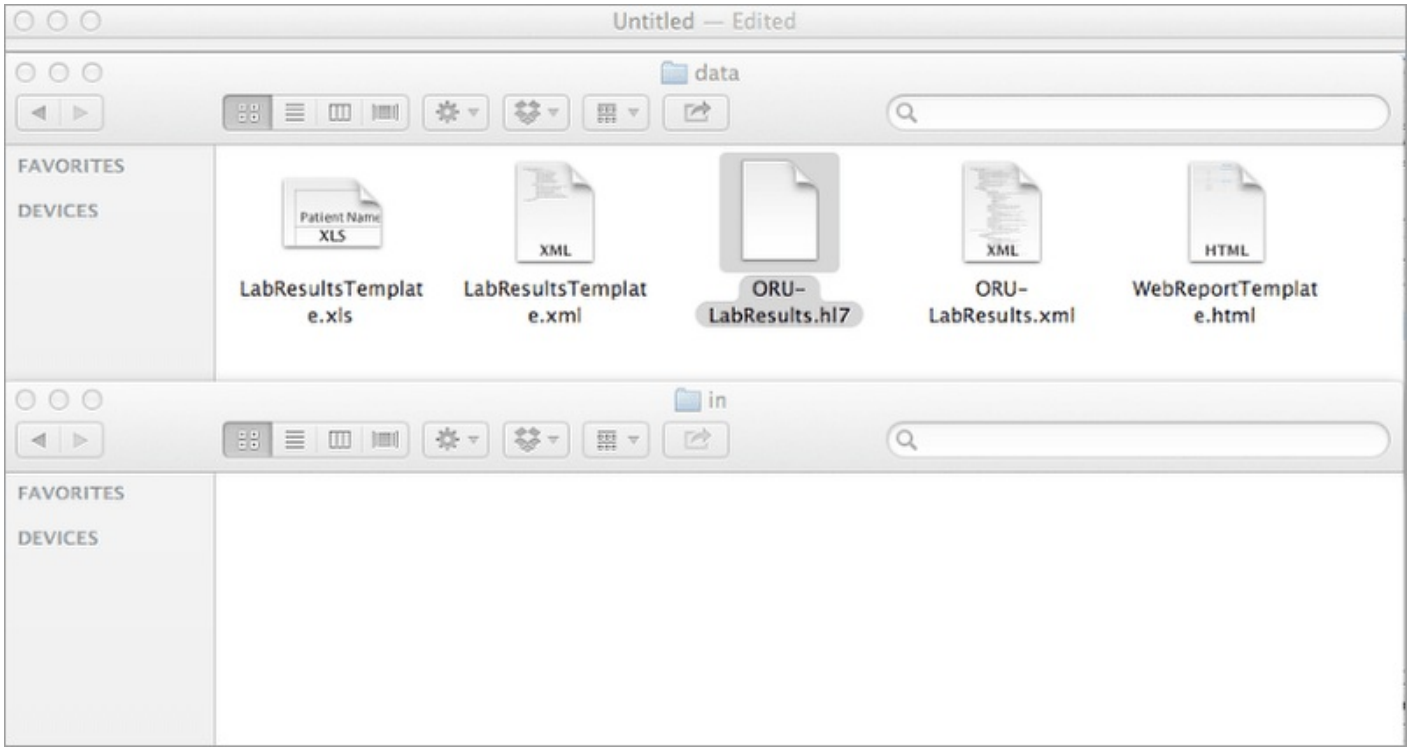
The eiConsole's inline **Testing Mode** opens. Here, we see our same route topology, Source System, Listener, Source Transform, etc., but this time our icons are replaced with question marks and arrows indicating stages of a test we can run and the path that it's configured to execute.



We'll start our test at the Listener stage, indicated by the green arrow.

Click the **Execute Test** button to start your test.

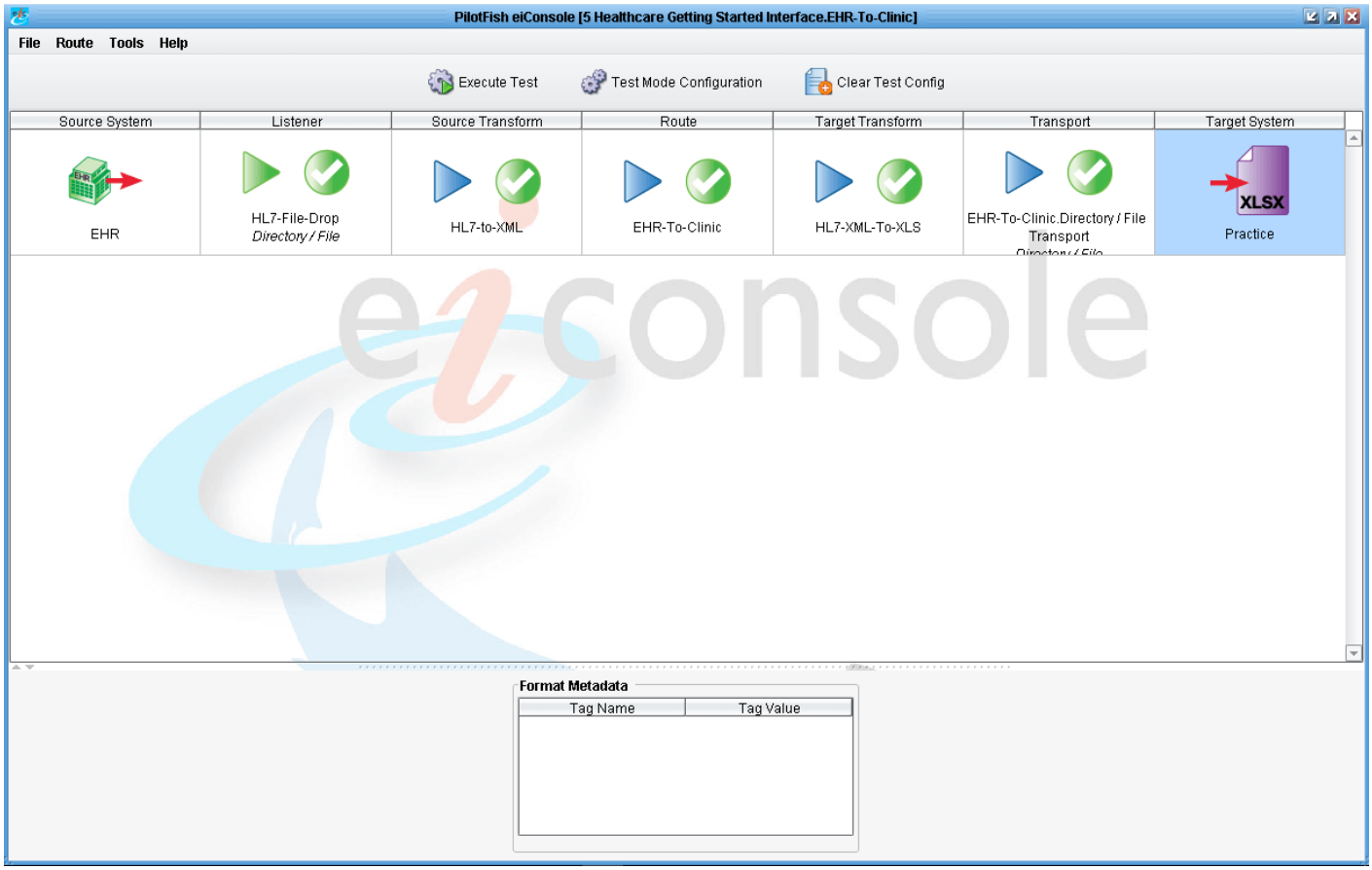
The Listener will start polling every 10 seconds waiting for an HL7 file to appear in the input folder that you designated.



In your distribution folder, one more time find the **ORU_LabResults.hl7** sample file (in the **data** folder) and copy it.

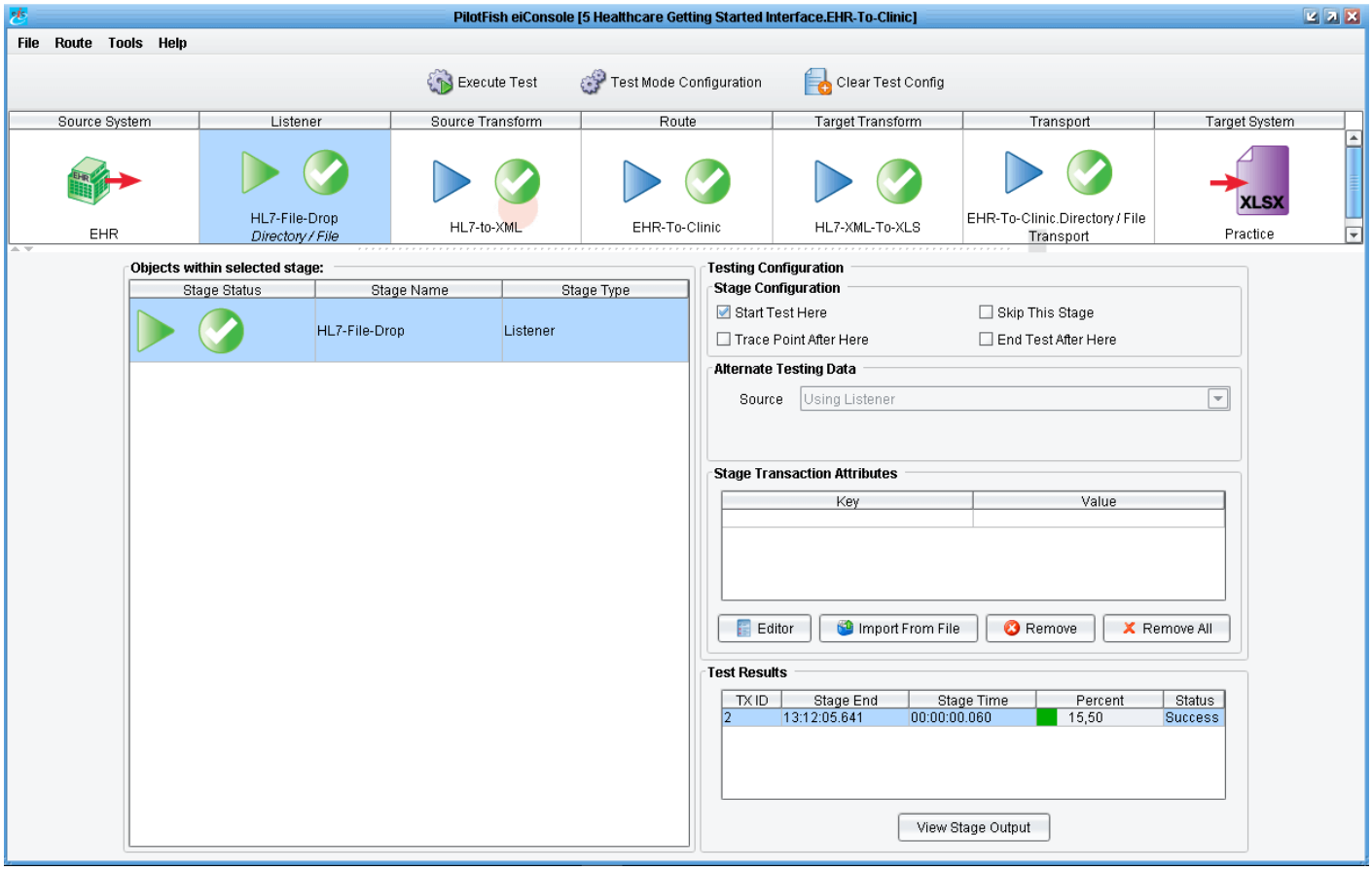
Then, paste it into the **"in"** folder of your Working Directory.

Within 10 seconds the file will disappear.

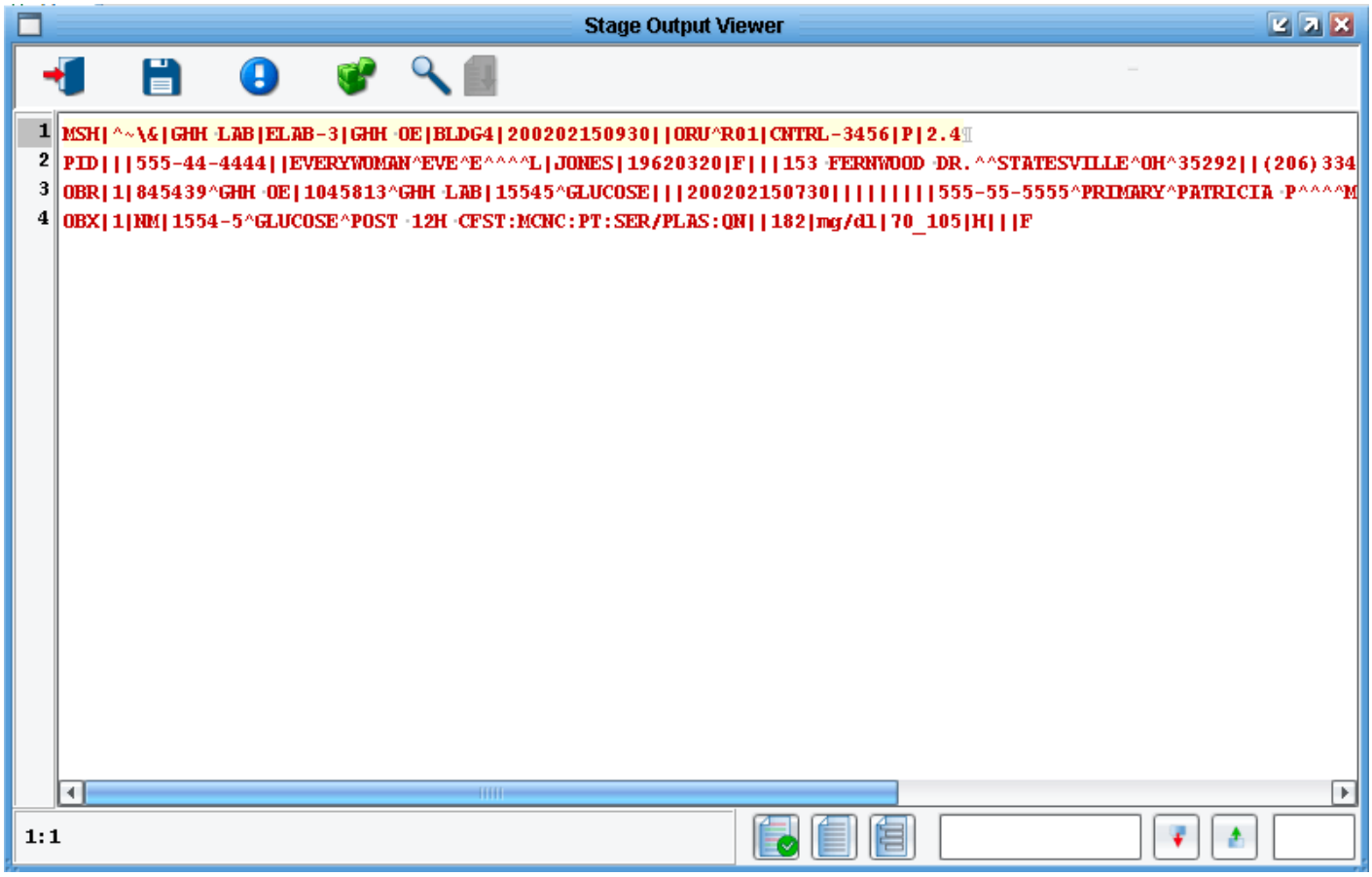


As each stage completes the question marks turn into **green check marks**. What a beautiful sight! Or, if you had a failure you will see a **red X**.

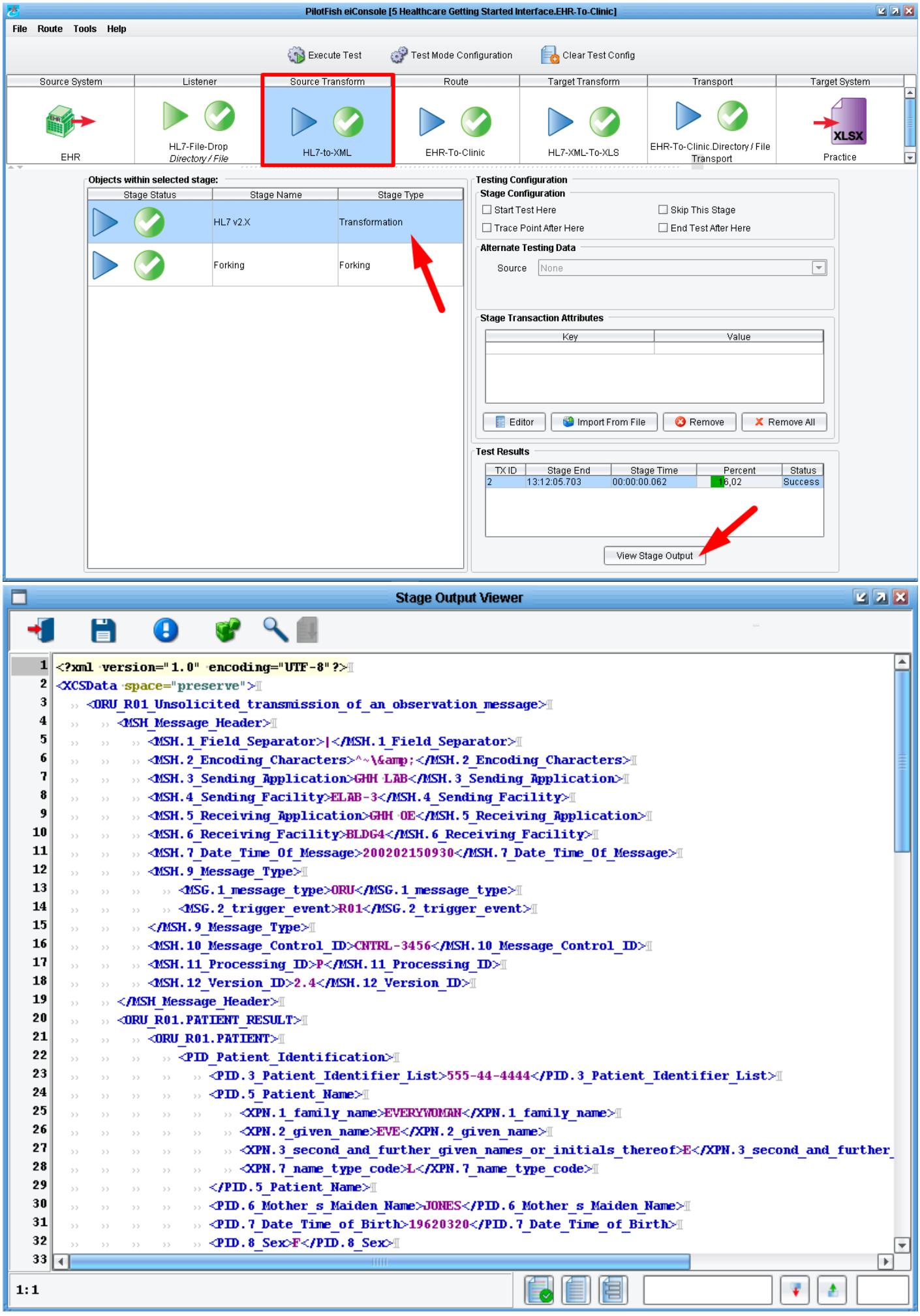
Here, we can take a look at how the data appeared at each point.



You can click on the **Listener** stage and double click the **HL7-File-Drop** stage name (in the Objects within selected stage grid.)

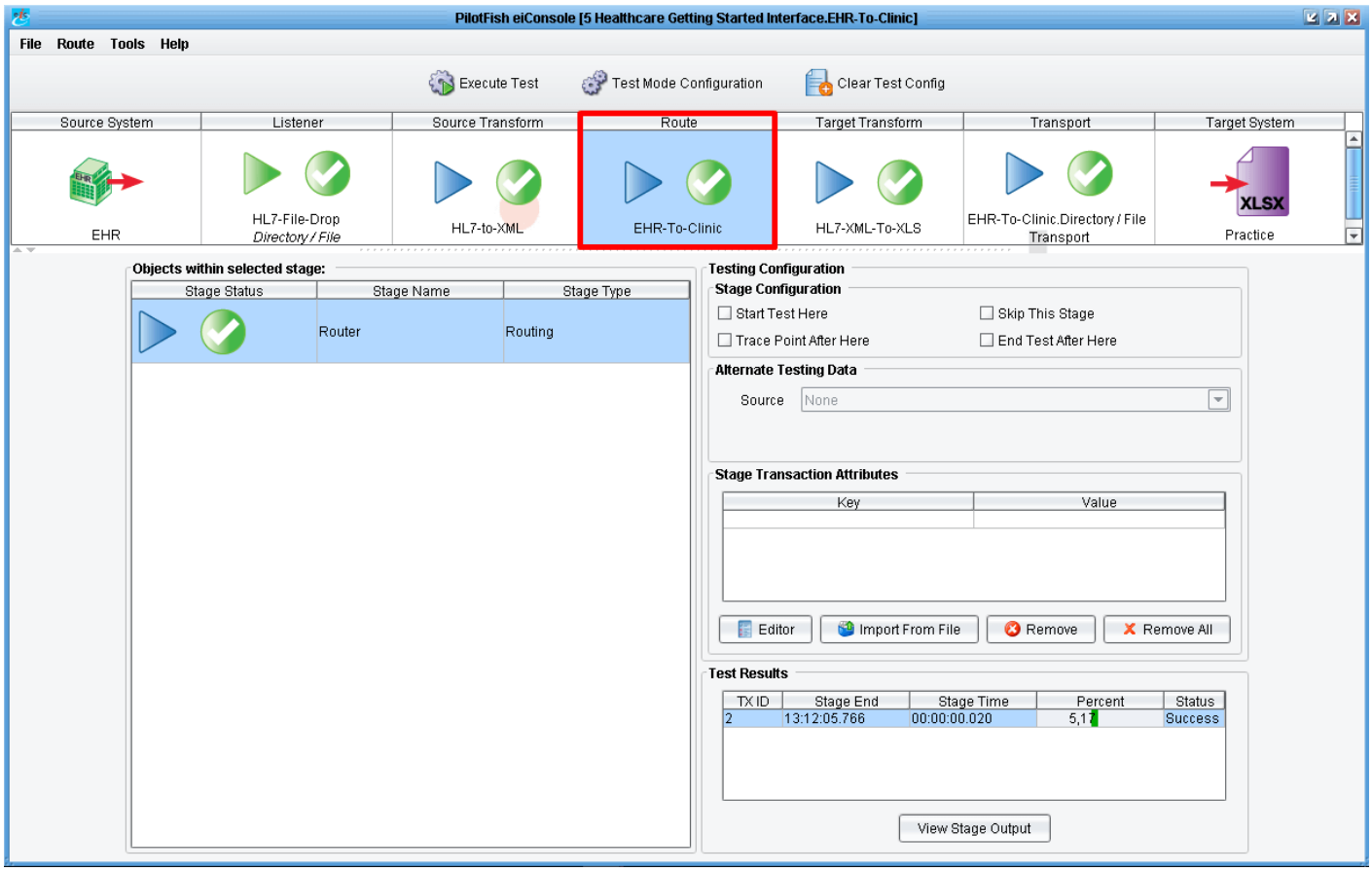


Here you can see the unaltered HL7 file as it was acquired.

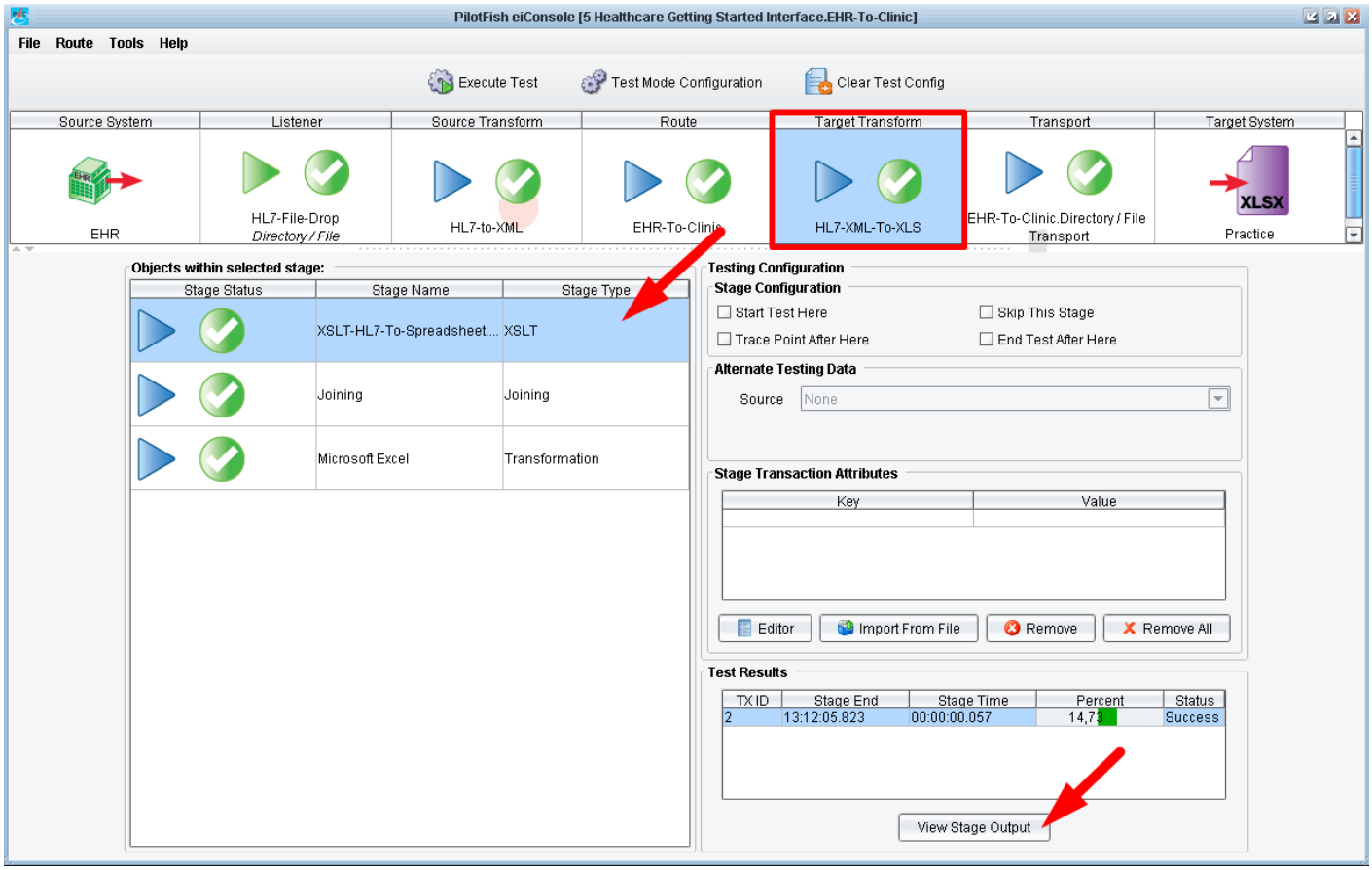


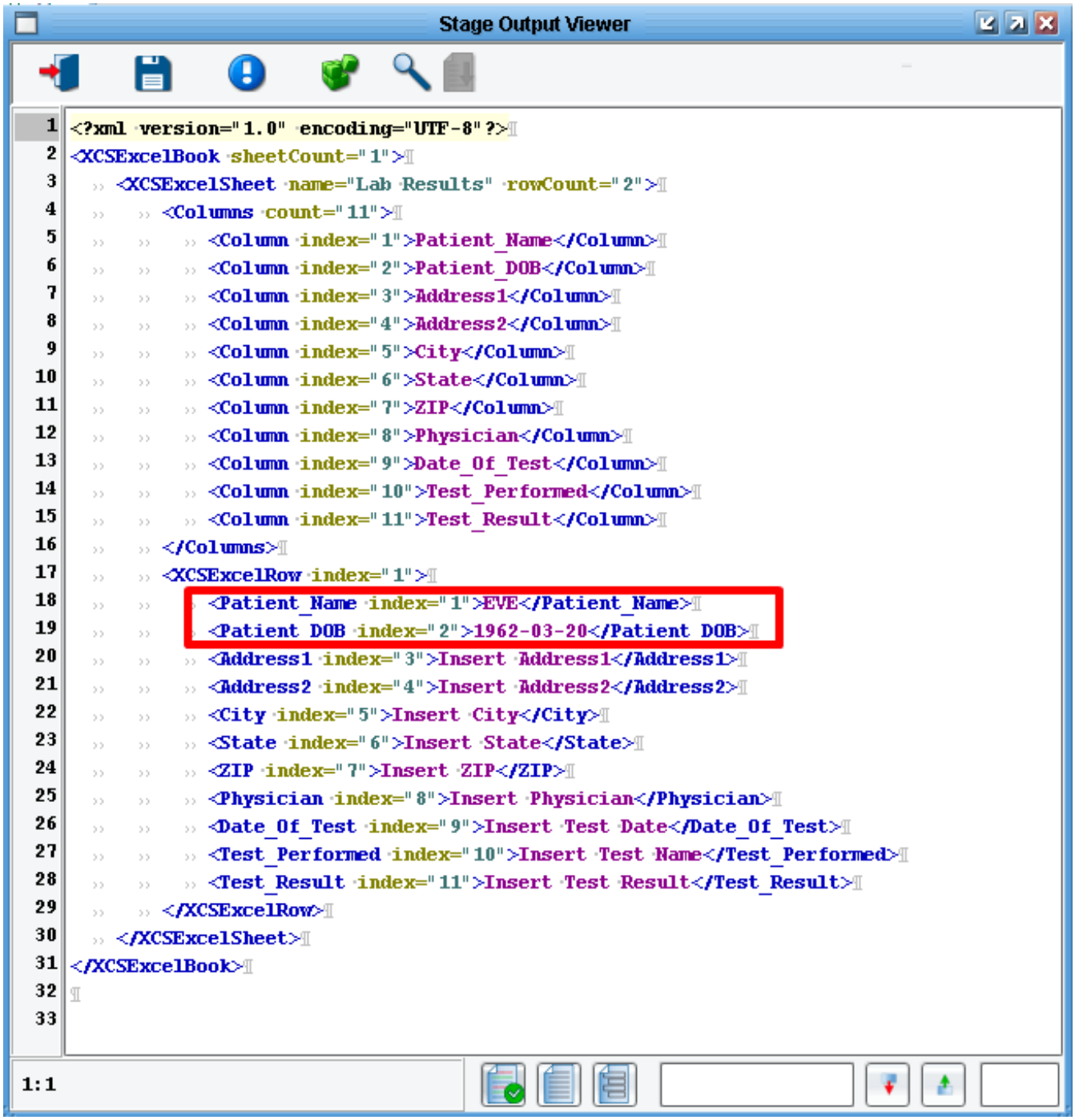
Next, you can click on the **Source Transform** stage and double click on the **HL7 v2.x Transformer** row or on

the **View Stage Output**. Here you can see the parsed HL7 message.



Nothing new to see in the **Route** stage.



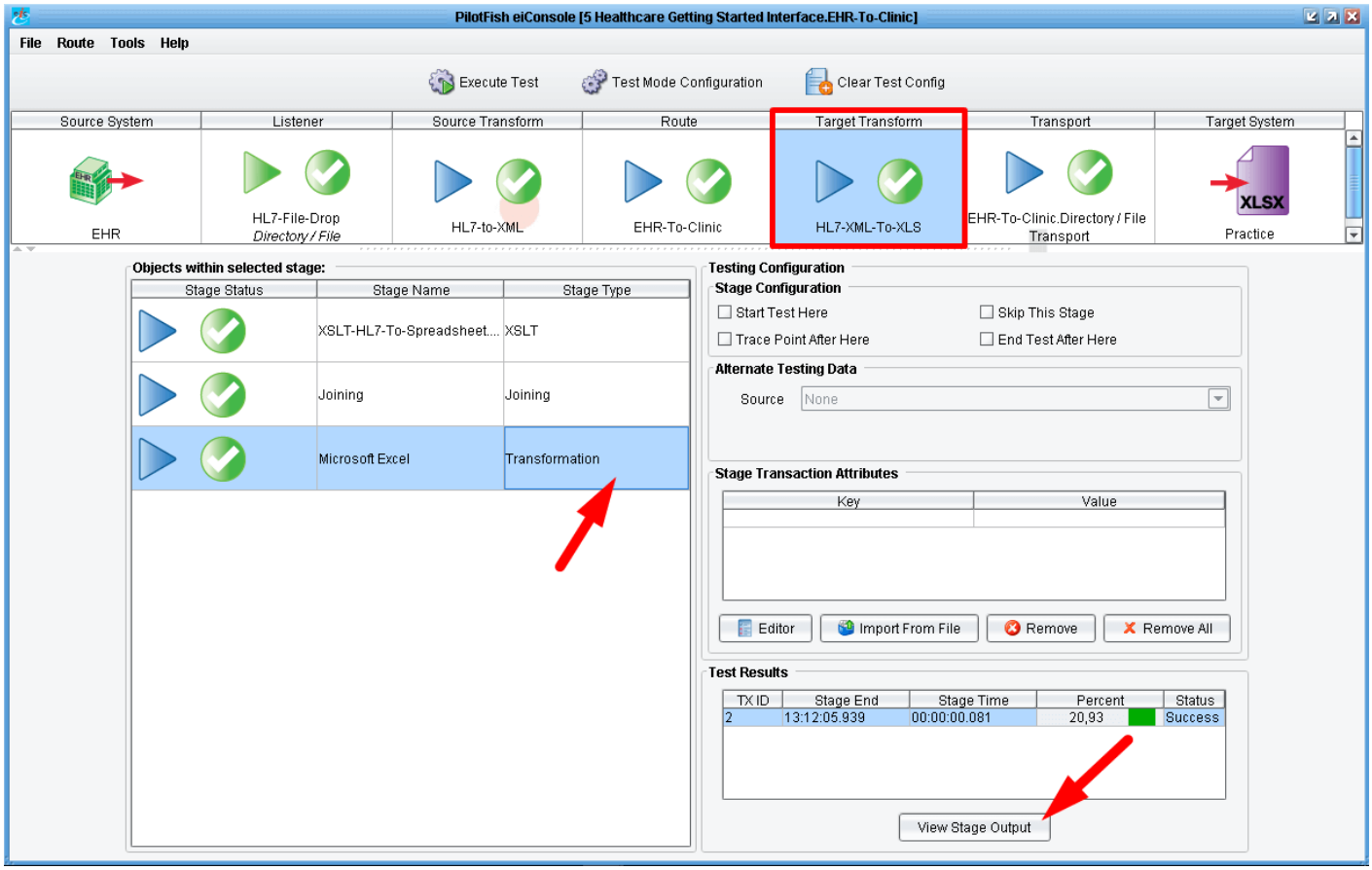


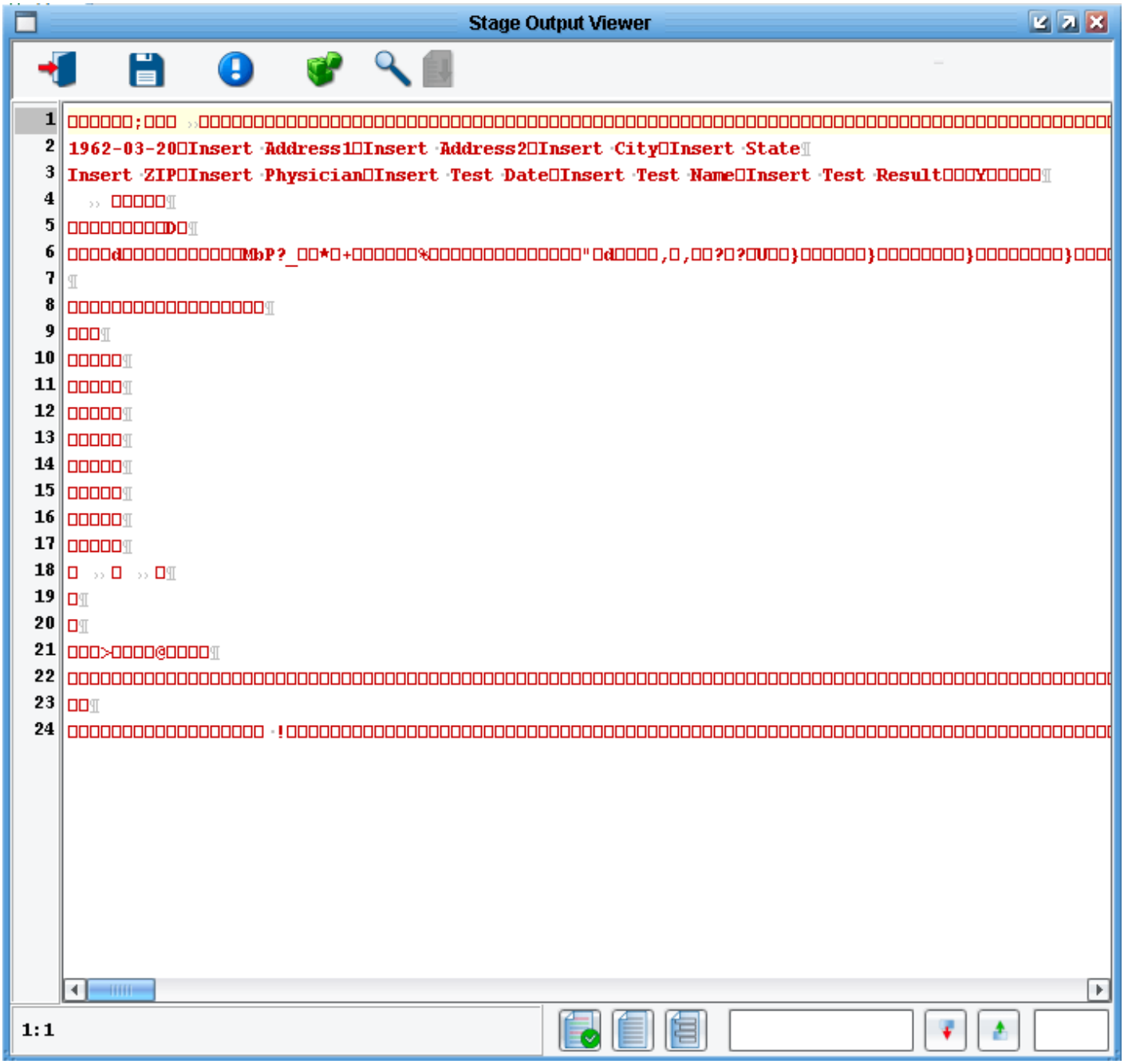

```
<?xml version="1.0" encoding="UTF-8"?>
<XCSExcelBook sheetCount="1">
  <XCSExcelSheet name="Lab Results" rowCount="2">
    <Columns count="11">
      <Column index="1">Patient_Name</Column>
      <Column index="2">Patient_DOB</Column>
      <Column index="3">Address1</Column>
      <Column index="4">Address2</Column>
      <Column index="5">City</Column>
      <Column index="6">State</Column>
      <Column index="7">ZIP</Column>
      <Column index="8">Physician</Column>
      <Column index="9">Date_Of_Test</Column>
      <Column index="10">Test_Performed</Column>
      <Column index="11">Test_Result</Column>
    </Columns>
    <XCSExcelRow index="1">
      <Patient_Name index="1">EVE</Patient_Name>
      <Patient_DOB index="2">1962-03-20</Patient_DOB>
      <Address1 index="3">Insert Address1</Address1>
      <Address2 index="4">Insert Address2</Address2>
      <City index="5">Insert City</City>
      <State index="6">Insert State</State>
      <ZIP index="7">Insert ZIP</ZIP>
      <Physician index="8">Insert Physician</Physician>
      <Date_Of_Test index="9">Insert Test Date</Date_Of_Test>
      <Test_Performed index="10">Insert Test Name</Test_Performed>
      <Test_Result index="11">Insert Test Result</Test_Result>
    </XCSExcelRow>
  </XCSExcelSheet>
</XCSExcelBook>
```

We then move on to the **Target Transform** stage. Here we can double click the **XSLT** row or the **View Stage Output** to see the output of our transformation.

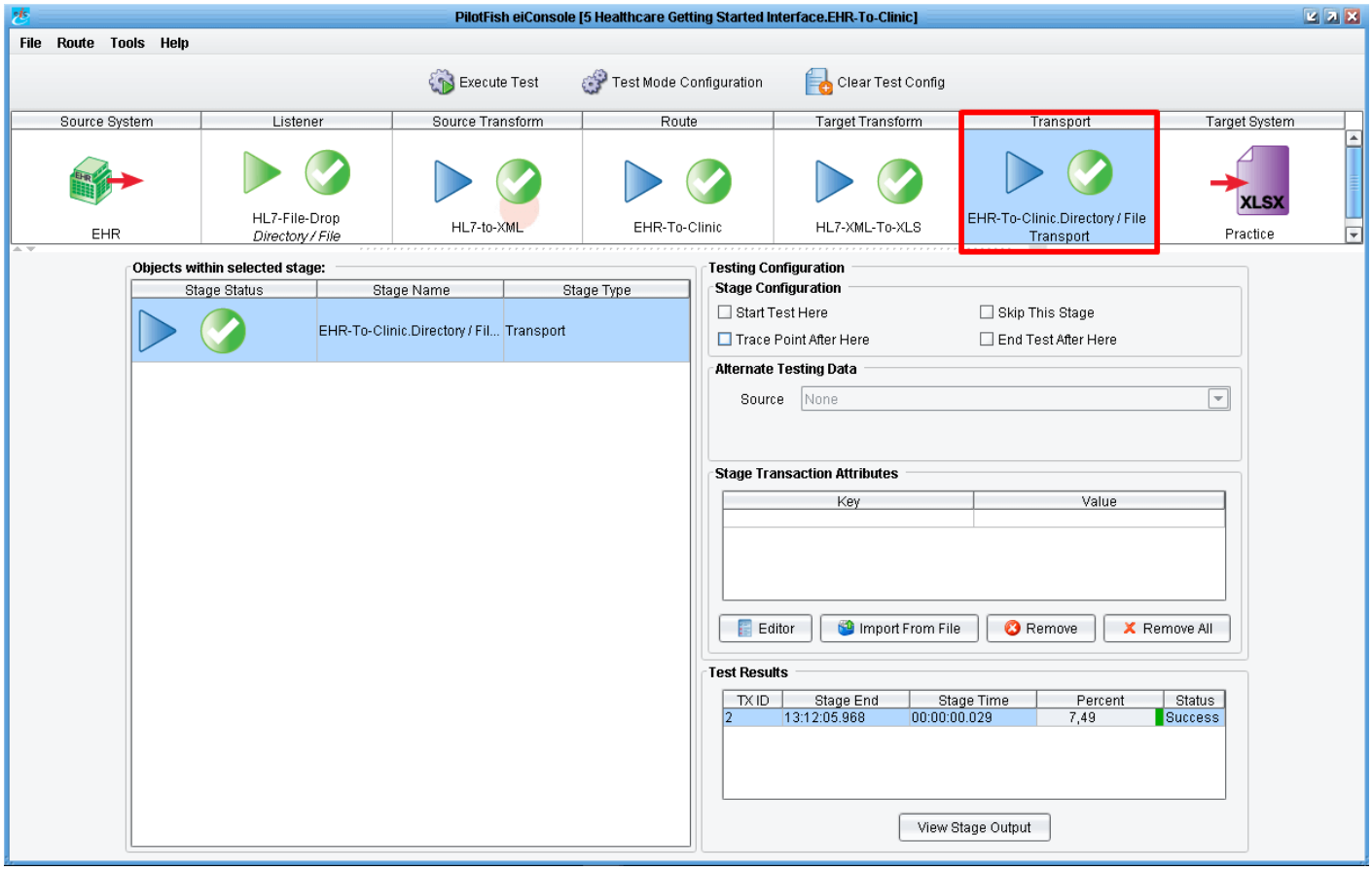
You can see the two fields that we've mapped, the **Patient_Name** and the **Patient_DOB** appear in the output.

All of the other fields are defaulted to the same information that we had in the sample Excel spreadsheet.

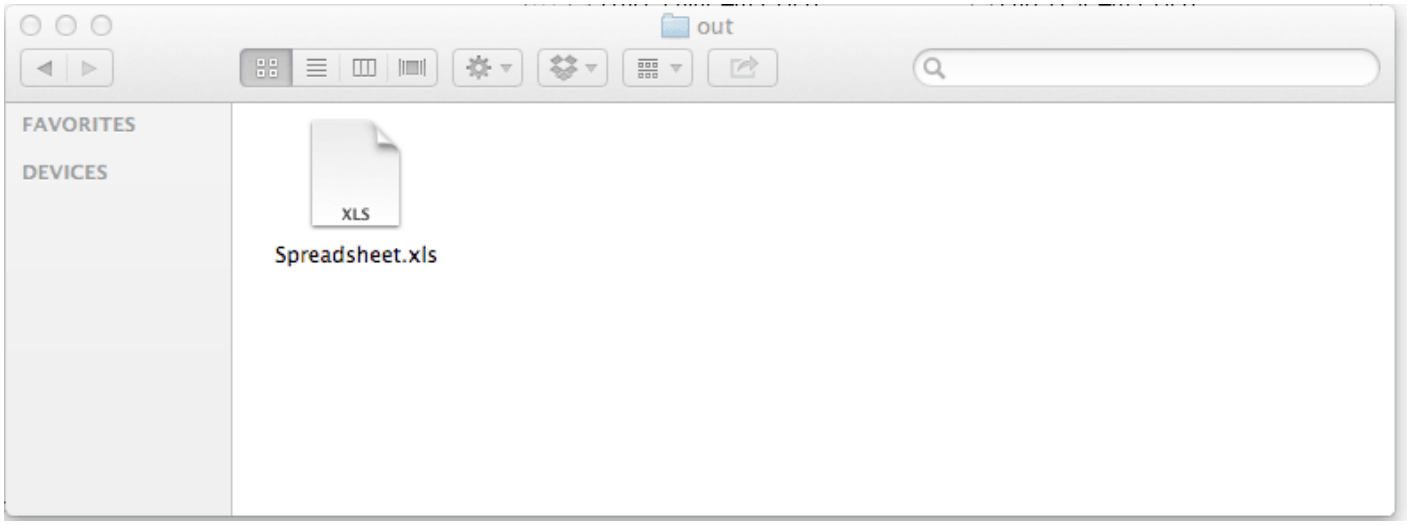

PilotFish eiConsole [5 Healthcare Getting Started Interface.EHR-To-Clinic]
File
Route
Tools
Help
Execute Test
Test Mode Configuration
Clear Test Config
Source System
Listener
Source Transform
Route
Target Transform
Transport
Target System
EHR
HL7-File-Drop
Directory / File
HL7-to-XML
EHR-To-Clinic
HL7-XML-To-XLS
EHR-To-Clinic.Directory / File
Transport
XLSX
Practice
Objects within selected stage:
Stage Status
Stage Name
Stage Type
XSLT-HL7-To-Spreadsheet....
XSLT
Joining
Joining
Microsoft Excel
Transformation
Testing Configuration
Stage Configuration
Start Test Here
Skip This Stage
Trace Point After Here
End Test After Here
Alternate Testing Data
Source
None
Stage Transaction Attributes
Key
Value
Editor
Import From File
Remove
Remove All
Test Results
TX ID
Stage End
Stage Time
Percent
Status
2
13:12:05.939
00:00:00.081
20,93
Success
View Stage Output



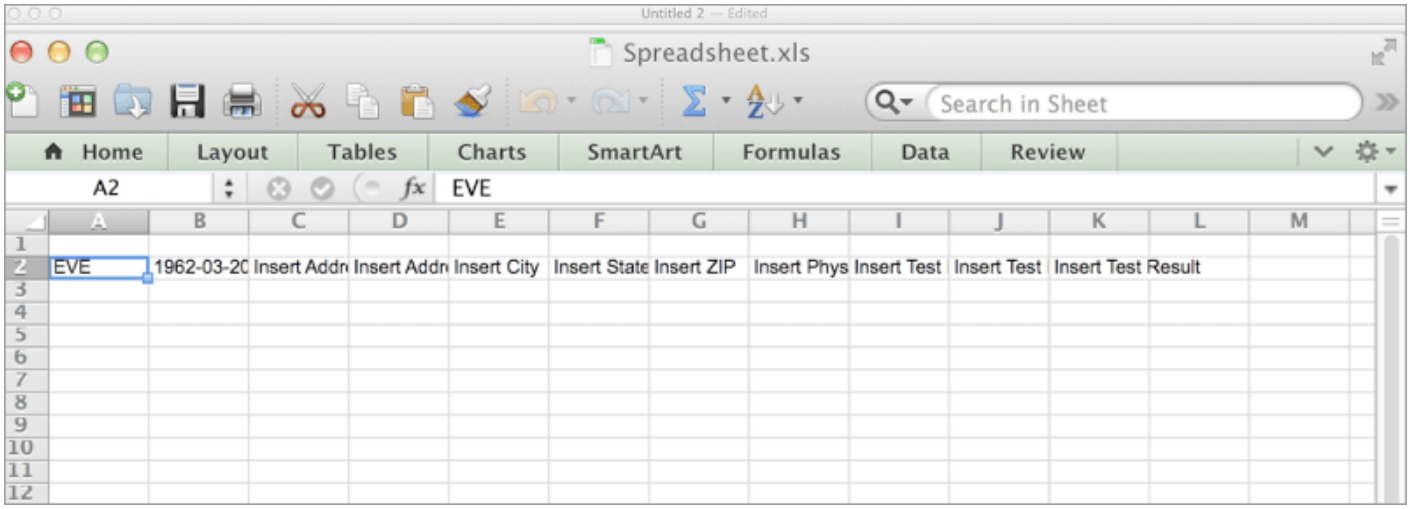
Next, the **Microsoft Excel Transformation** row. You can see it did its job. Double click the row and view the binary form of the Excel spreadsheet itself. It's an ugly sight!



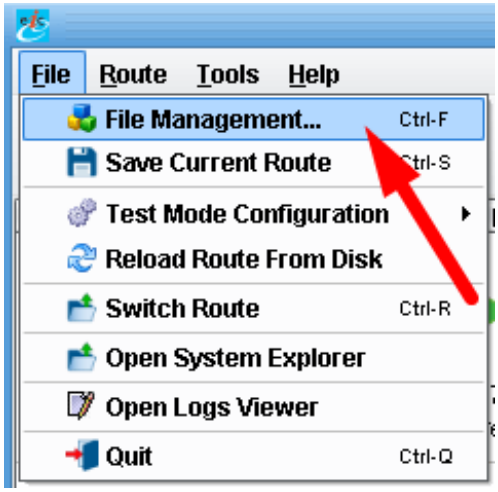
Finally, we can click on the **Transport** stage where we can see that the file was successfully dropped in a directory.



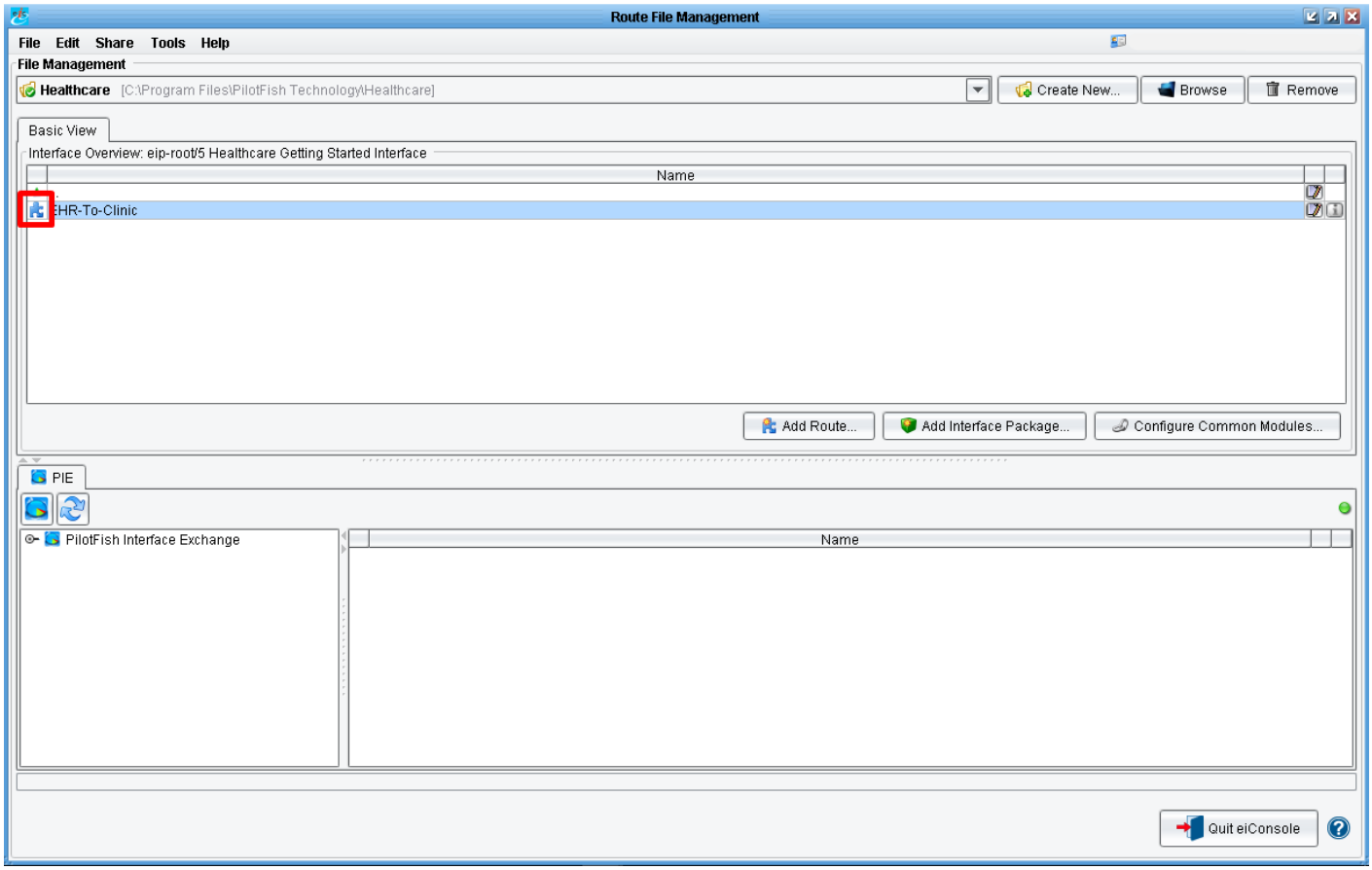
If we navigate to the **out** folder of our Working Directory, we'll see a newly created file called **spreadsheet.xls**.



If we open the spreadsheet we can see cells populated with our mapped data.



Now, you've completed your first interface using the eiConsole for Healthcare. Now that you have completed testing your interface, select **File Management** from the File menu.



The icon next to **EHR-To-Clinic** is now blue indicating it is a fully configured interface. Typically, the final step is to deploy the interface to production. This can be done by copying the configuration files that now exist in the Working Directory, or by connecting to an eiPlatform server, and dragging & dropping into an eiPlatform panel, accessed via a tab next to the PIE tab.

Note: an eiPlatform server needs to be configured in order for this option to be visible in this window.

That's all there is to it. You've successfully built your first HL7 interface using the PilotFish eiConsole for Healthcare. Browse through the demos available on the PilotFish Product Online Resource Center's Healthcare page to learn more about how you can leverage the HL7 specific features and components of the eiConsole for Healthcare. Also check out Levels I-IV for advanced topics to handle virtually any integration scenario with the eiConsole.