General Quick Start Tutorial

https://cms.pilotfishtechnology.com/node/4084

eiConsole v.23R1

The General Quick Start Tutorial is the prerequisite for all new users. It is a simple interface where you'll be taking a flat file from a directory and converting it to XML. In 15-20 minutes you will have configured an end-to-end interface with the eiConsole.

Before you begin the Quick Start Tutorial we suggest that you browse through the eiConsole **OVERVIEW** for a quick look at the overall process for configuring an interface. 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 Bundles which include all the sample files that you'll need to do the industry-specific tutorials. For additional information, click here for tutorials on Importing a **Bundle**. After that, you'll be ready to start.

eiConsole Basic Terminology

Before we begin configuring an interface let's review some eiConsole terminology:

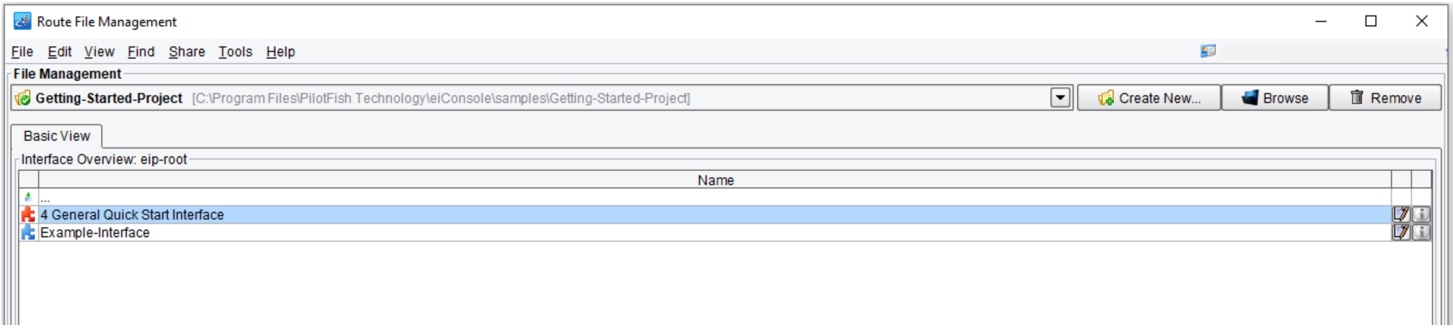
- A **Working Directory (aka Workspace):** is a project directory that contains at a minimum the 4 directories required by the eiConsole: data, formats, lib and routes.
- An **Interface:** is one or more routes.
- A **Route:** is an interface or a part of the interface that defines how the data moves from any number of source systems to any number of target systems.
- An **Interface Package (Interface Context):** is a group of routes and interfaces that can be categorized by functional tasks. (An Interface Package can consist of any number of routes, a collection of interface templates, and partially or fully configured interfaces.)

Next, let's expand on the **Working Directory**. In the eiConsole, a Working Directory is a set of directories and files utilized by eiConsole instances and by a running eiPlatform. The directory structure dictates a hierarchy to be used for determining which Routes and Formats belong to particular interfaces. If you select an empty directory, the eiConsole will ask you if you wish to initialize it. In doing so, it will create the necessary directories – data, formats, lib and routes:

- The "**data"** folder is used to store user-defined and managed data, such as sample files for testing or documents describing interfaces.
- The "**Lib"** directory may contain JAR files (files with a ".jar" extension), which are used for defining new types or newer versions of eiPlatform / eiConsole modules.
- Each **"routes"** folder (under “routes”) contains a "route.xml" file, which is where the definition of a particular Route is stored (including configuration settings and topology).
- Each **"formats"** folder (under “formats”) contains a "format.xml" file, which is where the definition for a particular Format is stored. This folder will also include XSLT transformations and File Specification Editor definitions (in XML).

For additional information on the Working Directory, visit the Module: **(1) The Working Directory**.

Identifying Interfaces



File Management

Getting-Started-Project [C:\Program Files\PilotFish Technology\eiConsole\samples\Getting-Started-Project]

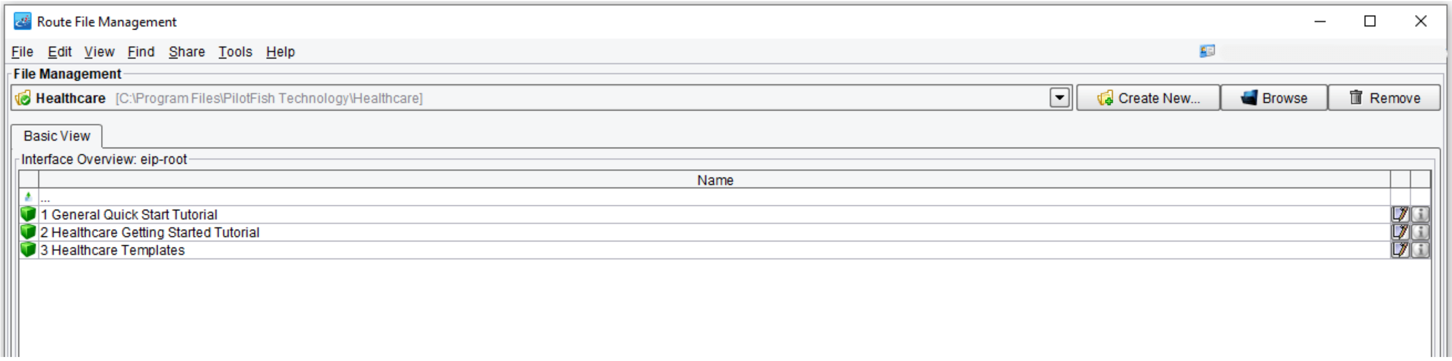
Interface Overview: eip-root

Name
...
4 General Quick Start Interface
Example-Interface

In the **Route File Management** window, the red icon, next to **4 General Quick Start Interface**, indicates an unconfigured interface. The blue icon, next to the **Example Interface**, indicates a fully configured, end-to-end interface.

The Route File Management Window

PC Users View



File Management

Healthcare [C:\Program Files\PilotFish Technology\Healthcare]

Interface Overview: eip-root

Name
...
1 General Quick Start Tutorial
2 Healthcare Getting Started Tutorial
3 Healthcare Templates

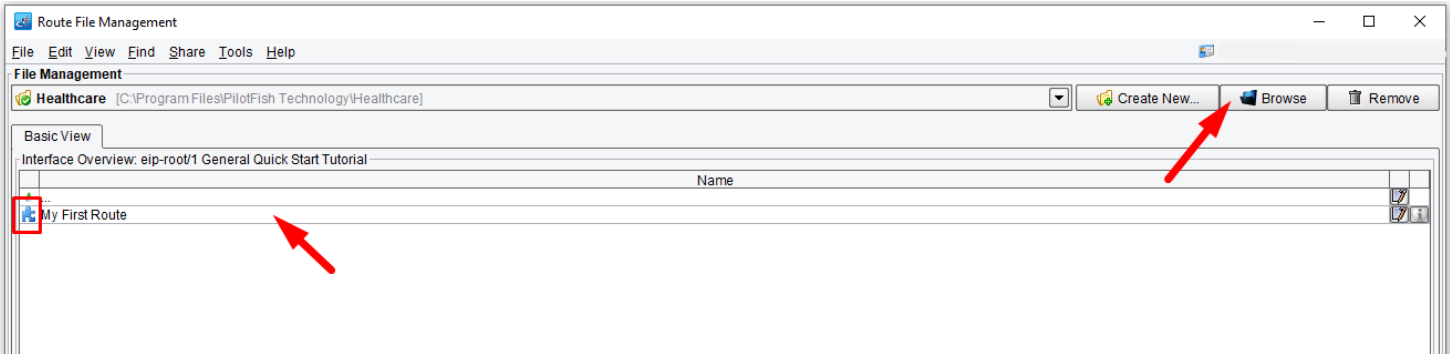
When you launch an eiConsole bundle, the **Route File Management** window opens. In the Basic View of the Interface Overview grid, you'll see 1-3 **"Packages",** depending on which bundle you have downloaded. In the eiConsole, packages are identified by a **green cube icon.** Here we see the contents of the eiConsole for Healthcare bundle (the bundles for ACORD are similar.)

The 3 Packages that an eiConsole Industry-Specific Bundle may include:

- **1 General Quick Start Tutorial** includes a simple, fully configured interface and all of the sample files you will need to complete your first interface.
- **2 Industry Specific (Healthcare, ACORD PCS, etc.) Getting Started Tutorial** includes a more complex, and fully configured industry-specific interface along with all the sample files that you will need to configure the interface.
- **3 Interface (Healthcare, ACORD, etc.) Templates** include examples of interface templates that provide users with a head start for configuring an interface. Free interface templates are available for download off of the PilotFish Interface Exchange (PIE). Click this **LINK** for how to download additional templates from the PIE and to learn how to leverage these.

To begin to configure an interface, double click the first row **1 General Quick Start Tutorial.**

Configuring an Interface



Healthcare [C:\Program Files\PilotFish Technology\Healthcare]

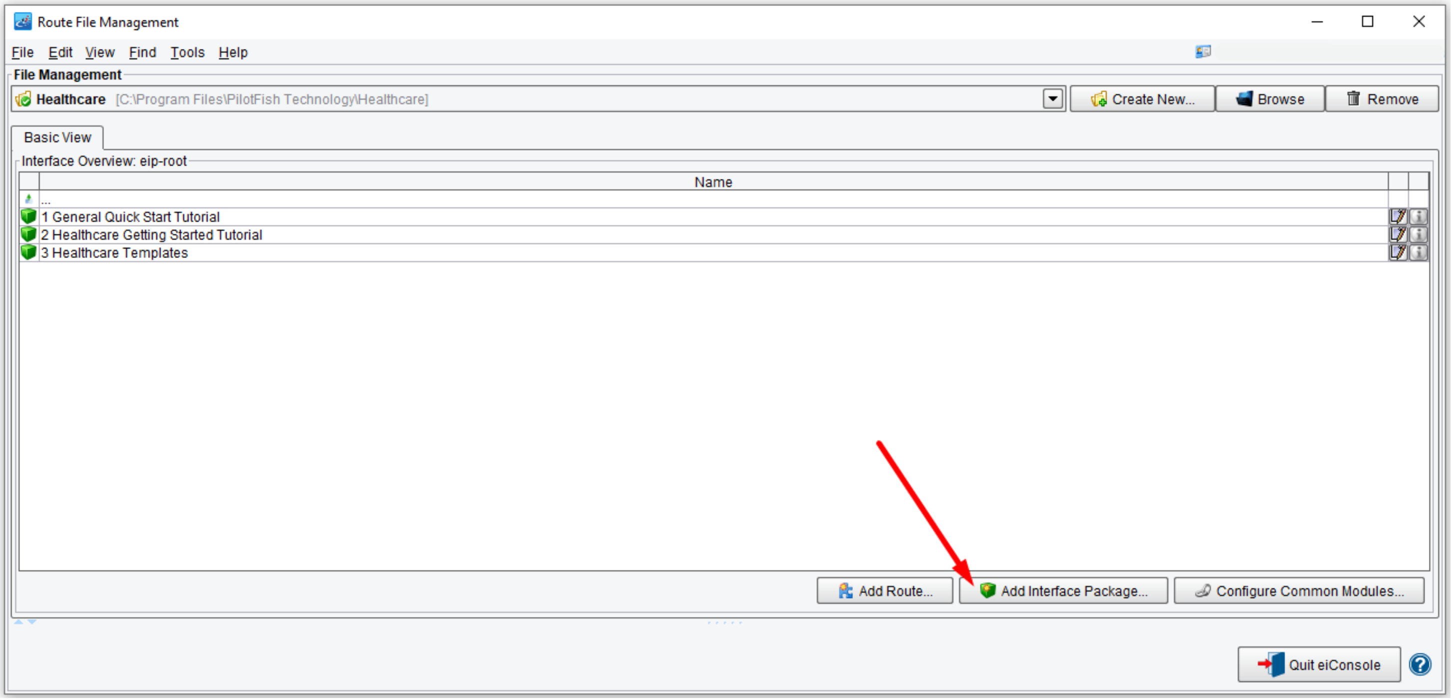
Interface Overview: eip-root/1 General Quick Start Tutorial

..
My First Route

Double-clicking the package reveals **My First Route,** the example Route or fully configured interface (note the blue icon).

Make note of the location of the sample files as you will need them to configure your first Route. If you downloaded an eiConsole bundle, the sample route is located in your distribution, on a Mac: **/Contents/eip-root** folder and on a PC: **eiConsoleeip-root folder**. This sample configured interface consists of a single route.

To return to the main Working Directory (eip-root), double click anywhere in the row above **My First Route**.

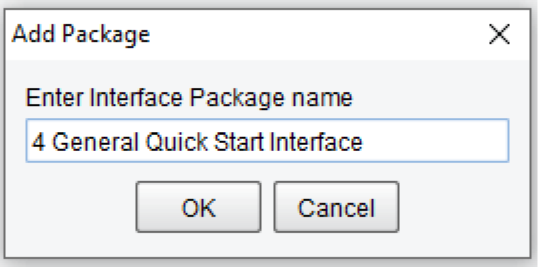


Healthcare [C:\Program Files\PilotFish Technology\Healthcare]

Interface Overview: eip-root

..
1 General Quick Start Tutorial
2 Healthcare Getting Started Tutorial
3 Healthcare Templates

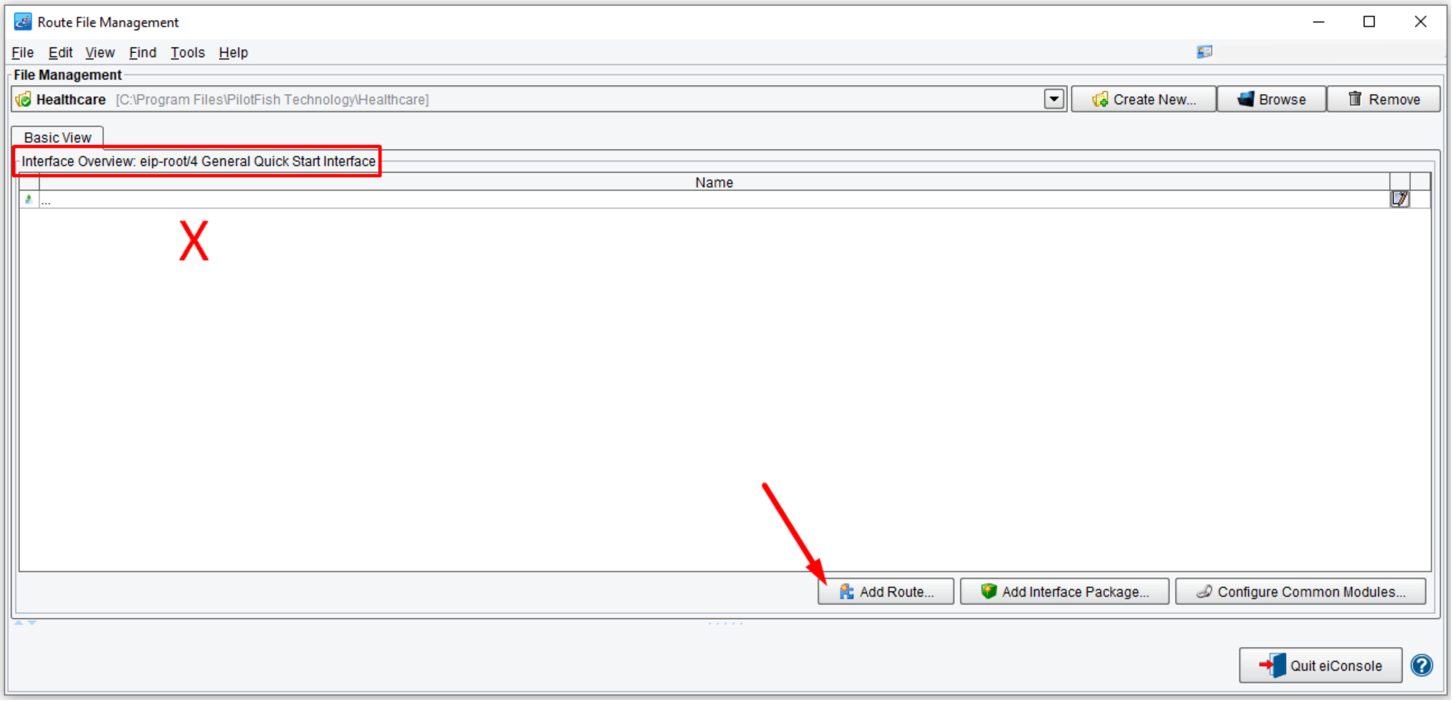
You'll return the original **Route File Management** window, showing Packages 1-3 in the **Interface Overview** grid. The next thing you'll want to do is to create a new Interface Package. Click the **Add Interface Package** button.



Enter Interface Package name

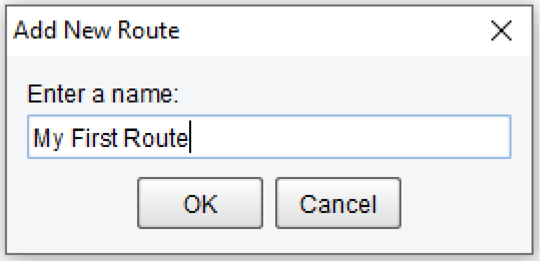
4 General Quick Start Interface

When the **Add Package** dialogue opens, enter the name "**4 General Quick Start Interface",** and click **OK**.

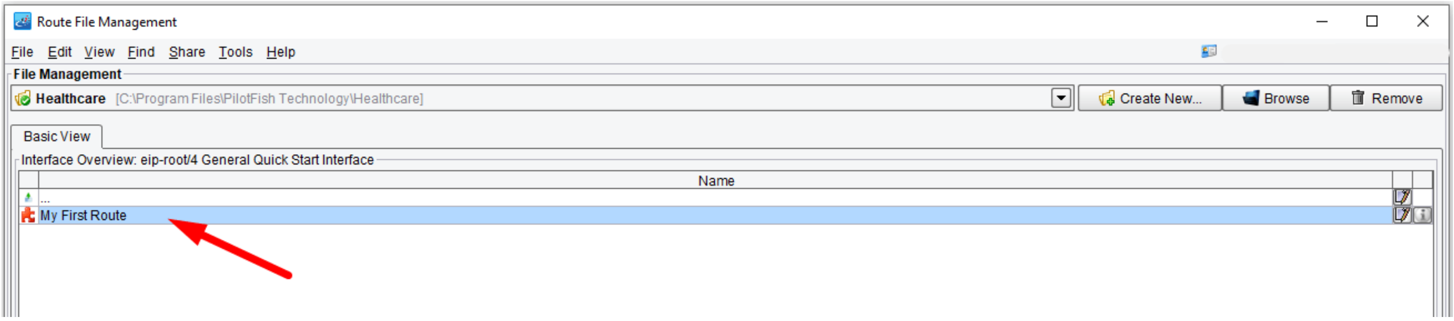

Route File Management
File Edit View Find Tools Help
File Management
Healthcare [C:\Program Files\PilotFish Technology\Healthcare]
Create New...
Browse
Remove
Basic View
Interface Overview: eip-root/4 General Quick Start Interface
Name
...
X
Add Route...
Add Interface Package...
Configure Common Modules...
Quit eiConsole

The **Route File Management** window opens. Notice the new Working Directory appears in the Basic View tab, **Interface Overview: eip-root/4 General Quick Start Interface**. You'll also be presented with an empty **Interface Overview** grid.

To create a new interface or route, click anywhere in the interface Overview grid, then, click the **Add Route** button.



When the **Add New** dialogue opens, enter the name "**My First Route**", and click **OK**.

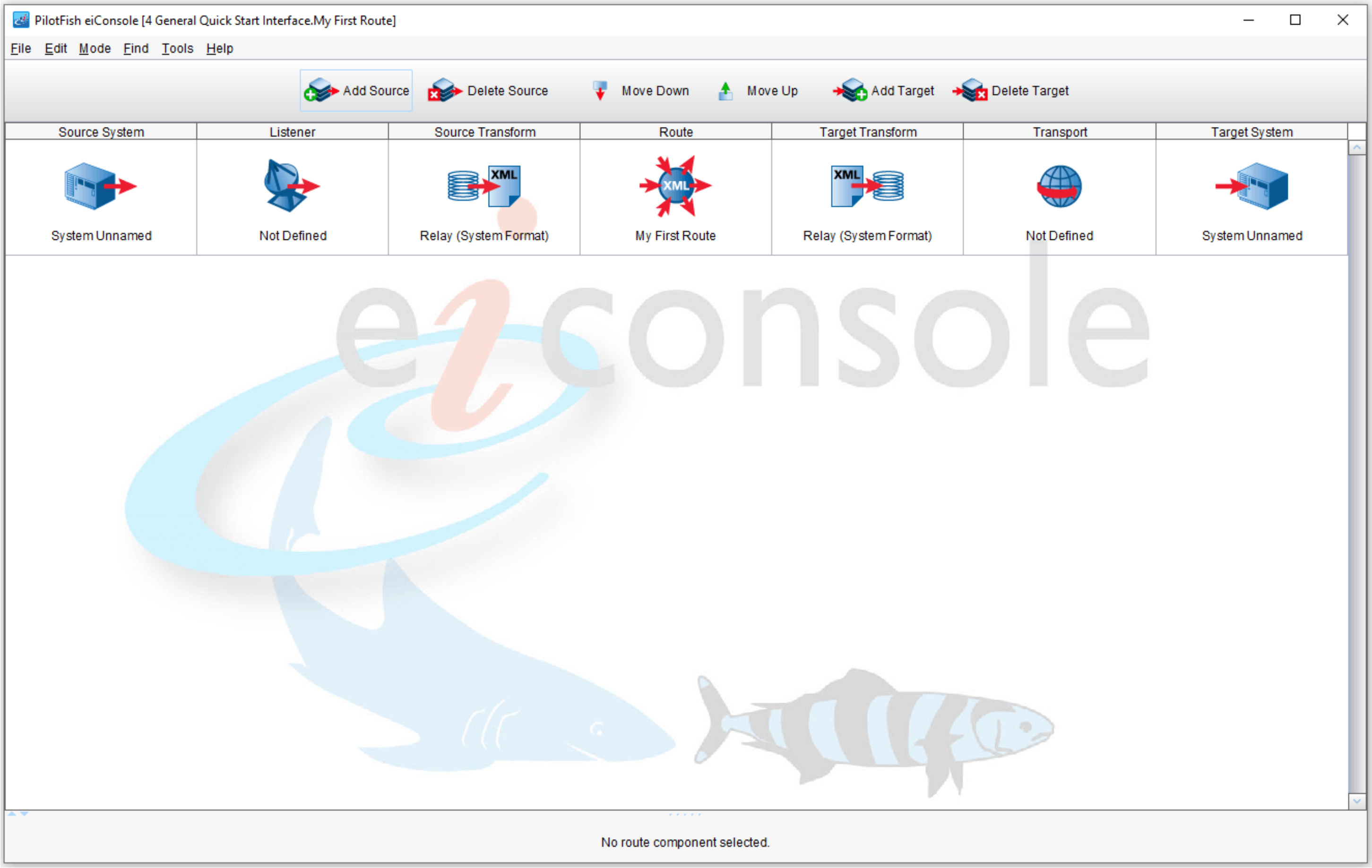

Route File Management
File Edit View Find Share Tools Help
File Management
Healthcare [C:\Program Files\PilotFish Technology\Healthcare]
Create New...
Browse
Remove
Basic View
Interface Overview: eip-root/4 General Quick Start Interface
Name
...
My First Route

The newly created route will appear in the Interface Overview grid.

Note: The icon next to your route name is red, indicating an unconfigured or incomplete interface.

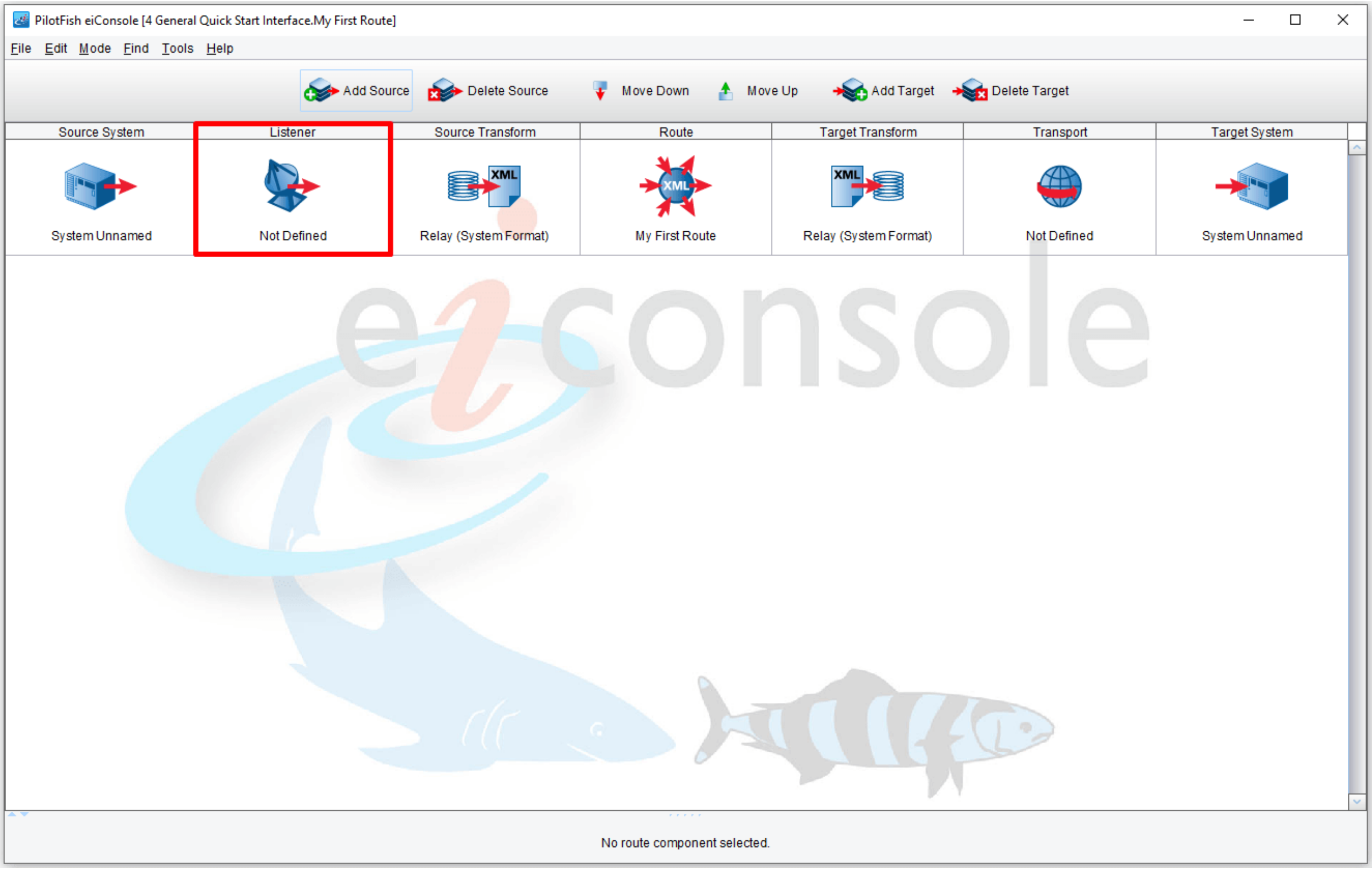
Now, to edit the route and to begin to configure your interface, select the row **My First Route** in the grid and double click.

The eiConsole's Main Route Grid

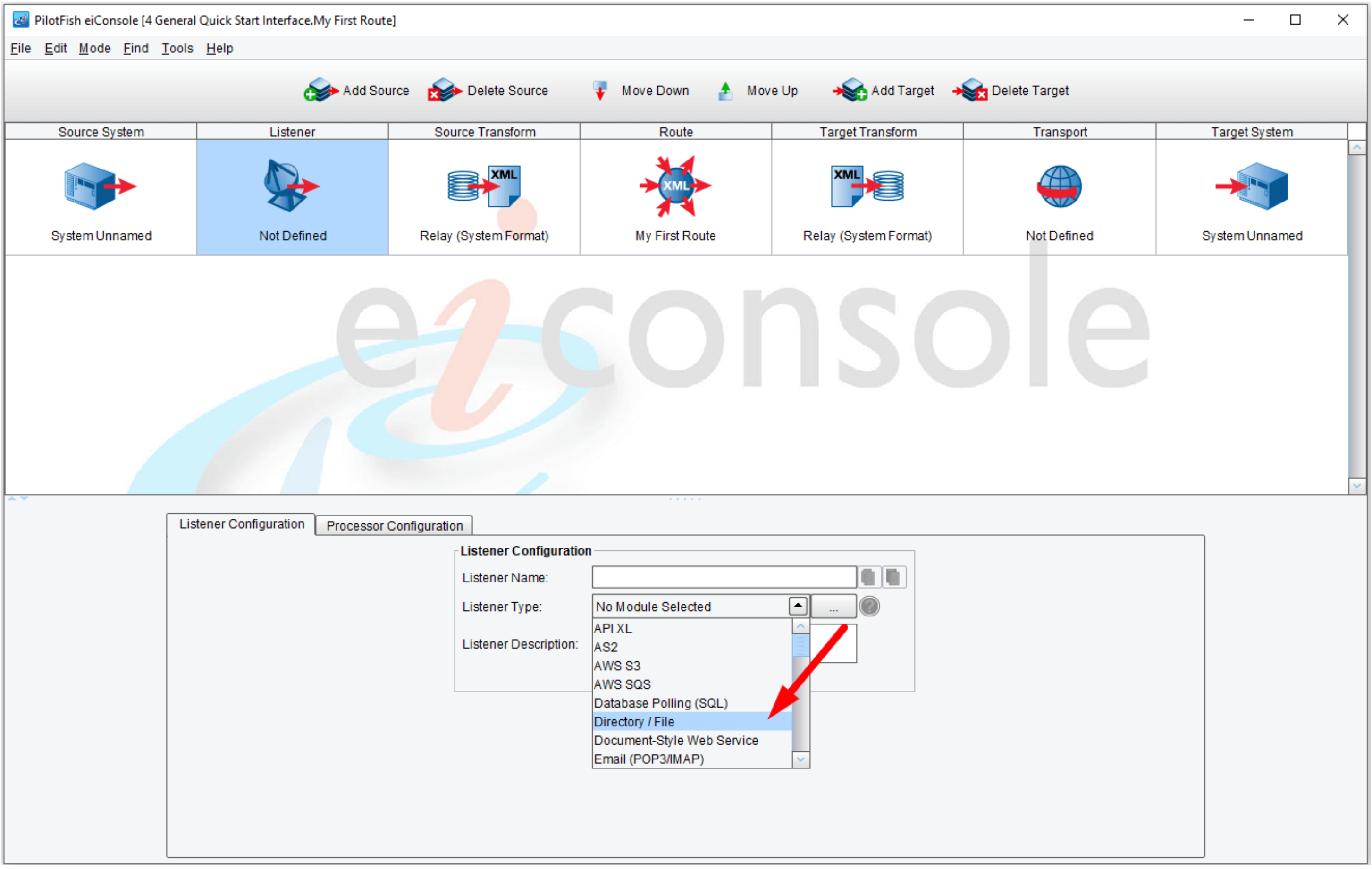


The eiConsole's Main Route grid will open. Now, define a Source and Target for your route. For this first interface, the general idea is to pick up a flat file from a directory, convert it to XML, and then drop it in the Target directory. For this example, the route will use one Source and one Target.

The Listener Stage



Click on the **Listener** stage.



Listener Configuration

Listener Name:

Listener Type: No Module Selected

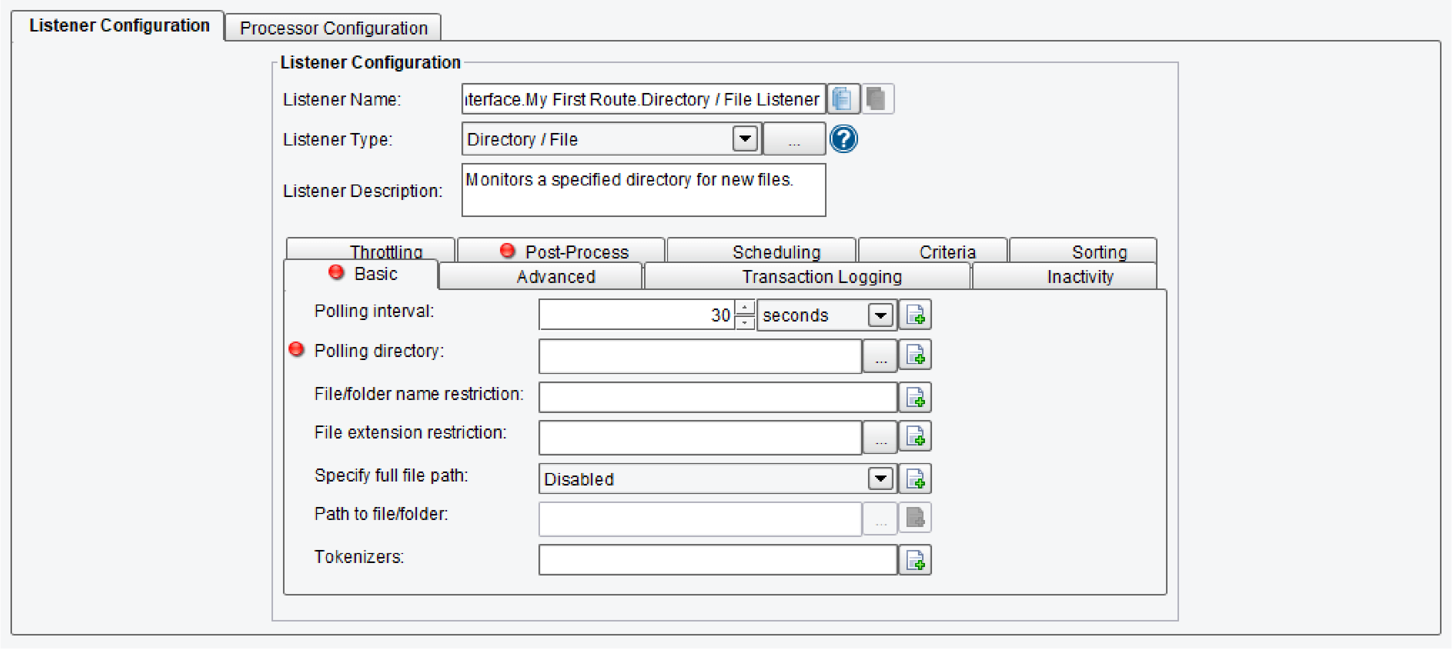
Listener Description:

- API XL
- AS2
- AWS S3
- AWS SQS
- Database Polling (SQL)
- Directory / File
- Document-Style Web Service
- Email (POP3/IMAP)

The Listener configuration panel will appear.

Note: When you select a stage, a configuration panel for that particular stage will appear in the bottom half of the screen.

Next, select **Directory / File** from the Listener Type drop-down.



Listener Configuration

Listener Name: iterface.My First Route.Directory / File Listener

Listener Type: Directory / File

Listener Description: Monitors a specified directory for new files.

Polling interval: 30 seconds

Polling directory:

File/folder name restriction:

File extension restriction:

Specify full file path: Disabled

Path to file/folder:

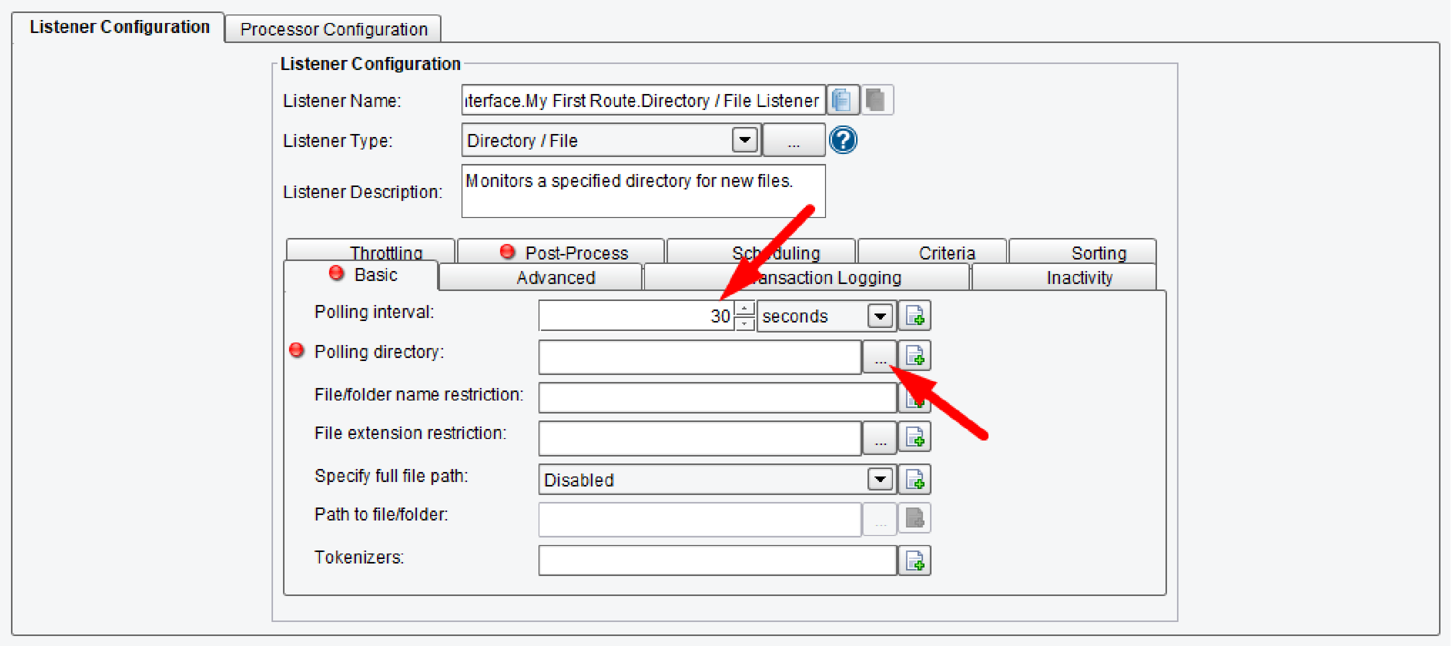
Tokenizers:

A set of configuration options for the Listener Module will appear.

Note: The eiConsole makes interface configuration very intuitive and provides the user with visual cues. In the panels, items marked with a red dot require configuration.

Enter **"10"** in the polling interval text box. (The drop-down lets you select minutes, hours, days or weeks for polling intervals.)

Note: After you have completed a required field and navigate away, the red dot will disappear.



Listener Configuration

Listener Name: ıterface.My First Route.Directory / File Listener

Listener Type: Directory / File

Listener Description: Monitors a specified directory for new files.

Polling interval: 30 seconds

● Polling directory:

File/folder name restriction:

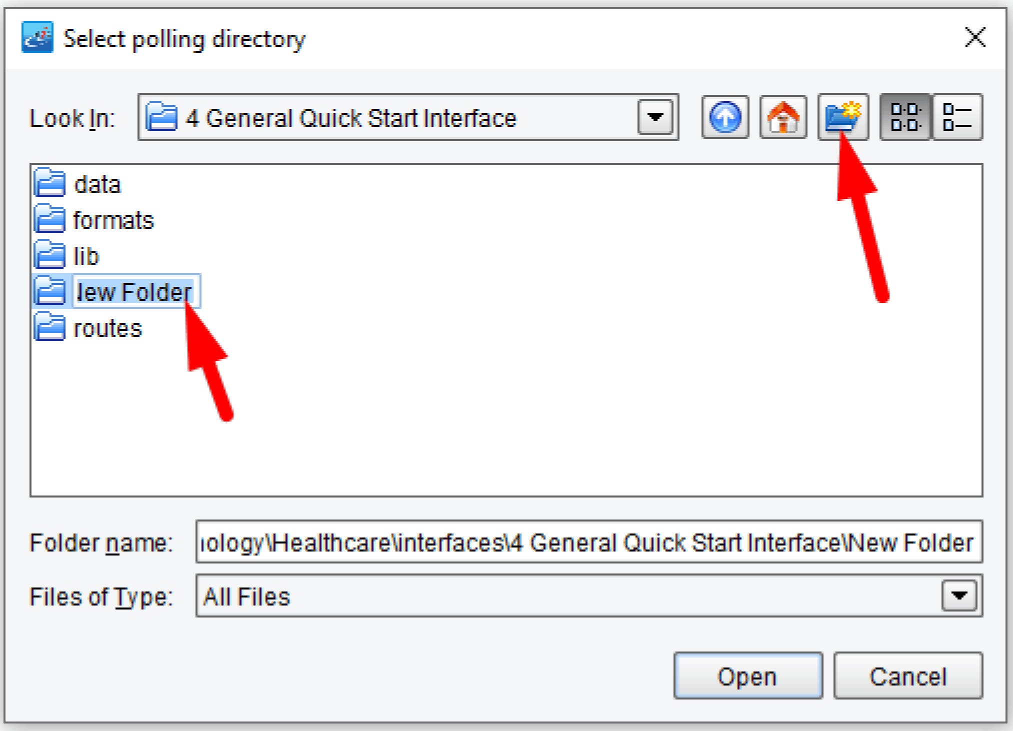
File extension restriction:

Specify full file path: Disabled

Path to file/folder:

Tokenizers:

Next, configure the Polling Directory (the directory from which files will be taken). Click the **Ellipsis** button next to the Polling Directory configuration field.



Select polling directory

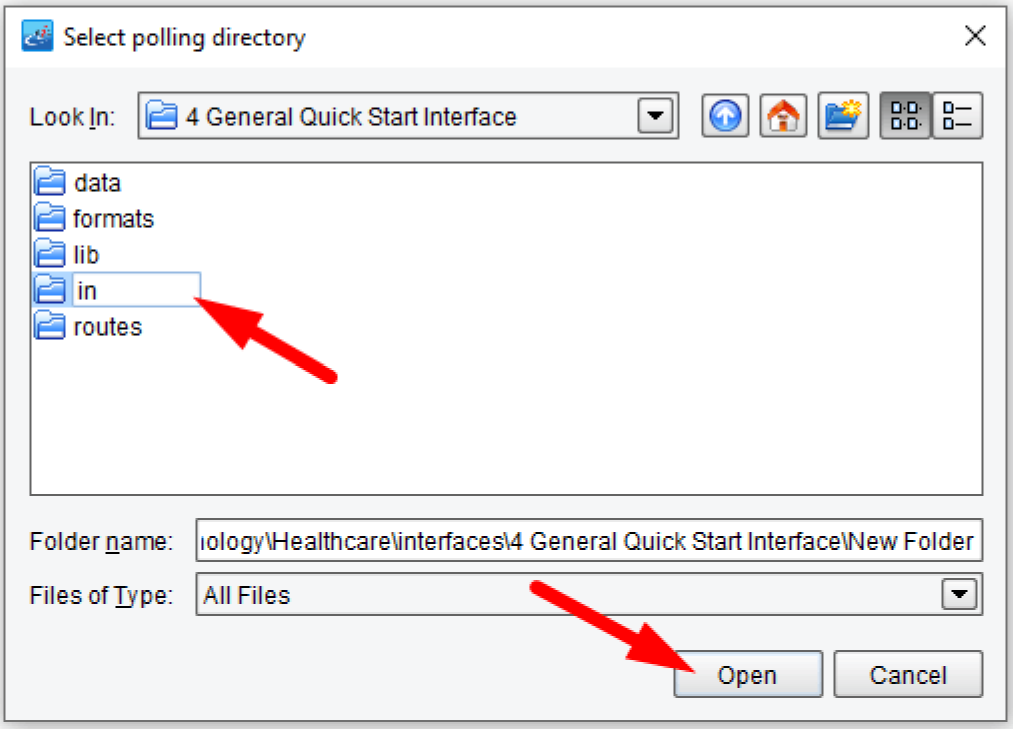
Look In: 4 General Quick Start Interface

- data
- formats
- lib
- lew Folder
- routes

Folder name: ıology\Healthcare\interfaces\4 General Quick Start Interface\New Folder

Files of Type: All Files

In your Working Directory, you'll find the **4 General Quick Start Interface** folder (or whatever you named your folder). Use the **Create New Folder** button and create a new folder.

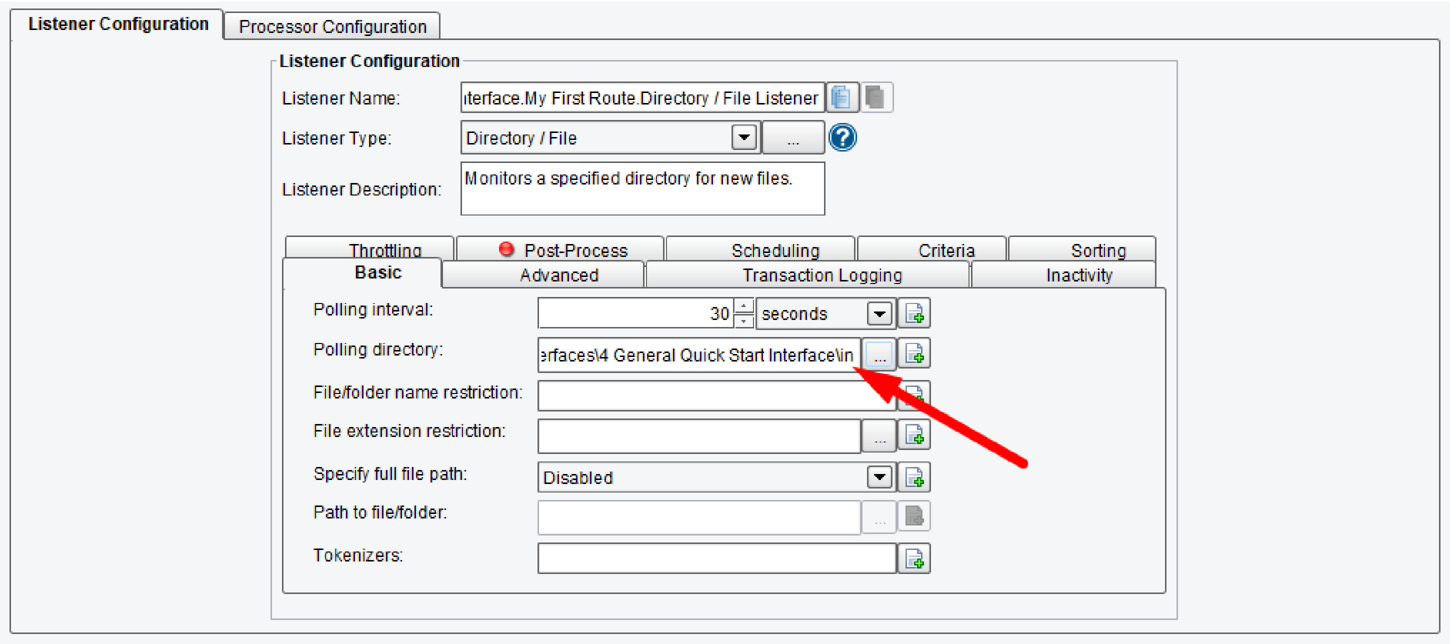


Look In: 4 General Quick Start Interface

Folder name: ıology\Healthcare\interfaces\4 General Quick Start Interface\New Folder

Files of Type: All Files

Select it and name it **"in"**. Select it again to highlight and click **Open.**



Listener Configuration

Listener Name: ıterface.My First Route.Directory / File Listener

Listener Type: Directory / File

Listener Description: Monitors a specified directory for new files.

Polling interval: 30 seconds

Polling directory: ərfaces\4 General Quick Start Interface\in

File/folder name restriction:

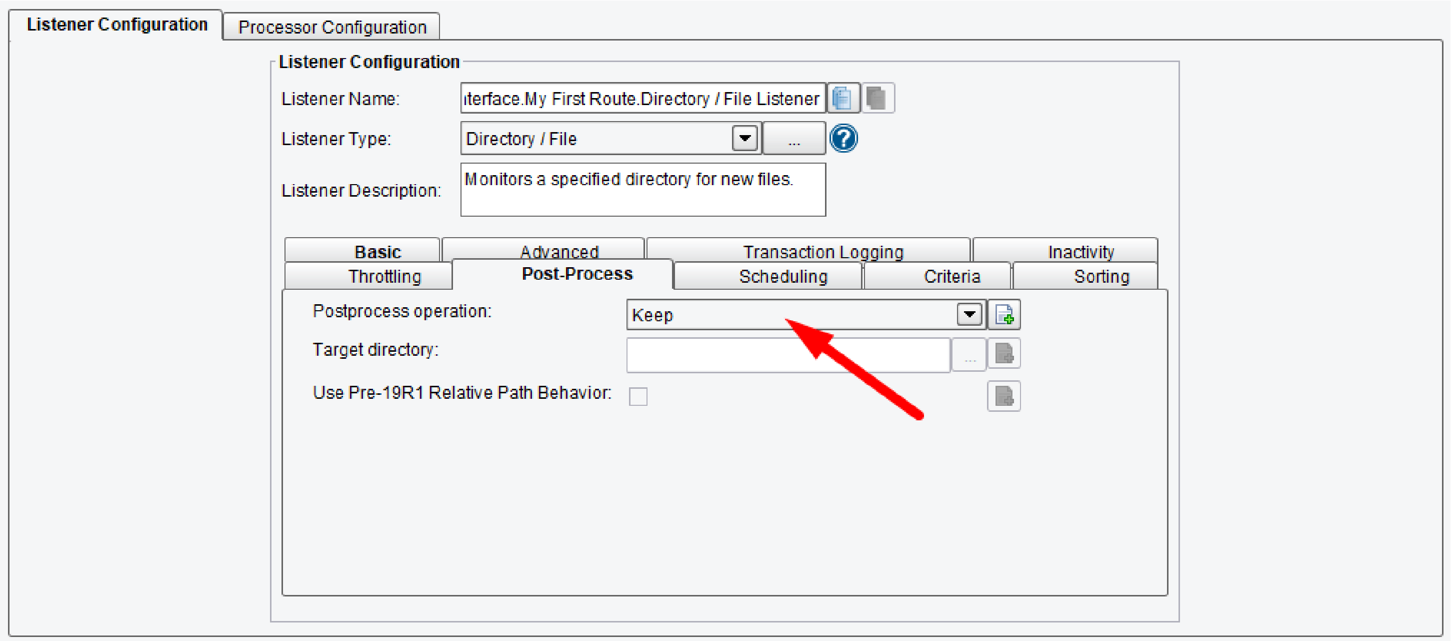
File extension restriction:

Specify full file path: Disabled

Path to file/folder:

Tokenizers:

The path to the **"in"** folder will appear in the Polling Directory configuration field. The red dot has now disappeared next to Polling Directory, and since all required items have been filled in within the Basic tab, the red dot in the Basic tab is gone, as well.



Listener Configuration

Listener Name: ıterface.My First Route.Directory / File Listener

Listener Type: Directory / File

Listener Description: Monitors a specified directory for new files.

Postprocess operation: Keep

Target directory:

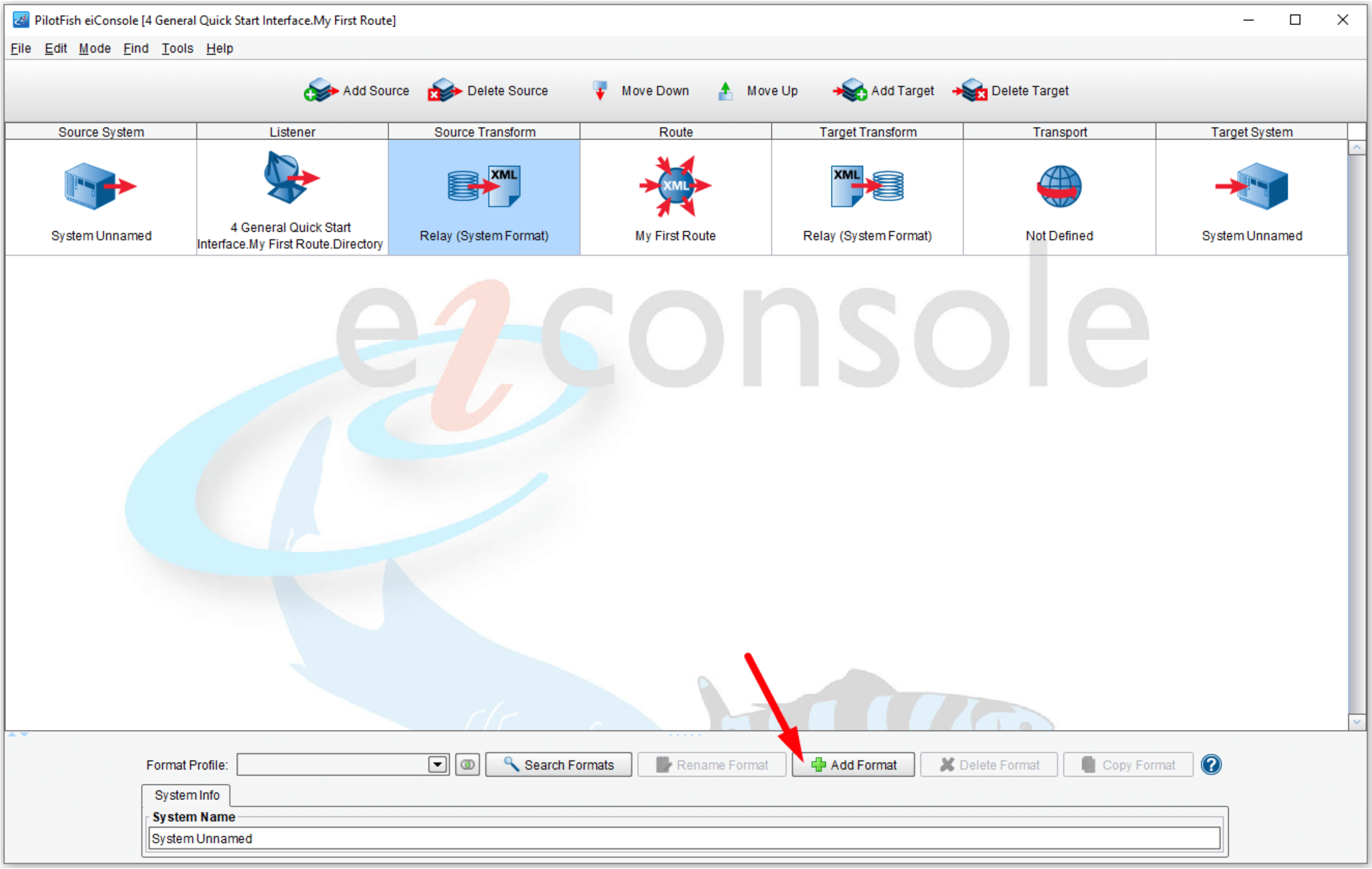
Use Pre-19R1 Relative Path Behavior: □

Set the Post-Process 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.

The Source Transform Stage



PilotFish eiConsole [4 General Quick Start Interface.My First Route]

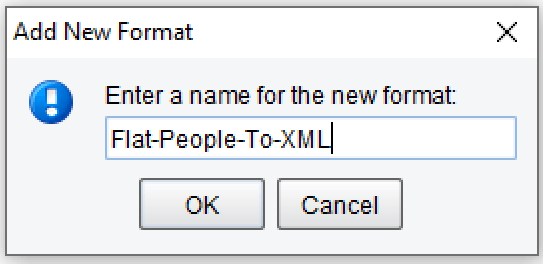
Source System	Listener	Source Transform	Route	Target Transform	Transport	Target System
System Unnamed	4 General Quick Start Interface.My First Route.Directory	Relay (System Format)	My First Route	Relay (System Format)	Not Defined	System Unnamed

Format Profile:

System Name: System Unnamed

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

To create a new transformation, click the **Add Format** button.

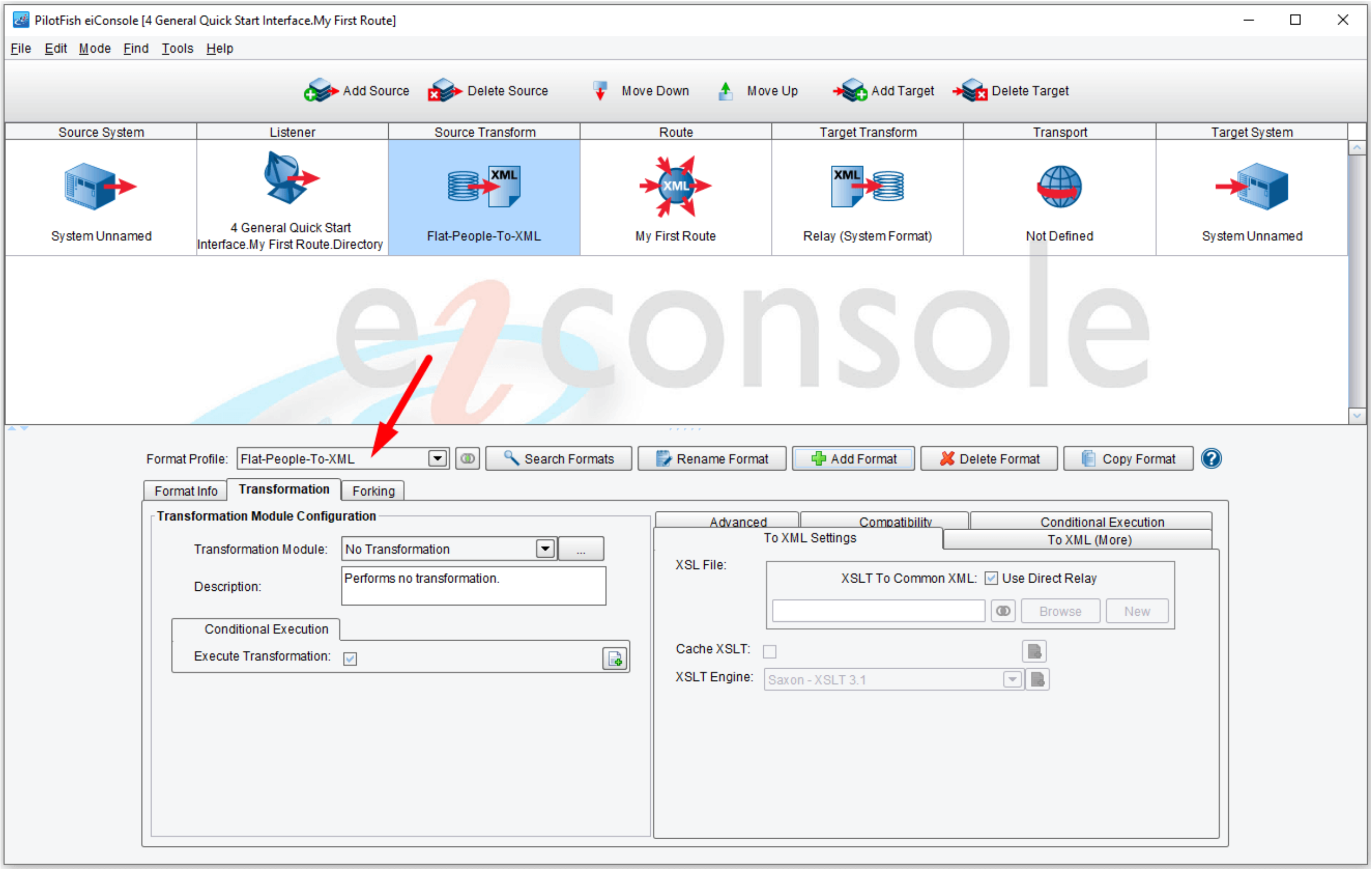


Add New Format

Enter a name for the new format:

Flat-People-To-XML

Our first transformation will be responsible for converting a flat file to XML. When prompted, add the format name **"Flat-People-To-XML"** and click **OK**.



PilotFish eiConsole [4 General Quick Start Interface.My First Route]

Format Profile: Flat-People-To-XML

Transformation Module Configuration

Transformation Module: No Transformation

Description: Performs no transformation.

Conditional Execution

Execute Transformation: ☑

XSL File:

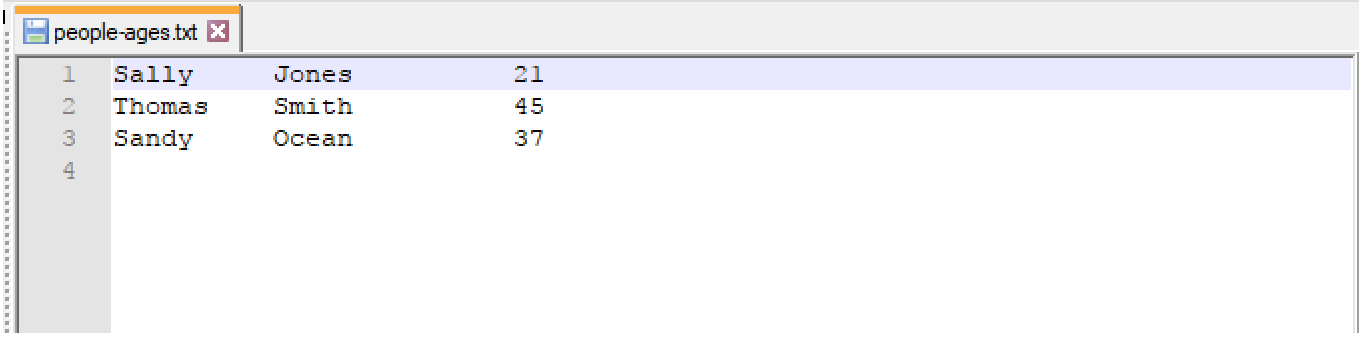
XSLT To Common XML: ☑ Use Direct Relay

Cache XSLT: □

XSLT Engine: Saxon - XSLT 3.1

The "Flat-People-To-XML" will become selected in the Format Profile drop-down. And now the Transformation Module and XSLT Configuration panels will appear. Configure both the Transformation Module and an XSLT data mapping.

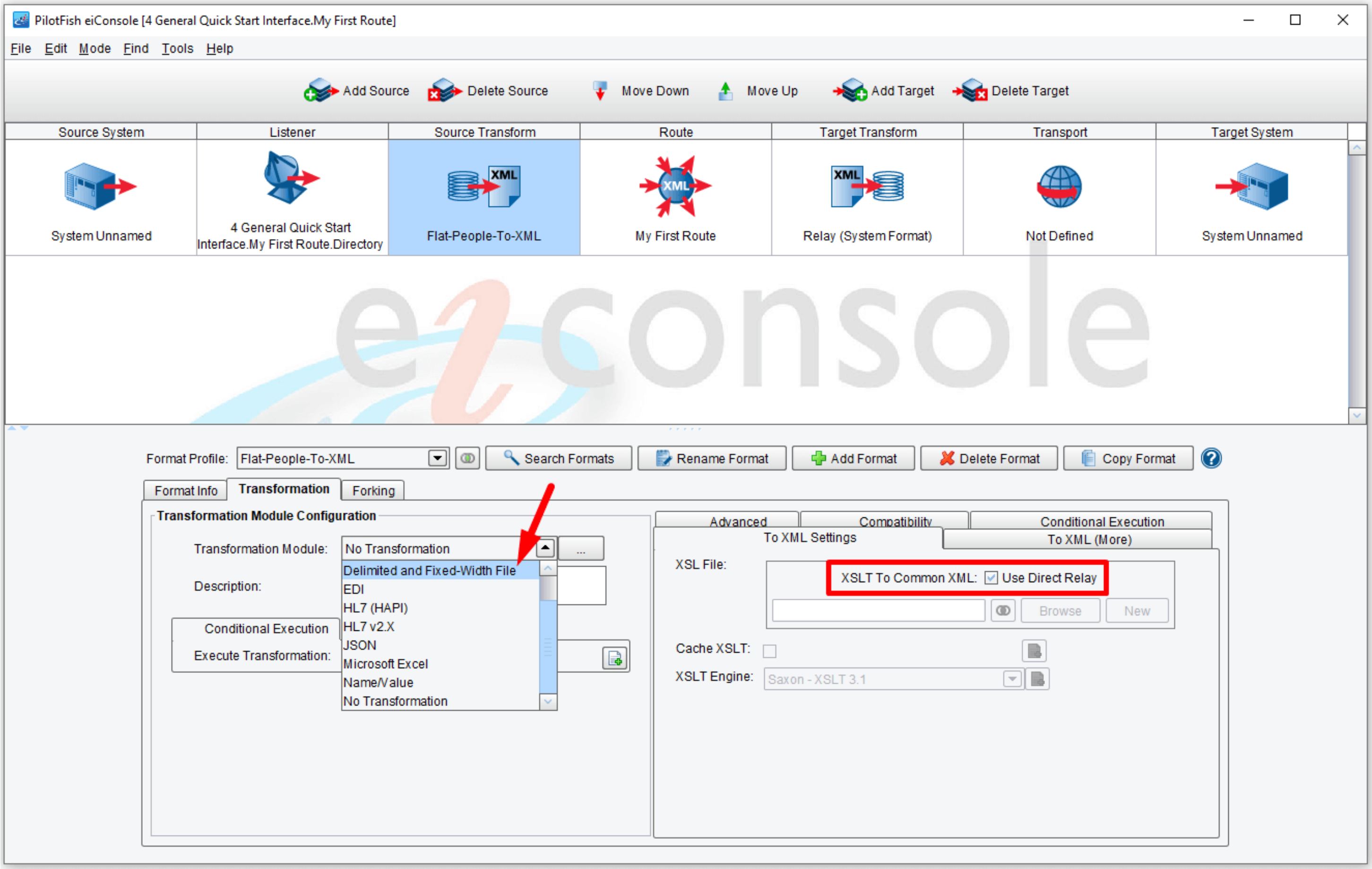
Note: Transformation Modules are responsible for converting non-XML data into an XML format so they can later be mapped.



people-ages.txt

```
Sally     Jones     21
Thomas    Smith     45
Sandy     Ocean     37
```

Our input document will be a text file, **people-ages.txt,** from the distribution **data** folder. This distribution data folder is your {eiConsole-installation-directory}/samples/Getting-Started-Project/interfaces/1 General Quick Start Tutorial/data. It contains a set of fixed-width records containing a first name, last name, and age.



Format Profile: Flat-People-To-XML

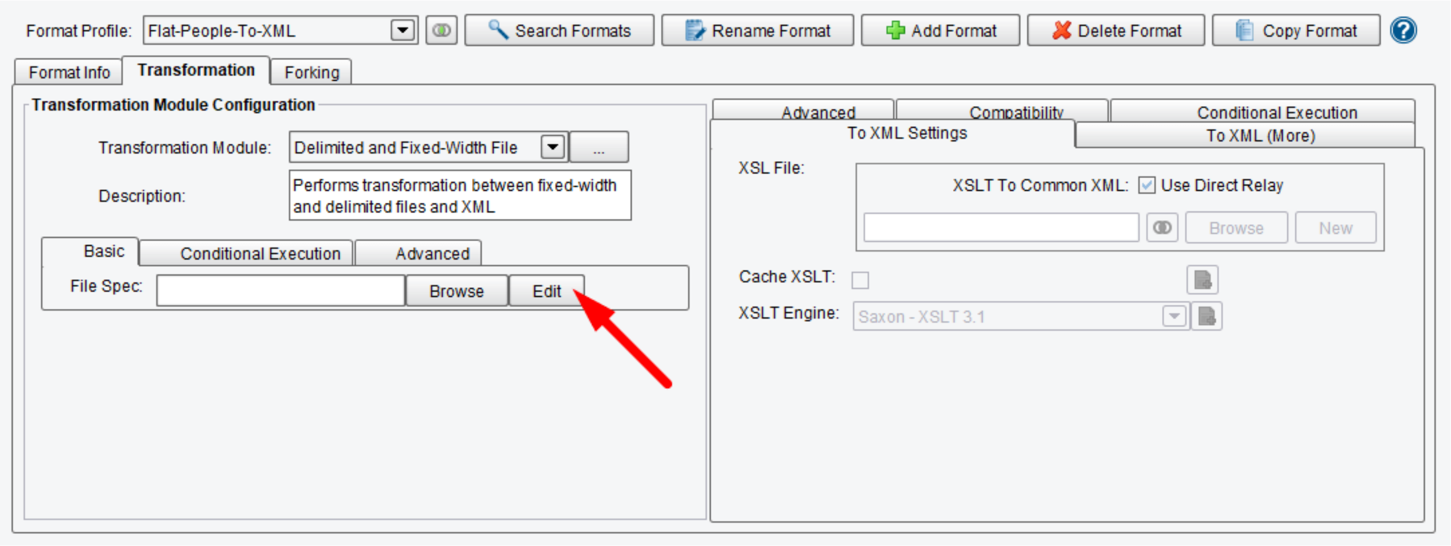
Transformation Module: No Transformation

XSLT To Common XML: ☑ Use Direct Relay

Cache XSLT: □

XSLT Engine: Saxon - XSLT 3.1

To work with this type of data, choose the **Delimited and Fixed-Width File Transformation Module** from the drop-down. Leave the **Use Direct Relay** box checked.



Format Profile: Flat-People-To-XML

Transformation Module: Delimited and Fixed-Width File

Description: Performs transformation between fixed-width and delimited files and XML

XSLT To Common XML: ☑ Use Direct Relay

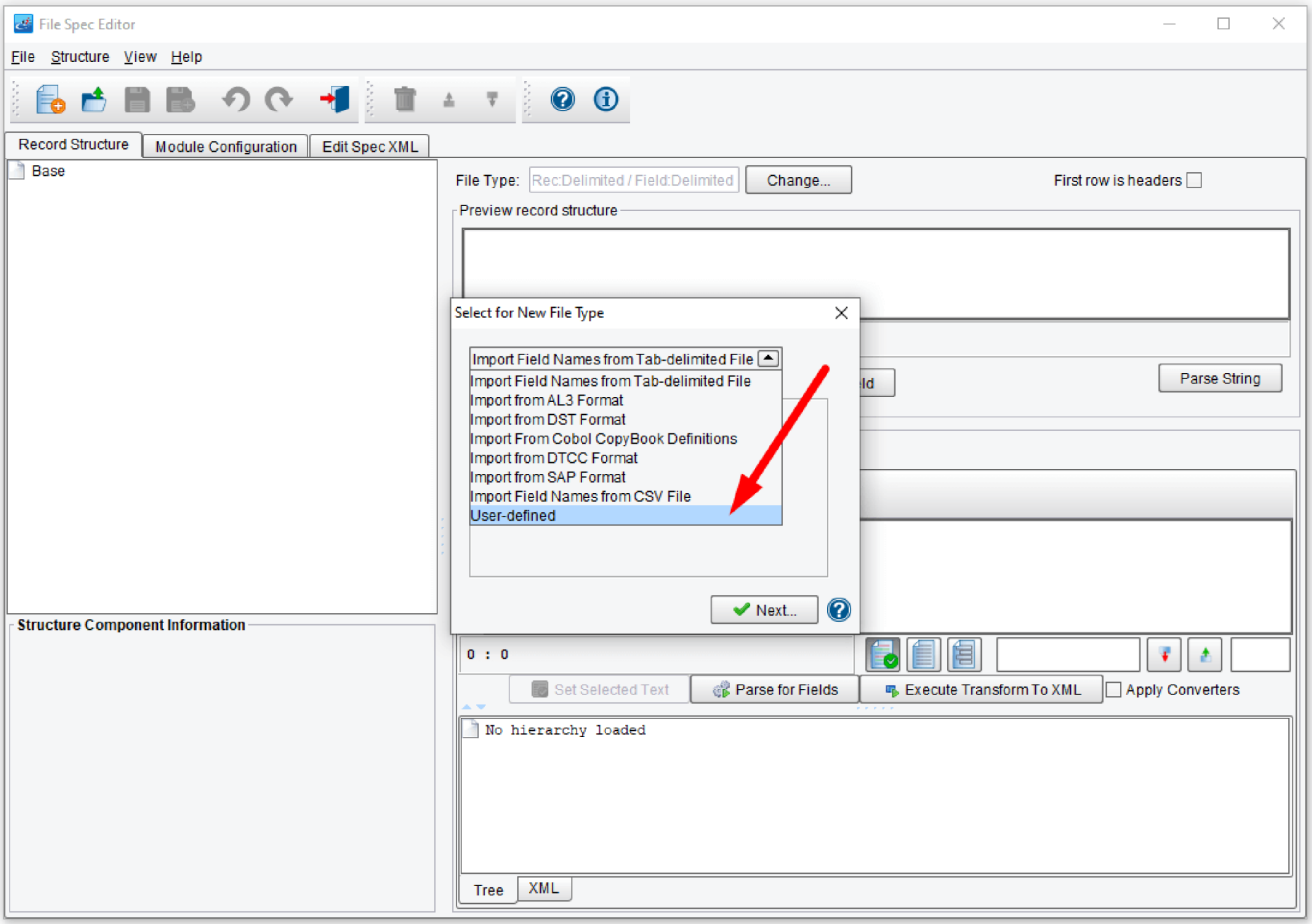
Cache XSLT: □

XSLT Engine: Saxon - XSLT 3.1

Next, click the **Edit** button in the File Spec configuration panel.

This will launch the File Specification Editor.

The eiConsole File Specification Editor



File Type: Rec:Delimited / Field:Delimited

First row is headers □

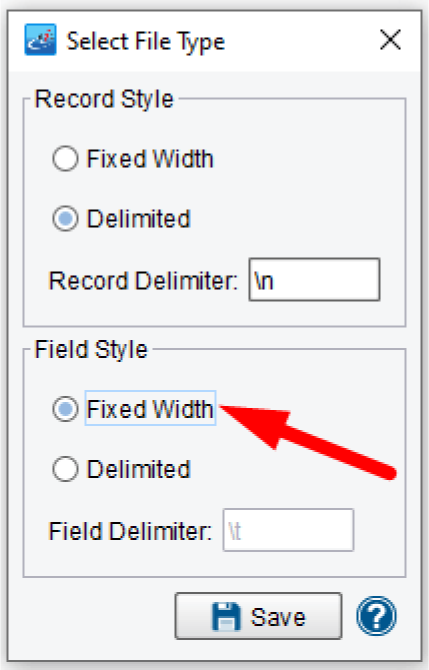
Select for New File Type: User-defined

0 : 0

□ Apply Converters

```
No hierarchy loaded
```

The Select for New File Type wizard will appear. Select **User-defined** from the drop-down, and click **Next**.



Select File Type

Record Style

○ Fixed Width

● Delimited

Record Delimiter: \n

Field Style

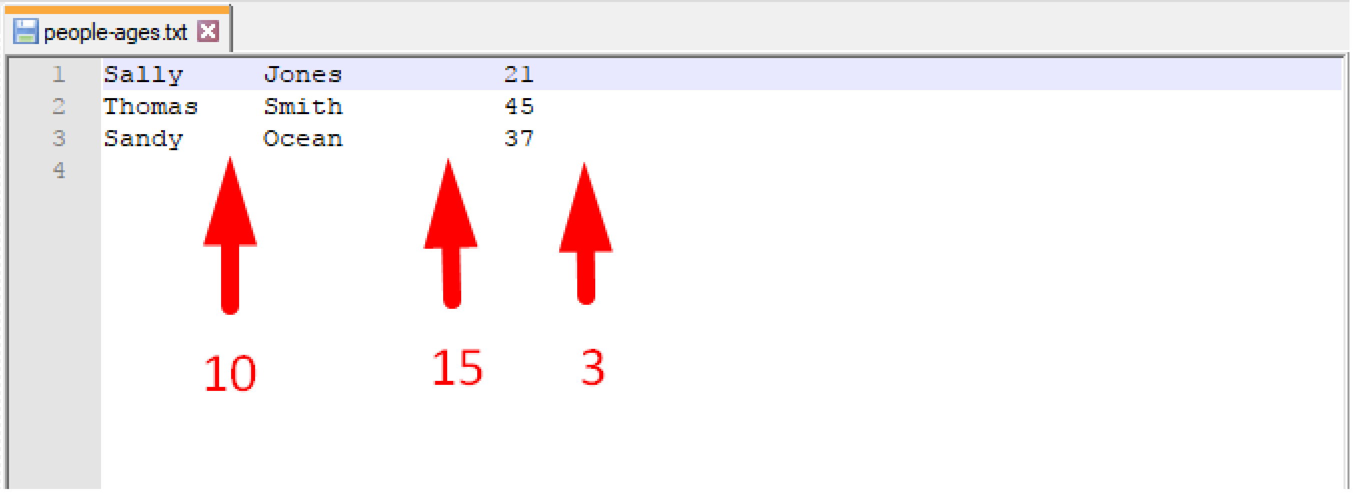
● Fixed Width

○ Delimited

Field Delimiter: \t

Now, describe the type of file that the transformation module will be parsing. Since the file contains records that are Delimited with new lines, leave that option as is.

However, our fields are fixed width. So, select the Fixed Width Field Style radio box and click **Save**.

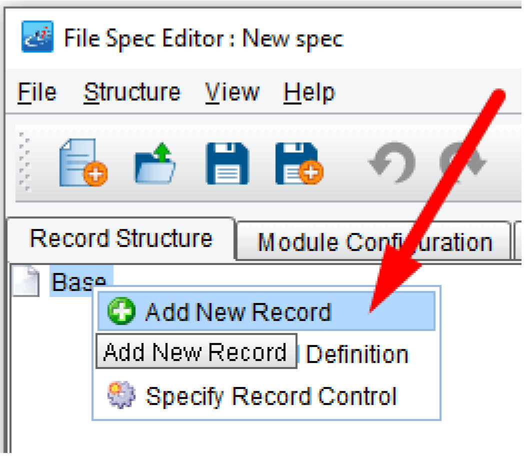


people-ages.txt

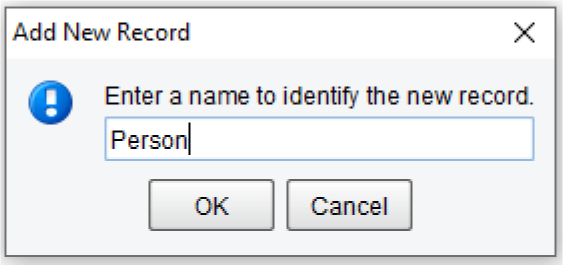
```
Sally     Jones          21
Thomas    Smith          45
Sandy     Ocean          37
```

To describe the structure of the flat file that will be parsed, let's take a look at the **people.ages.txt** file.

The first 10 bytes of each line will be the first name. The following 15 bytes are the last name. The final 3 are the age.



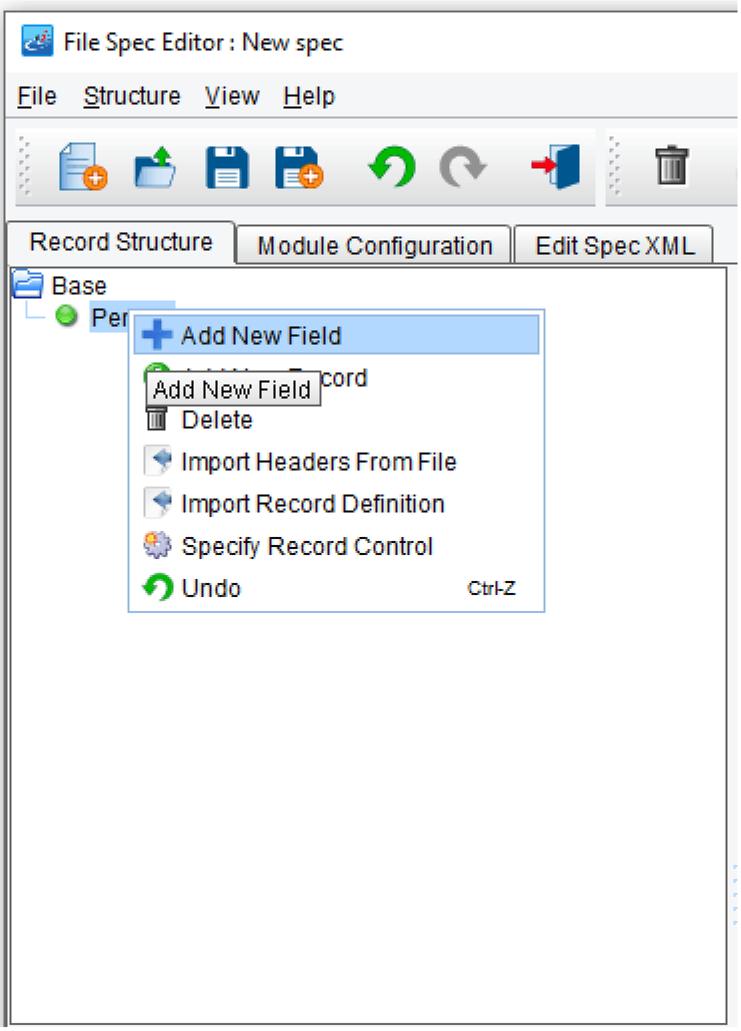
To configure this in the File Specification Editor, right-click the Base icon in the record structure panel, then click **Add New Record**.


Add New Record

Enter a name to identify the new record.

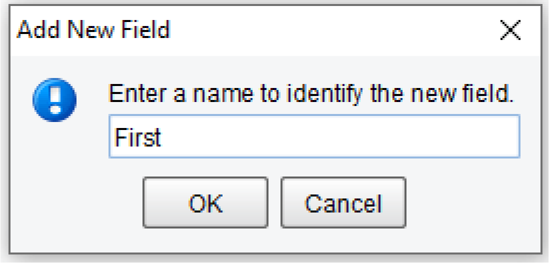
Person

Since each record represents a person, enter the name **"Person"** in the Add New Record dialogue and click **OK**.



A green node will appear in the record structure tree.

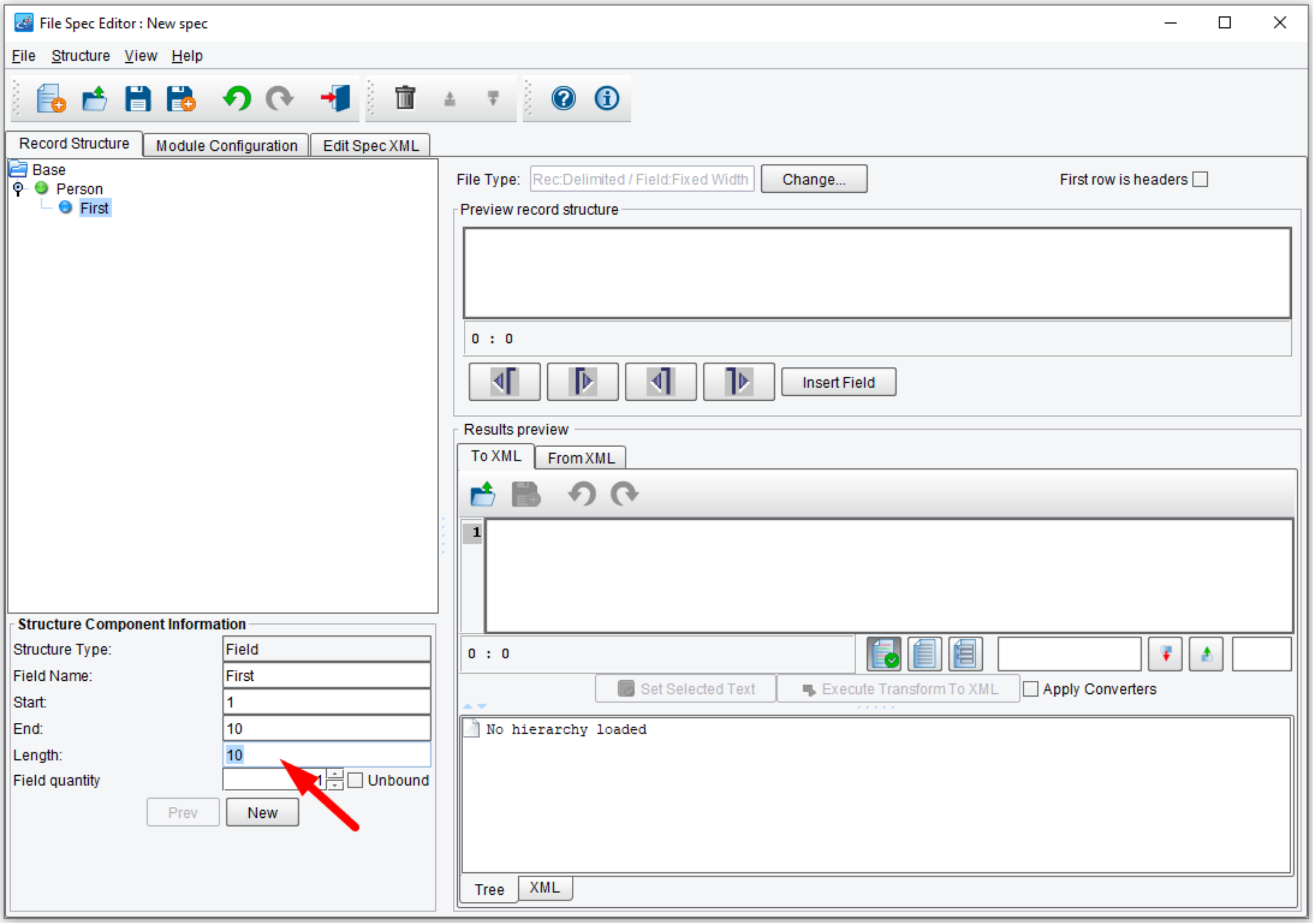
To describe each field in the Person record, right-click the Person node and select **Add New Field**.



Enter a name to identify the new field.

First

Provide the field name "**First**" in the Add New Field Dialogue, and click **OK**.



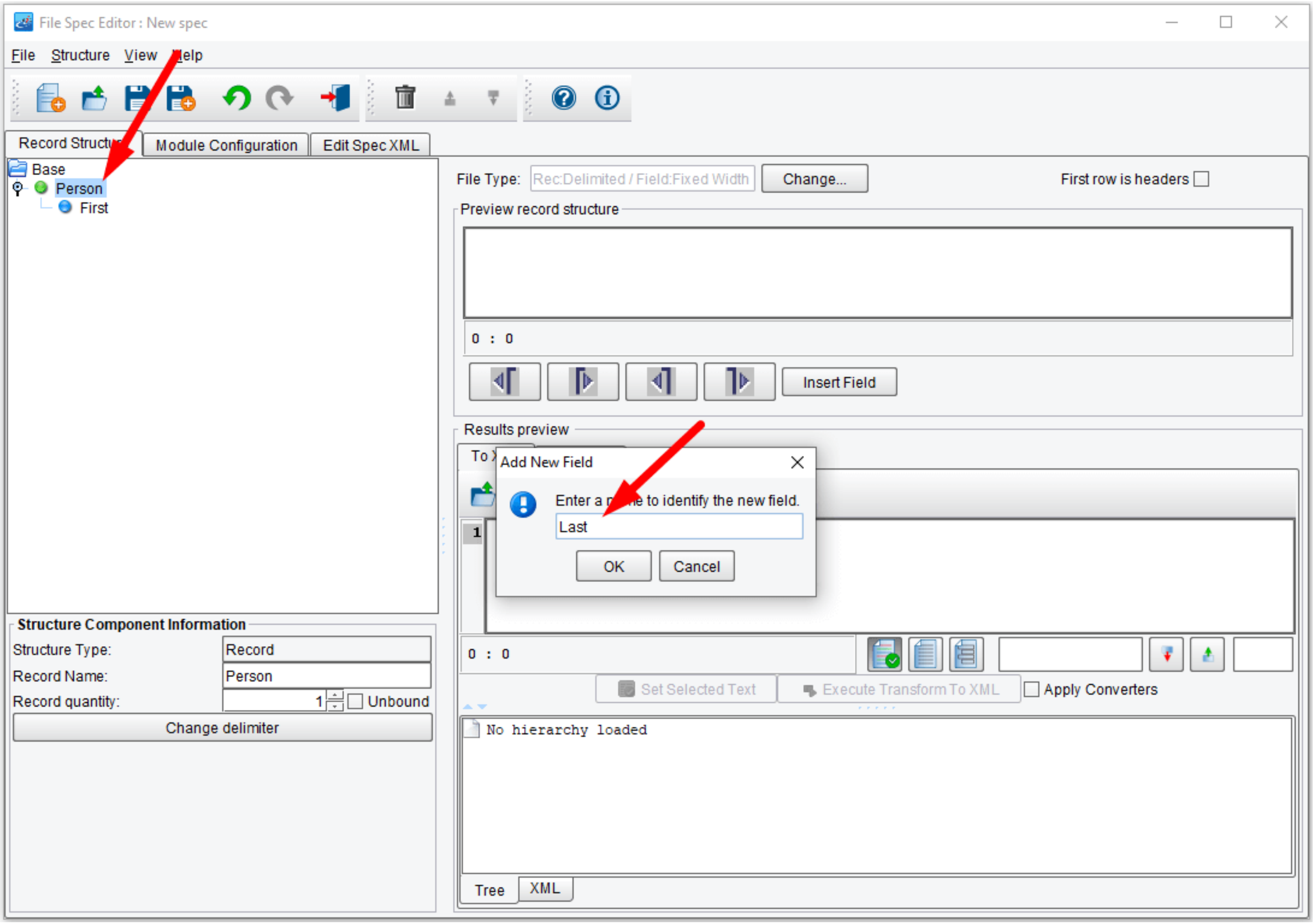
Structure Component Information

Structure Type:	Field
Field Name:	First
Start:	1
End:	10
Length:	10
Field quantity	1 □ Unbound

In the Structure Component Information area underneath the Record Structure panel, specify the start position, end position, or length of the field.

Type in **"10"** bytes for length and click enter/return.

Note: The end positions are automatically updated based on the length entered.



File Type: Rec:Delimited / Field:Fixed Width

First row is headers □

Preview record structure

0 : 0

Add New Field

Enter a name to identify the new field.

Last

Structure Component Information

Structure Type: Record

Record Name: Person

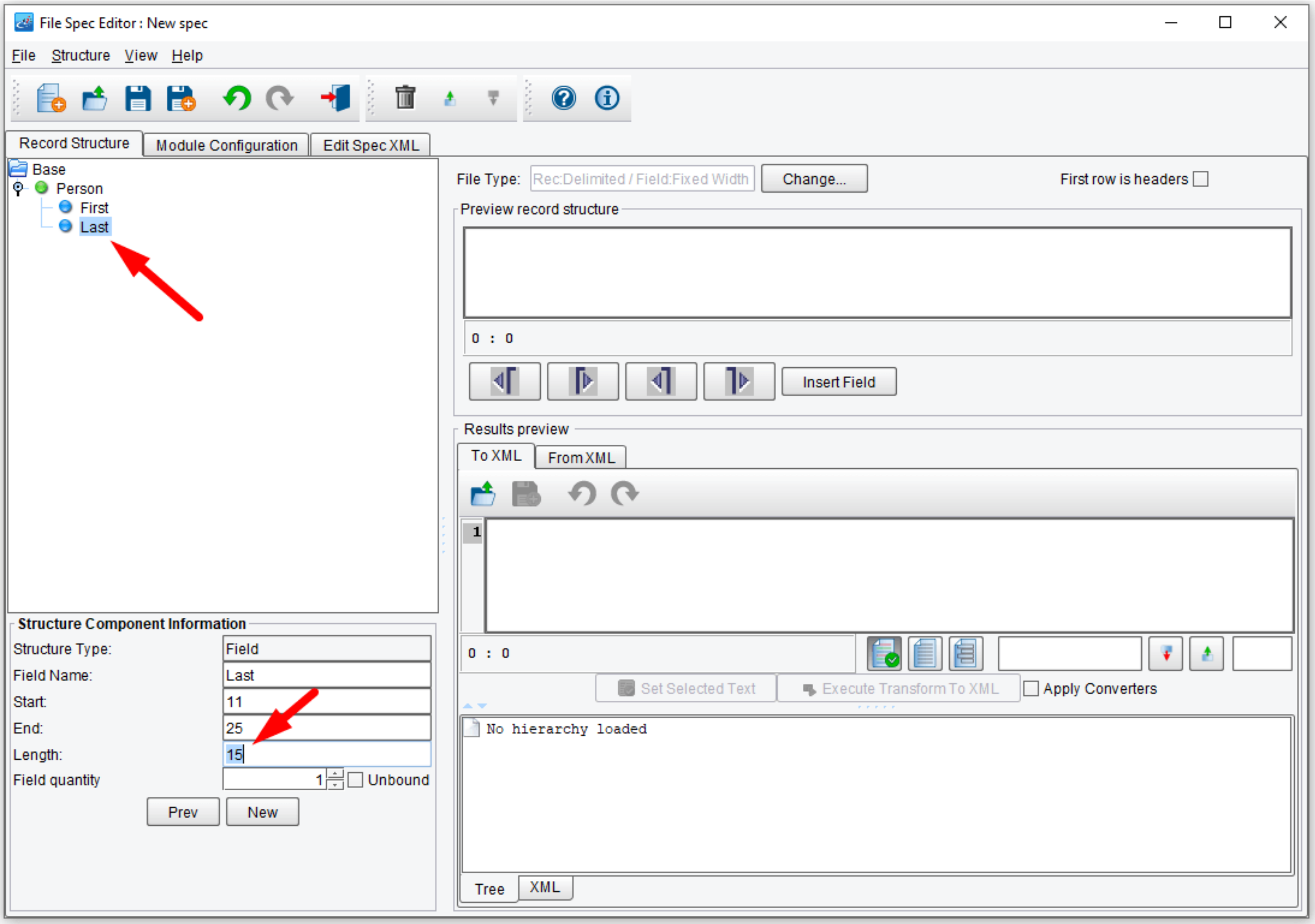
Record quantity: 1 □ Unbound

□ Apply Converters

```
No hierarchy loaded
```

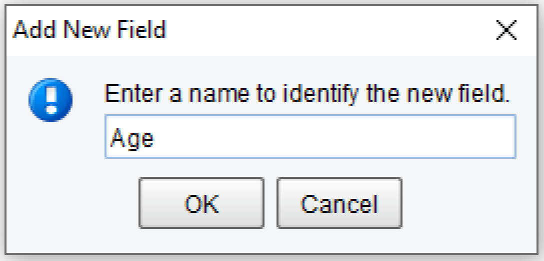
Right-click the person record and select **Add New Field**.

Type the field name **"Last"** and click **OK.**



Structure Type:	Field
Field Name:	Last
Start:	11
End:	25
Length:	15
Field quantity	1 □ Unbound

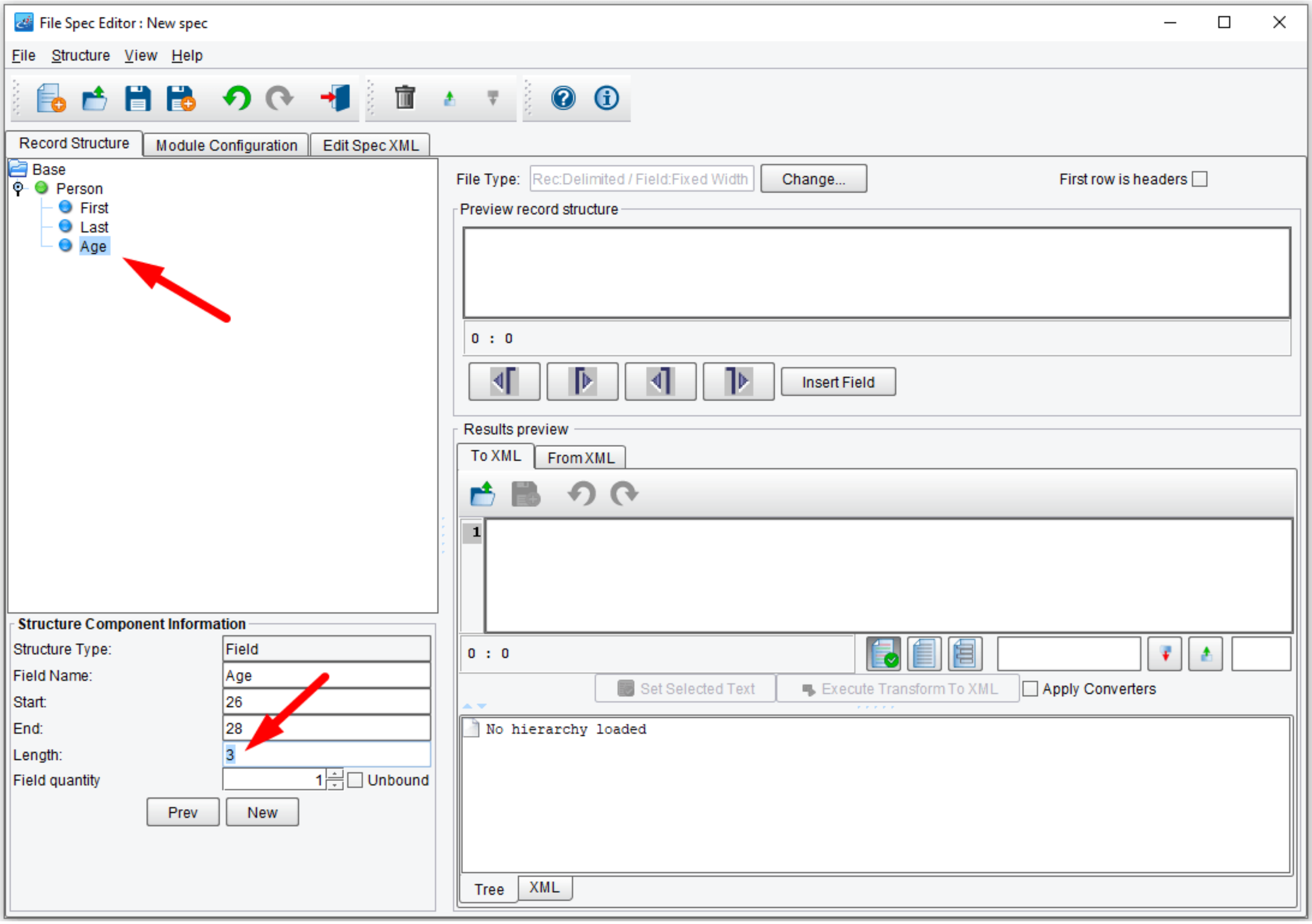
Select the blue Last node and type **"15"** into the length field.


Add New Field

Enter a name to identify the new field.

Age

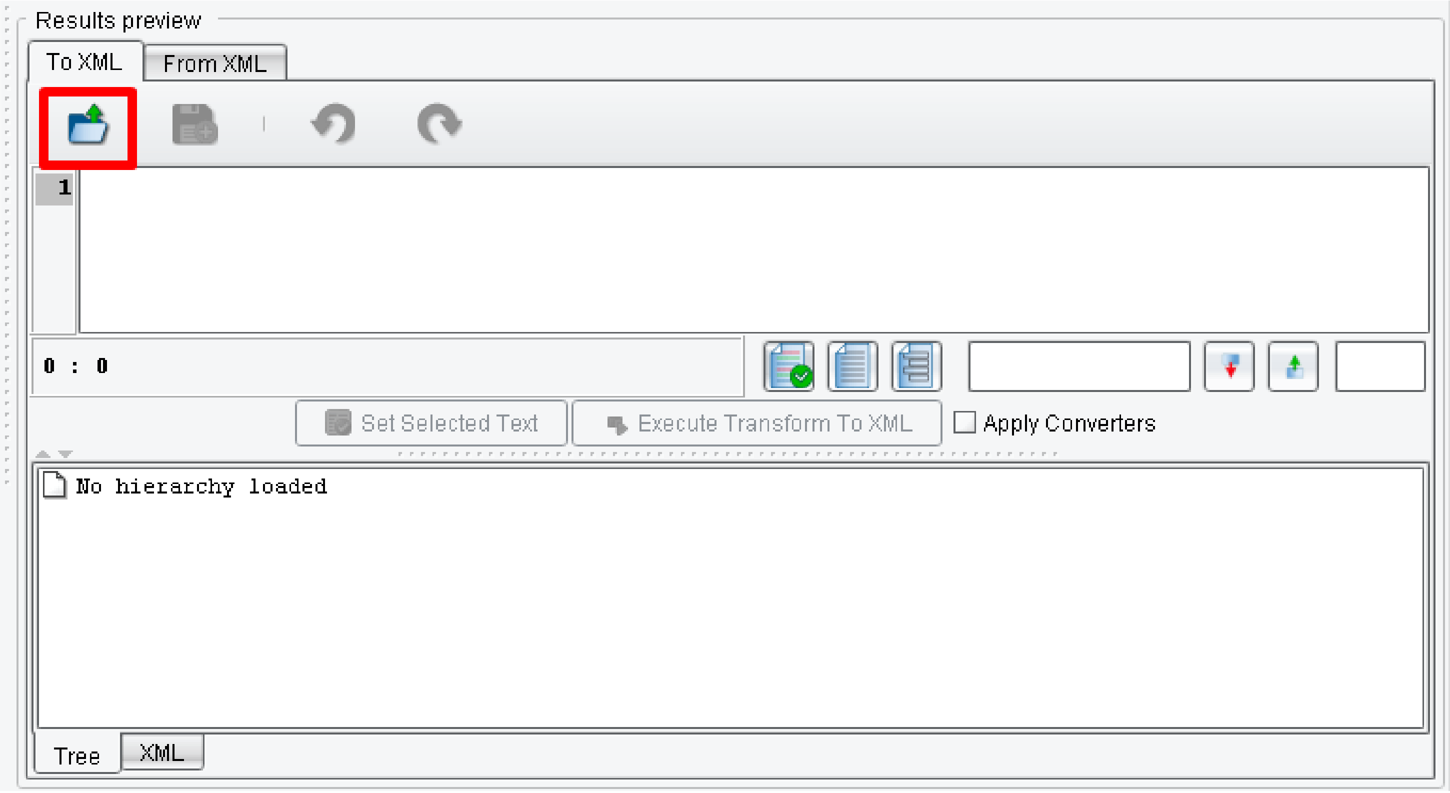
Finally, repeat the process one more time. Select the green Person node, right click and select **Add New Field**, type in **"Age"**, click **OK**.



Structure Component Information

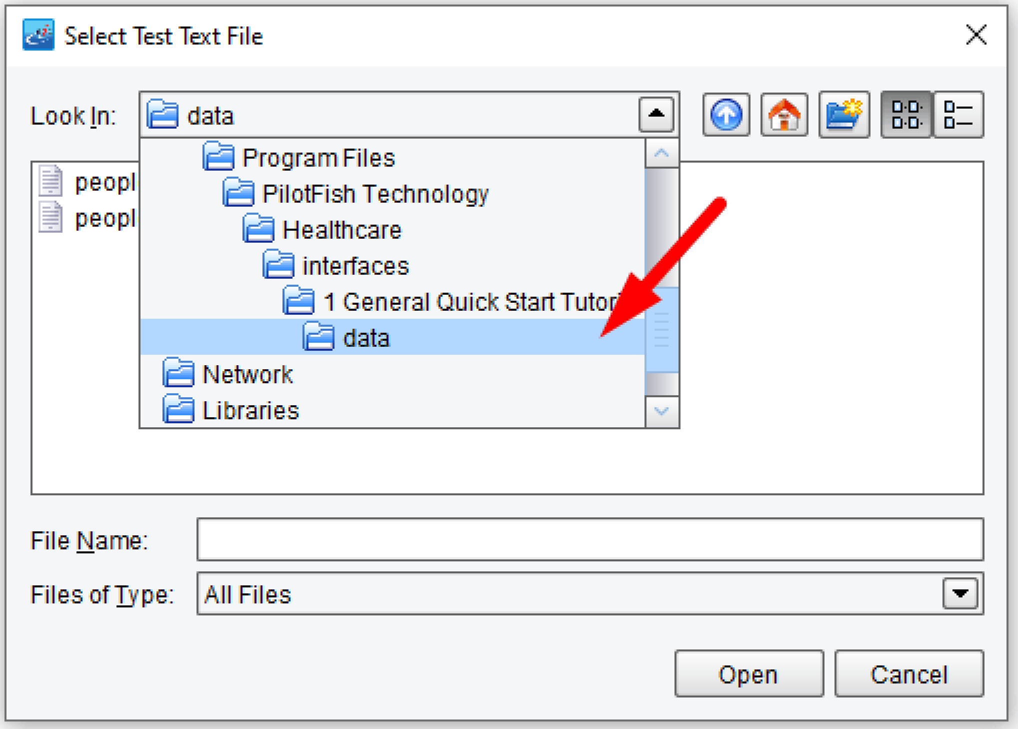
Structure Type:	Field
Field Name:	Age
Start:	26
End:	28
Length:	3
Field quantity	1 □ Unbound

Next, select the blue **Age** node and type in **"3"** in the length field. Then click enter/return.

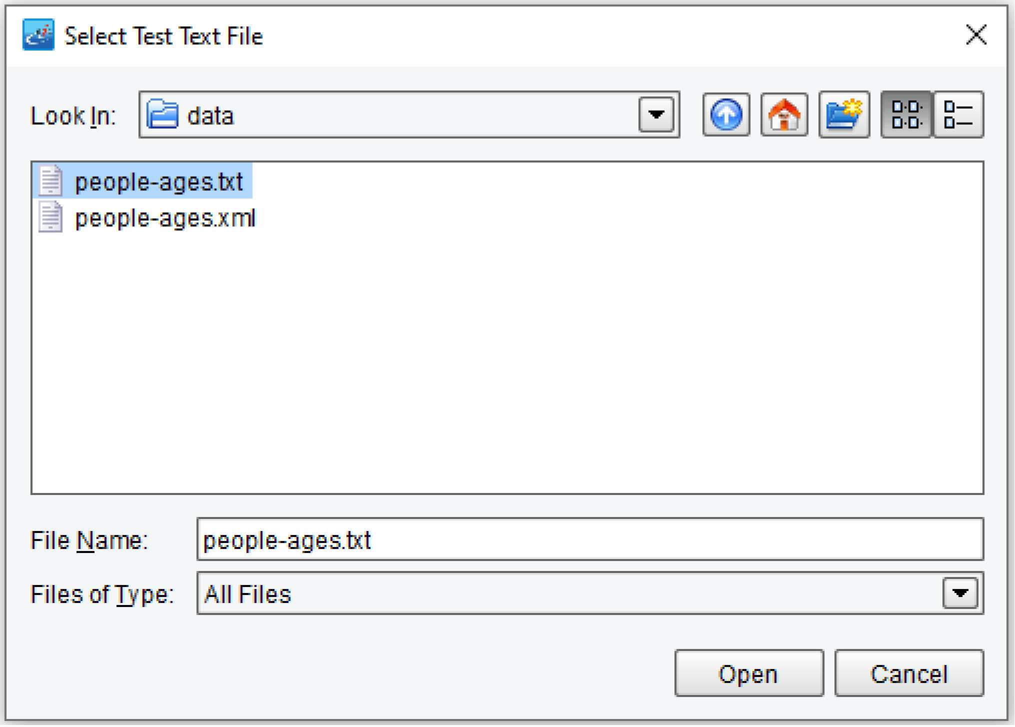


With our record defined, let's test the parsing. To do this, load a sample file into the results preview area.

Click the **Folder** icon.



Navigate to the **people-ages.txt** file in the data folder of your distribution.

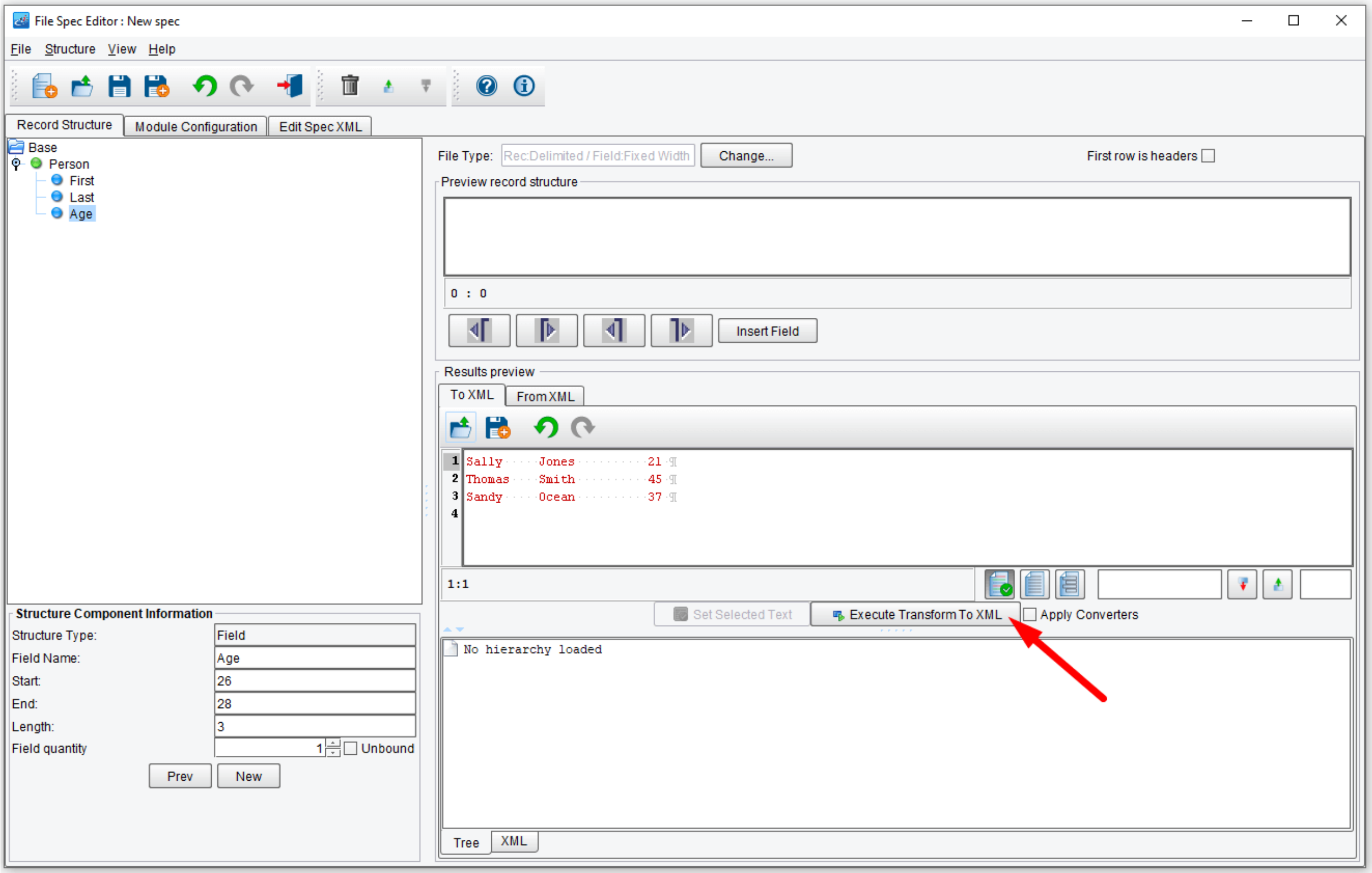


Look In: data

File Name: people-ages.txt

Files of Type: All Files

Click the **Open** button.



File Type: Rec:Delimited / Field:Fixed Width

First row is headers □

```
0 : 0
```

```
1 Sally    Jones          21 ¶
2 Thomas   Smith          45 ¶
3 Sandy    Ocean          37 ¶
4
```

```
1:1
```

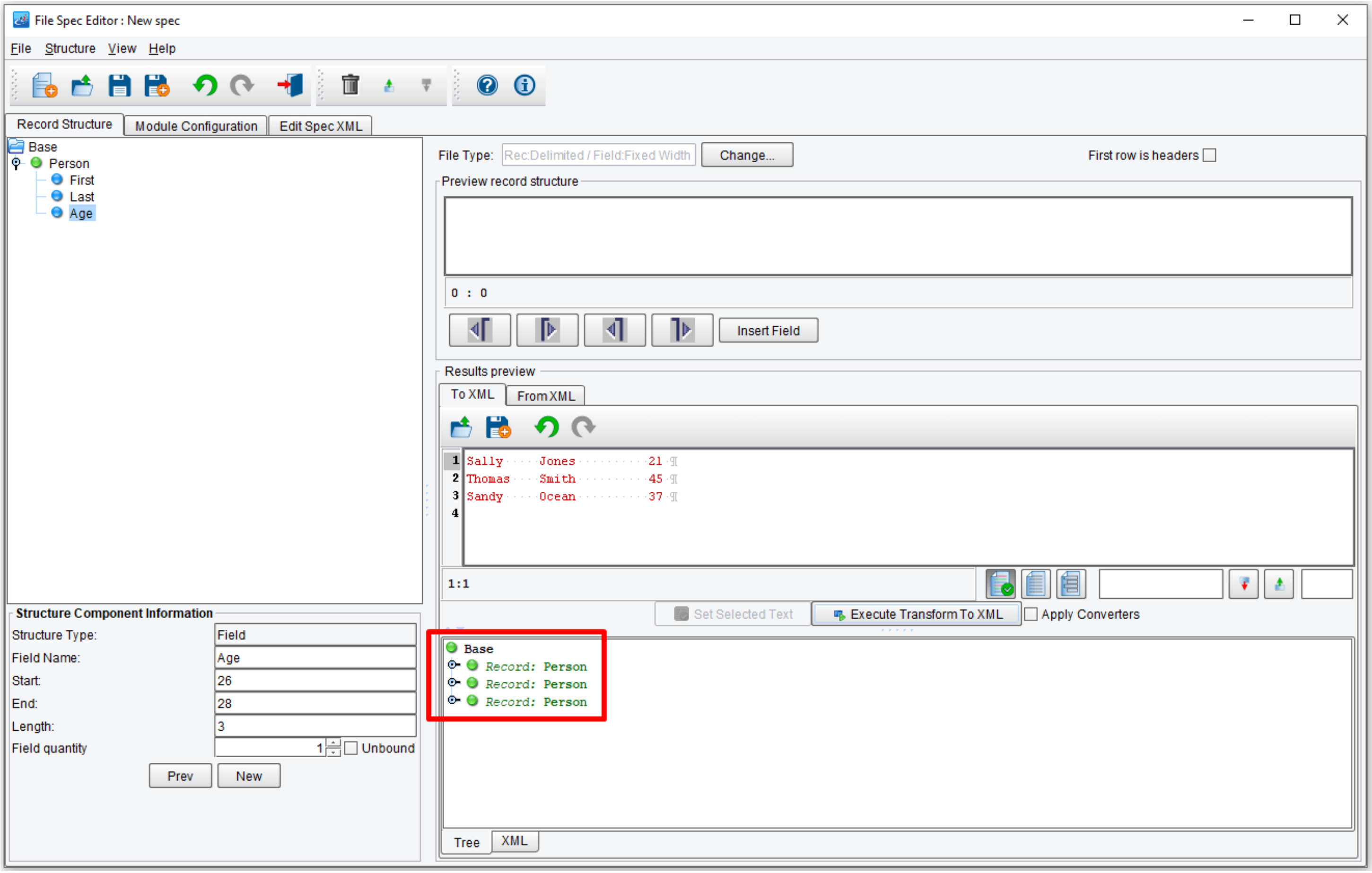
□ Apply Converters

```
No hierarchy loaded
```

Structure Component Information

Structure Type:	Field
Field Name:	Age
Start:	26
End:	28
Length:	3
Field quantity	1 □ Unbound

Finally, to test the XML transformation, click on the **Execute Transform to XML** button. The parsed representation of the data will be displayed.



- Base
  - Person
    - First
    - Last
    - Age

0 : 0

```
1 Sally    Jones        21
2 Thomas   Smith        45
3 Sandy    Ocean        37
4
```

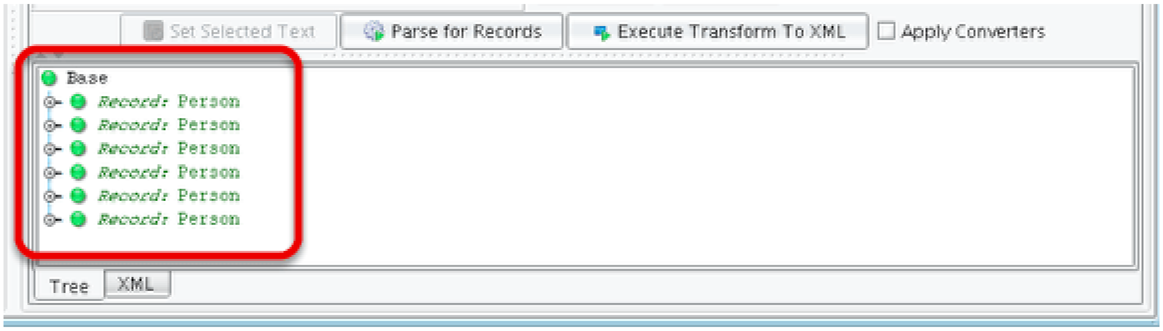
1:1

Structure Component Information

Structure Type:	Field
Field Name:	Age
Start:	26
End:	28
Length:	3
Field quantity	1 □ Unbound

```
Base
  Record: Person
  Record: Person
  Record: Person
```

Displayed are all 3 person records from our sample file. If it doesn't look like this, still go ahead and move on to the next step.

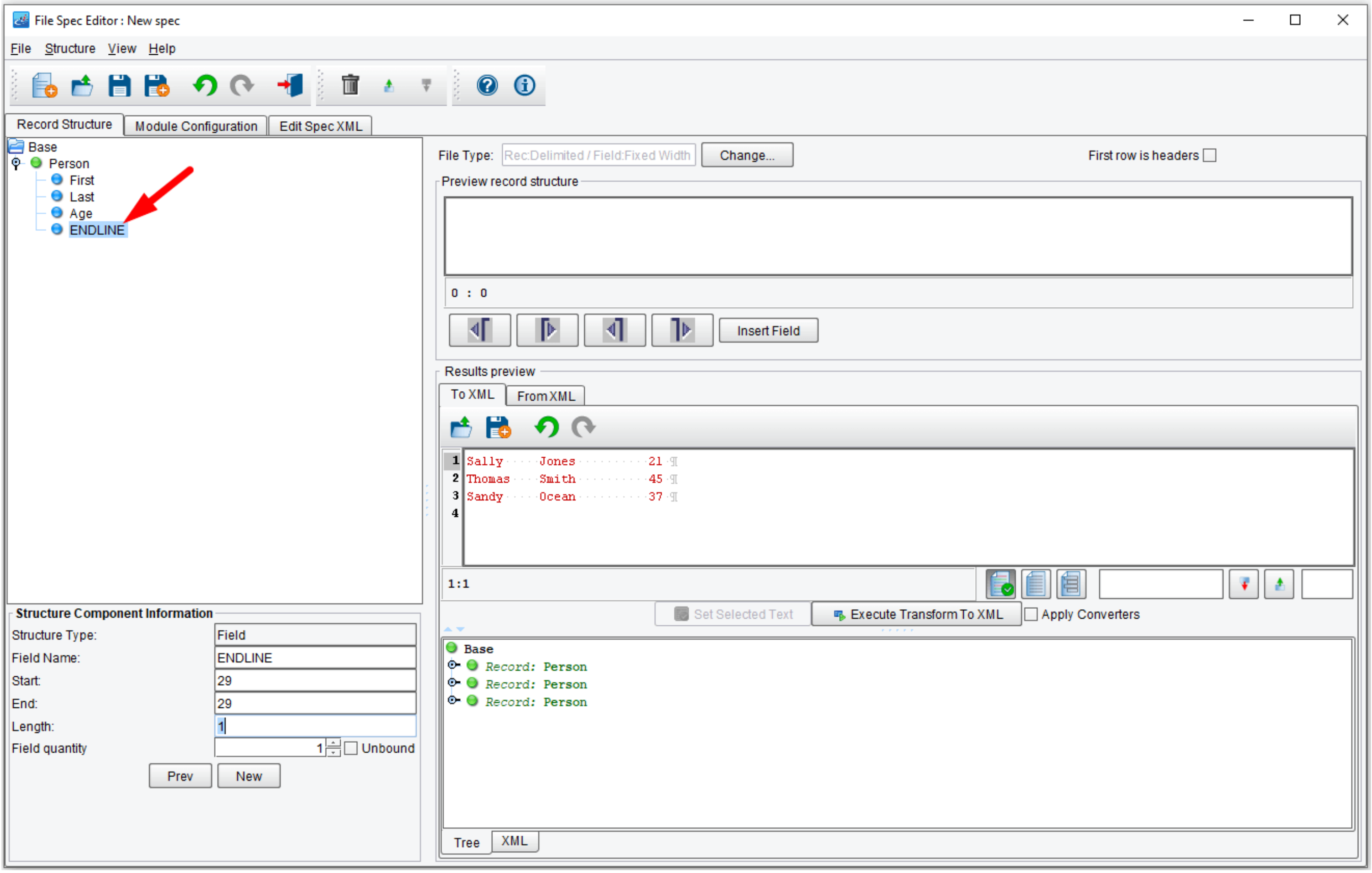


```
Base
  Record: Person
  Record: Person
  Record: Person
  Record: Person
  Record: Person
  Record: Person
```

If your output looks like this instead, you'll need to add one more step.

NOTE: Some filesystems use a 2-character endline sequence (carriage return + linefeed), where others use a single character. For this reason, you may be required to add an extra field to your record definition to consume the additional character. If empty records appear in your output, add an additional record called "ENDLINE" with a length of 1 byte. This should stop the empty records from appearing in your output.

Repeat the previous processes by selecting the green Person node. Right-click and select **Add New Field**, type in **"ENDLINE"**, click **OK**. Leave the Length at "1". Click enter/return then click **Execute Transform to XML**.



- Base
  - Person
    - First
    - Last
    - Age
    - ENDLINE

File Type: Rec:Delimited / Field:Fixed Width

First row is headers □

0 : 0

```
1 Sally     Jones          21
2 Thomas    Smith          45
3 Sandy     Ocean          37
4
```

1:1

□ Apply Converters

- Base
  - Record: Person
  - Record: Person
  - Record: Person

Structure Type: Field

Field Name: ENDLINE

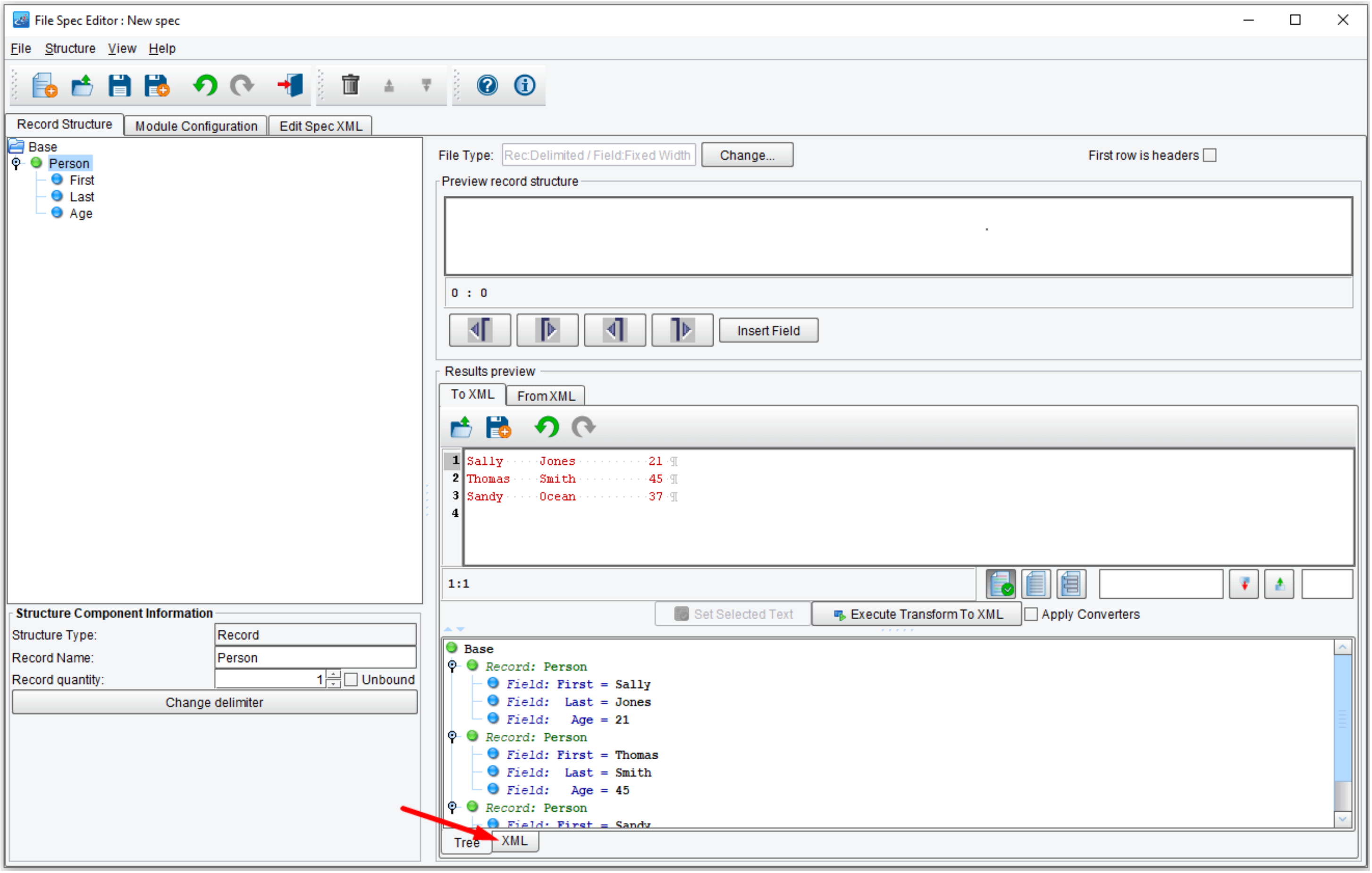
Start: 29

End: 29

Length: 1

Field quantity 1 □ Unbound

Your output should now look like that above. You can now proceed to the next step.



File Type: Rec:Delimited / Field:Fixed Width

First row is headers □

Structure Type: Record

Record Name: Person

Record quantity: 1 □ Unbound

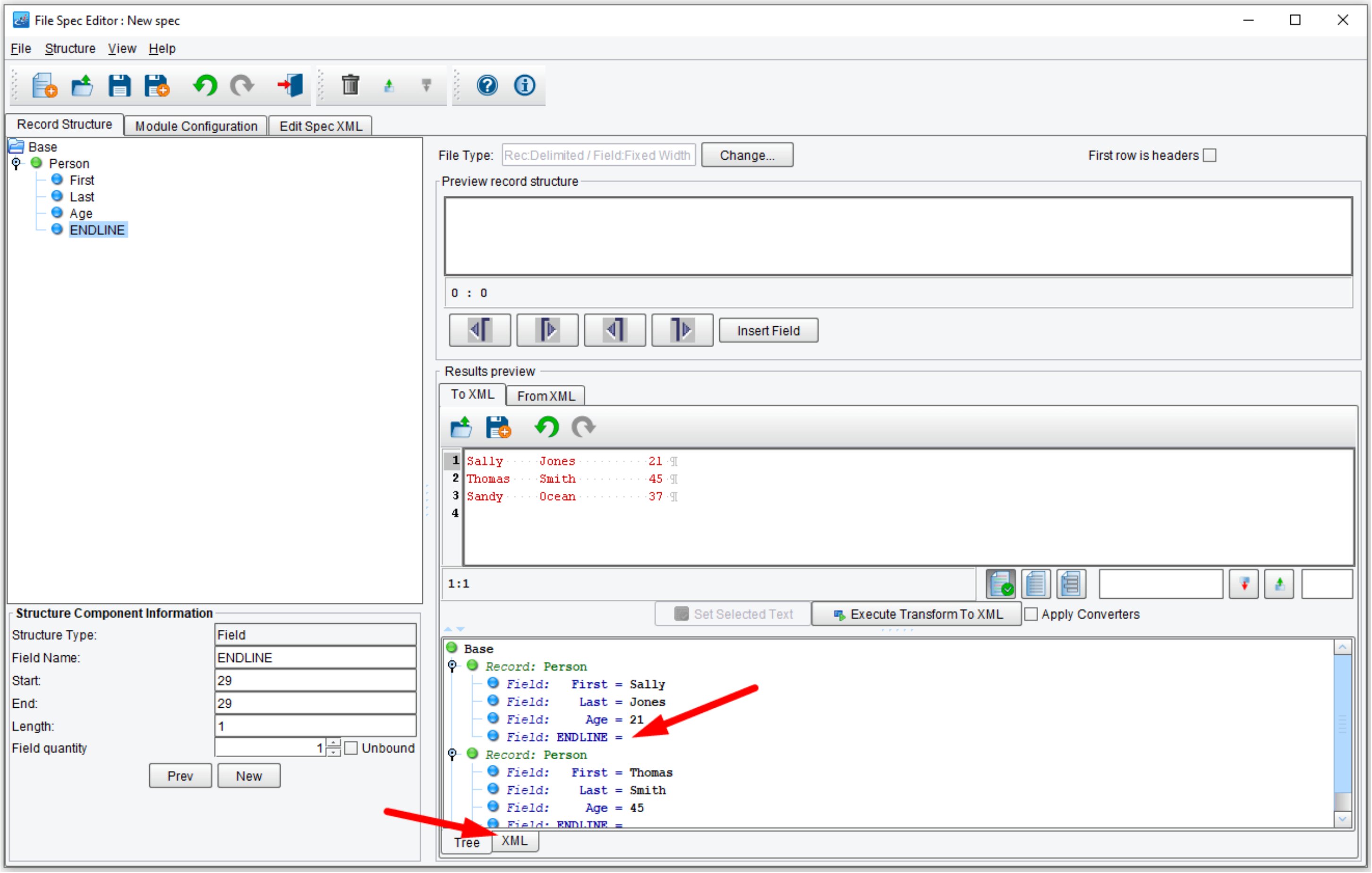
```
0 : 0
```

```
1 Sally    Jones          21 ¶
2 Thomas   Smith          45 ¶
3 Sandy    Ocean          37 ¶
4
```

```
1:1
```

```
Base
  Record: Person
    Field: First = Sally
    Field:  Last = Jones
    Field:   Age = 21
  Record: Person
    Field: First = Thomas
    Field:  Last = Smith
    Field:   Age = 45
  Record: Person
    Field: First = Sandy
```

Click the nodes. You see that each is correctly parsed, containing a first, last, and age field. To view the XML representation, click on the **XML** tab.



- Base
  - Person
    - First
    - Last
    - Age
    - ENDLINE

File Type: Rec:Delimited / Field:Fixed Width

Structure Component Information

Structure Type:	Field
Field Name:	ENDLINE
Start:	29
End:	29
Length:	1
Field quantity	1 □ Unbound

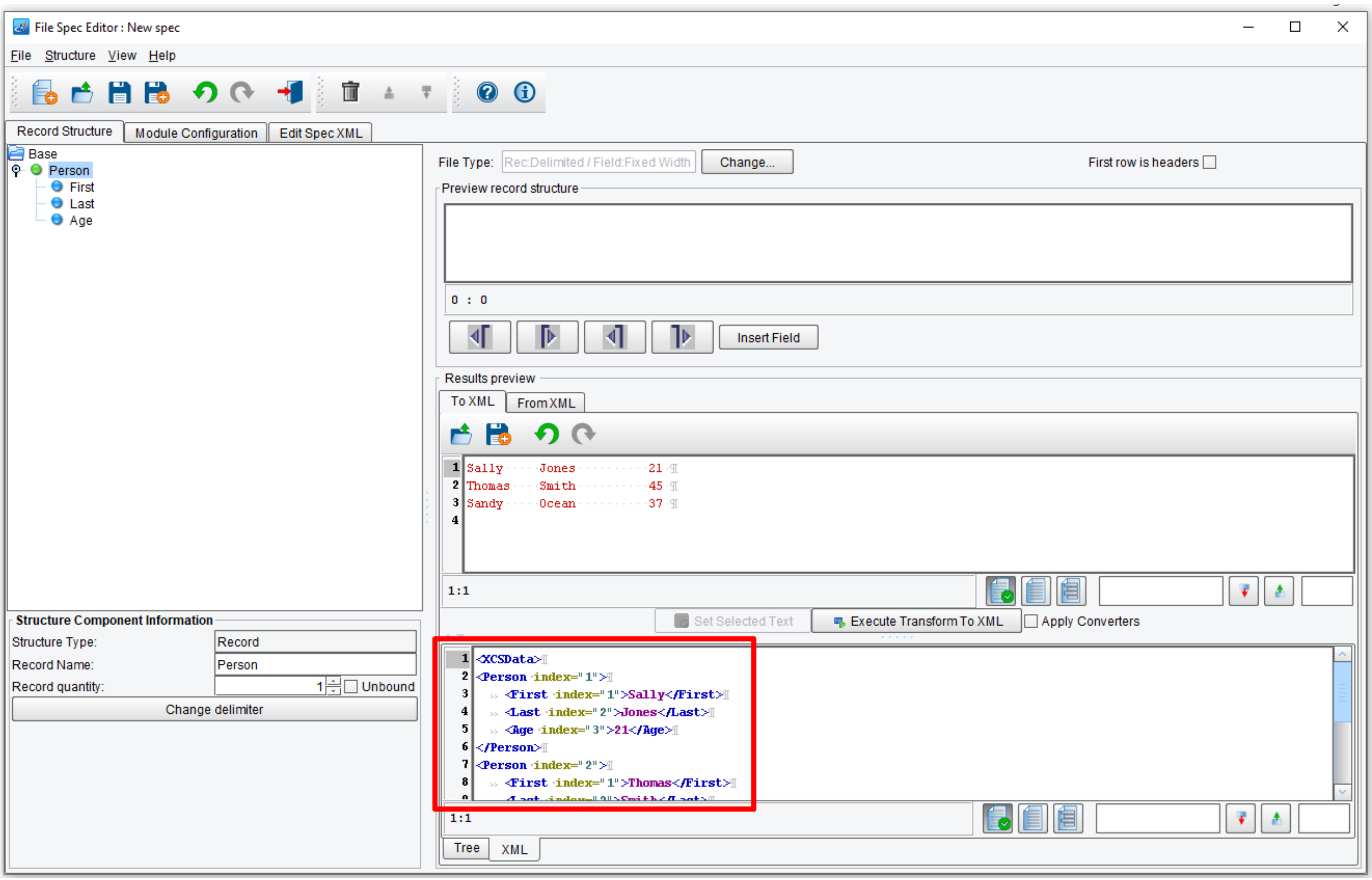
```
0 : 0
```

```
Sally    Jones         21 ¶
Thomas   Smith         45 ¶
Sandy    Ocean         37 ¶

1:1
```

```
Base
  Record: Person
    Field:   First = Sally
    Field:    Last = Jones
    Field:     Age = 21
    Field: ENDLINE =
  Record: Person
    Field:   First = Thomas
    Field:    Last = Smith
    Field:     Age = 45
    Field: ENDLINE =
```

For those who added "ENDLINE", clicking the nodes will reveal the output above. To view the XML representation we can click on the XML tab.



File Type: Rec:Delimited / Field:Fixed Width

Structure Type: Record

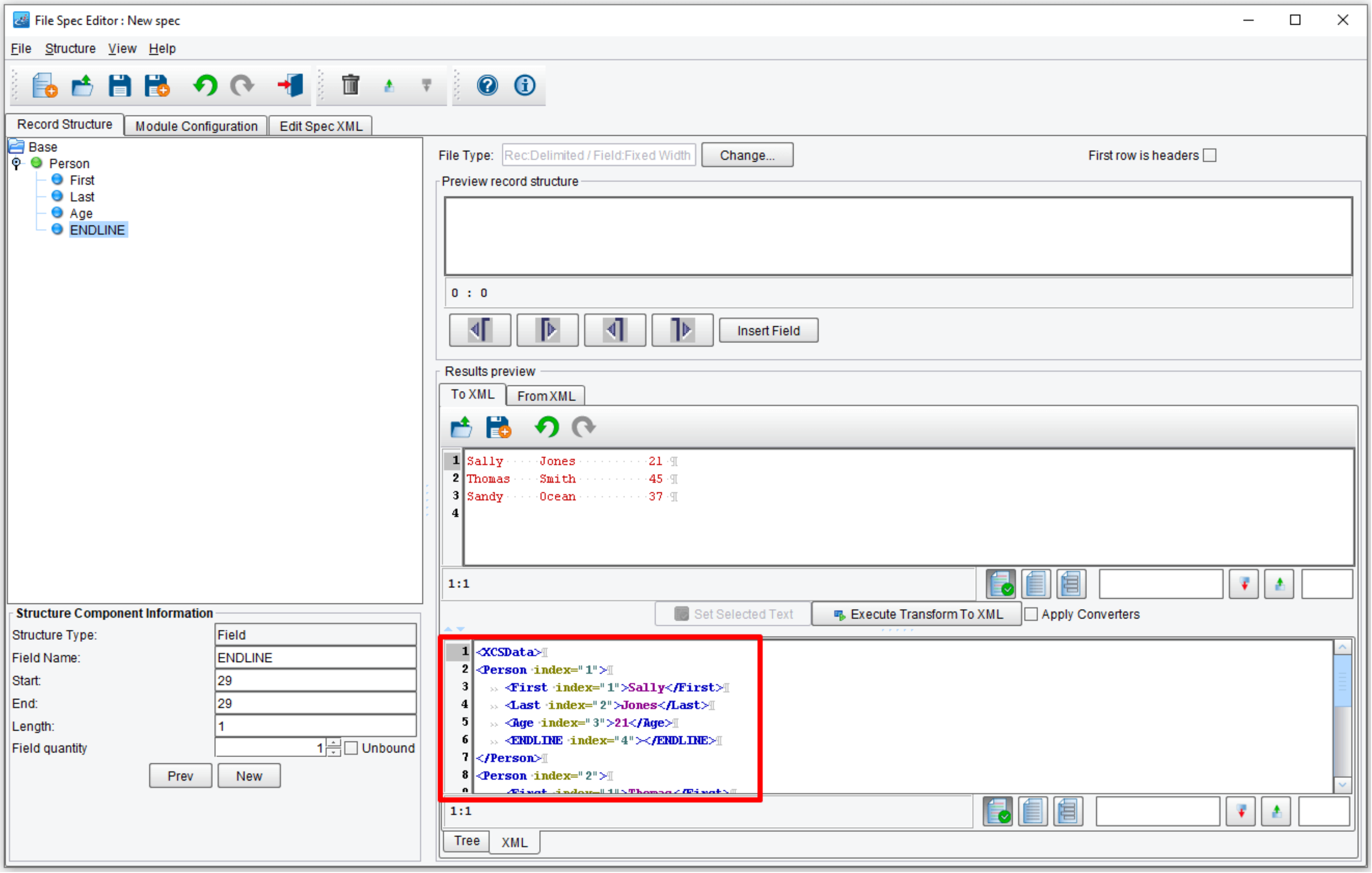
Record Name: Person

Record quantity: 1 □ Unbound

```
Sally    Jones         21
Thomas   Smith         45
Sandy    Ocean         37
```

```
<XCSData>
<Person index="1">
    <First index="1">Sally</First>
    <Last index="2">Jones</Last>
    <Age index="3">21</Age>
</Person>
<Person index="2">
    <First index="1">Thomas</First>
    <Last index="2">Smith</Last>
```

Use the scroll bar to view the XML.



File Type: Rec:Delimited / Field:Fixed Width

First row is headers □

Structure Component Information

Structure Type:	Field
Field Name:	ENDLINE
Start:	29
End:	29
Length:	1
Field quantity	1 □ Unbound

```
Sally    Jones          21
Thomas   Smith          45
Sandy    Ocean          37
```

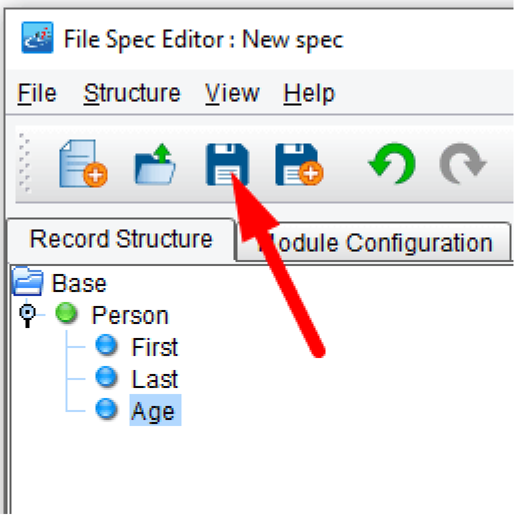
1:1

□ Apply Converters

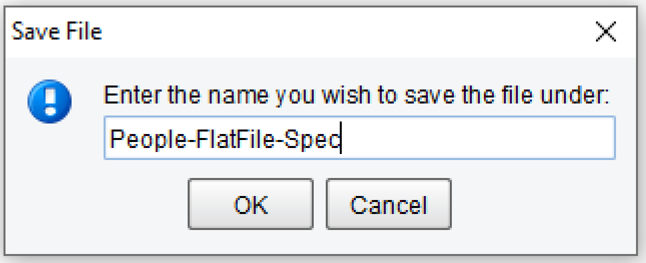
```
<XCSData>
<Person index="1">
    <First index="1">Sally</First>
    <Last index="2">Jones</Last>
    <Age index="3">21</Age>
    <ENDLINE index="4"></ENDLINE>
</Person>
<Person index="2">
    <First index="1">Thomas</First>
```

1:1

And for those with the added field "ENDLINE", your output will look like that above.



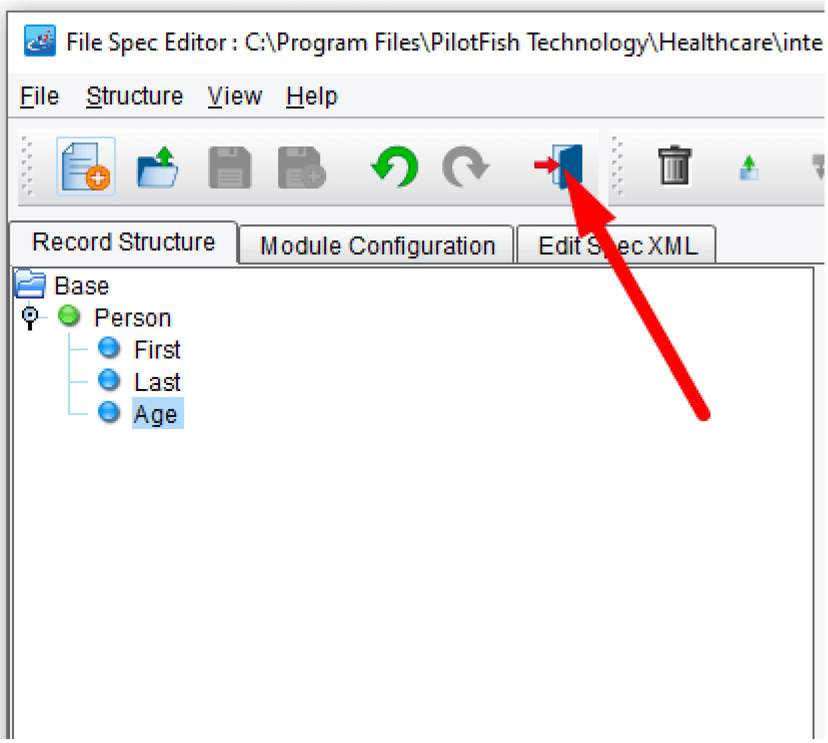
Our file specification is complete. Save it by clicking the **Save** icon.



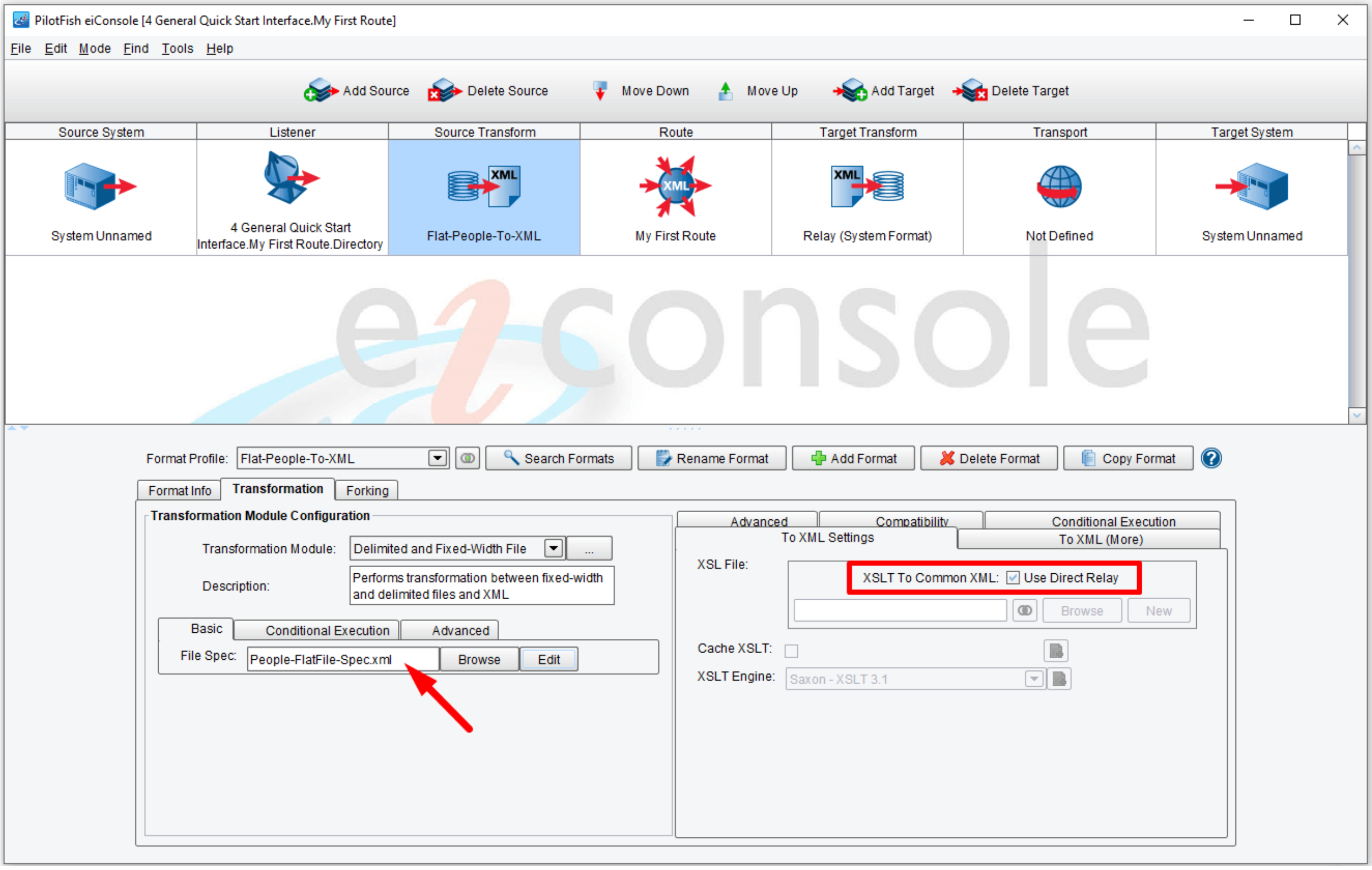
Enter the name you wish to save the file under:

People-FlatFile-Spec

The Save File dialogue will appear. You'll be prompted to enter a name. Type **"People-FlatFile-Spec"** and click **Ok**.



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



Format Profile: Flat-People-To-XML

Transformation Module: Delimited and Fixed-Width File

Description: Performs transformation between fixed-width and delimited files and XML

File Spec: People-FlatFile-Spec.xml

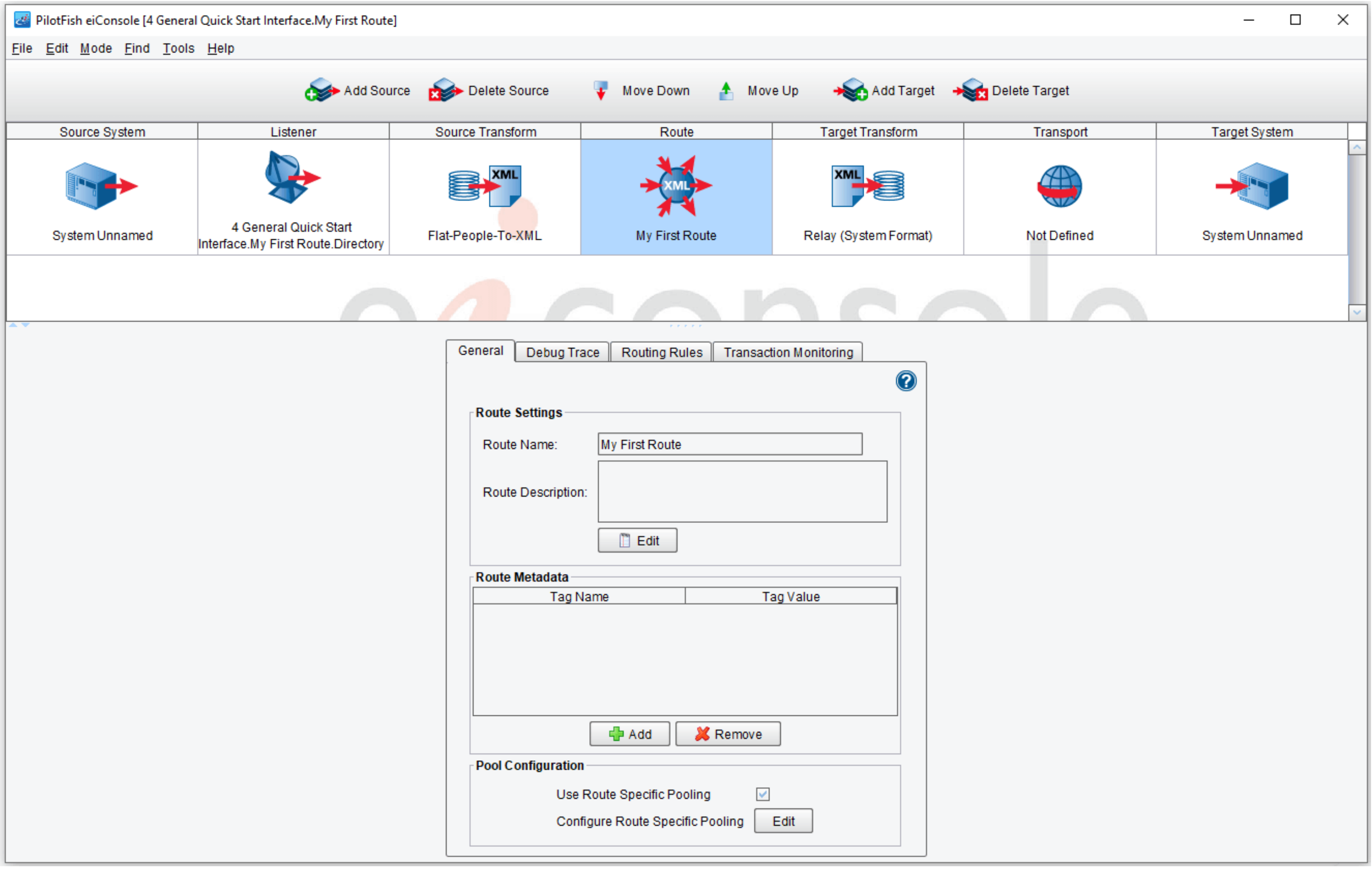
XSLT To Common XML: ☑ Use Direct Relay

Cache XSLT: □

XSLT Engine: Saxon - XSLT 3.1

Note that the new file specification now appears in the Transformation Module Configuration area. Since there won't be any additional transformation, leave the XSL File set to **Use Direct Relay**, checked.

The Route Stage



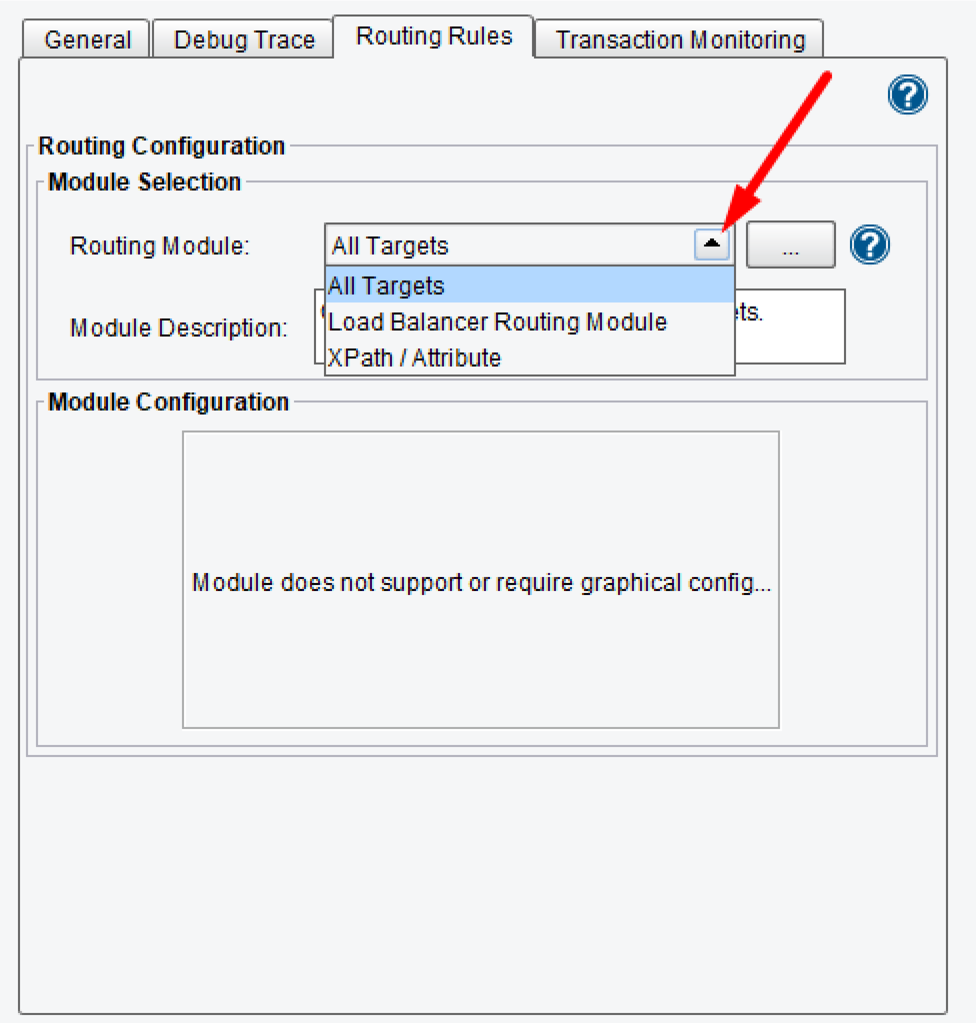
Route Name: My First Route

Route Description:

Use Route Specific Pooling ☑

Configure Route Specific Pooling

Click on the **Route** stage. The General tab will open with the configuration settings. Leave these settings as is.

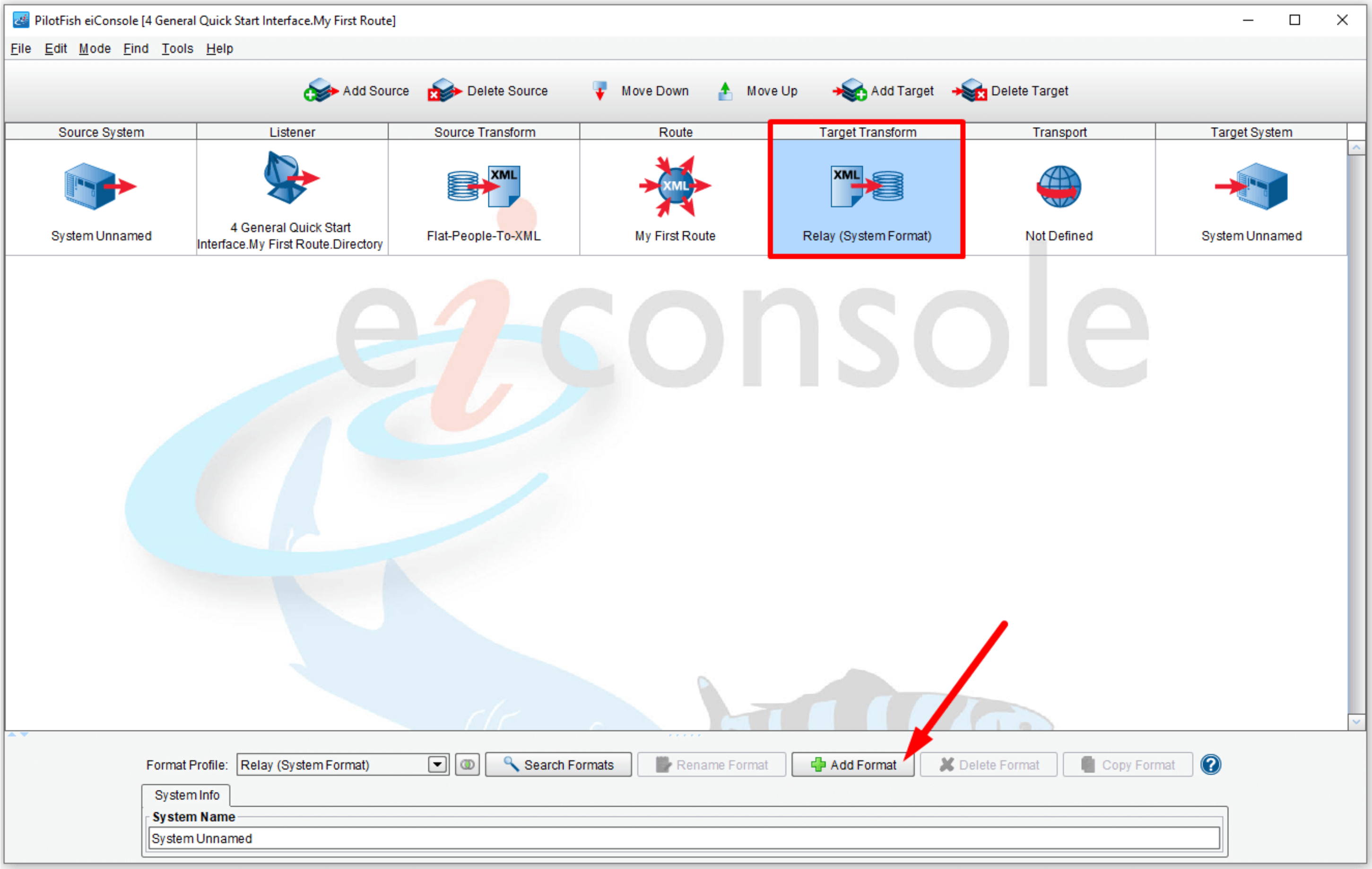


In the Route stage, we can optionally implement Routing Rules.

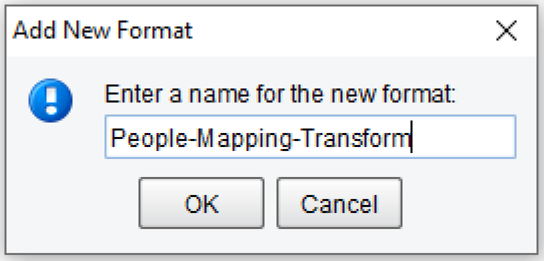
Select the Routing Rules tab. Note your options in the drop-down. In our case, you are sending all data from the Source to the Target System. So, the default Routing Module selection of **All Targets** applies. Leave these settings as is.

Next, click on the **Target Transform** stage.

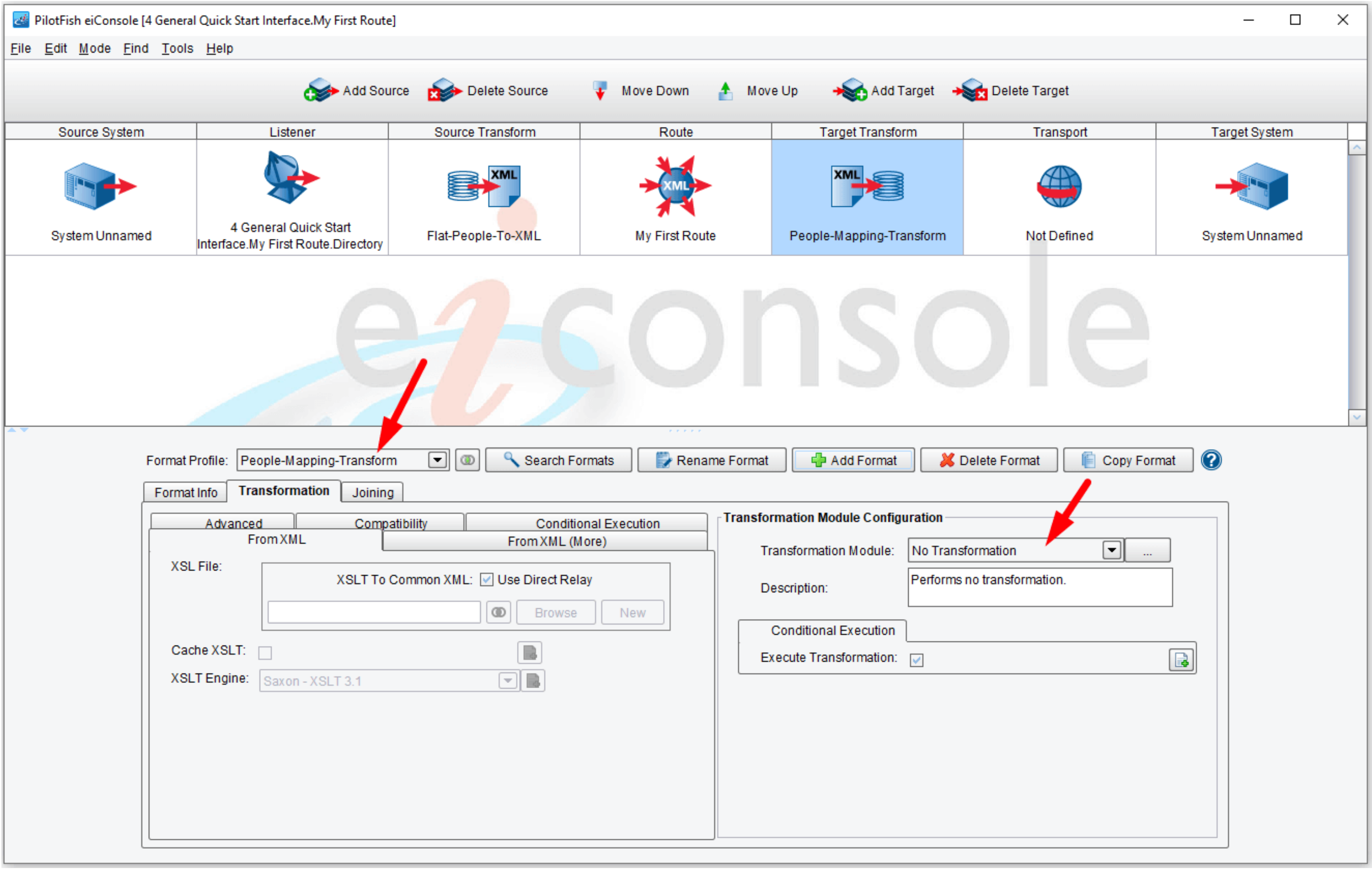
The Target Transform Stage



To add another format, click the **Add Format** button.



Once the Add New Format dialog appears type in **"People-Mapping-Transform"** and click **OK**.


PilotFish eiConsole [4 General Quick Start Interface.My First Route]

Format Profile: People-Mapping-Transform

XSL File: XSLT To Common XML: ☑ Use Direct Relay

Cache XSLT: □

XSLT Engine: Saxon - XSLT 3.1

Transformation Module Configuration

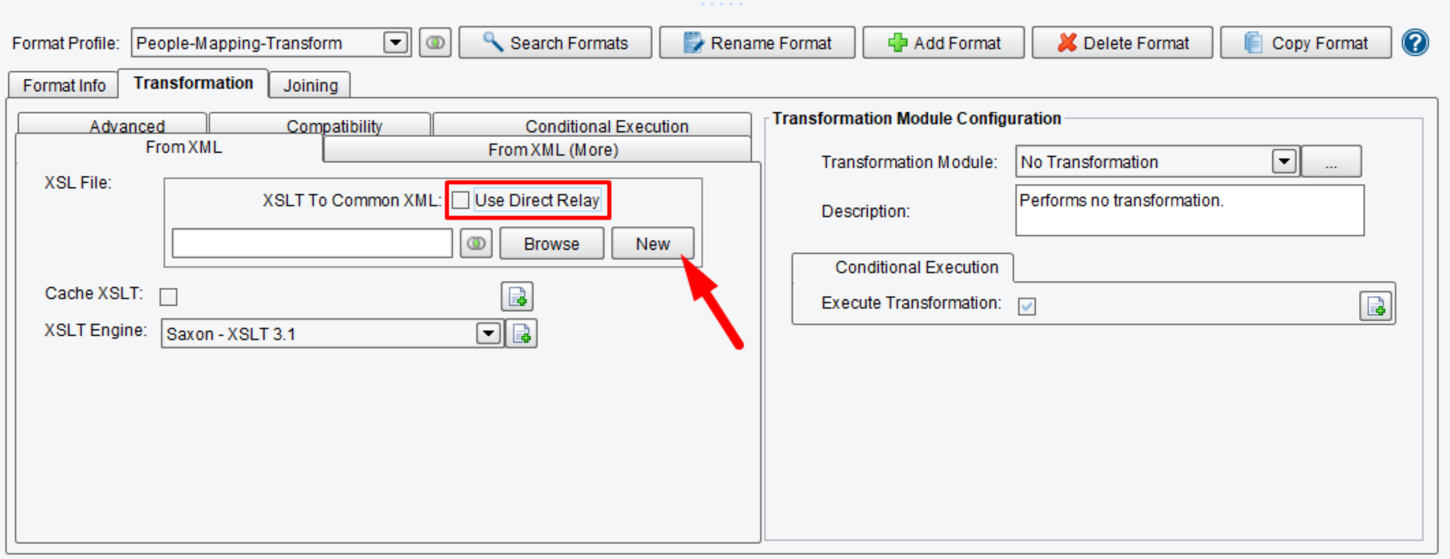
Transformation Module: No Transformation

Description: Performs no transformation.

Execute Transformation: ☑

People-Mapping-Transform will be selected in the Format Profile drop-down. Leave that as is.

Next, configure the XSL File and leave the Transformation Module Configuration set at **No Transform**.



Format Profile: People-Mapping-Transform

XSL File: XSLT To Common XML: □ Use Direct Relay

Cache XSLT: □

XSLT Engine: Saxon - XSLT 3.1

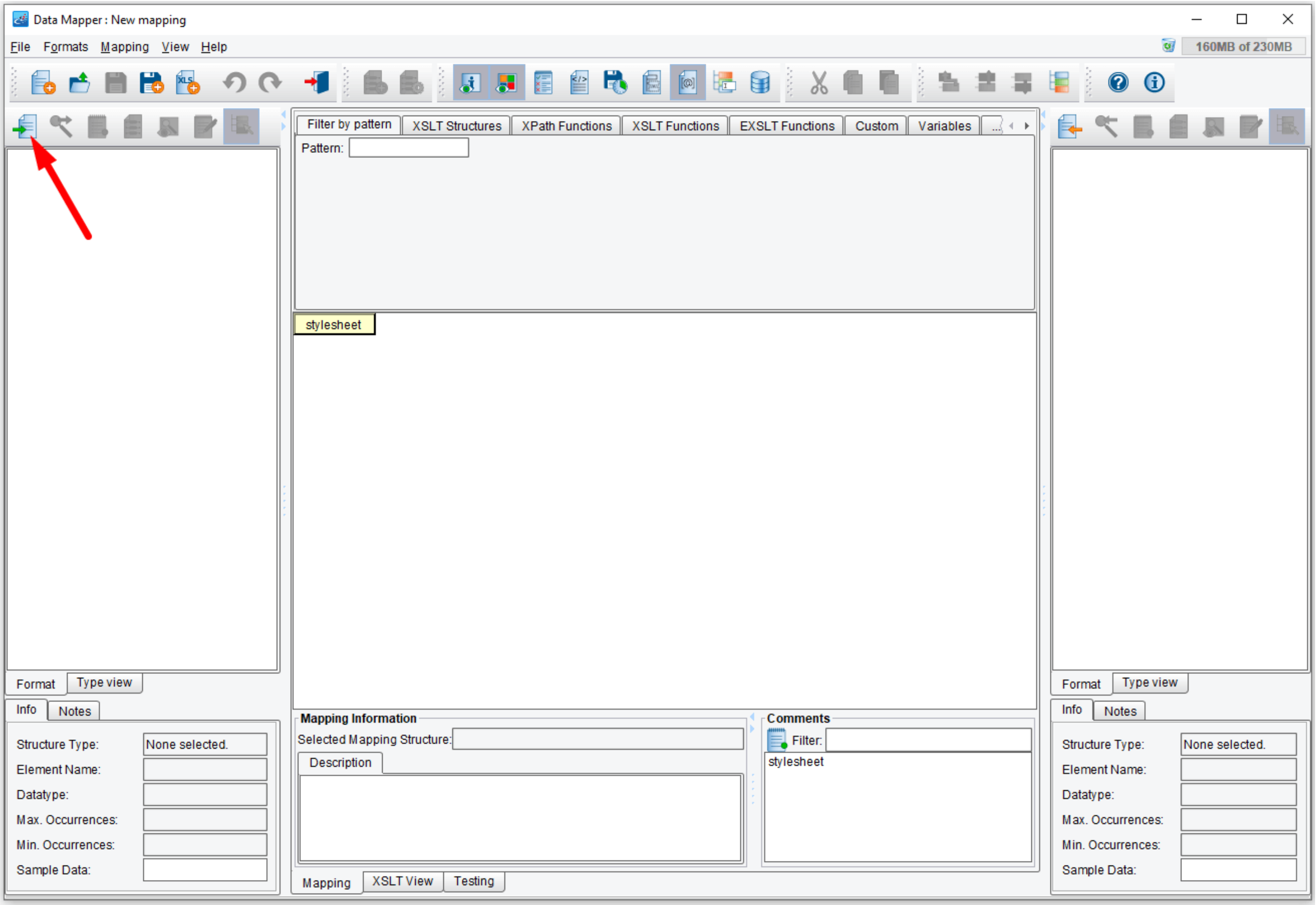
Transformation Module Configuration

Transformation Module: No Transformation

Description: Performs no transformation.

Execute Transformation: ☑

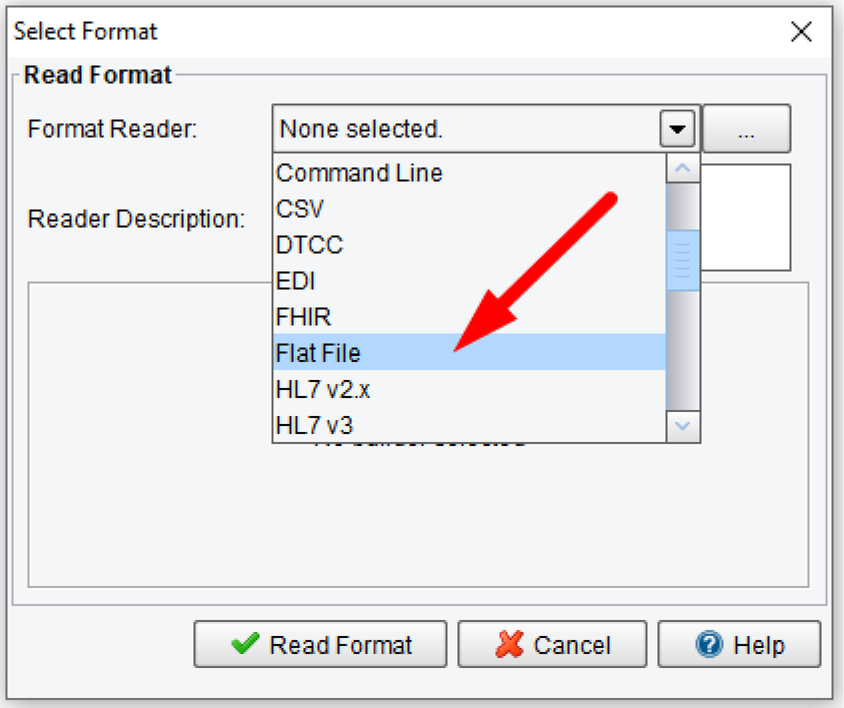
Uncheck the Use Direct Relay box in the XSLT Configuration panel. This will enable the Browse and Edit buttons in configuration items. Click the **New** button to launch the Data Mapper.



Structure Type: None selected.
Element Name:
Datatype:
Max. Occurrences:
Min. Occurrences:
Sample Data:

The Data Mapper opens. The Data Mapper is the eiConsole's 3 pane mapping tool. It generates XSLT transformations and enables you to transform any data format to any other data format. In the Data Mapper, you can load the Source Format, load the TargetFormat, and then graphically map between the two.

Click the **Open Source Format** icon above the Source panel.



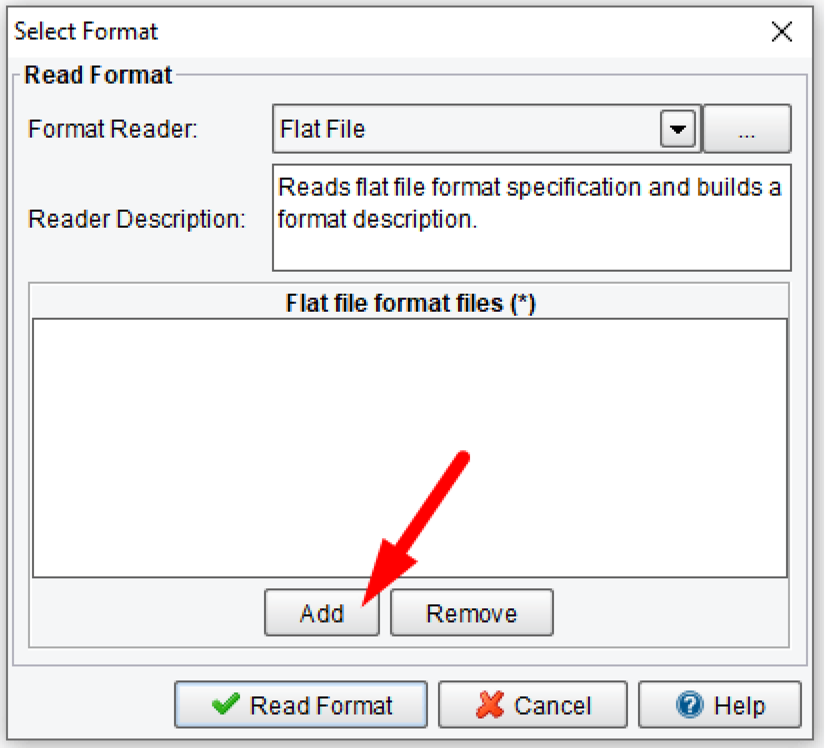
Select Format

Read Format

Format Reader: None selected.

Reader Description:

The Select Format dialog will open. To map from a flat file format, choose the **Flat File** from the drop-down and click **Read Format**.



Select Format

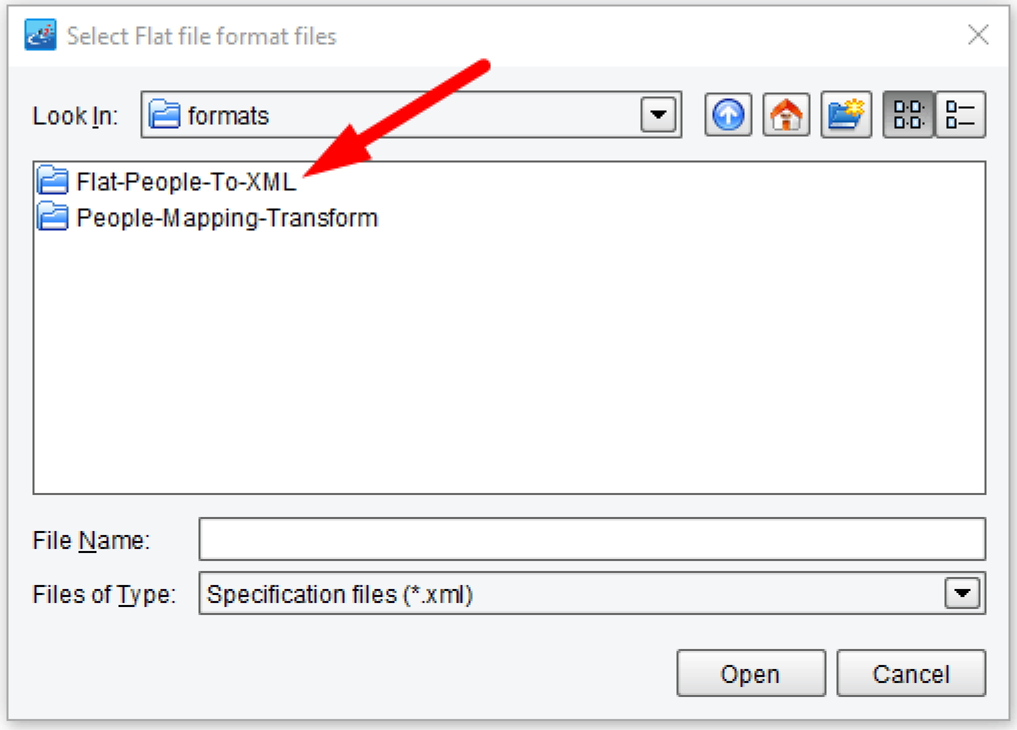
Read Format

Format Reader: Flat File

Reader Description: Reads flat file format specification and builds a format description.

Flat file format files (*)

Then, the **Add** button below the Flat file format files panel.



Select Flat file format files

Look In: formats

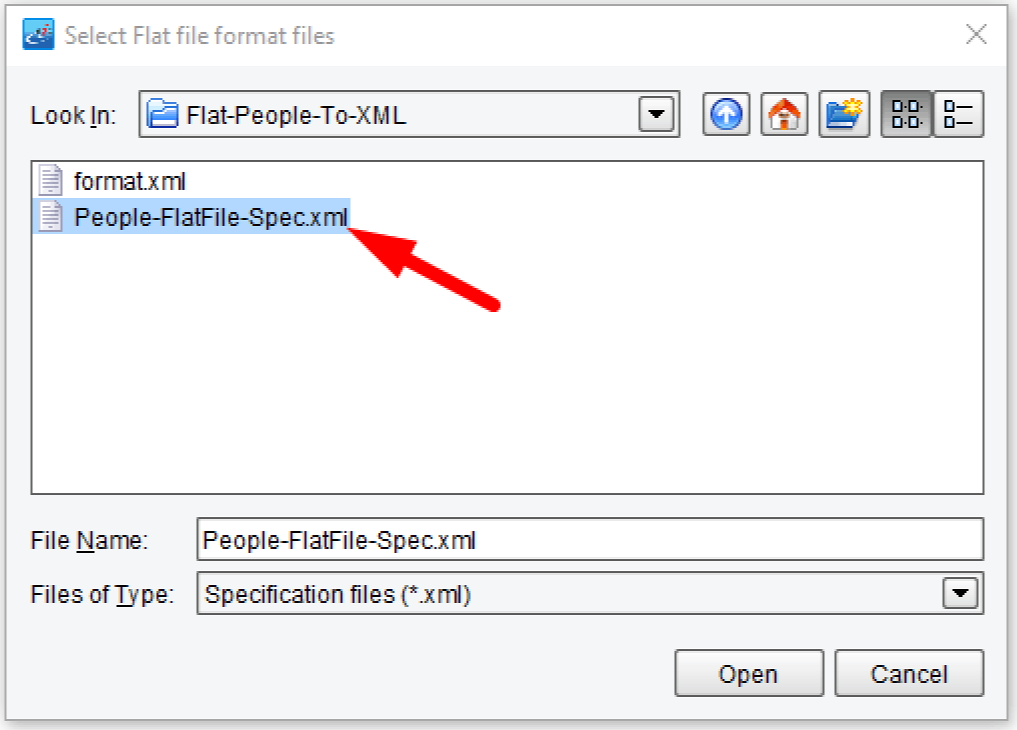
Flat-People-To-XML
People-Mapping-Transform

File Name:

Files of Type: Specification files (*.xml)

When the Select Flat file format files panel opens, navigate to the 4 General Quick Start Interface directory, or your Working Directory, and double click on the **formats** folder to open.

Select the **Flat-People-To-XML** folder and double click to open.



Select Flat file format files

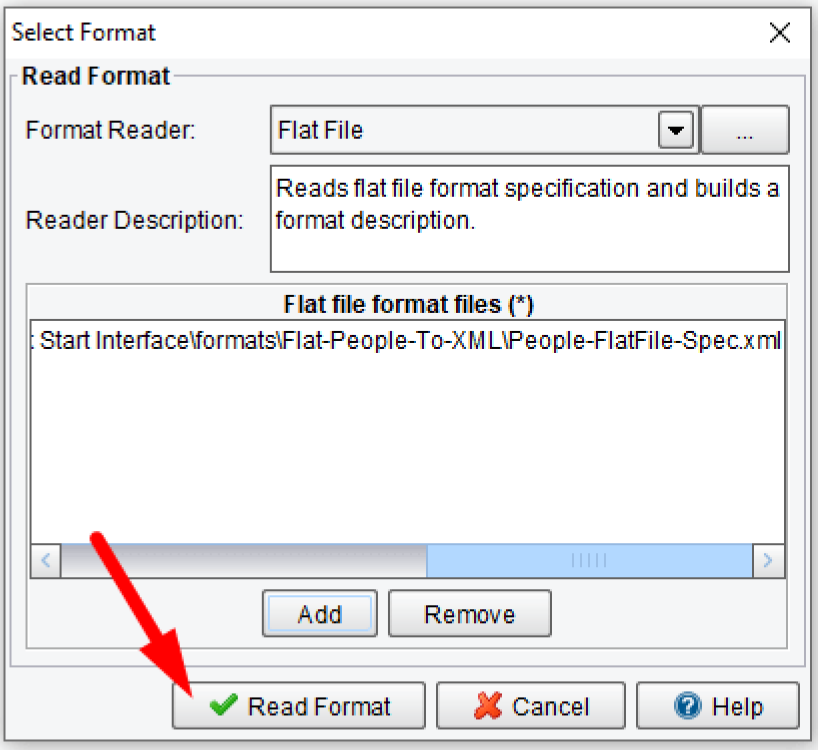
Look In: Flat-People-To-XML

format.xml
People-FlatFile-Spec.xml

File Name: People-FlatFile-Spec.xml

Files of Type: Specification files (*.xml)

Inside that folder, you'll find the file specification that was completed earlier. Select the **People-FlatFile-Spec.xml** file and click **Open**.



Select Format

Read Format

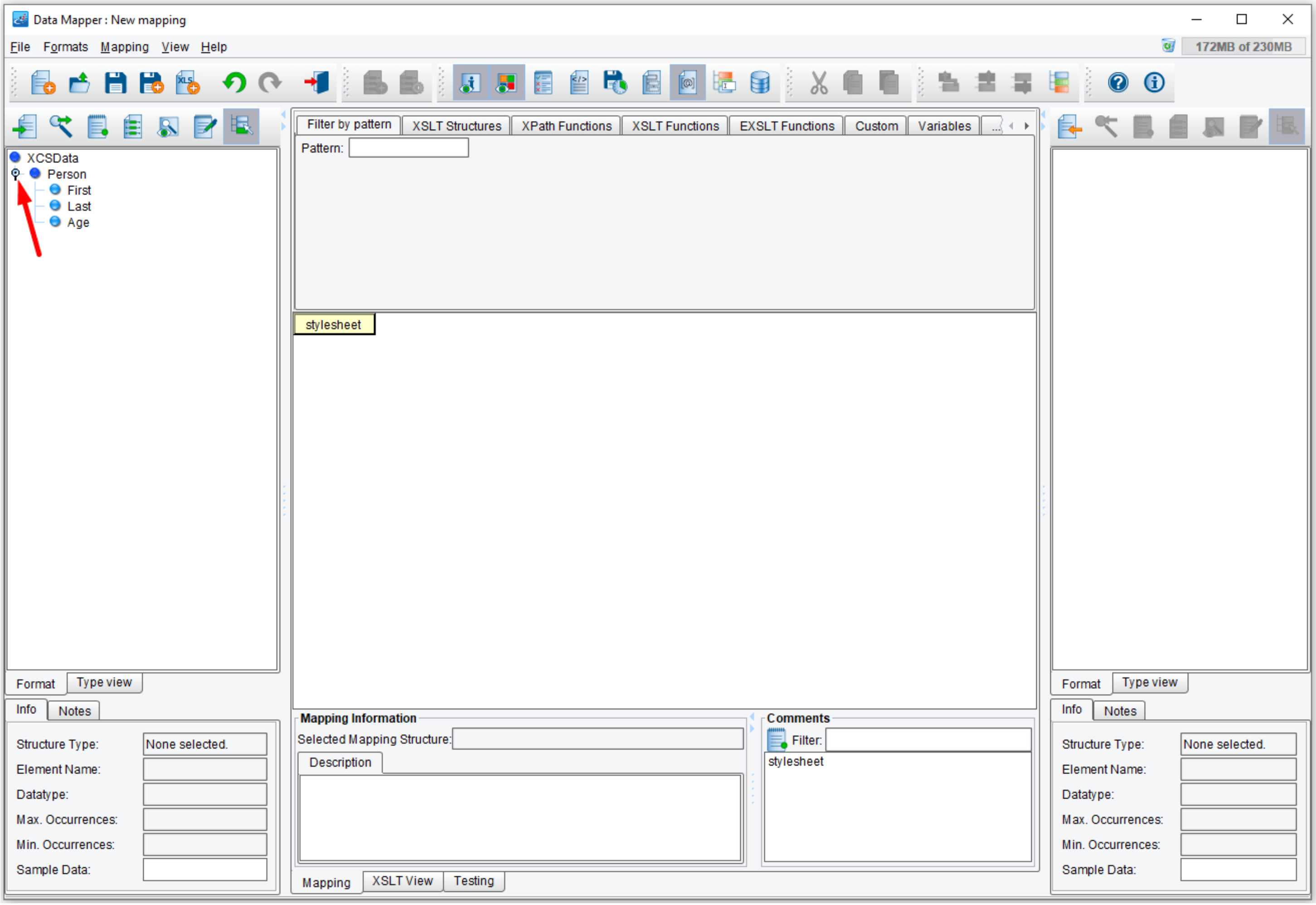
Format Reader: Flat File

Reader Description: Reads flat file format specification and builds a format description.

Flat file format files (*)

: Start Interface\formats\Flat-People-To-XML\People-FlatFile-Spec.xml

When the Select Format window opens click the **Read Format** button.



Data Mapper : New mapping

- XCSData
  - Person
    - First
    - Last
    - Age

Pattern:

stylesheet

Mapping Information

Selected Mapping Structure:

Comments

Filter:

stylesheet

Structure Type: None selected.

Element Name:

Datatype:

Max. Occurrences:

Min. Occurrences:

Sample Data:

Structure Type: None selected.

Element Name:

Datatype:

Max. Occurrences:

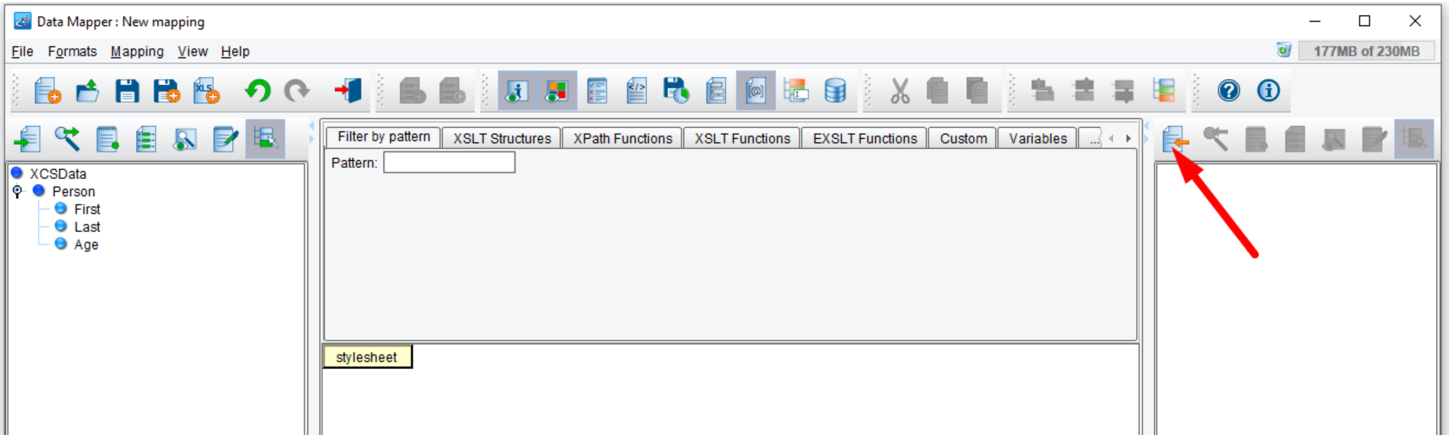
Min. Occurrences:

Sample Data:

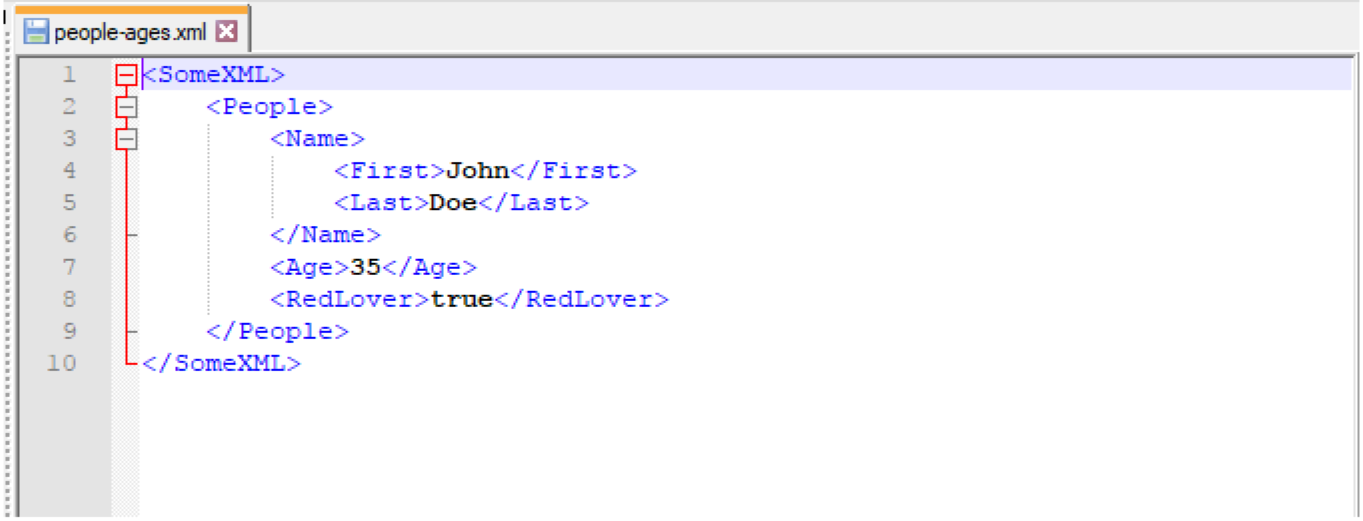
The structure of the flat file will now appear in the tree in the left panel of the Data Mapper. Click the node to expand the tree. On the right-hand side, load the TargetFormat. (Note, there might be the last element "ENDLINE" in the source format. It would only appear if it has been added in the File Specification step)

Note: What is a "node" vs. an element. The nomenclature comes from the XML standard itself, where any object within an XML tree is called a "node." "Elements" are nodes that have attributes (also nodes) and child nodes. Generally, they're pretty interchangeable. A good analogy, you can think of it this way. A "node is like a fruit, an element is like an apple. An apple is a kind of fruit; an element is a kind of node."

Note: A node has a bullet to the left, it's expandable/collapsable and it's technically an element. But everything - attributes, XSLT instructions, tabular mappings, etc. are all "nodes".

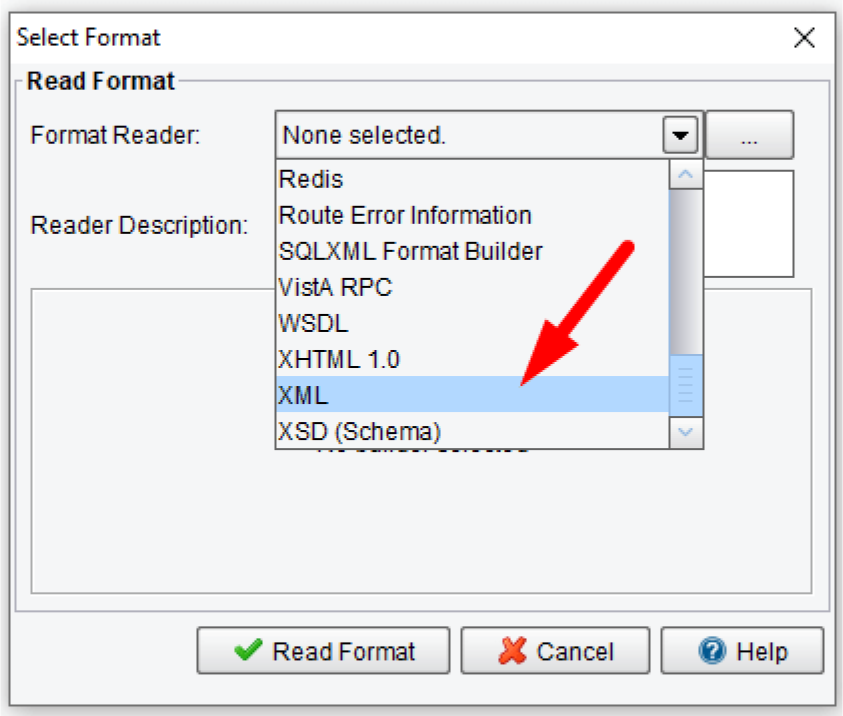


To load the Target Format, click the **Open Target Format** icon.



```
<SomeXML>
    <People>
        <Name>
            <First>John</First>
            <Last>Doe</Last>
        </Name>
        <Age>35</Age>
        <RedLover>true</RedLover>
    </People>
</SomeXML>
```

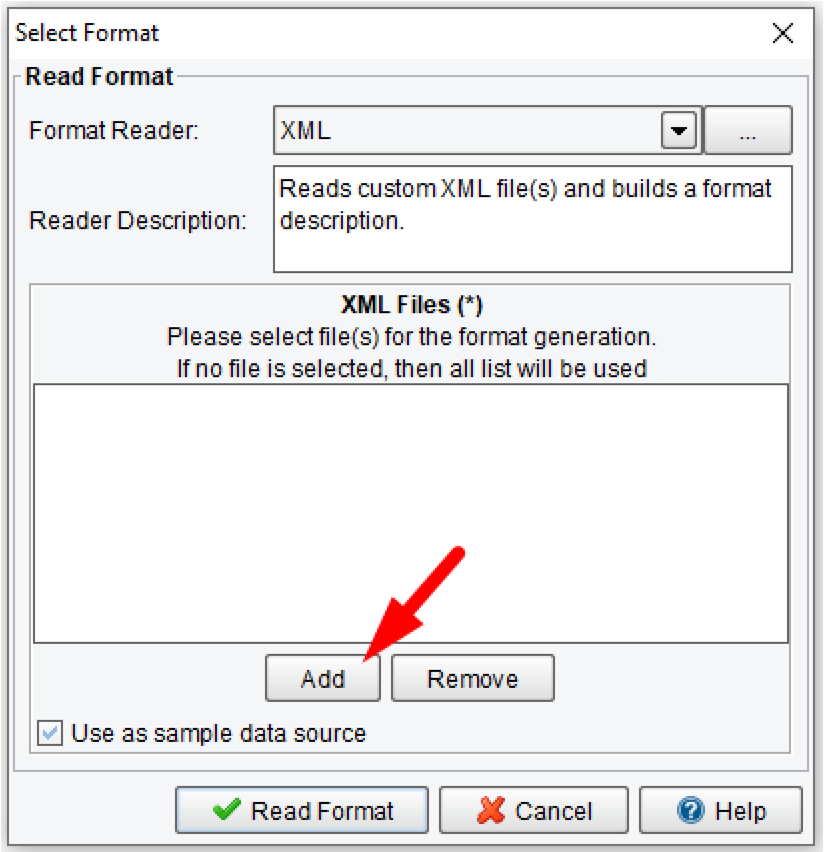
Take a look at the desired Target Format. It is this XML structure shown here. (The **people-ages.xml** file in the distribution data folder.)



Format Reader: None selected.

Reader Description:

Since there is no schema, and only a sample, choose the **XML** option from the drop-down.



Format Reader: XML

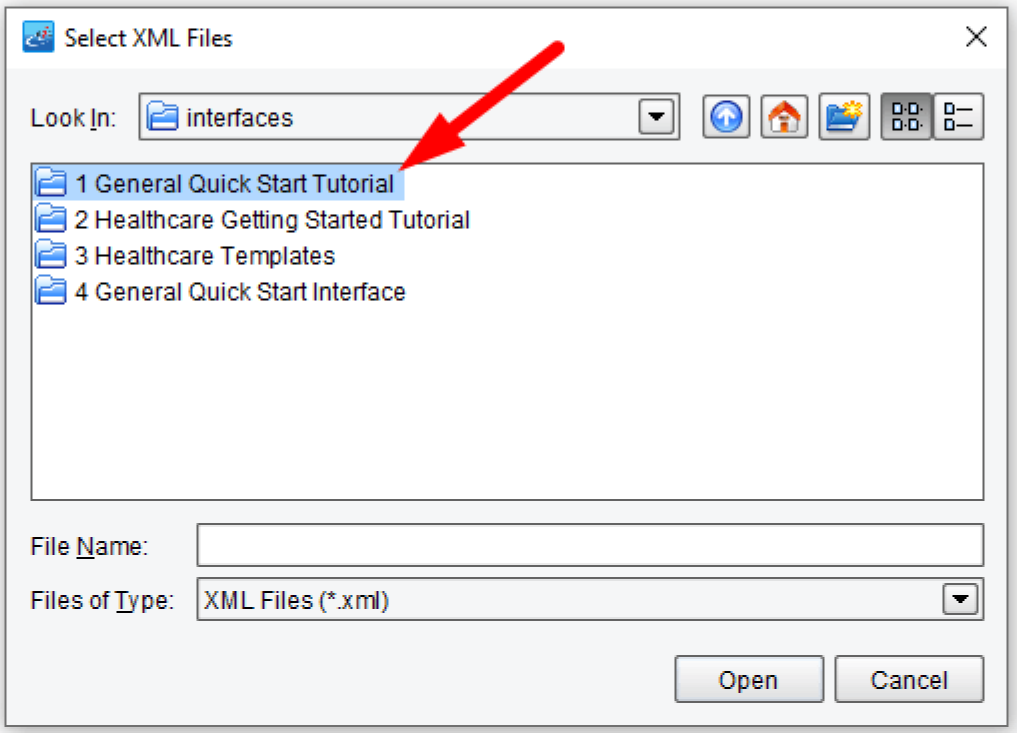
Reader Description: Reads custom XML file(s) and builds a format description.

XML Files (*)

Please select file(s) for the format generation.
If no file is selected, then all list will be used

☑ Use as sample data source

Next, click the **Add** button under the XML Files list.



Select XML Files

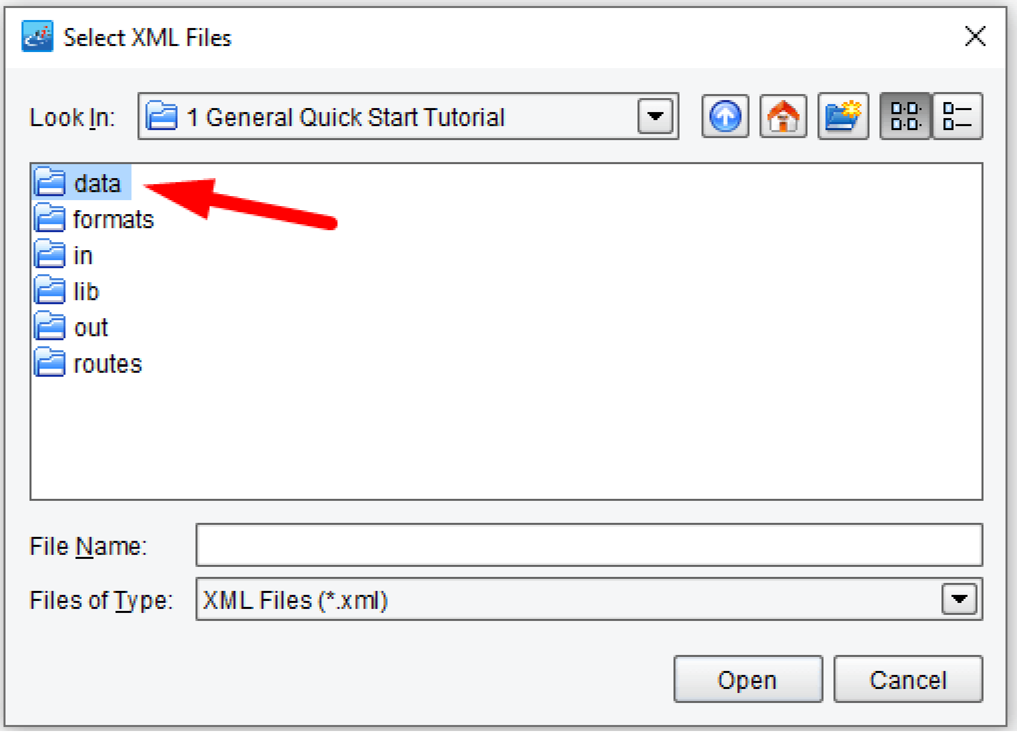
Look In: interfaces

File Name:

Files of Type: XML Files (*.xml)

1 General Quick Start Tutorial
2 Healthcare Getting Started Tutorial
3 Healthcare Templates
4 General Quick Start Interface
Open
Cancel

The **Select XML Files** window will open. Navigate to the distribution folder - **1 General Quick Start Tutorial.** Select and double click to open.



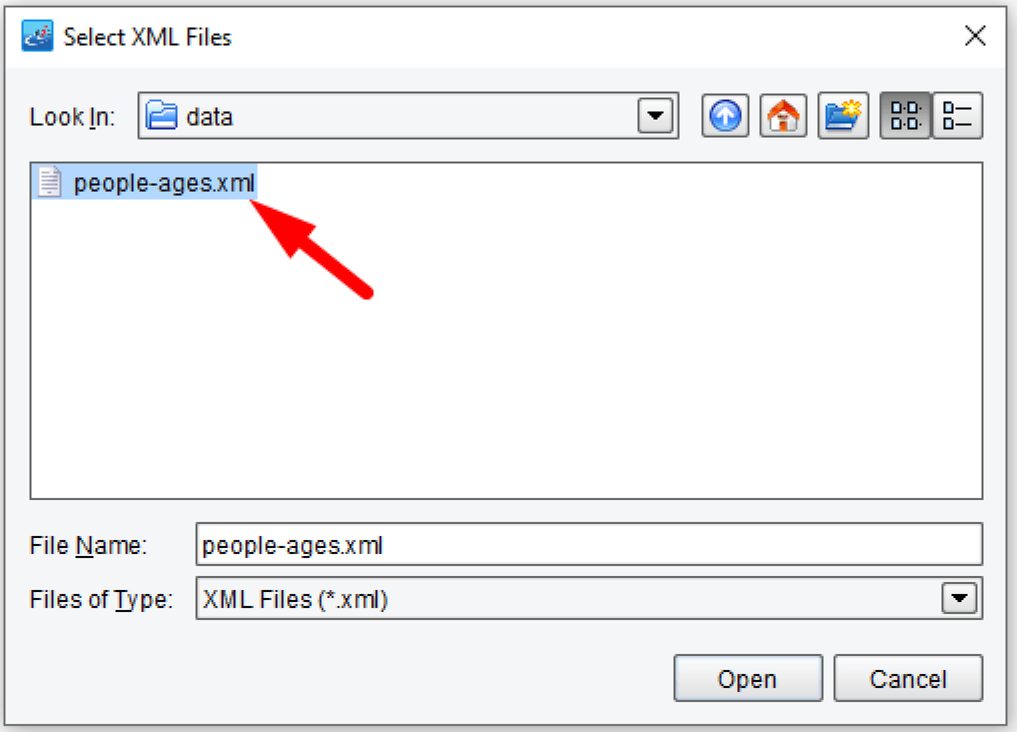
Select XML Files

Look In: 1 General Quick Start Tutorial

File Name:

Files of Type: XML Files (*.xml)

Next, select the **data** folder and double click to open.



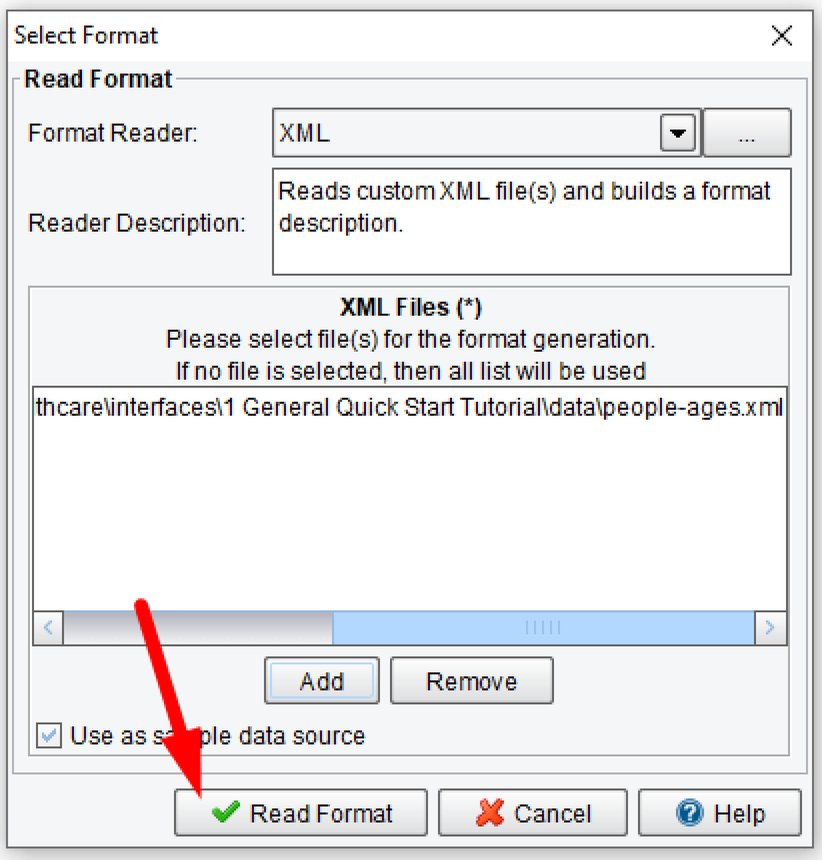
Look In: data

people-ages.xml

File Name: people-ages.xml

Files of Type: XML Files (*.xml)

Next, select the **people-ages.xml** file and click **Open**.



Format Reader: XML

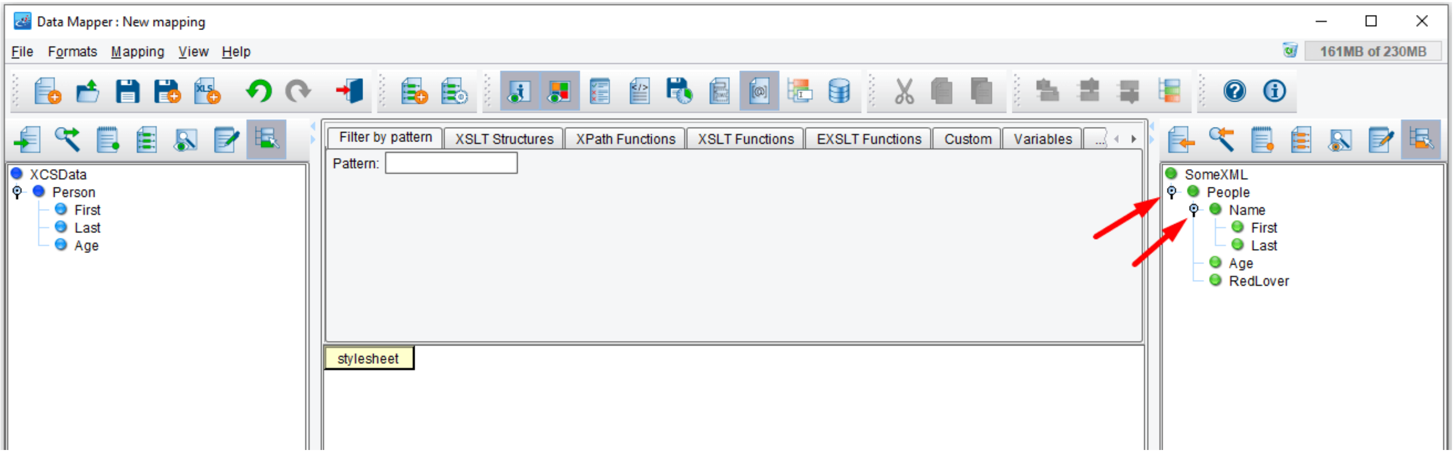
Reader Description: Reads custom XML file(s) and builds a format description.

XML Files (*)
Please select file(s) for the format generation.
If no file is selected, then all list will be used

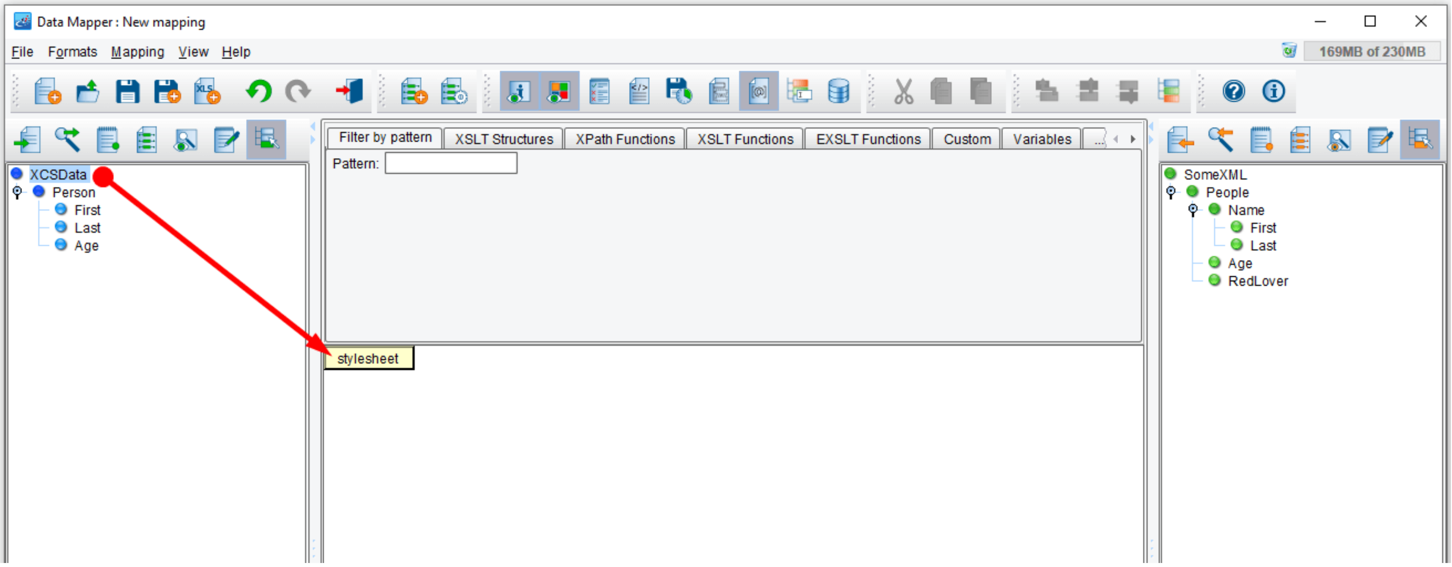
thcare\interfaces\1 General Quick Start Tutorial\data\people-ages.xml

☑ Use as sample data source

Then click **Read Format.**

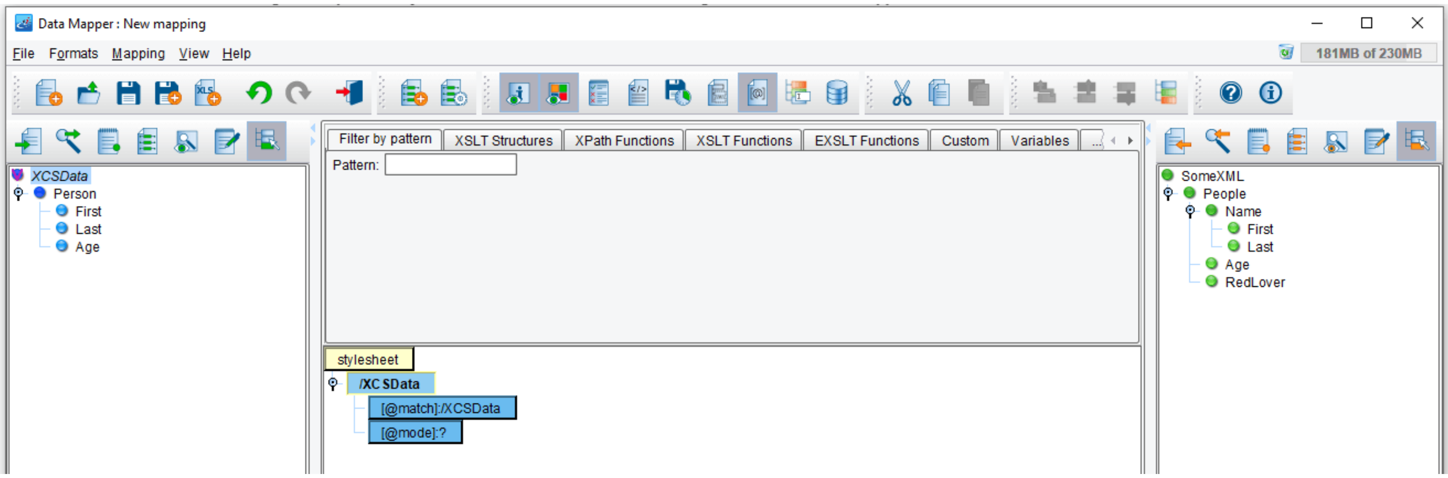


Once the Source and Target structures are set, create a mapping from one to another. Click the node to expand the tree.

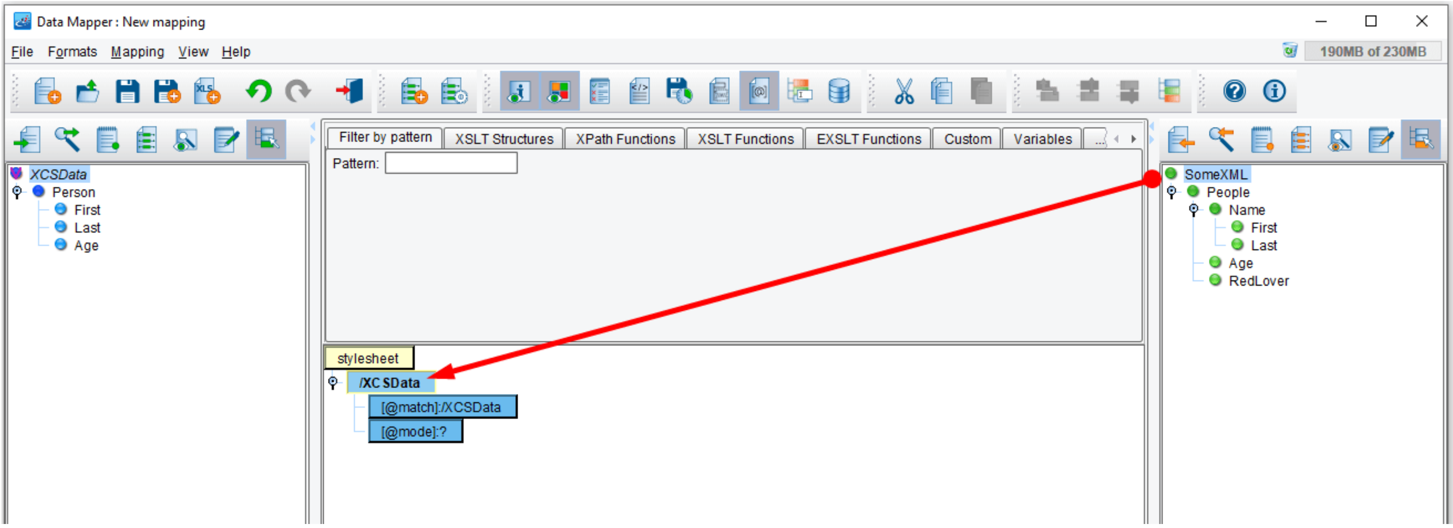


To begin our mapping, drag the root node of the Source Format onto the stylesheet.

TIP: Select the node, click and drag it on top of the style sheet element and release once the green circle with the + appears.

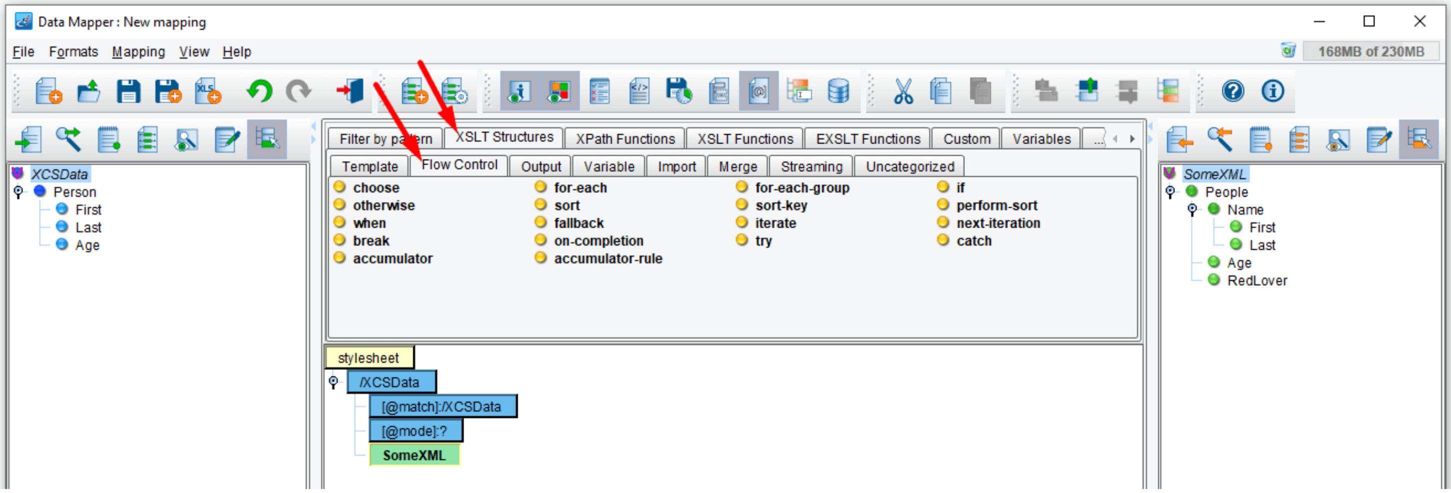


The blue **XCSData** element should appear under **stylesheet**.



Next, drag the root node of the Target format onto your newly created **XCSData** node. This indicates that we'll be creating a SomeXML tag each time we encounter an XCSData tag in our Source.

TIP: When dragging a node onto your mapping, release your mouse once the gray bar appears above the element in your mapping. This indicates the correct position.



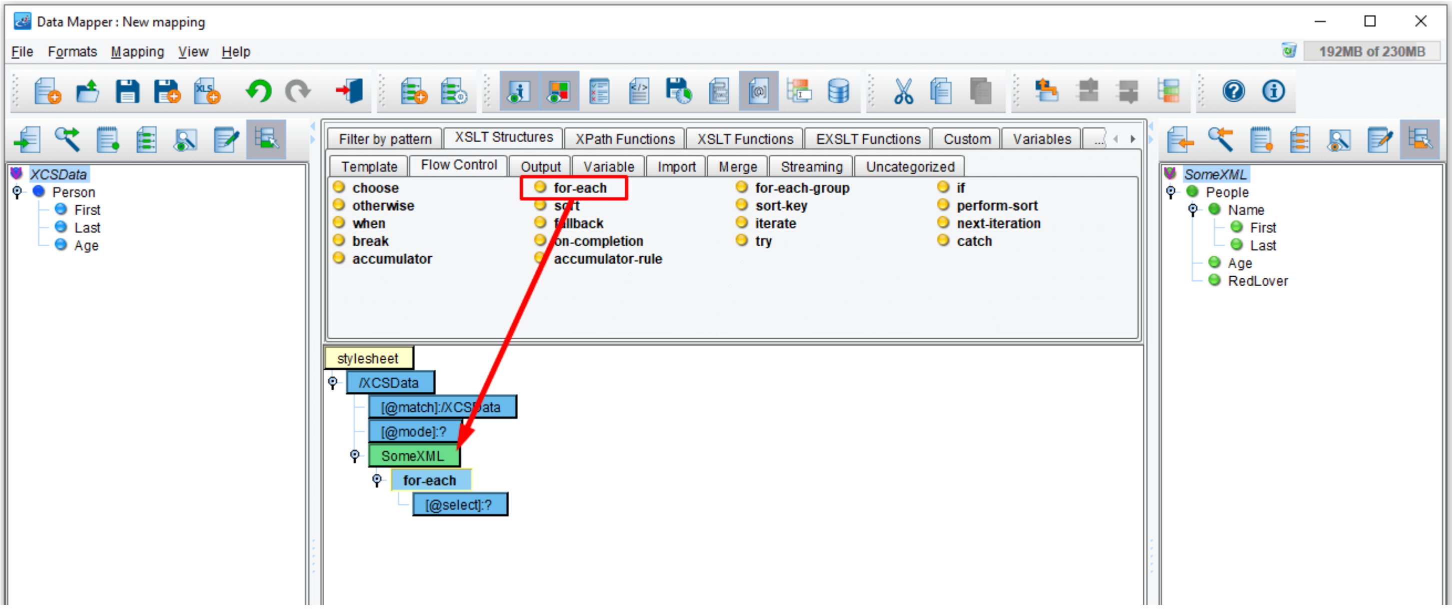
Notice the red V that appears over the **SomeXML** node? This indicates it has been mapped.

In order to output a People node for each Person in the Source, select the Flow Control tab in the pallet of XSLT structures and functions above the mapping.

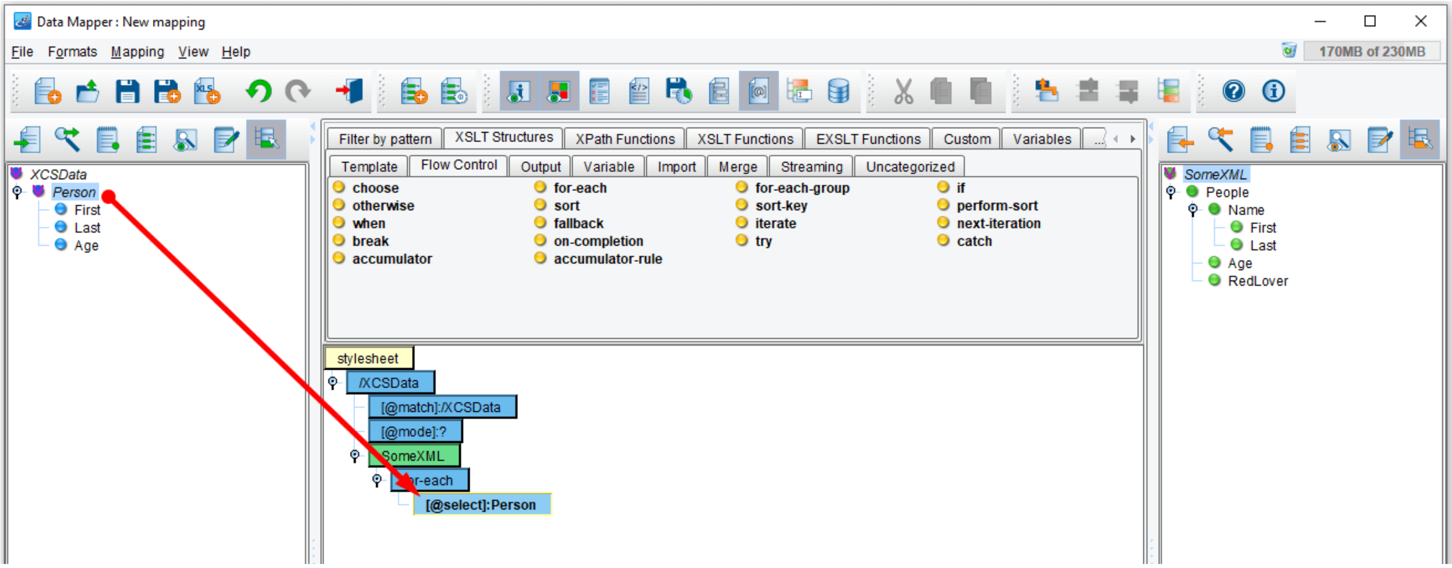
Note: The Data Mapper's XSLT palette of structures and functions allows you to do anything you can do by programming via the eiConsole's drag & drop process.

This tool palette provides a convenient location for all of these XSLT and XPath structures and functions. It allows users to drag & drop these directly into the mapping. Mousing over these palette tabs and their contents provides descriptions of what each item does.

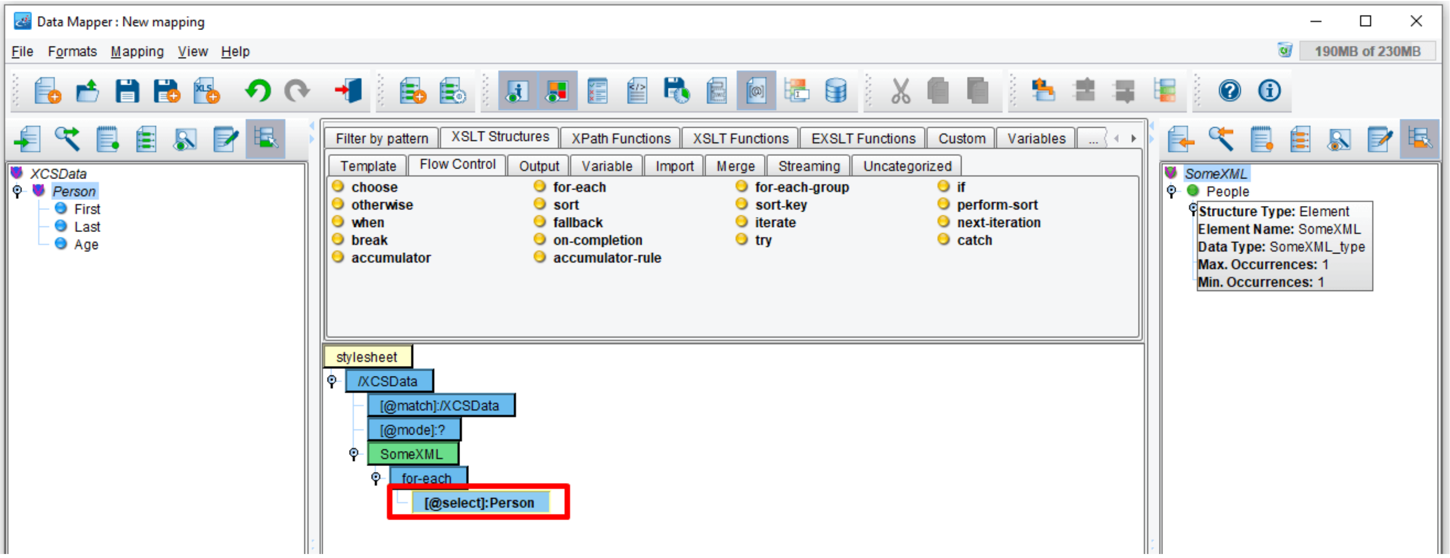
These structures allow users to perform many-to-one, one-to-many, or many-to-many mappings, manipulate string contents (concatenation, trimming, replacements), date formatting, and many other powerful operations that normally require users to write XSLT directly.



Under the **Flow Control** tab, select the **for-each** tool and drag & drop it on top of the **SomeXML** node. (Once you see the gray bar appears above SomeXML, it's in the right location and you can release the mouse.)

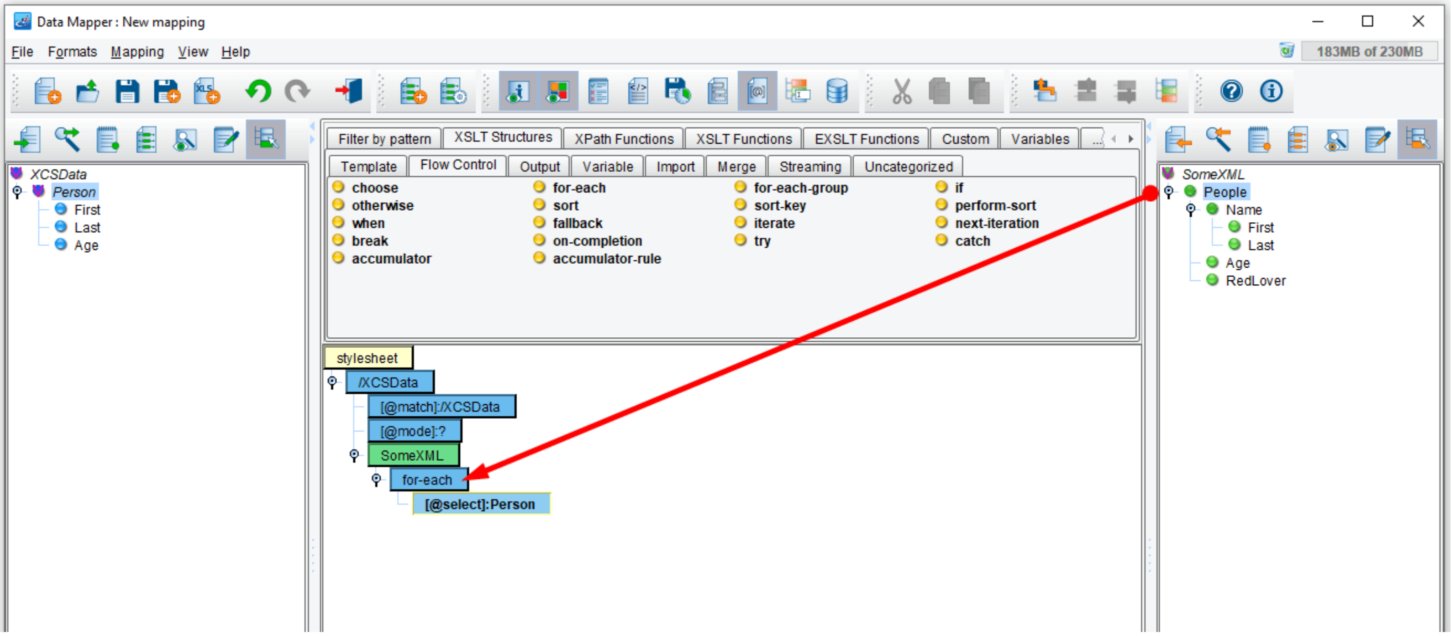


In order to iterate over each instance of Person in our Source, drag **Person** onto the **select** node underneath the for-each. Make sure the gray bar appears above the select node before you release the mouse. This indicates that some structure will be created underneath SomeXML for each instance of Person encountered in our Source.

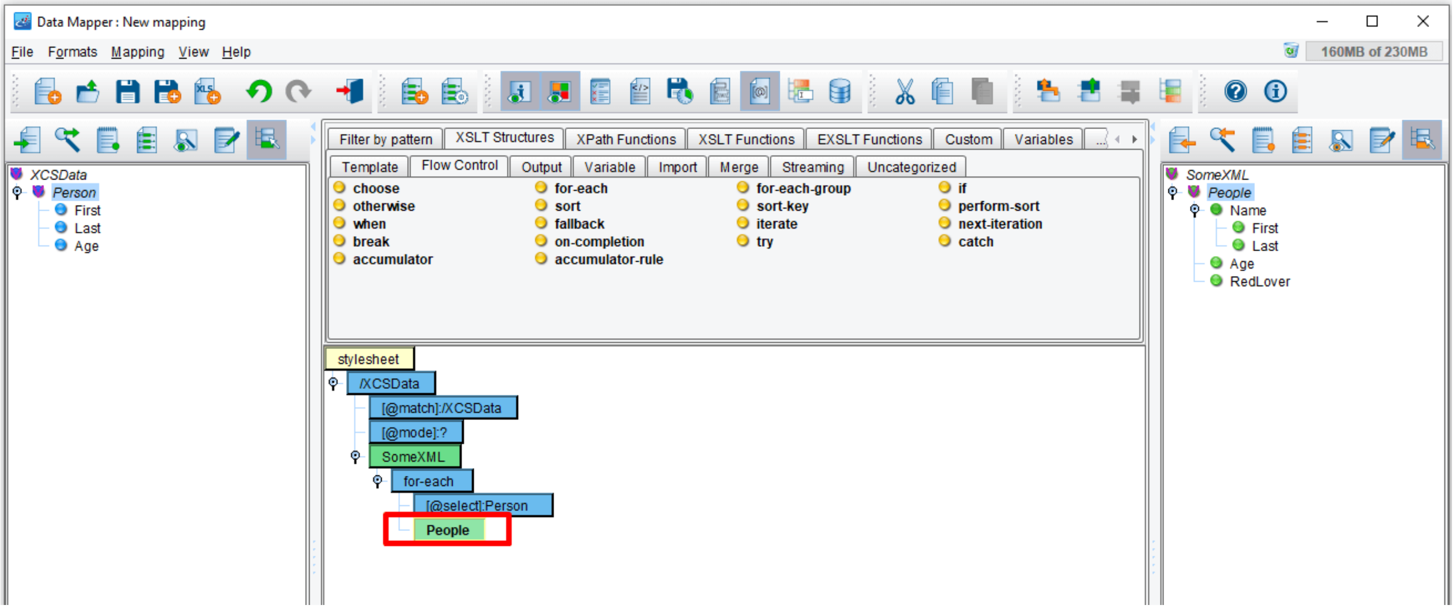


Structure Type: Element
Element Name: SomeXML
Data Type: SomeXML_type
Max. Occurrences: 1
Min. Occurrences: 1

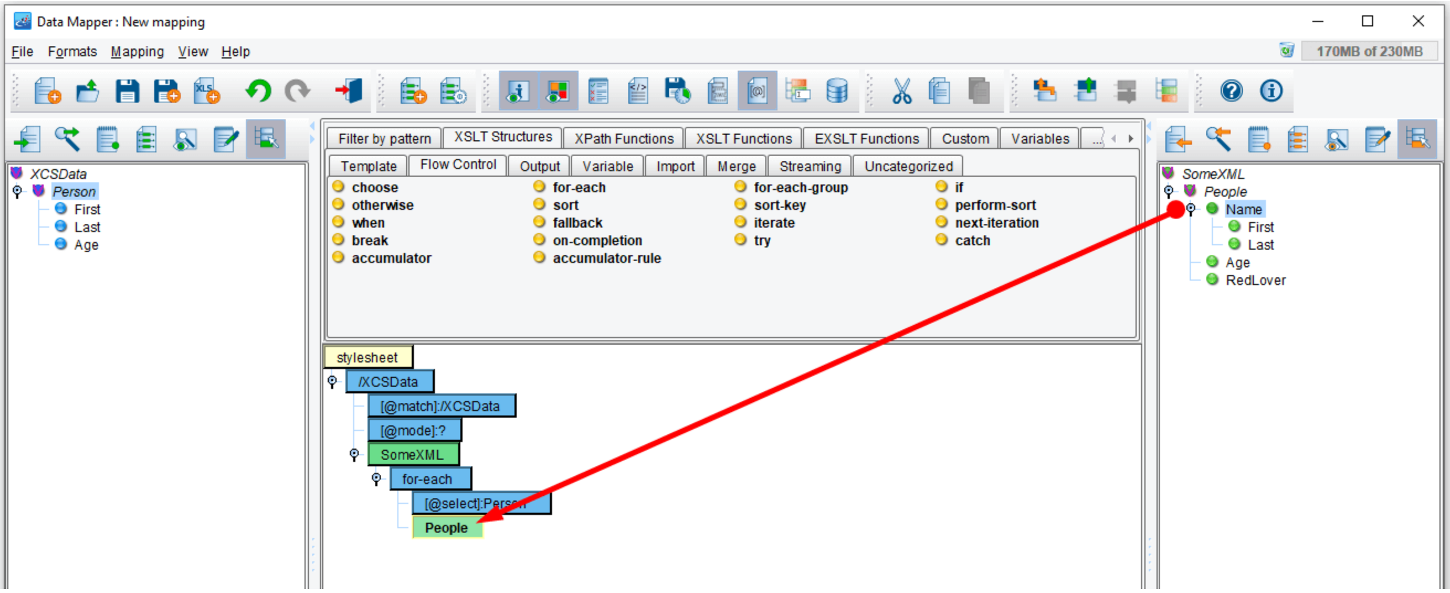
Mouse over the **SomeXML** node and click F1 to view the Help ToolTips.



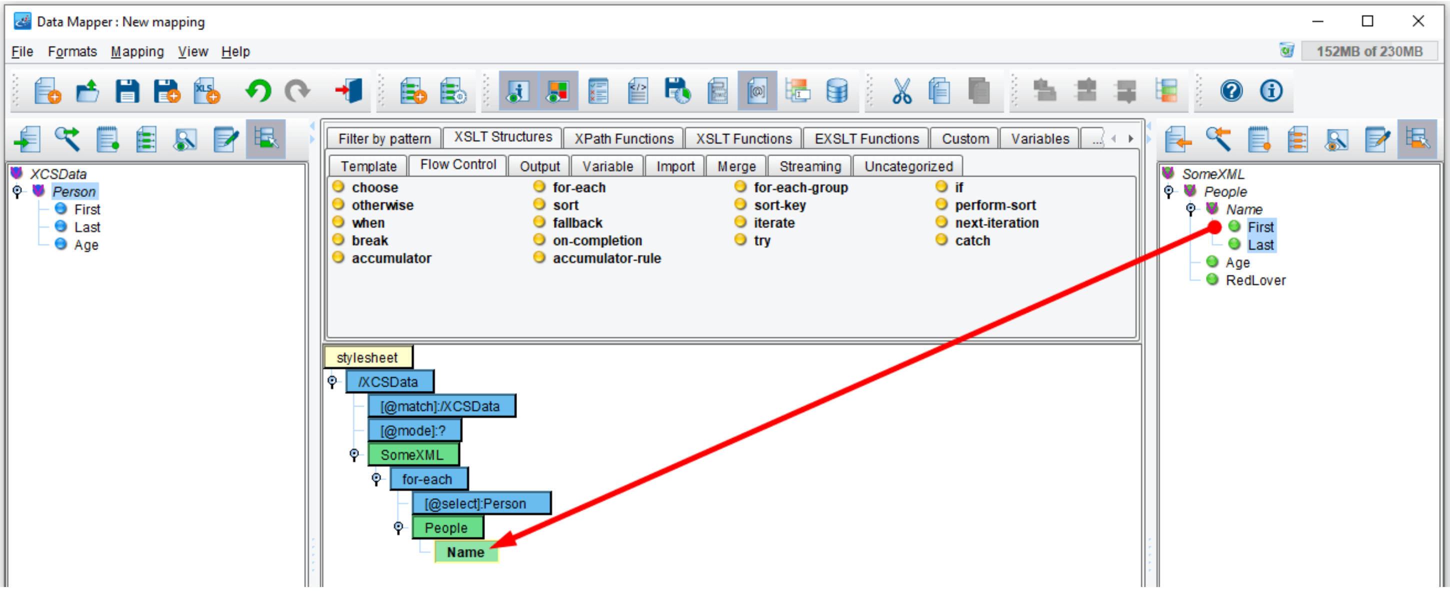
In order to generate a **People** node, drag this on top of the **for-each** and release the mouse button.



The green **People** node now appears in your mapping.

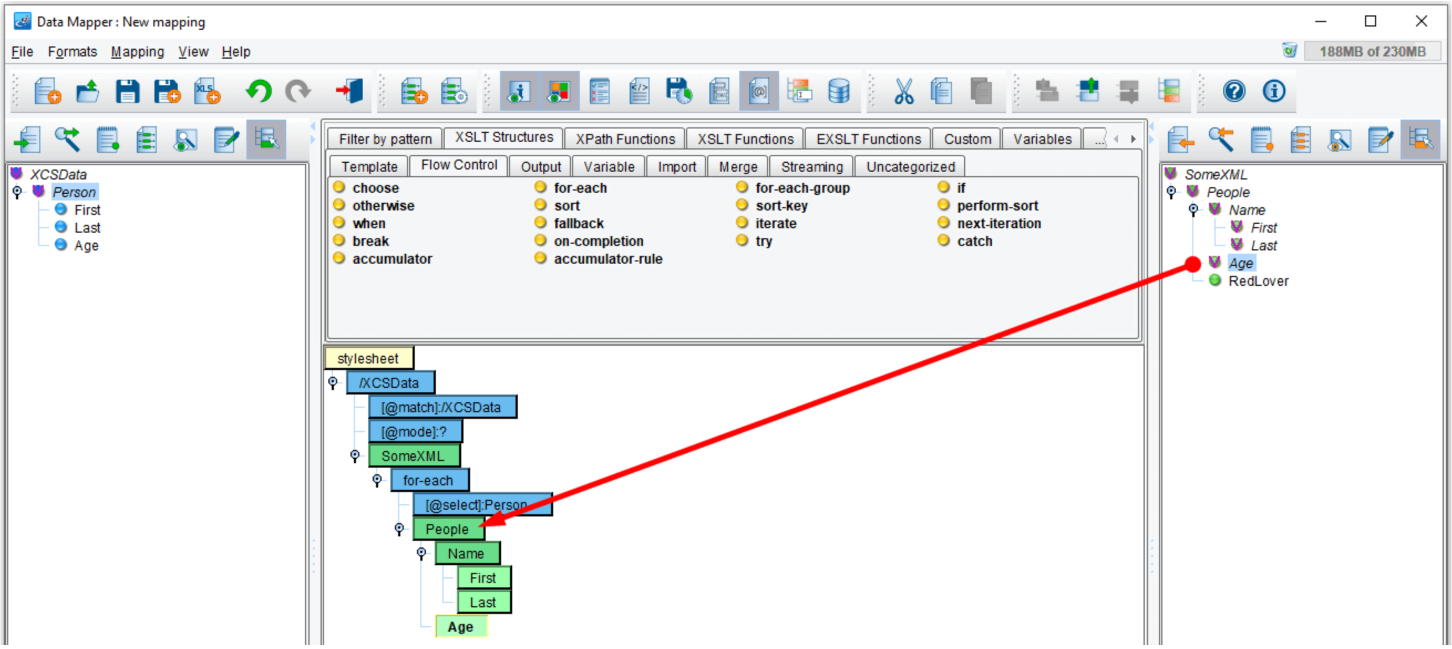


In order to create the Name, First, Last, Age and RedLover elements, do the following steps. Click the **Name** node to expand the tree and drag **Name** on top of **People**.

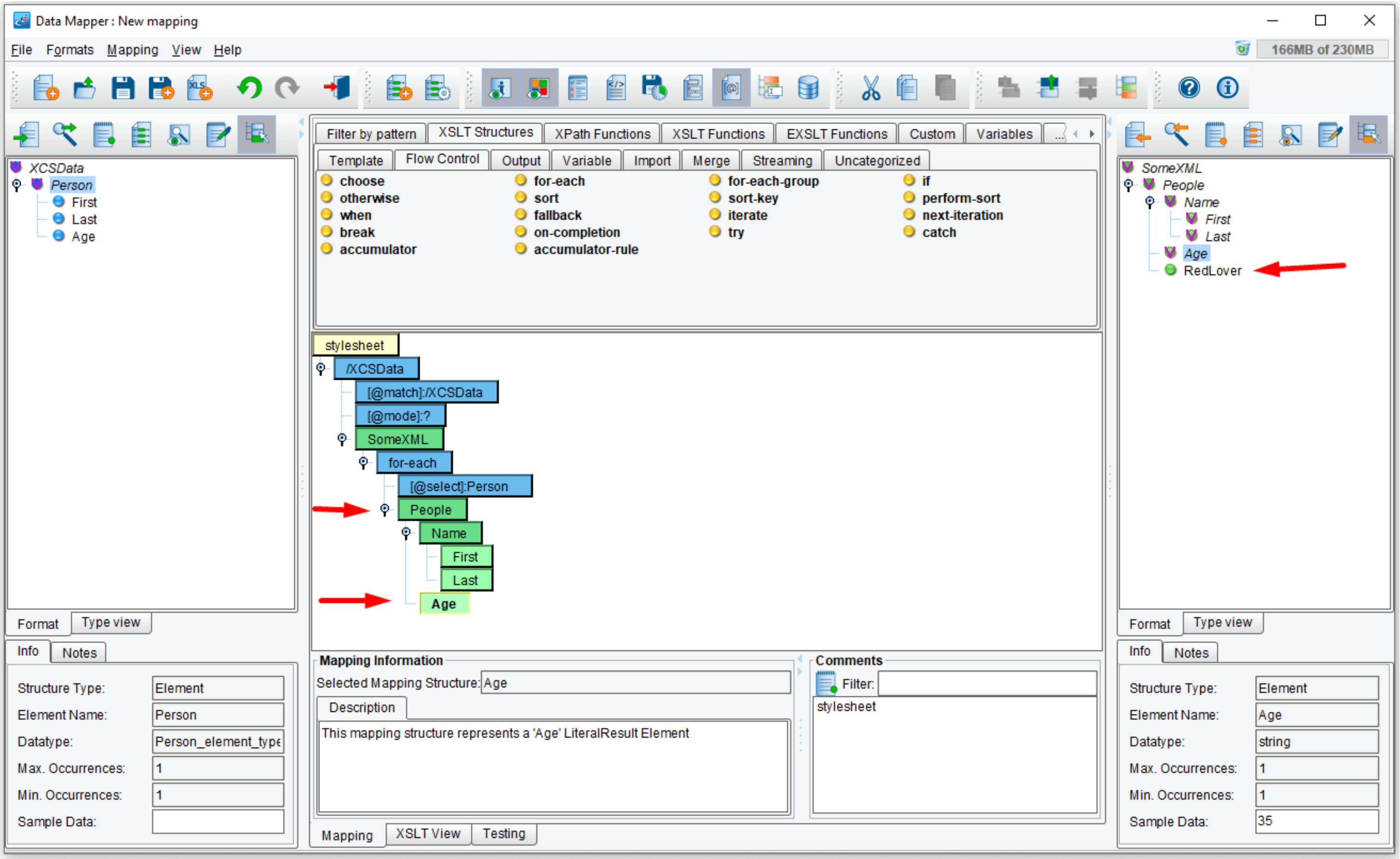


Then drag & drop **First** and **Last** on top of **Name**.

Note: You can drag elements individually or together. Select, shift click and drag.



Drag & drop **Age** on top of **People**.



Structure Type:	Element
Element Name:	Person
Datatype:	Person_element_type
Max. Occurrences:	1
Min. Occurrences:	1
Sample Data:	

Mapping Information

Selected Mapping Structure: Age

Description

This mapping structure represents a 'Age' LiteralResult Element

Comments

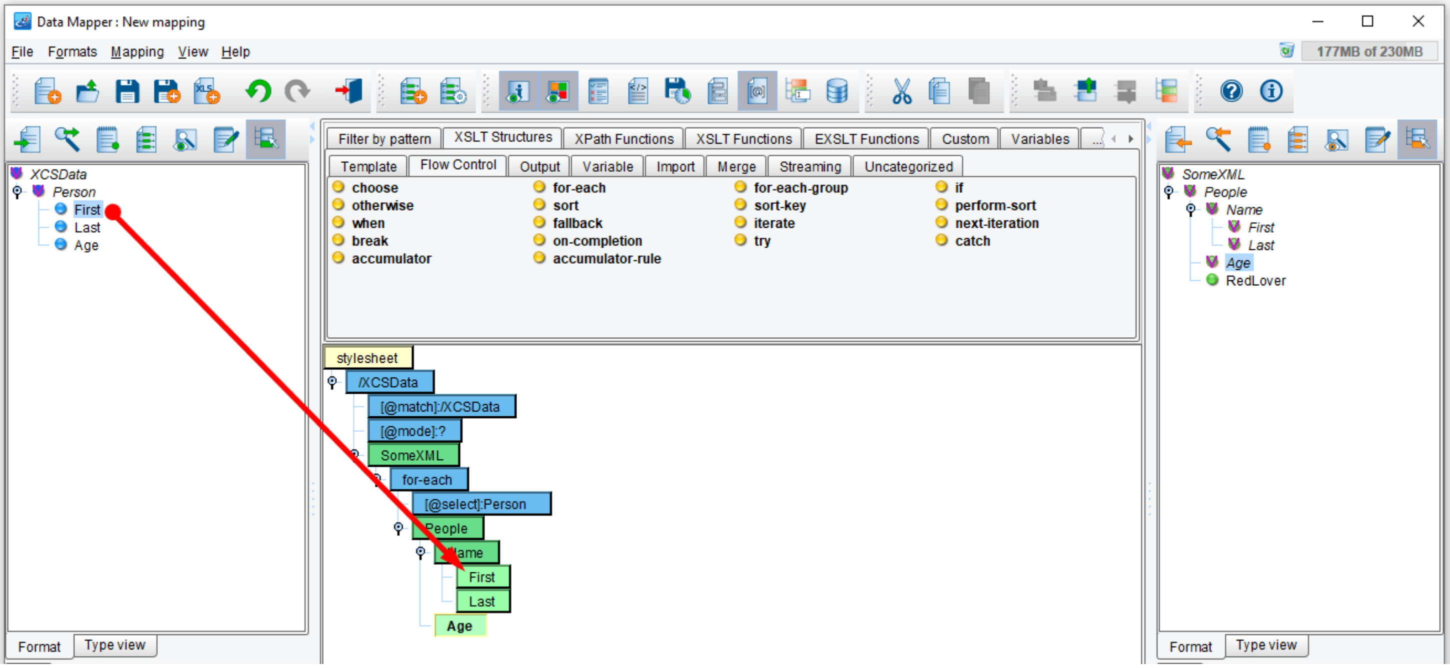
Filter:

stylesheet

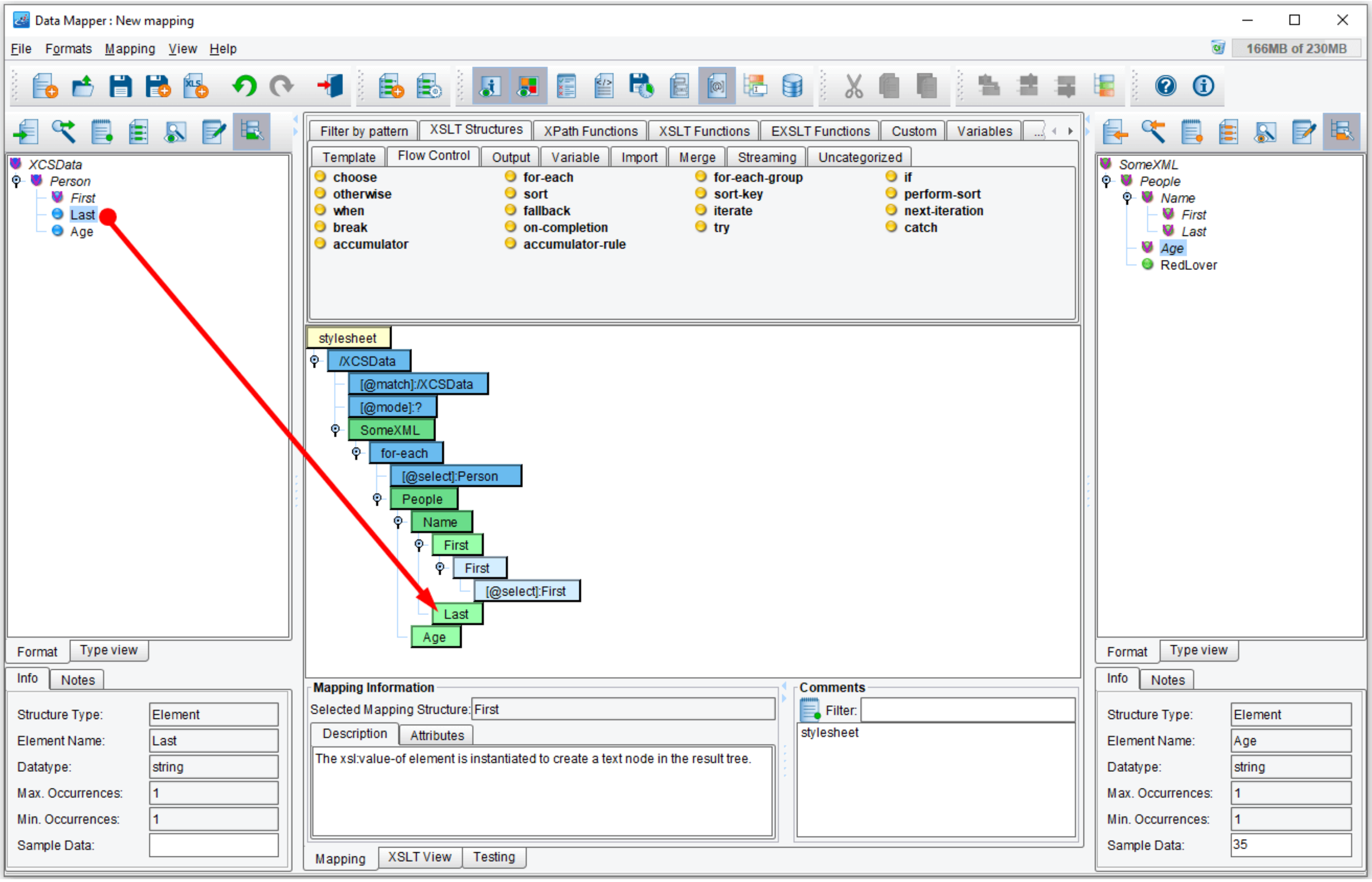
Structure Type:	Element
Element Name:	Age
Datatype:	string
Max. Occurrences:	1
Min. Occurrences:	1
Sample Data:	35

Even though you mapped Age onto People, **Age** appears in the correct position at the bottom of your mapping.

While **RedLover** is in the TargetFormat, ignore it for the purposes of this tutorial.



To map the First field from the Source onto the First field of the Target, drag **First** from the Source onto the green **First** node in the mapping.



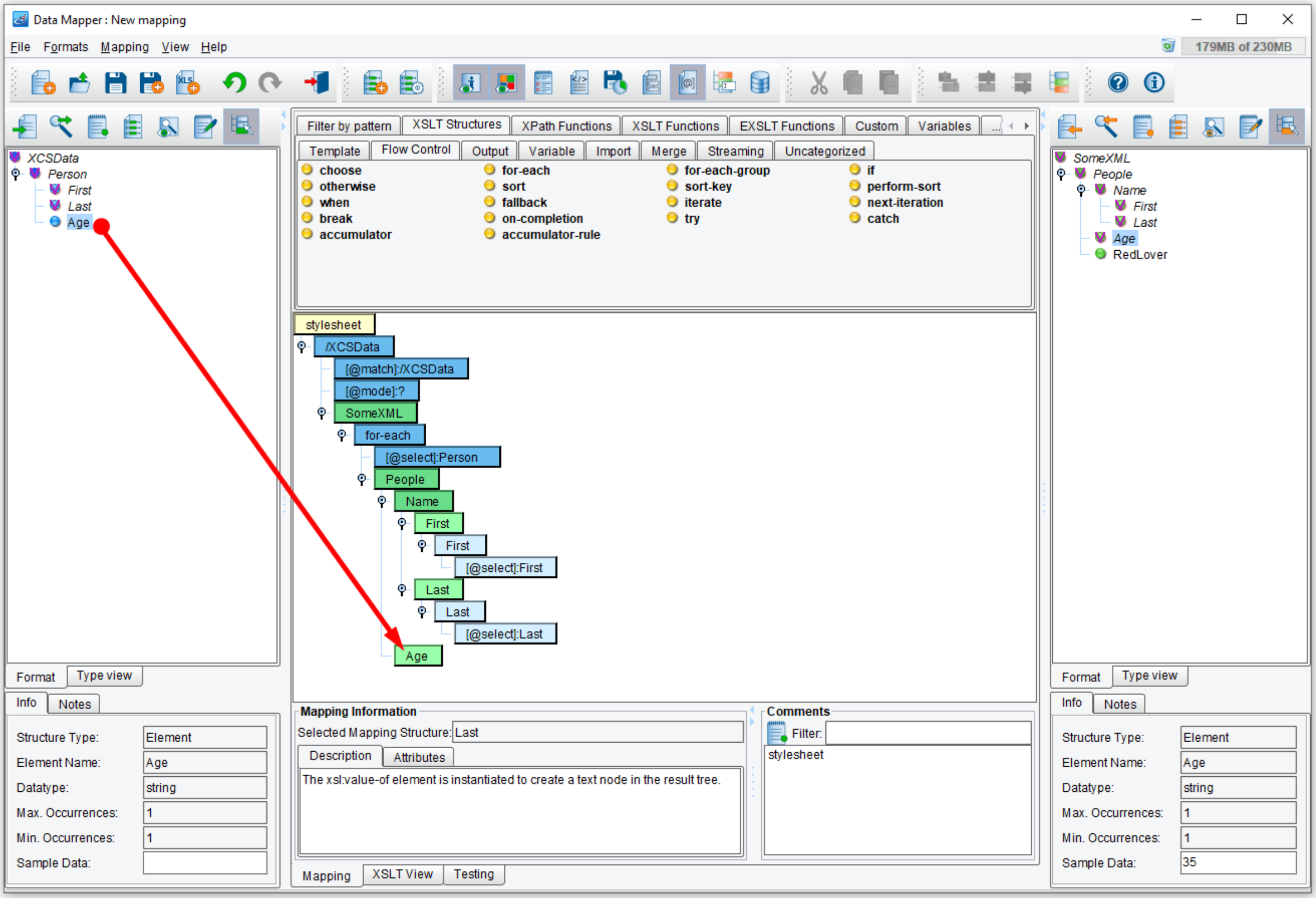
Structure Type: Element
Element Name: Last
Datatype: string
Max. Occurrences: 1
Min. Occurrences: 1
Sample Data:

Mapping Information
Selected Mapping Structure: First
Description | Attributes
The xsl:value-of element is instantiated to create a text node in the result tree.

Comments
Filter:
stylesheet

Structure Type: Element
Element Name: Age
Datatype: string
Max. Occurrences: 1
Min. Occurrences: 1
Sample Data: 35

Next, drag & drop **Last** from the source on to the **Last** element in your mapping.



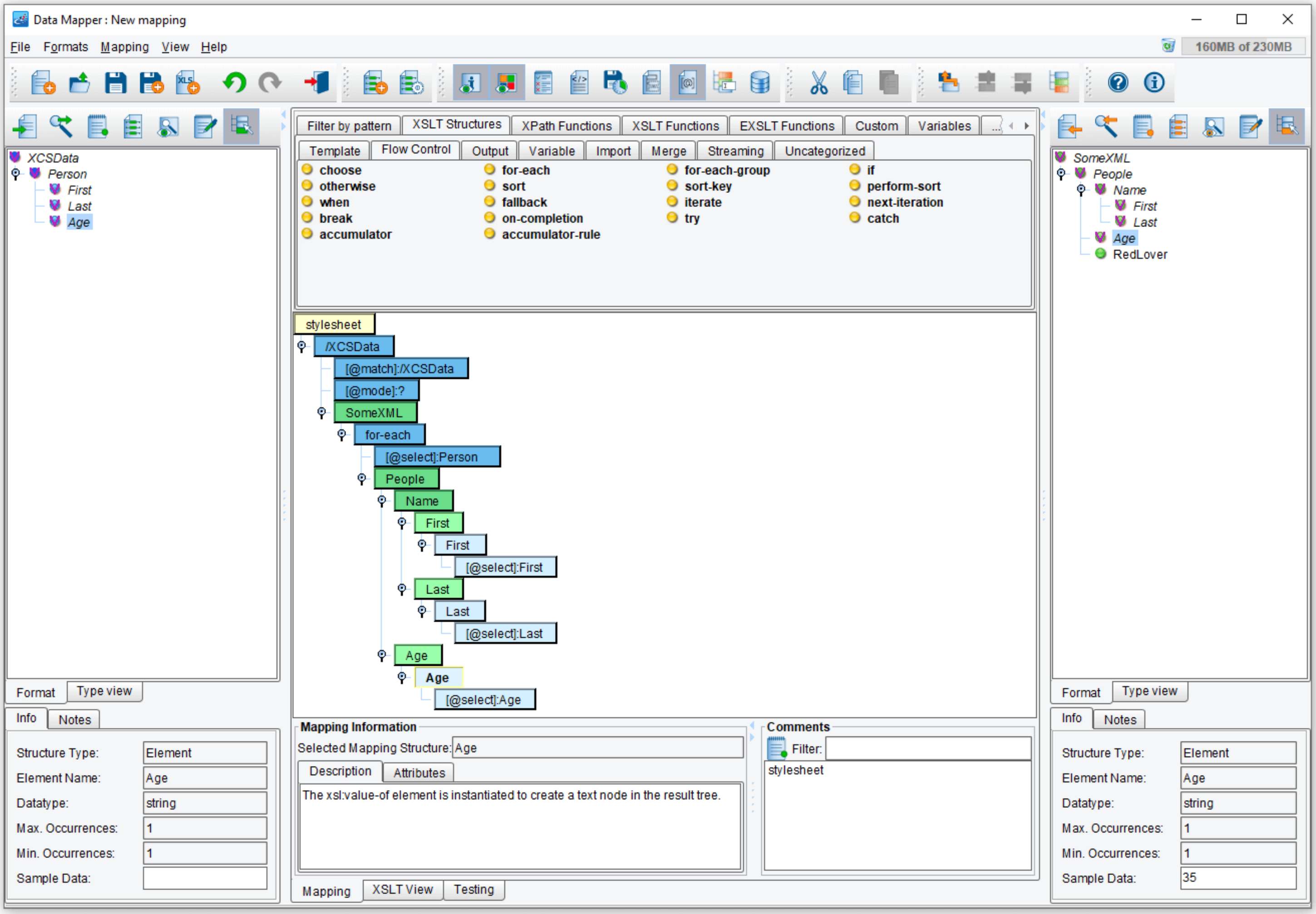
Structure Type:	Element
Element Name:	Age
Datatype:	string
Max. Occurrences:	1
Min. Occurrences:	1
Sample Data:	

Selected Mapping Structure: Last

The xsl:value-of element is instantiated to create a text node in the result tree.

Structure Type:	Element
Element Name:	Age
Datatype:	string
Max. Occurrences:	1
Min. Occurrences:	1
Sample Data:	35

Repeat the process by dragging and dropping **Age** from the source on top of **Age** in the mapping.



Structure Type: Element
Element Name: Age
Datatype: string
Max. Occurrences: 1
Min. Occurrences: 1
Sample Data:

Selected Mapping Structure: Age

The xsl:value-of element is instantiated to create a text node in the result tree.

stylesheet

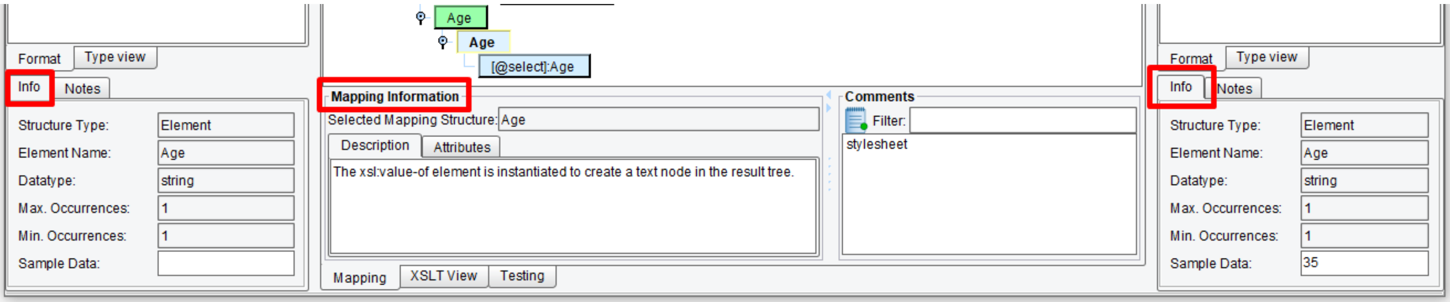
Structure Type: Element
Element Name: Age
Datatype: string
Max. Occurrences: 1
Min. Occurrences: 1
Sample Data: 35

Let's review what the various colored nodes represent: The blue nodes are from your source, the green from your target. Light blue elements are “childless” fields, and light green elements are “childless” elements. Dark blue and dark green elements have children (sub-elements). Attributes are always light green because they would never have children.

Other features available within the Data Mapper are:

- Information Panels
- Notes Tabs
- Type View Tabs

DATA MAPPER FEATURES – INFORMATION PANELS



Structure Type: Element
Element Name: Age
Datatype: string
Max. Occurrences: 1
Min. Occurrences: 1
Sample Data:

Selected Mapping Structure: Age

The xsl:value-of element is instantiated to create a text node in the result tree.

stylesheet

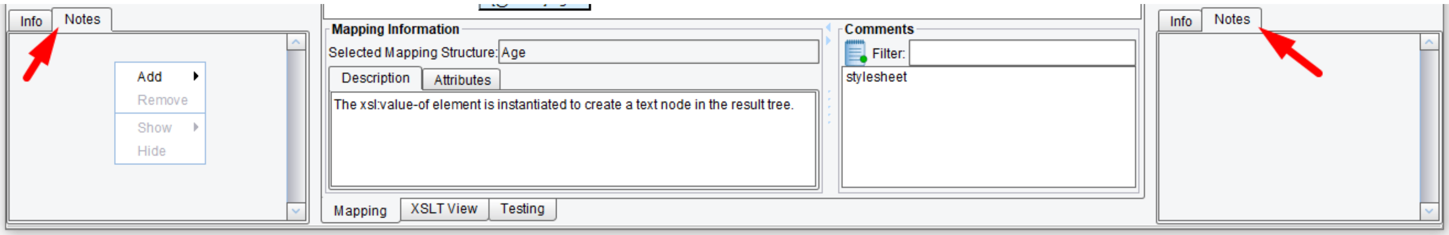
Structure Type: Element
Element Name: Age
Datatype: string
Max. Occurrences: 1
Min. Occurrences: 1
Sample Data: 35

The Data Mapper has three information panels. The information panel in the center provides high-level information about a selected node. It allows you to individually edit attributes for XSLT elements as well as to modify tabular mappings.

The information panels on the source and target show sample values for the selected node (called "sample data"); they also offer tabs for format descriptions and typecodes (for the ACORD standard, for example).

This applies to both source and target.

DATA MAPPER FEATURES – NOTES TABS



Selected Mapping Structure: Age

The xsl:value-of element is instantiated to create a text node in the result tree.

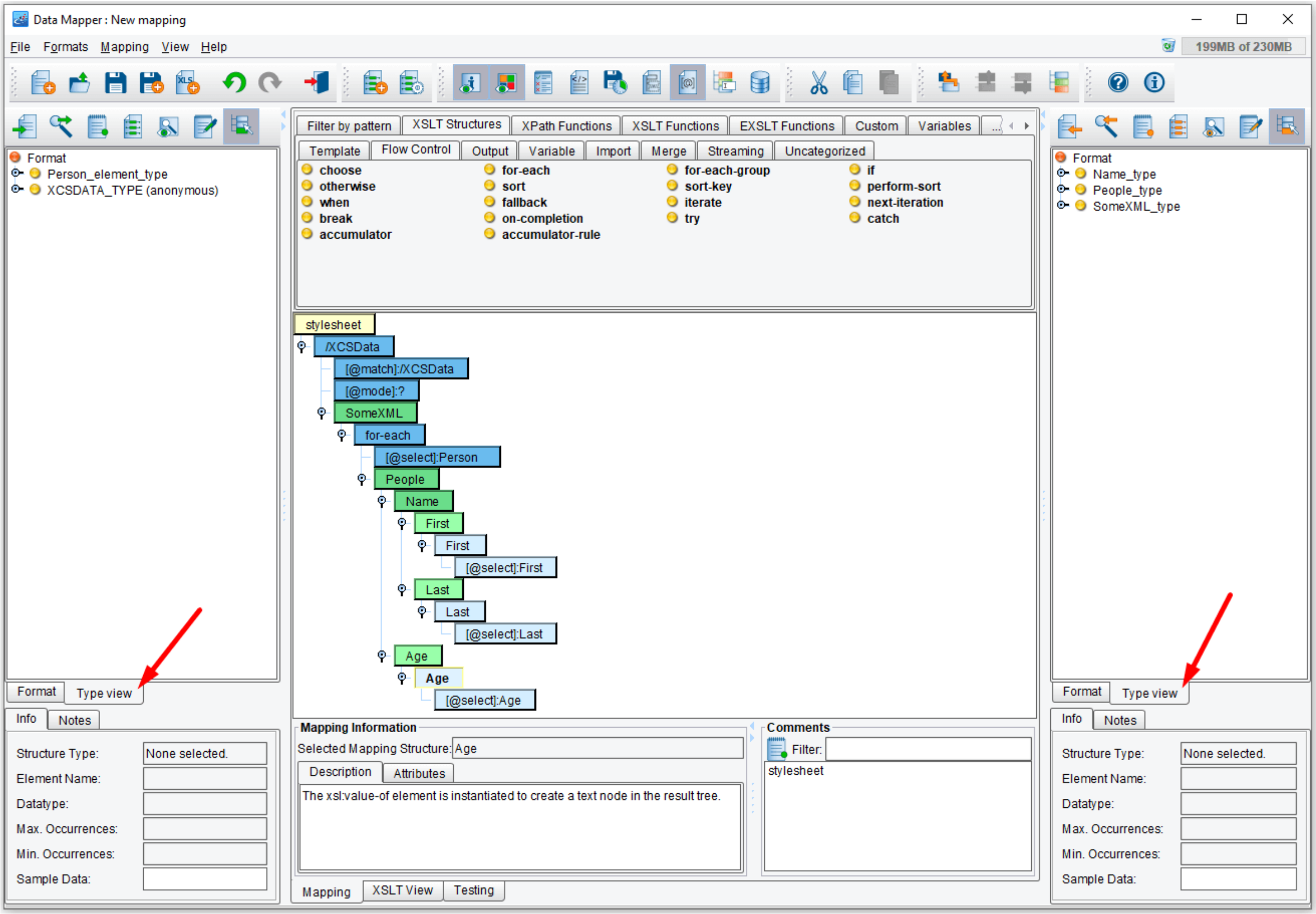
The notes tab, in source and target, allows users to associate comments with a particular node. You do this by selecting an item in the source or target, click the Notes tab, then right-click in the empty (grey) space and select Add > Define new. Then provide a name for your notes. This name applies to all notes defined by you, so you can think of it as a group (e.g. "John's Notes").

Once added to a node, you can type in the information in the Notes panel. You'll see a note icon in the source/target panel for any node you've added notes to.

You can also right-click in that panel again to add more notes (e.g. "Consultant Notes").

The functionality of notes is that if multiple users are managing a mapping, they can use these notes to provide comments or information to one another, or just as a convenient place holder for comments.

DATA MAPPER FEATURES - TYPE VIEW TABS



Structure Type: None selected.
Element Name:
Datatype:
Max. Occurrences:
Min. Occurrences:
Sample Data:

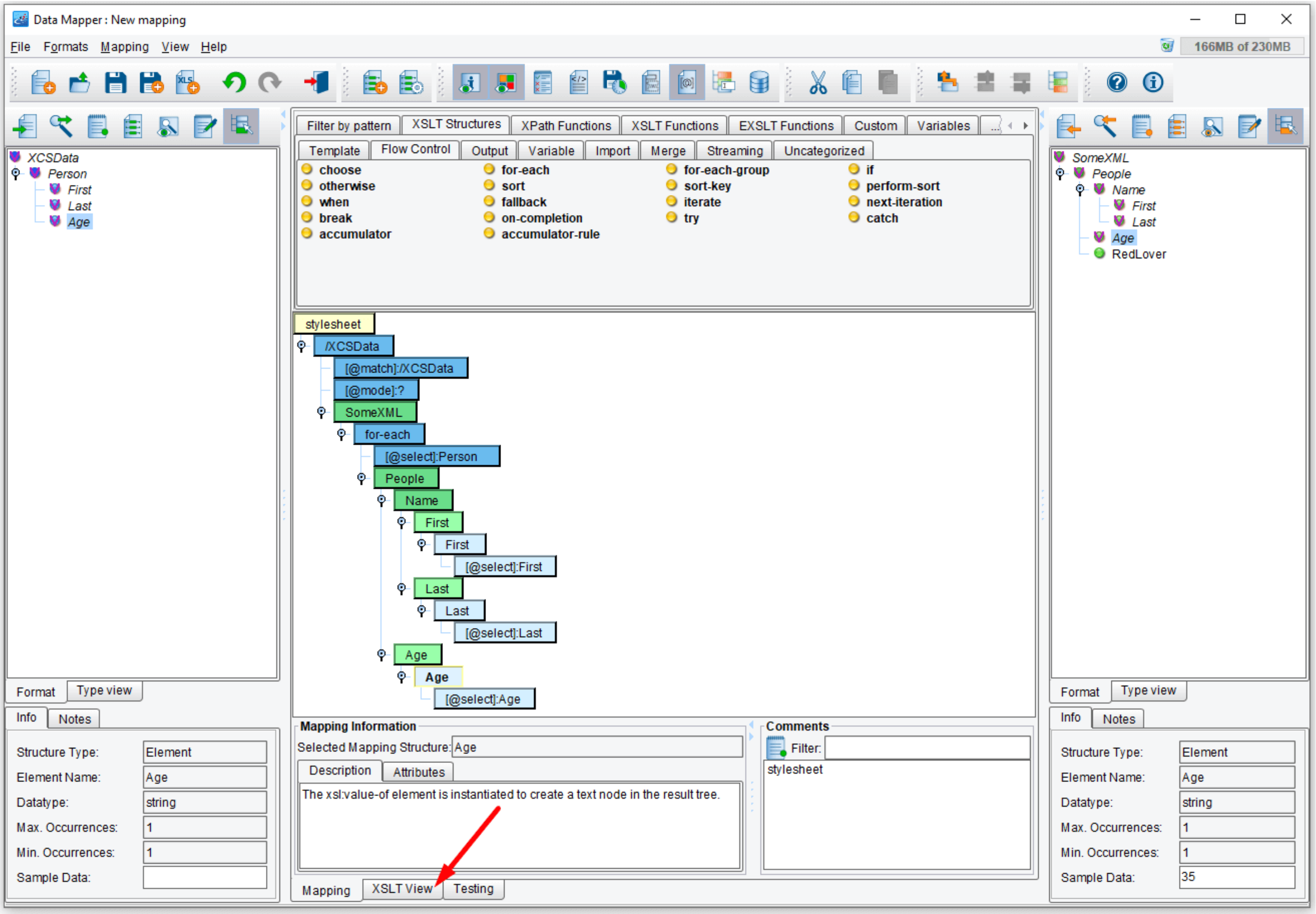
Selected Mapping Structure: Age

The xsl:value-of element is instantiated to create a text node in the result tree.

Structure Type: None selected.
Element Name:
Datatype:
Max. Occurrences:
Min. Occurrences:
Sample Data:

Type view, in source and target, is mainly for schema maintenance and editing, or for users who are really familiar with the format in its raw definition. For example, very experienced ACORD users can use it to find

or just look up information about types in the ACORD model.



Structure Type:	Element
Element Name:	Age
Datatype:	string
Max. Occurrences:	1
Min. Occurrences:	1
Sample Data:	

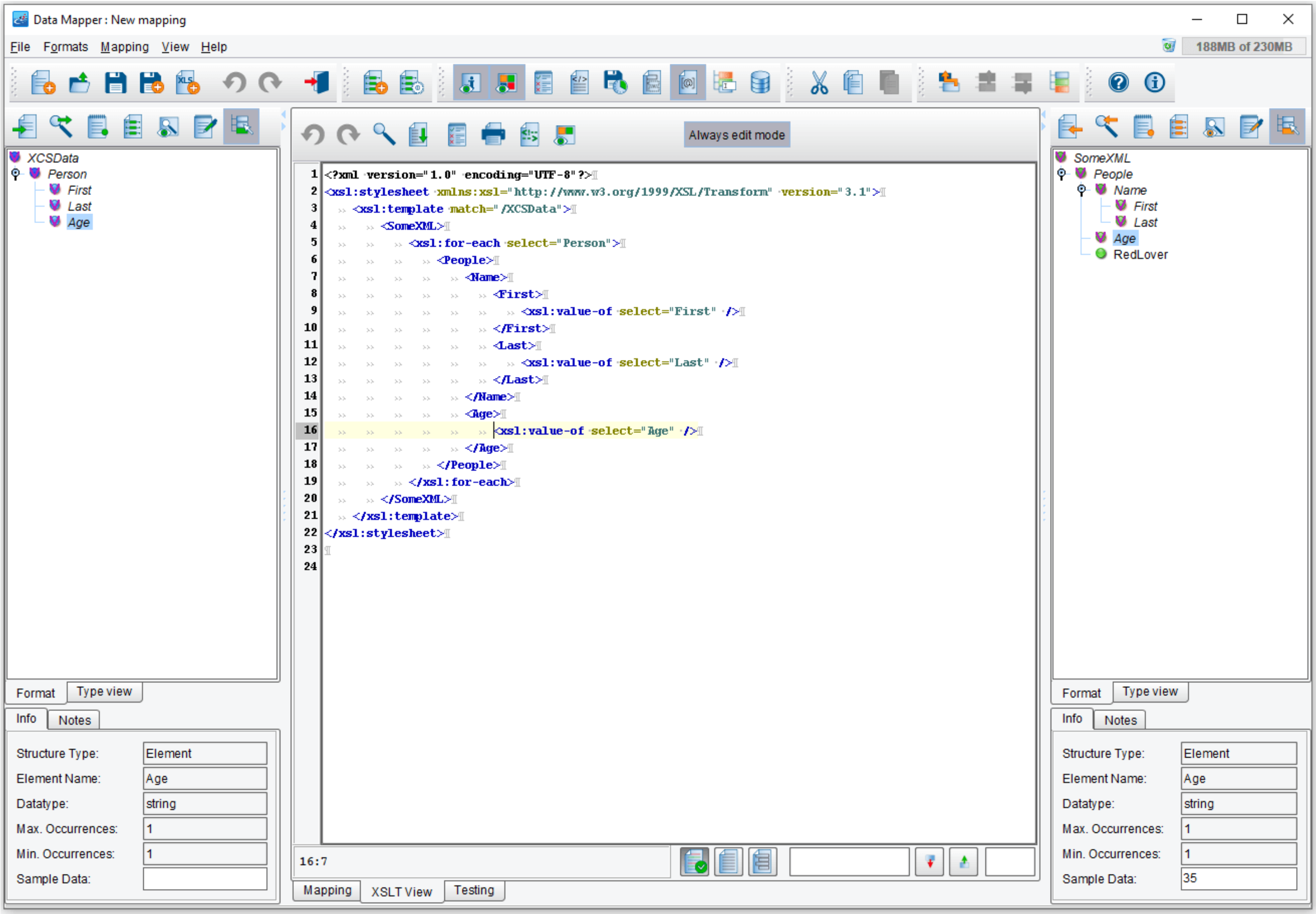
Selected Mapping Structure: Age

The xsl:value-of element is instantiated to create a text node in the result tree.

stylesheet

Structure Type:	Element
Element Name:	Age
Datatype:	string
Max. Occurrences:	1
Min. Occurrences:	1
Sample Data:	35

Click the **XSLT View** tab to view the generated XSLT.



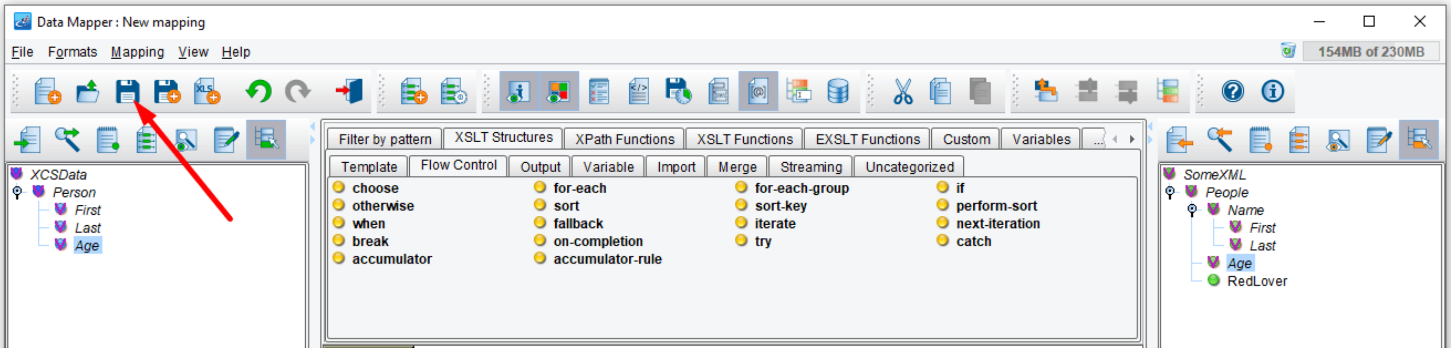
```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="3.1">
    <xsl:template match="/XCSData">
        <SomeXML>
            <xsl:for-each select="Person">
                <People>
                    <Name>
                        <First>
                            <xsl:value-of select="First" />
                        </First>
                        <Last>
                            <xsl:value-of select="Last" />
                        </Last>
                    </Name>
                    <Age>
                        <xsl:value-of select="Age" />
                    </Age>
                </People>
            </xsl:for-each>
        </SomeXML>
    </xsl:template>
</xsl:stylesheet>
```

Structure Type:	Element
Element Name:	Age
Datatype:	string
Max. Occurrences:	1
Min. Occurrences:	1
Sample Data:	

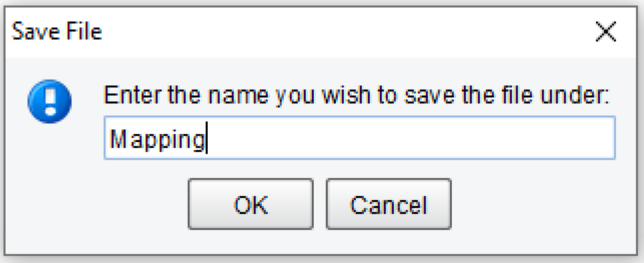
Structure Type:	Element
Element Name:	Age
Datatype:	string
Max. Occurrences:	1
Min. Occurrences:	1
Sample Data:	35

This is the XSLT generated by the Data Mapper.

Note: The Data Mapper generates, W3C compliant XSLT, in real-time, under the graphical view. There is no proprietary scripting or coding. Users can access the XSLT at any time by clicking the XSLT view tab. Should a user choose to work in the XSLT mode, the graphical view will be updated in real-time, as well. Users who wish to perform complex or specific functions not available via the eiConsole's wide range of XSLT structures and functions can leverage the virtually unlimited reference materials and many user groups on the web. Once they find what they want, they can simply copy and paste into the XSLT view.



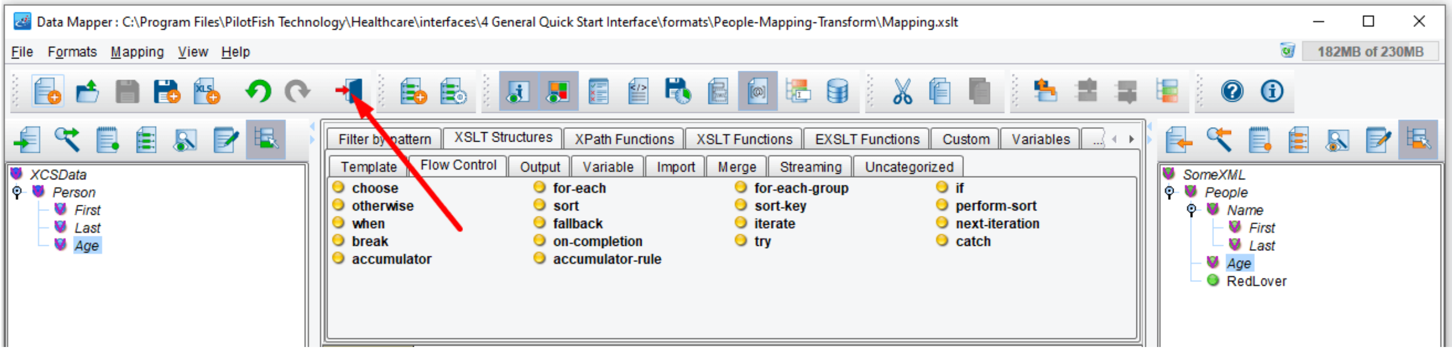
Save your mapping by clicking the **Save Current Mapping** button.



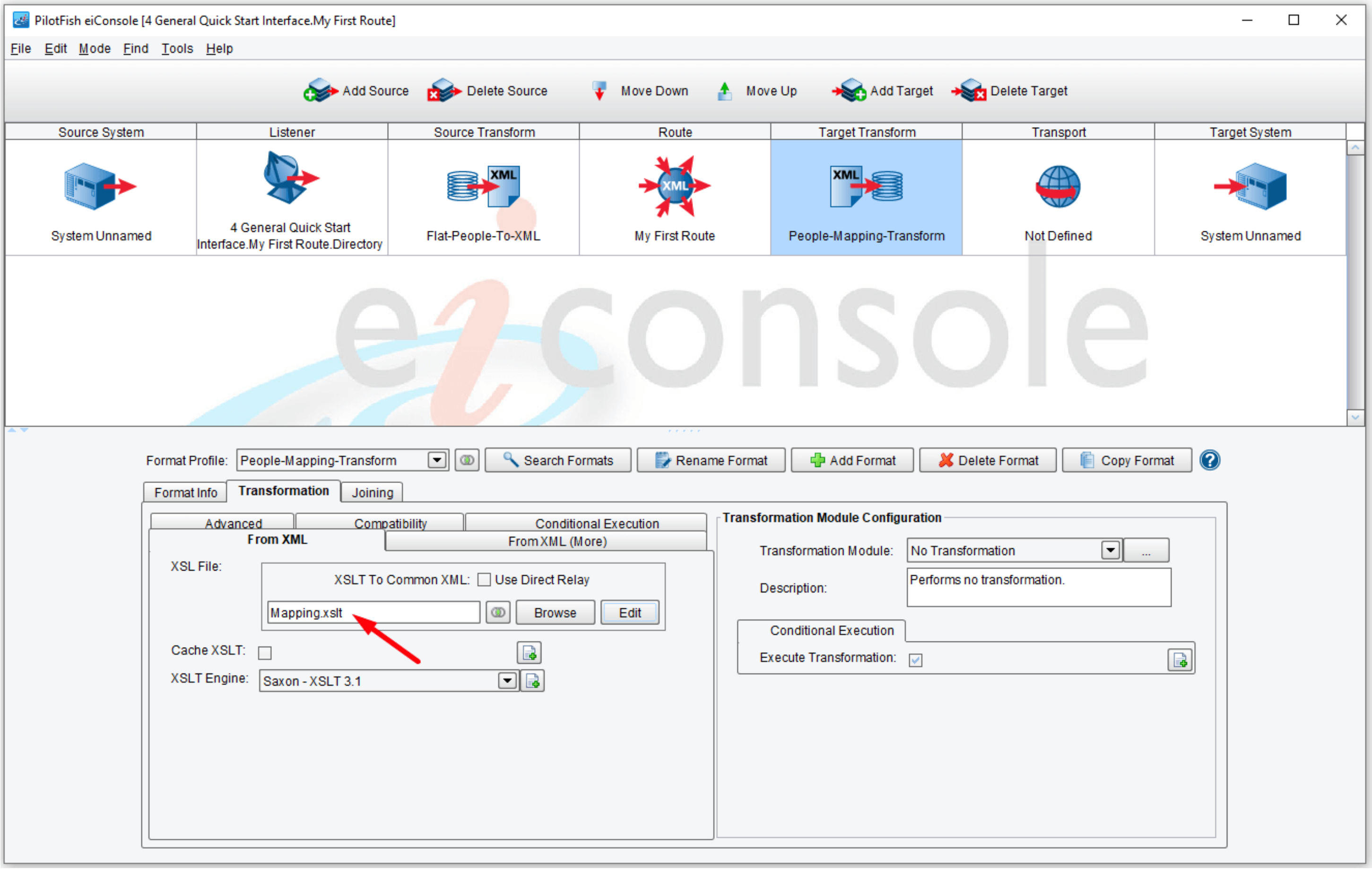
Enter the name you wish to save the file under:

Mapping

You’ll be prompted for a name in the Save File dialog. You can simply enter the name **"Mapping"** and click **OK.**


Data Mapper : C:\Program Files\PilotFish Technology\Healthcare\interfaces\4 General Quick Start Interface\formats\People-Mapping-Transform\Mapping.xslt

You can then return to the console by clicking the **Return to Console** button.


PilotFish eiConsole [4 General Quick Start Interface.My First Route]

Format Profile: People-Mapping-Transform

XSL File:

XSLT To Common XML: □ Use Direct Relay

Mapping.xslt

Cache XSLT: □

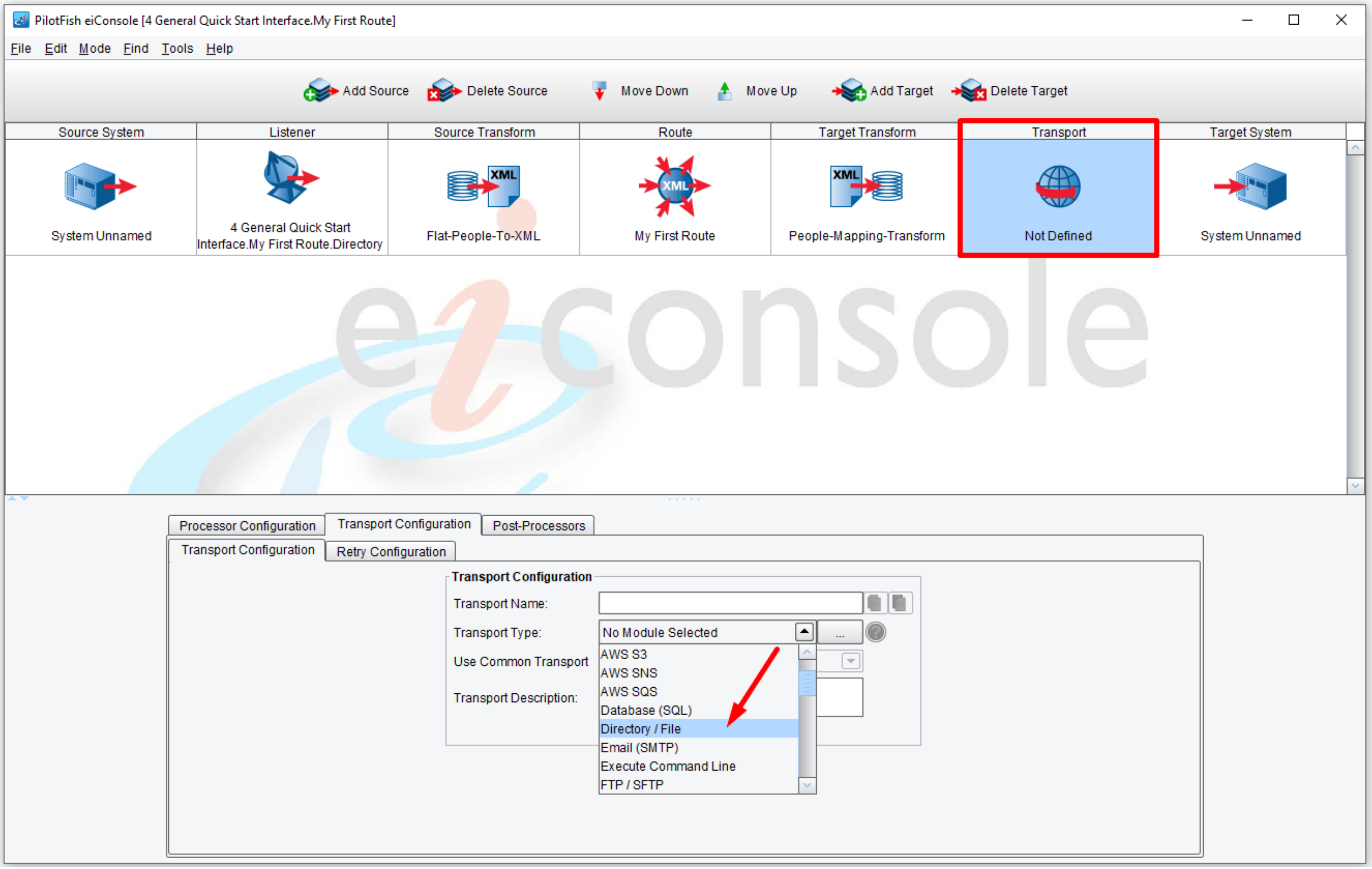
XSLT Engine: Saxon - XSLT 3.1

Transformation Module: No Transformation

Description: Performs no transformation.

Execute Transformation: ☑

You’ll see the name of the configured mapping now appears in the XSLT Configuration area. Now, click the **Transport** icon.



Transport Configuration

Transport Name:

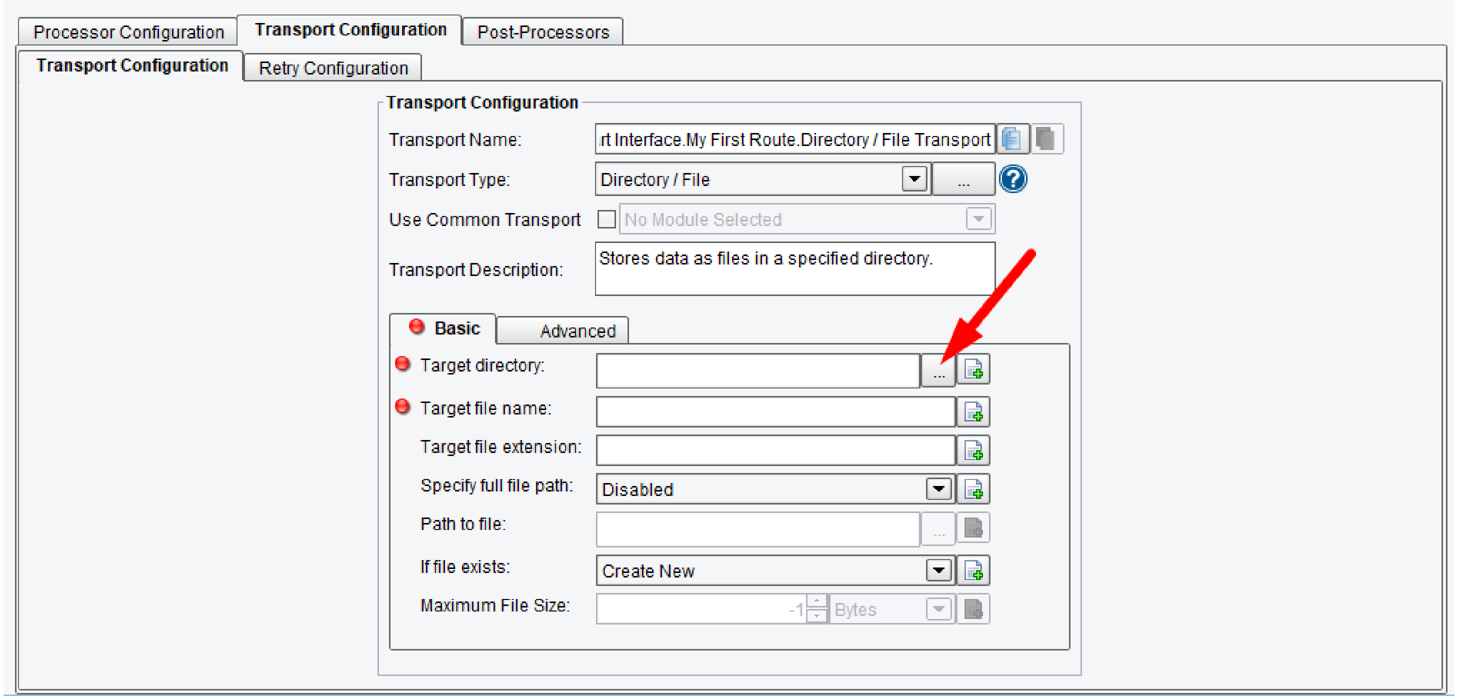
Transport Type: No Module Selected

Use Common Transport

Transport Description:

AWS S3
AWS SNS
AWS SQS
Database (SQL)
Directory / File
Email (SMTP)
Execute Command Line
FTP / SFTP

To drop the transformed XML off in a file, select **Directory / File** in the Transport Type drop-down.



Transport Configuration

Transport Name: rt Interface.My First Route.Directory / File Transport

Transport Type: Directory / File

Use Common Transport ☐ No Module Selected

Transport Description: Stores data as files in a specified directory.

● Target directory:

● Target file name:

Target file extension:

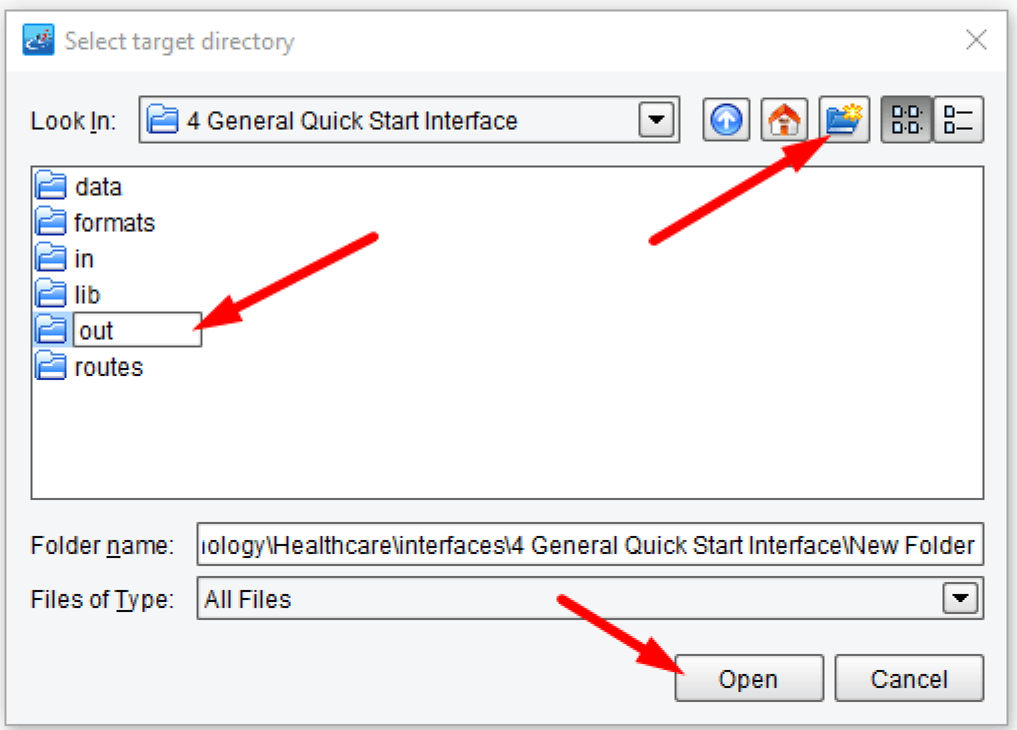
Specify full file path: Disabled

Path to file:

If file exists: Create New

Maximum File Size: -1 Bytes

When the Transport Configuration panel opens, configure the required items. Click the **Ellipsis** button next to the Target directory configuration item.



Select target directory

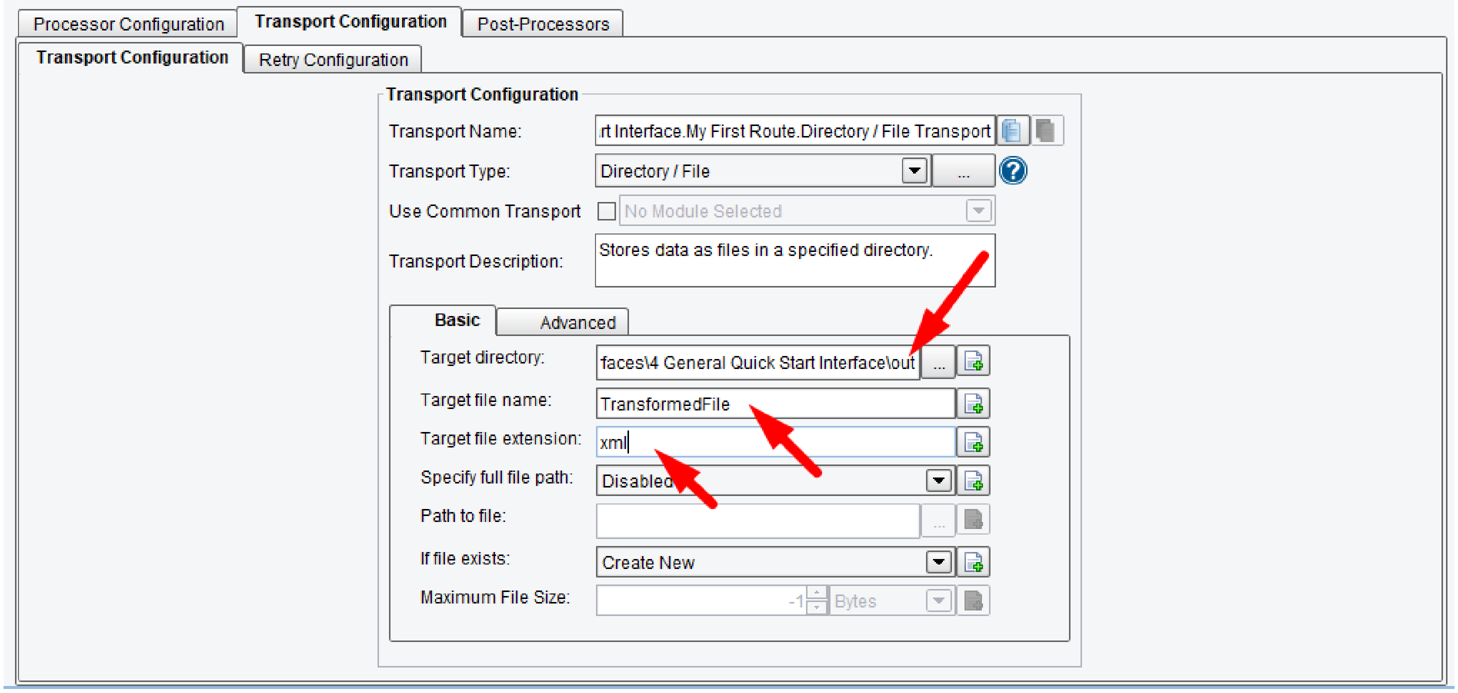
Look In: 4 General Quick Start Interface

- data
- formats
- in
- lib
- out
- routes

Folder name: ıology\Healthcare\interfaces\4 General Quick Start Interface\New Folder

Files of Type: All Files

Navigate to **4 General Quick Start Interface** directory and double click to open. Select the **New Folder** icon to create a new folder and name it **"out"**. Select it and click **Open.**



Transport Configuration

Transport Name: rt Interface.My First Route.Directory / File Transport

Transport Type: Directory / File

Use Common Transport □ No Module Selected

Transport Description: Stores data as files in a specified directory.

Target directory: faces\4 General Quick Start Interface\out

Target file name: TransformedFile

Target file extension: xml

Specify full file path: Disabled

Path to file:

If file exists: Create New

Maximum File Size: -1 Bytes

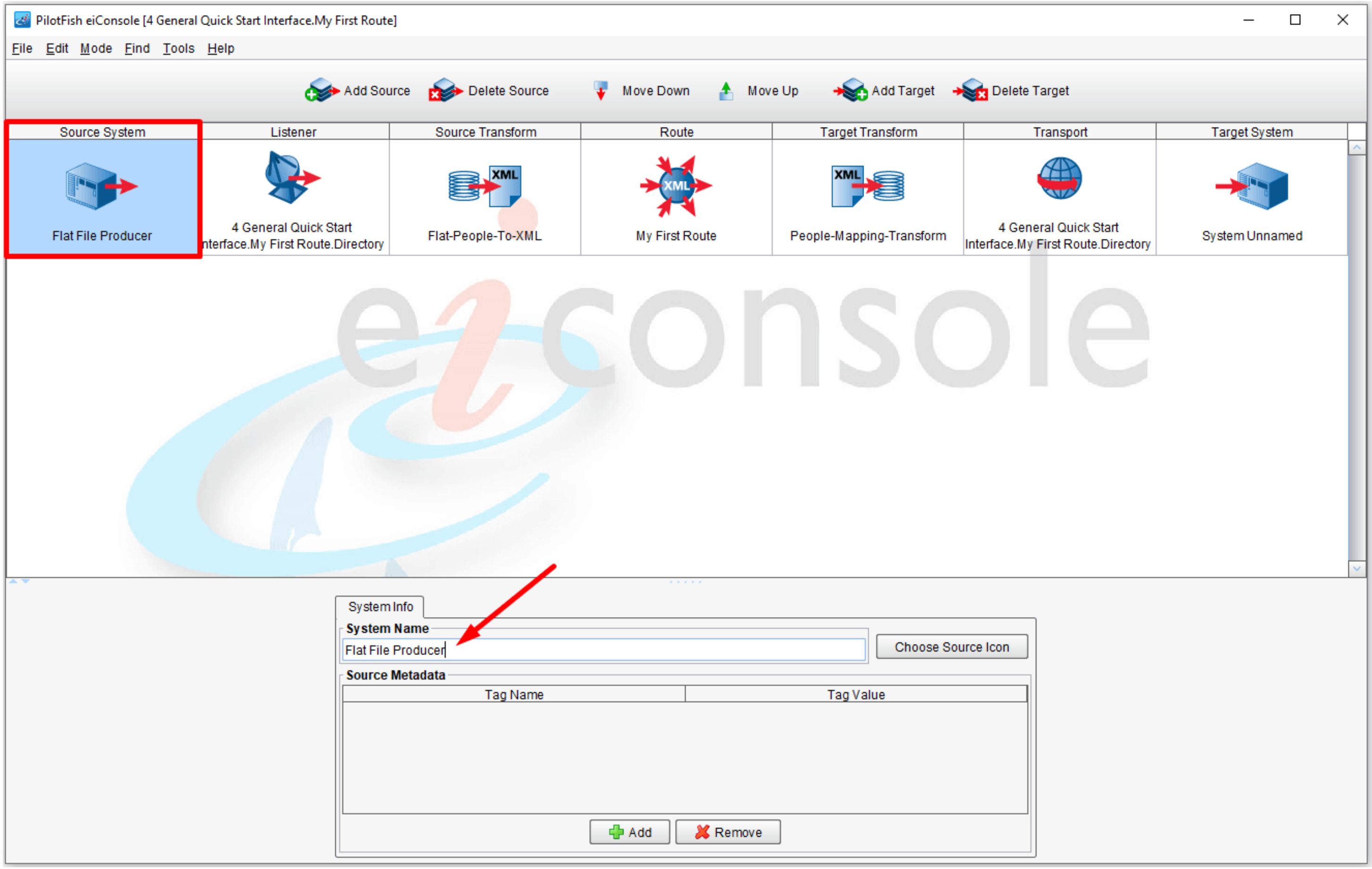
The out folder is now set in the Target directory.

In the Target file name area, type in **"TransformedFile"**.

In the Target File Extension, type in **"xml"**.

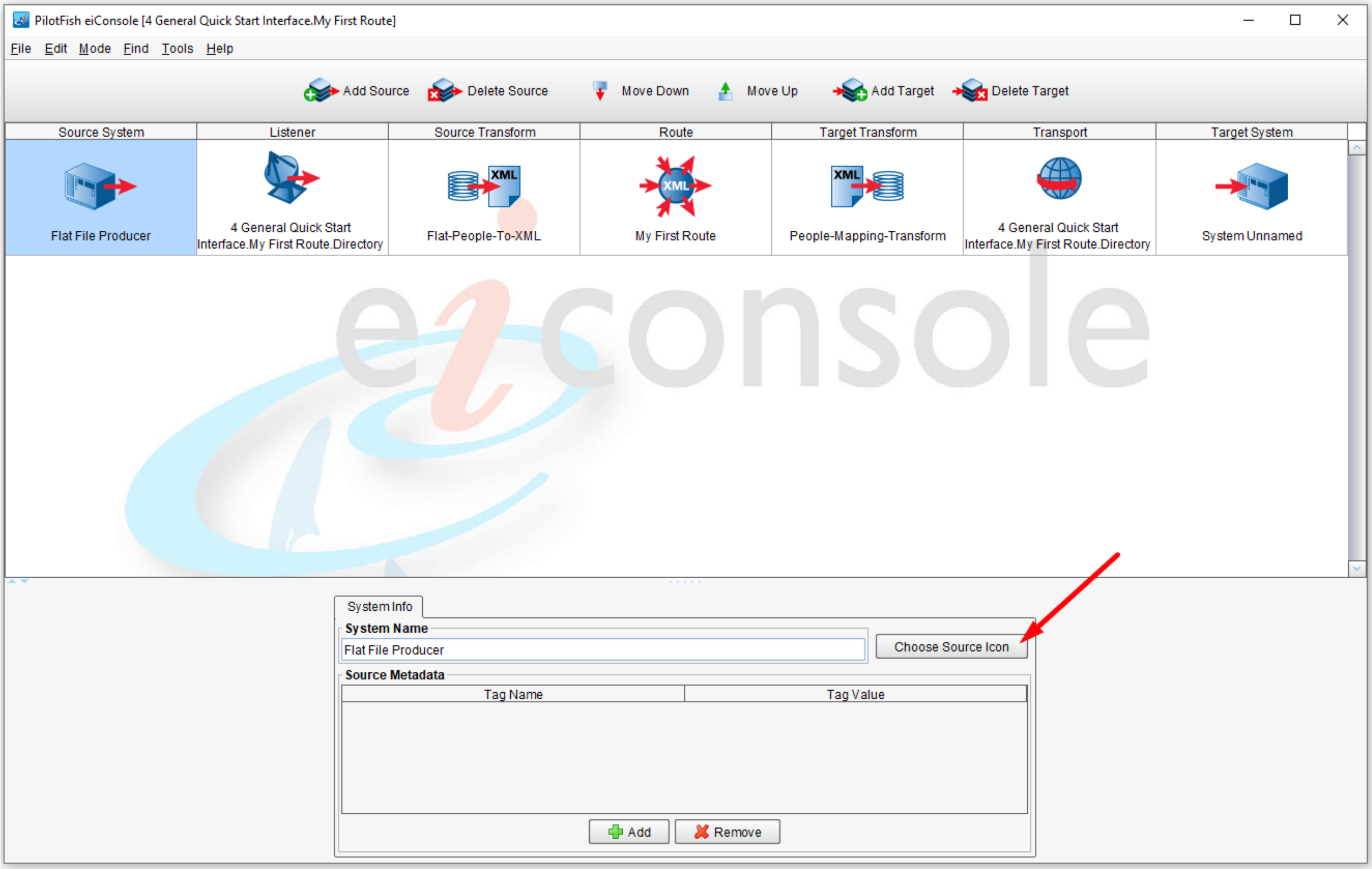
To make the interface a bit more understandable, provide Metadata for the Source and Target system.

Source & Target System Metadata



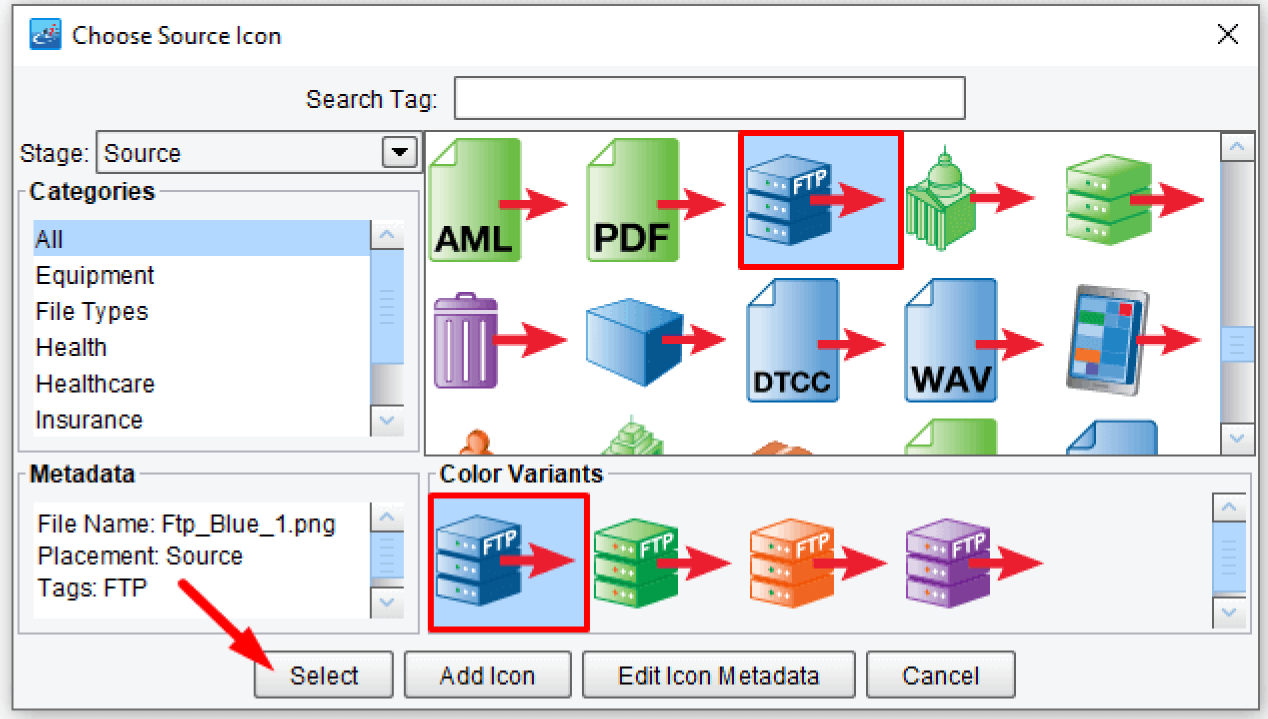
System Name: Flat File Producer

Click on the **Source System** stage and type in **"Flat File Producer"** in the System Name text box.



System Name: Flat File Producer

Also, you can set different icons for each of your Sources. In our example, we have only one source, so let's choose an icon for it. Click on the **Choose Source Icon** button.

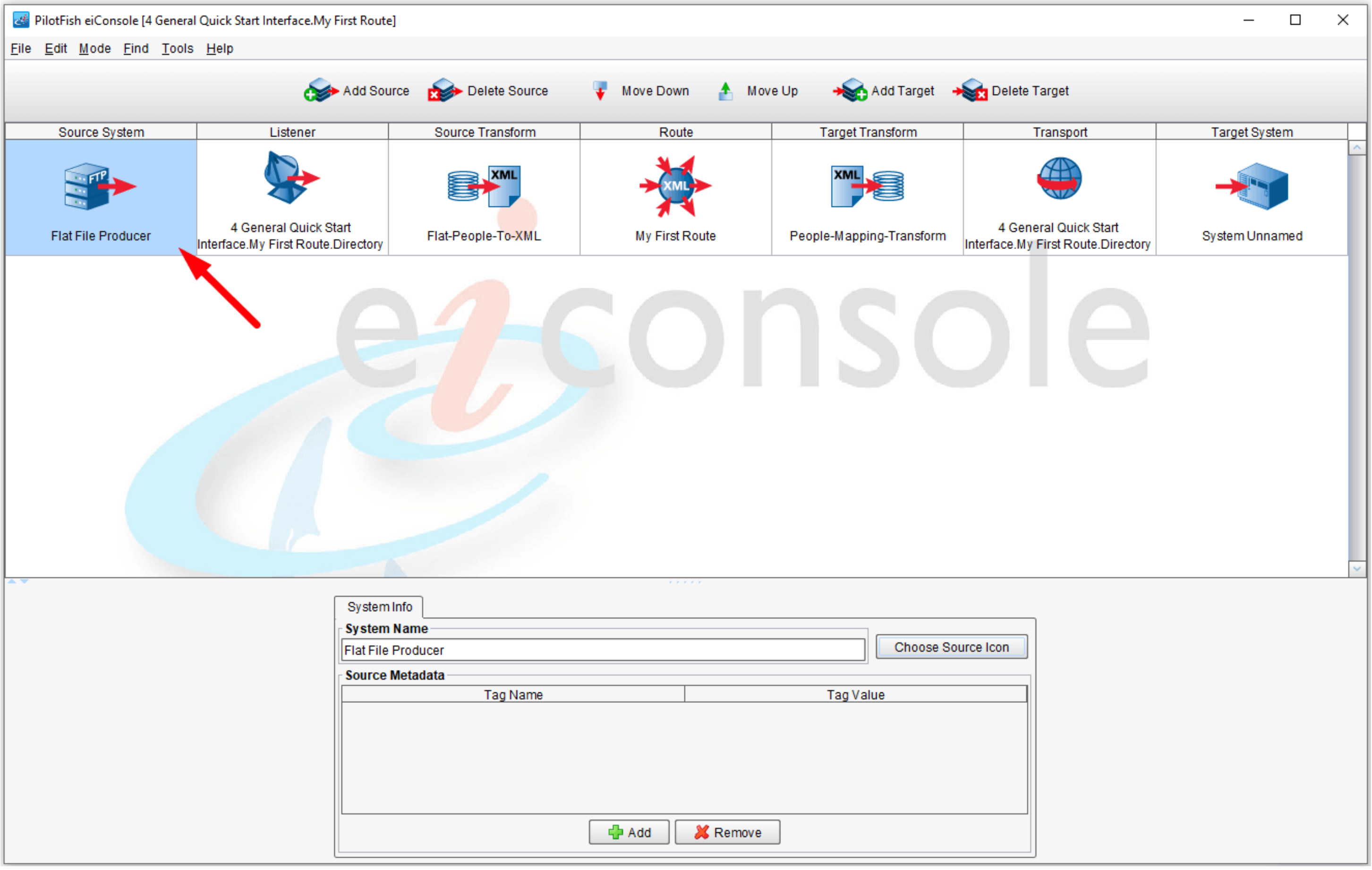


Stage: Source

Metadata

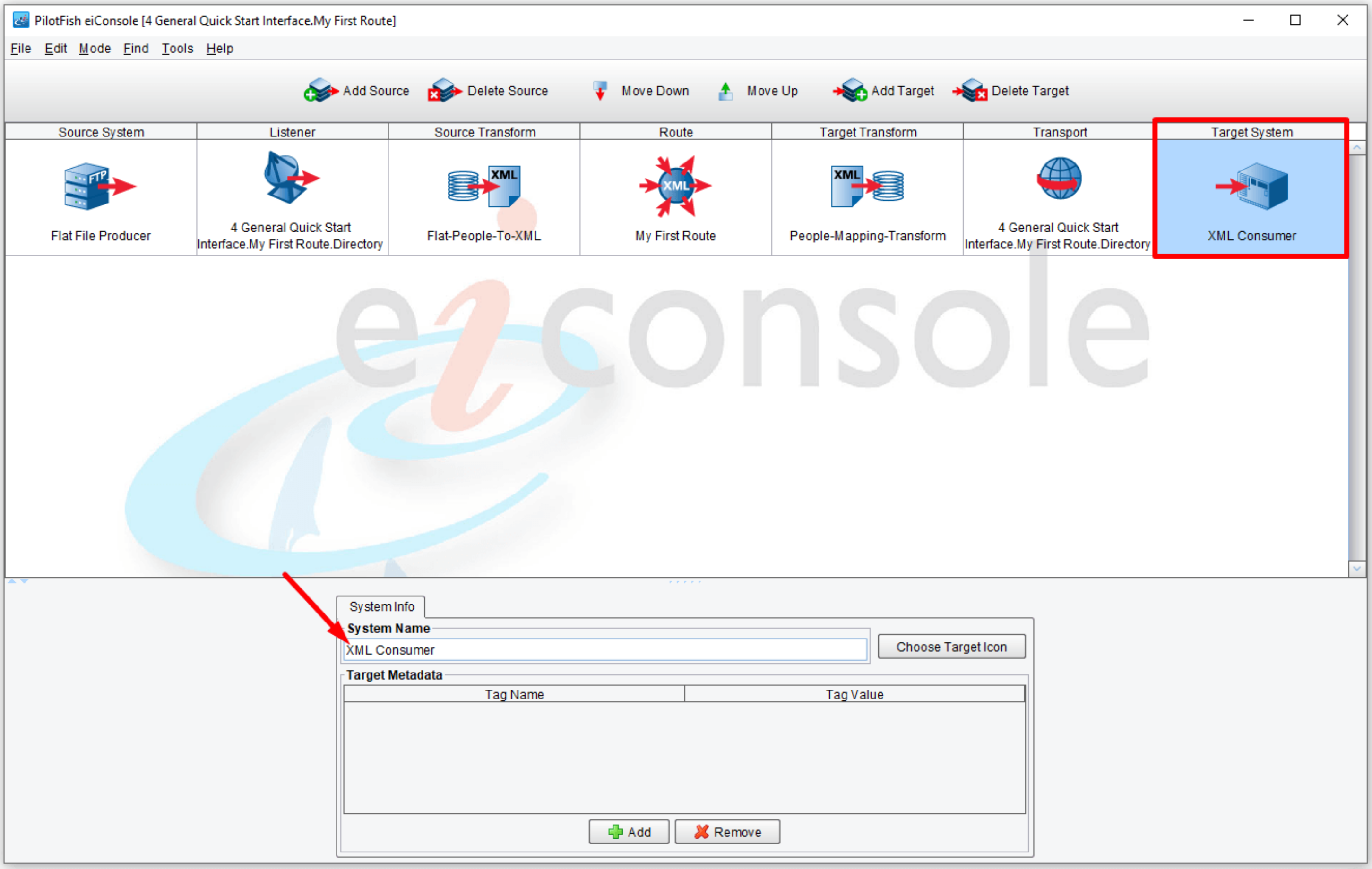
File Name: Ftp_Blue_1.png
Placement: Source
Tags: FTP

There is a number of icons in the **Choose Source Icon** dialog. Choose the **FTP Server** icon and click **OK**.



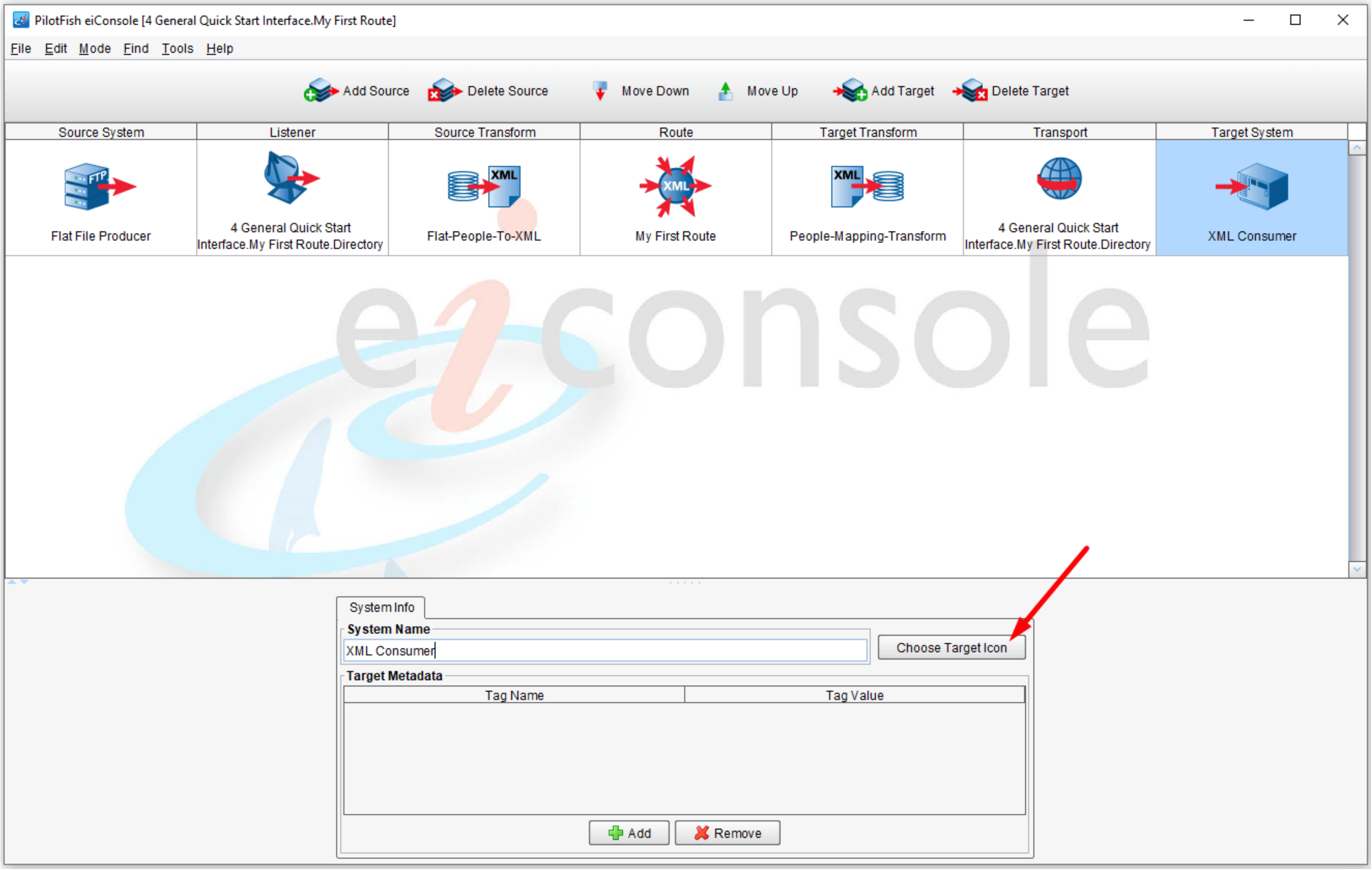
System Name: Flat File Producer

The new icon will appear in the main grid of the eiConsole.



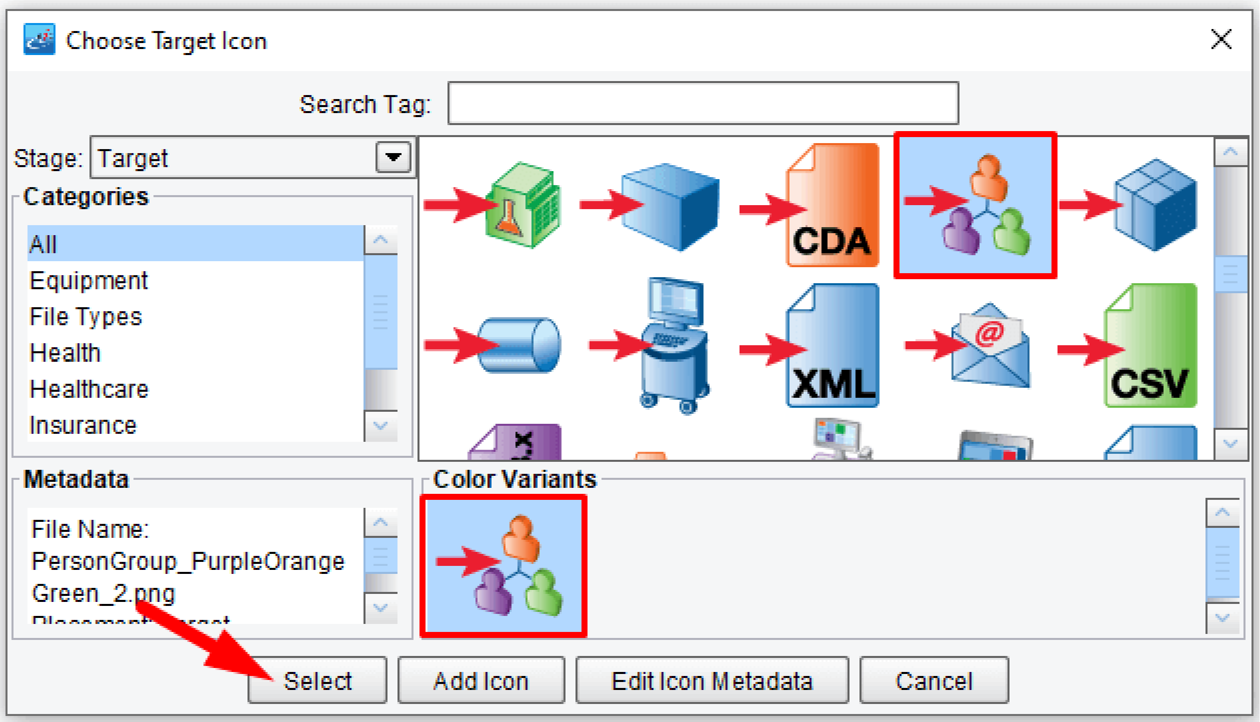
System Name: XML Consumer

Next, click on the **Target System** stage. Type in **"XML Consumer"** in the Target System Name text box.



System Name: XML Consumer

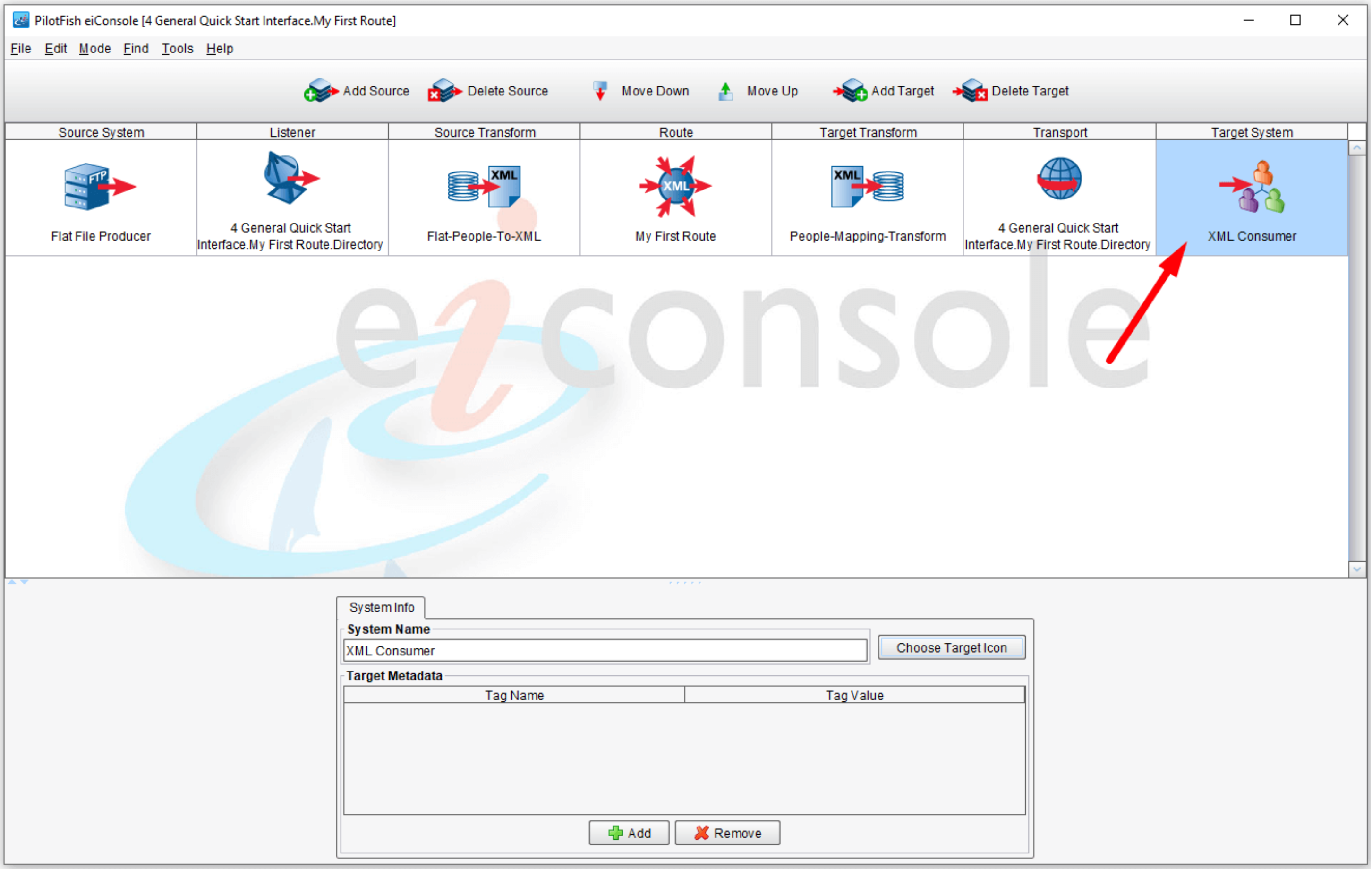
And click on the **Choose Target Icon** button to set a unique icon for your Target.



Stage: Target

File Name: PersonGroup_PurpleOrangeGreen_2.png

You can choose any relevant icon. We'll use the **Group People**.



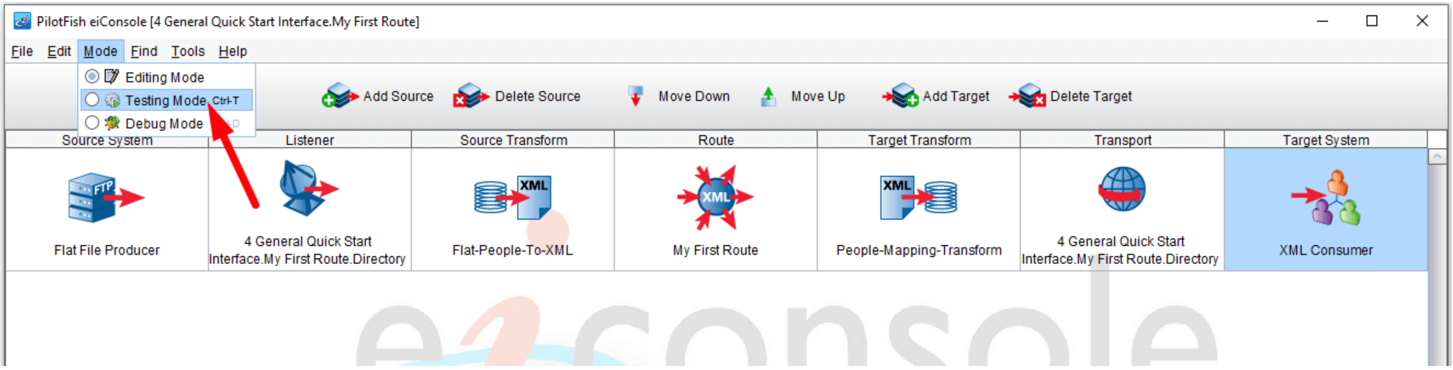
Source System	Listener	Source Transform	Route	Target Transform	Transport	Target System
Flat File Producer	4 General Quick Start Interface.My First Route.Directory	Flat-People-To-XML	My First Route	People-Mapping-Transform	4 General Quick Start Interface.My First Route.Directory	XML Consumer

System Name: XML Consumer

Target Metadata

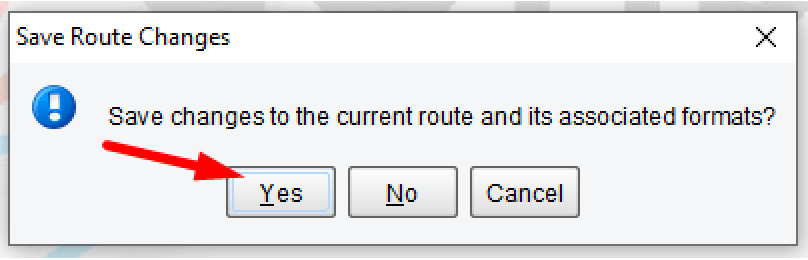
Tag Name	Tag Value

The new icon will appear in the main grid of the eiConsole.



Source System	Listener	Source Transform	Route	Target Transform	Transport	Target System
Flat File Producer	4 General Quick Start Interface.My First Route.Directory	Flat-People-To-XML	My First Route	People-Mapping-Transform	4 General Quick Start Interface.My First Route.Directory	XML Consumer

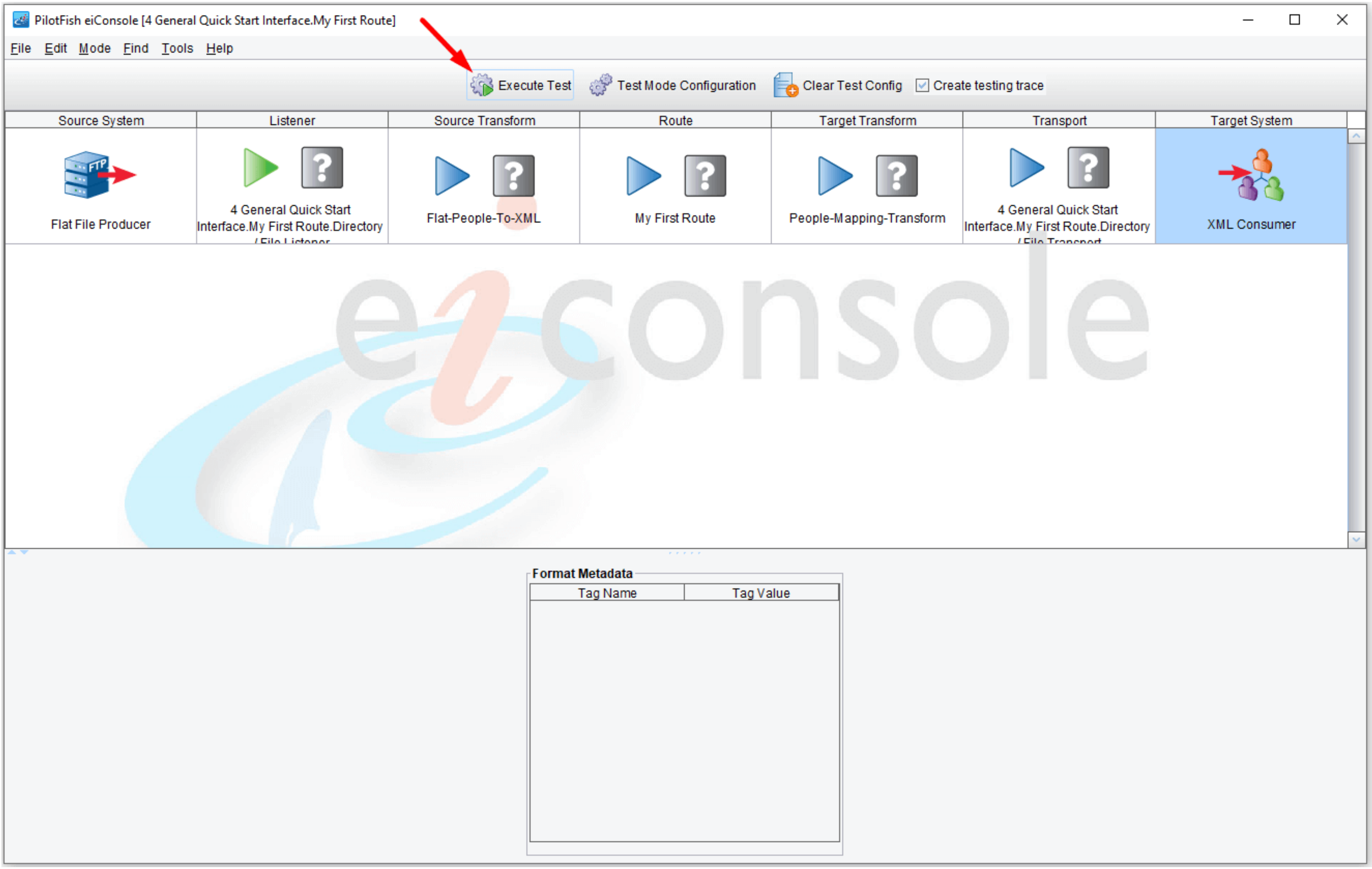
When the basic interface configuration is complete, move on to Testing Mode. From the Mode menu, select **Testing Mode**.



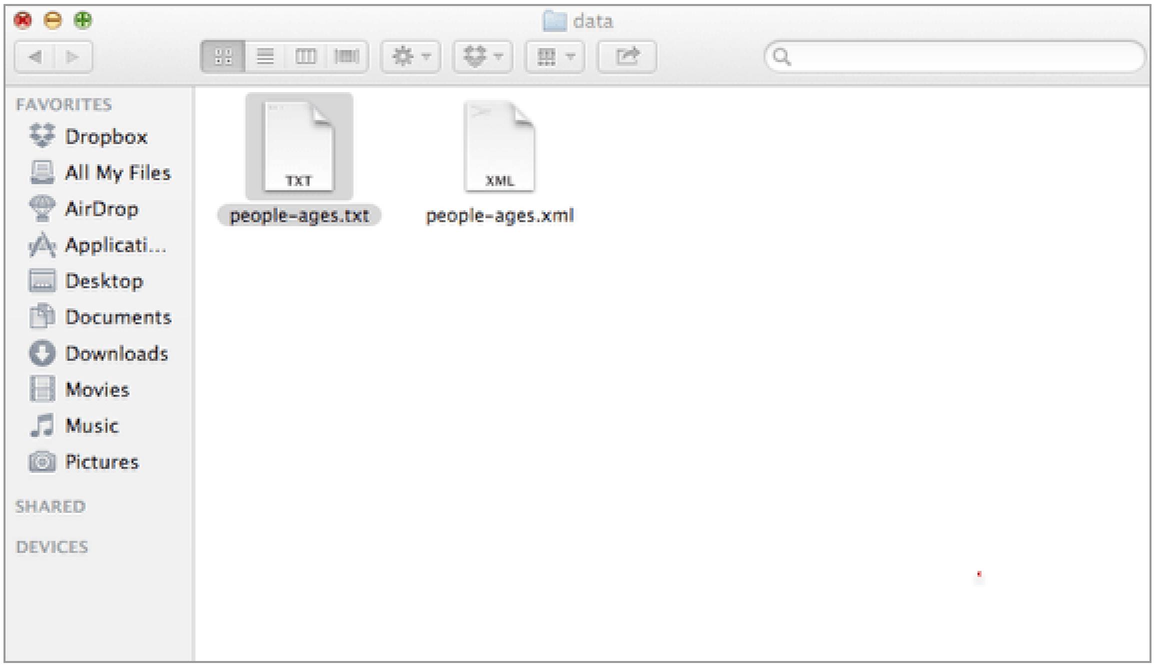
Save changes to the current route and its associated formats?

When the Save Route Changes pop up appears and you are prompted to save the route, click **Yes**.

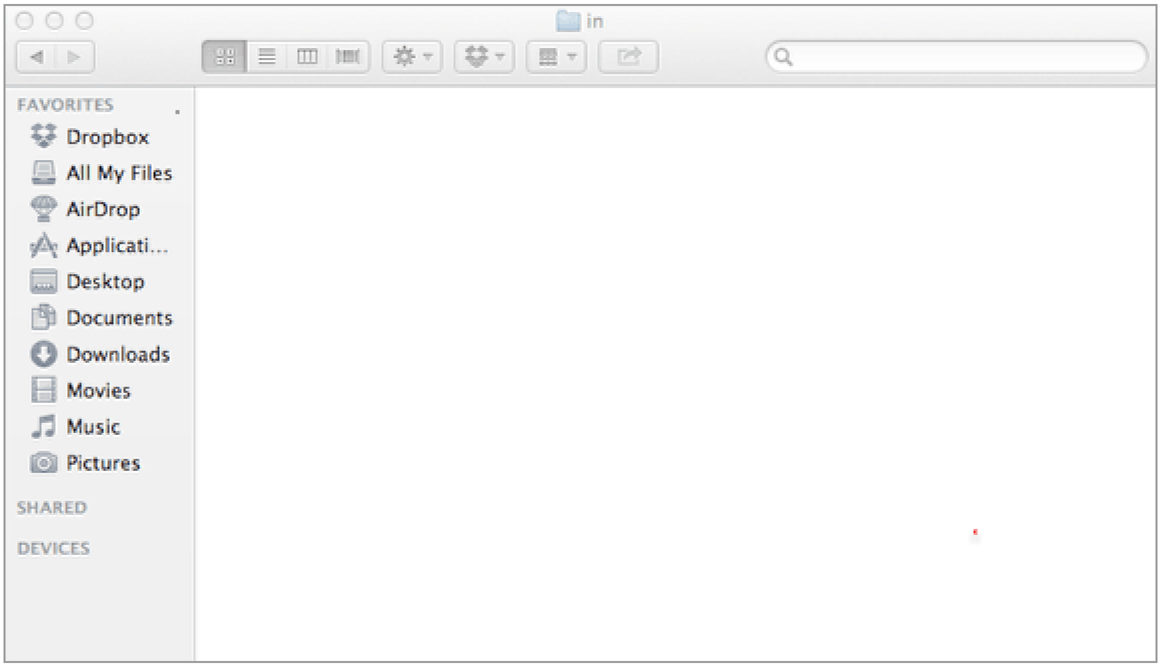
Testing Mode



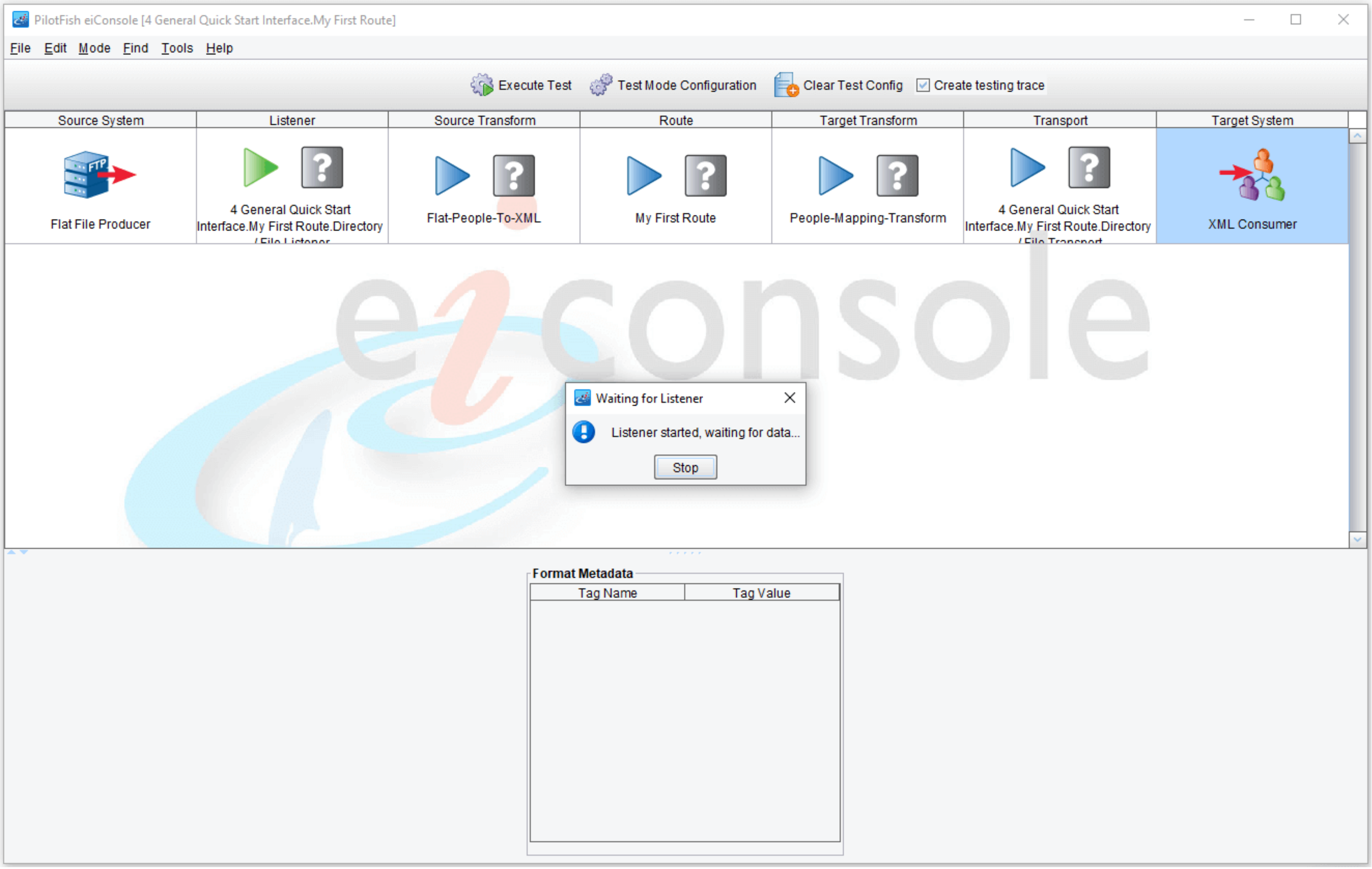
In Testing Mode, you'll see the arrows that indicate the path that the test may take. You can choose to start or end your test at any point. Here, start from the beginning (the Listener stage) and run the test all the way through to the end (the Transport stage).



For the test, navigate to the distribution folder. Double click the **data** folder, and select the **people-ages.txt** file and copy.

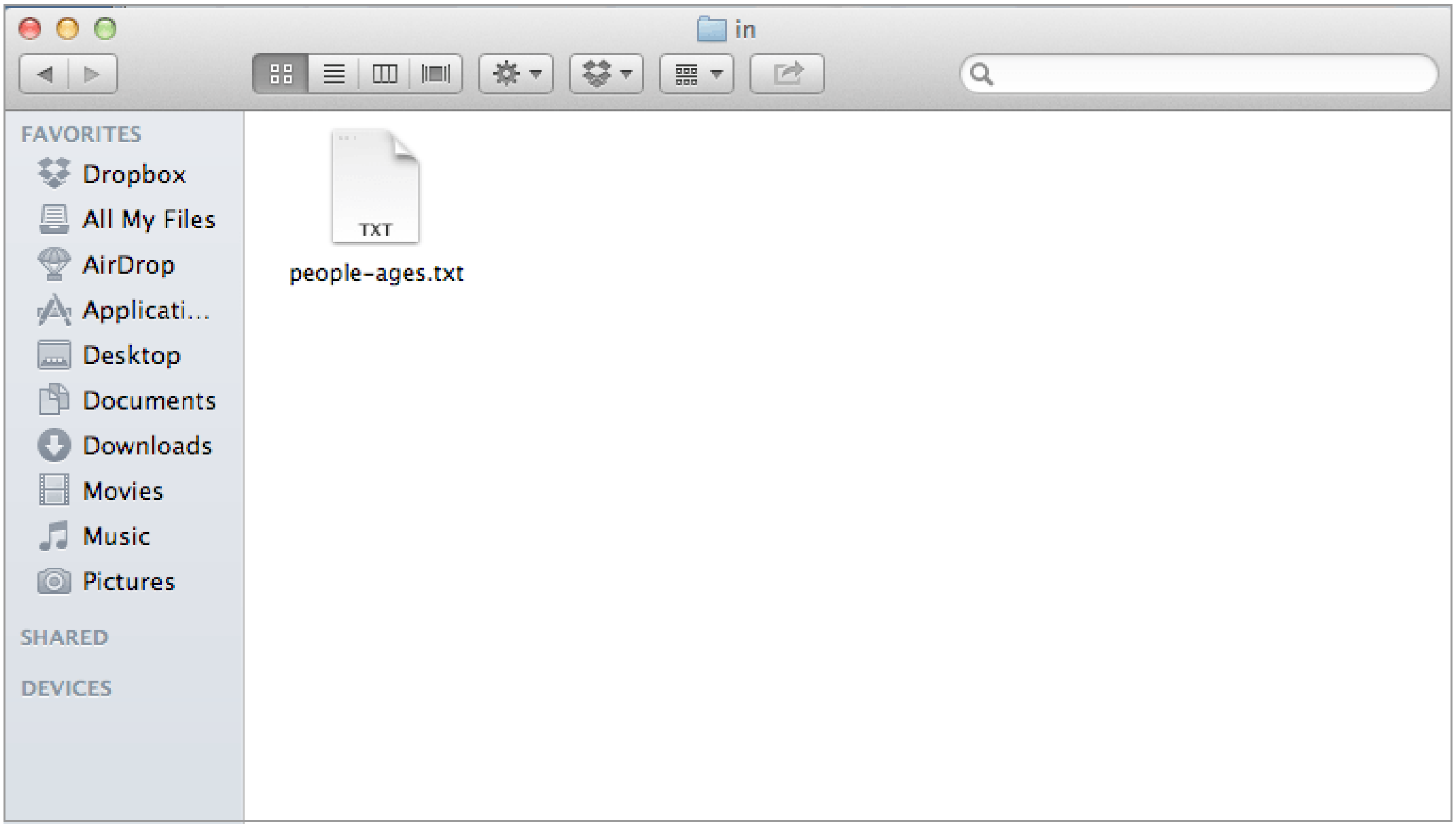


Next, navigate to the **4 General Quick Start Interface** directory or your Working Directory, double click to **Open**. Then double click the **"in"** folder to open.



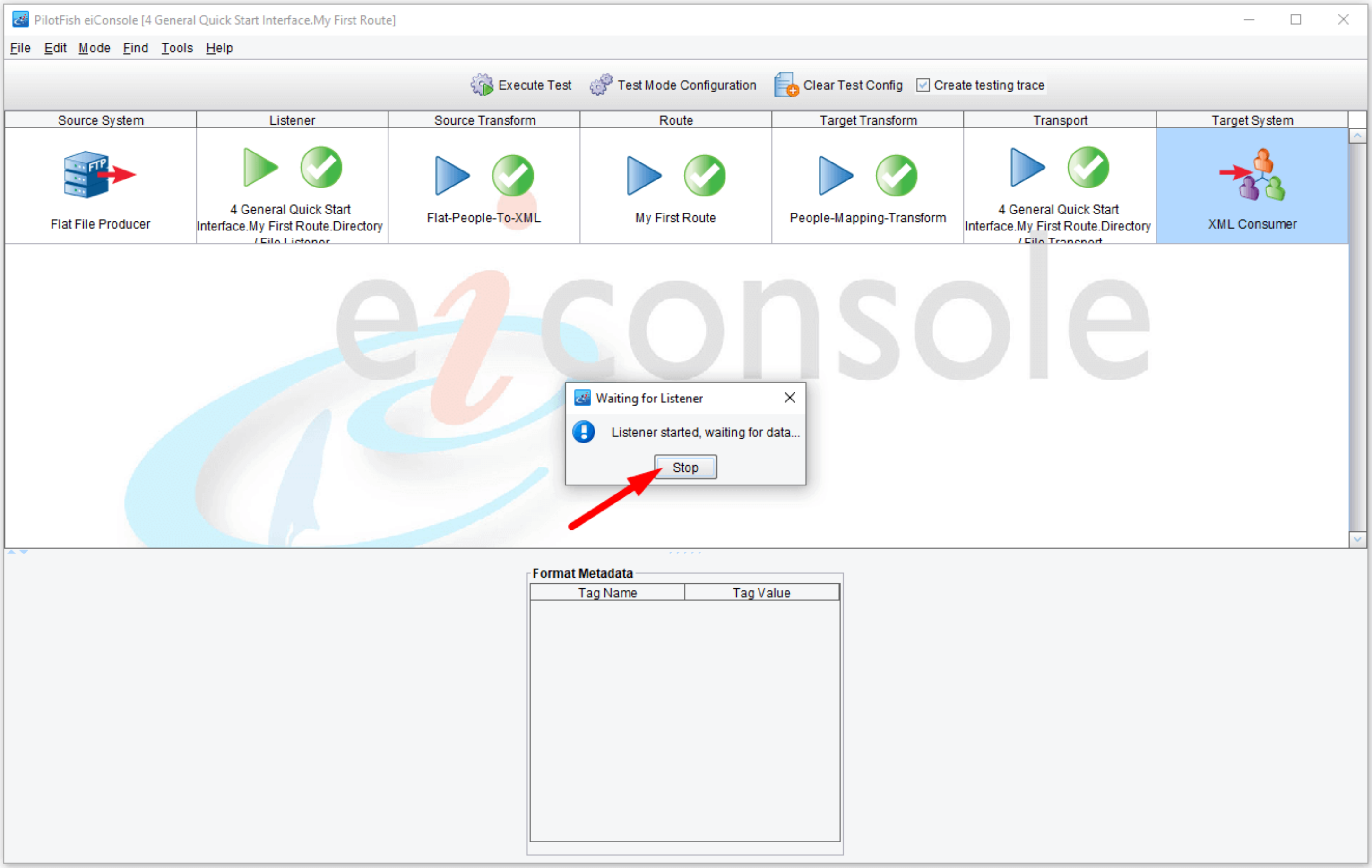
Listener started, waiting for data...

To begin testing click the **Execute Test** button. The Directory Listener will start waiting for data to appear in the input folder.



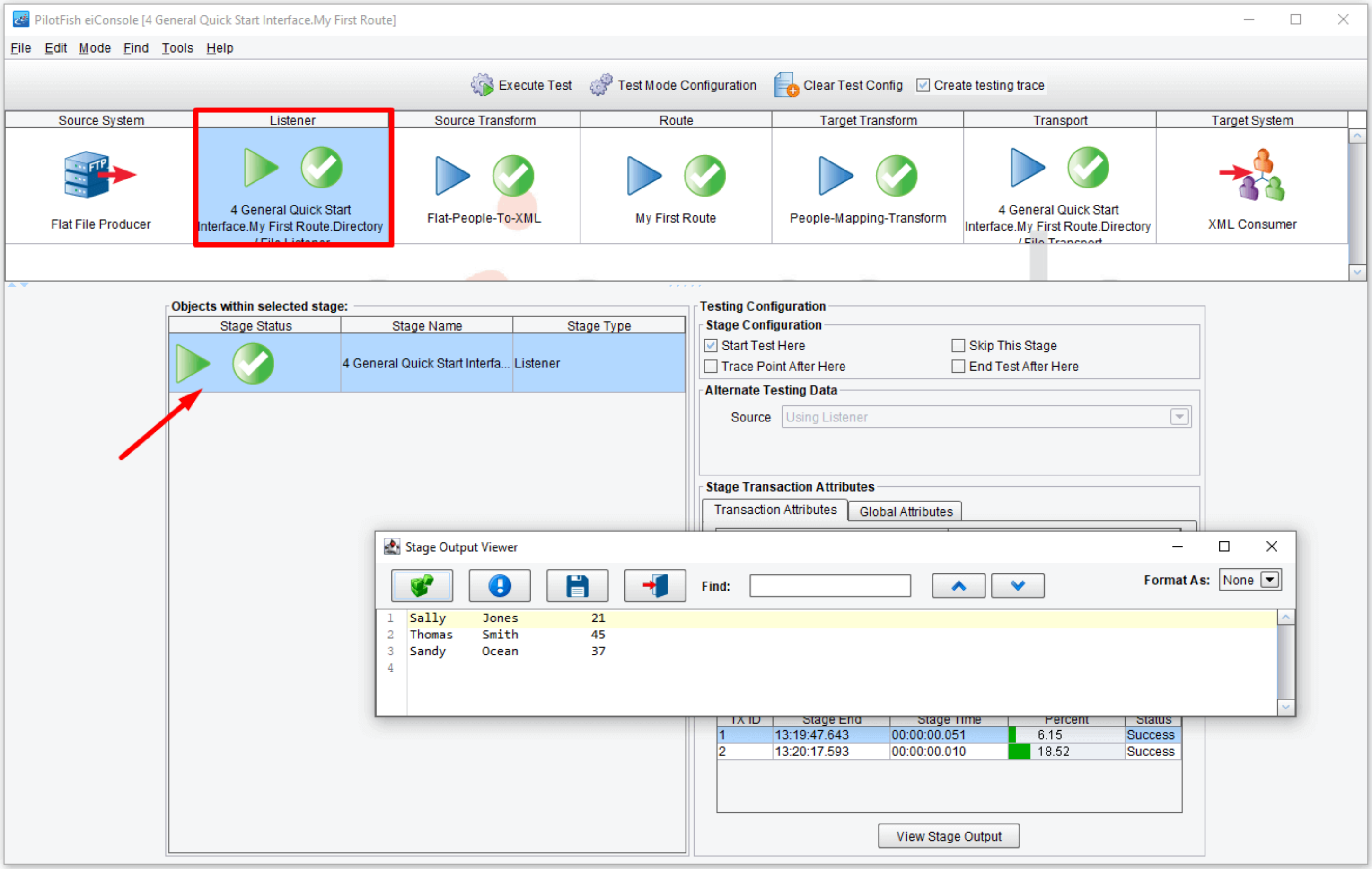
Paste the **"people-ages.txt"** file into the **"in"** folder within the **4 General Quick Start Interface** of your Working Directory.

Within 10 seconds the file should be picked up and processed.



Listener started, waiting for data...

As the file is processed the blue question marks are replaced with green checkmarks. Had there been an error, a red x would appear in one or more of the stage cells. Click on the **Stop** button.



Stage Status	Stage Name	Stage Type
▶ ✓	4 General Quick Start Interfa...	Listener

☑ Start Test Here
☐ Skip This Stage
☐ Trace Point After Here
☐ End Test After Here

Source: Using Listener

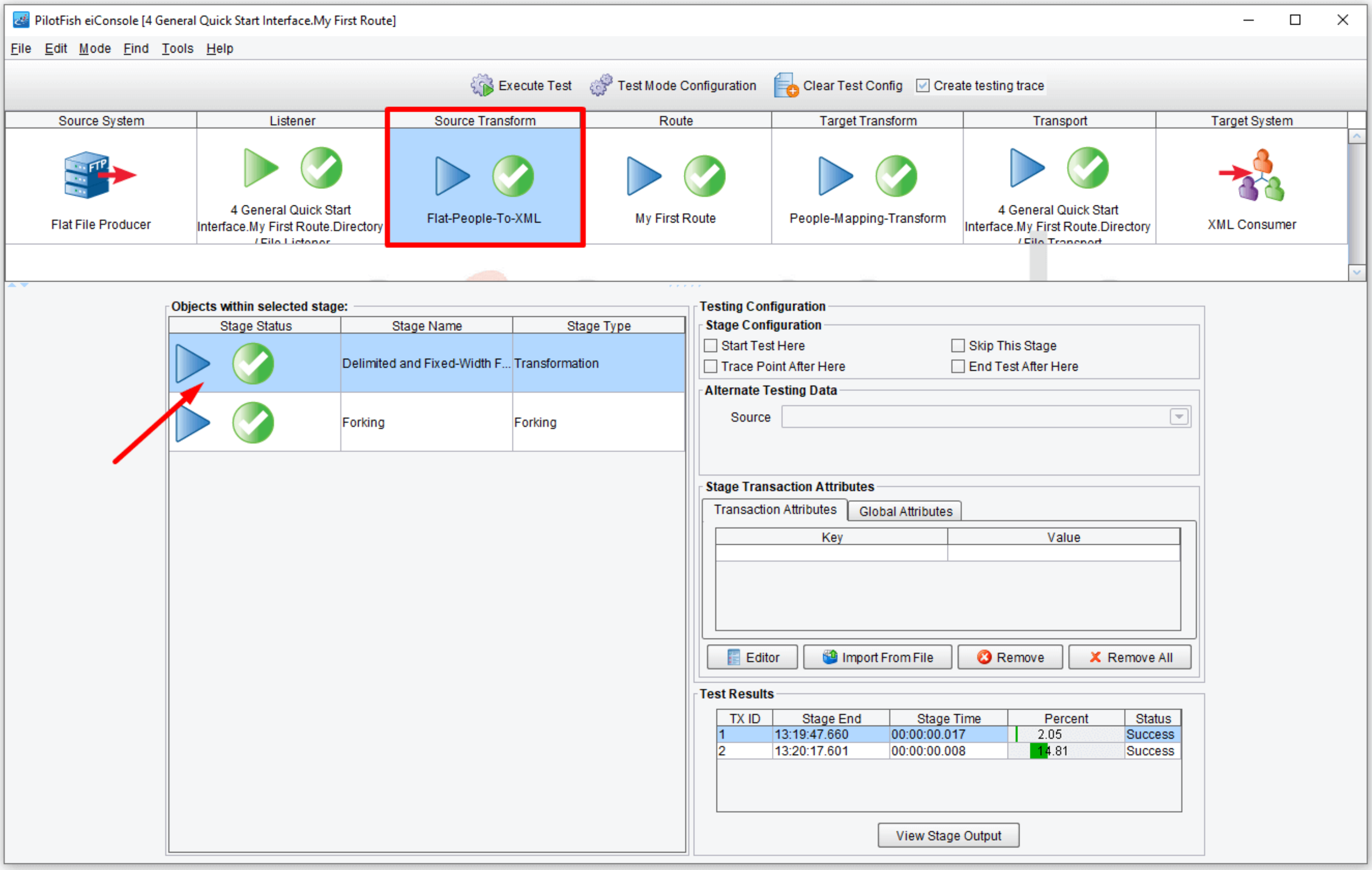
Format As: None

```
Sally     Jones       21
Thomas    Smith       45
Sandy     Ocean       37
```

TX ID	Stage End	Stage Time	Percent	Status
1	13:19:47.643	00:00:00.051	6.15	Success
2	13:20:17.593	00:00:00.010	18.52	Success

You can view the data at each part of the process. When you click on a stage, such as the Listener stage, you can double click on any sub-stage within the Objects Within Selected Stage grid to view the output as it appeared when it exited that stage.

Here, you can see the flat file input.



Objects within selected stage:

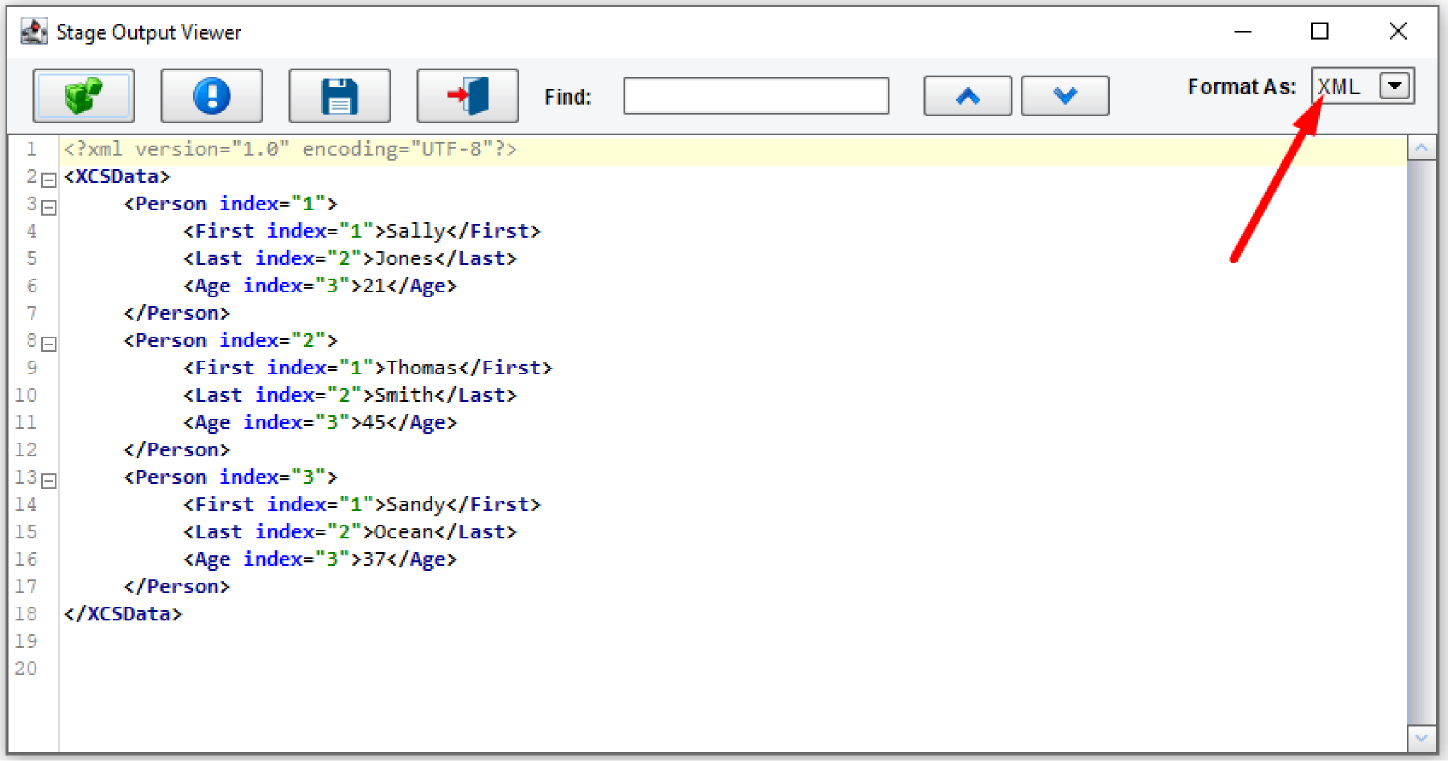
Stage Status	Stage Name	Stage Type
✓	Delimited and Fixed-Width F...	Transformation
✓	Forking	Forking

□ Start Test Here　□ Skip This Stage
□ Trace Point After Here　□ End Test After Here

Test Results

TX ID	Stage End	Stage Time	Percent	Status
1	13:19:47.660	00:00:00.017	2.05	Success
2	13:20:17.601	00:00:00.008	14.81	Success

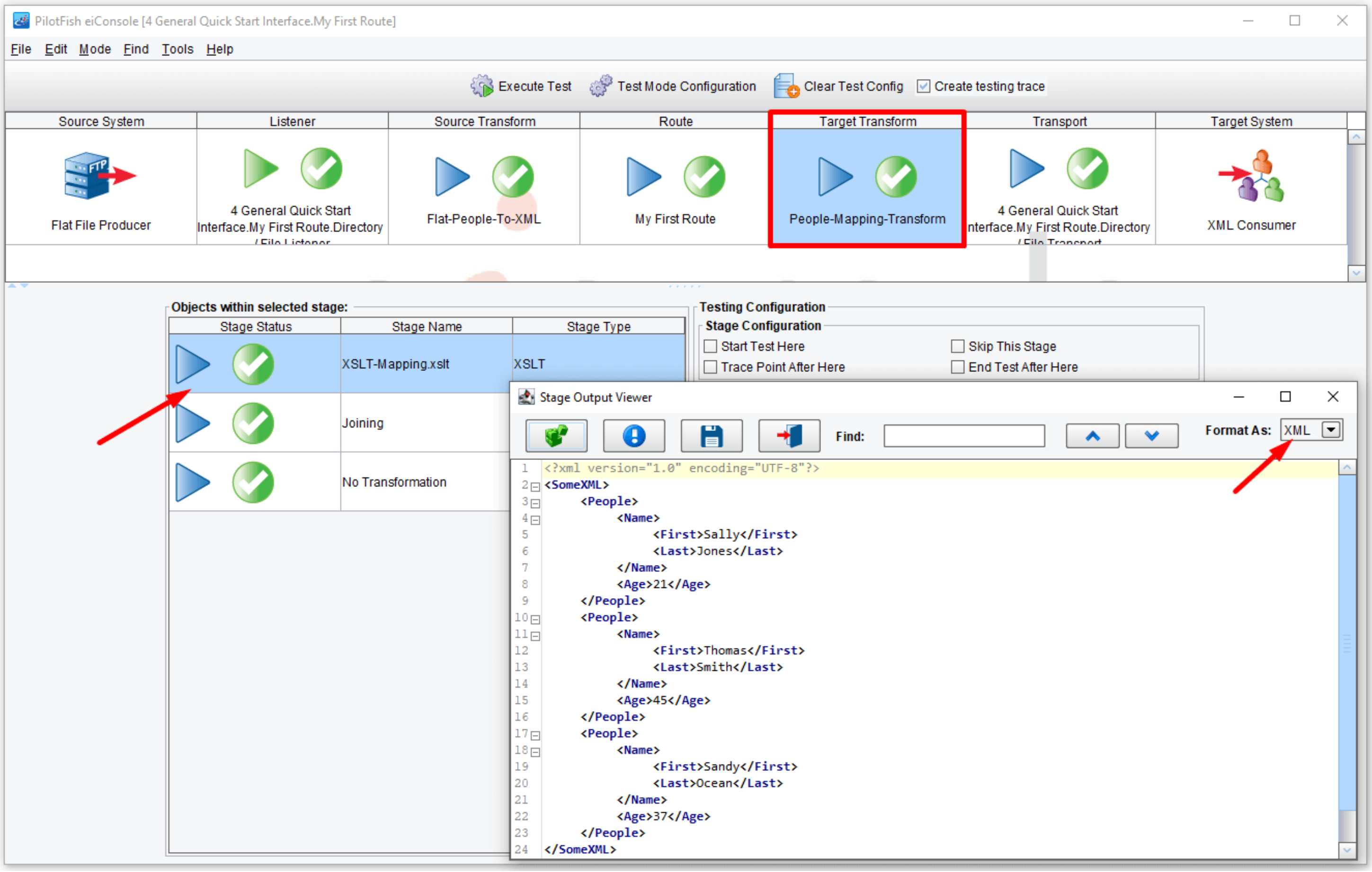
Click on the Source Transformation stage. Double click the **Delimited and Fixed-Width** row to view the output.



Format As: XML

```
<?xml version="1.0" encoding="UTF-8"?>
<XCSData>
    <Person index="1">
        <First index="1">Sally</First>
        <Last index="2">Jones</Last>
        <Age index="3">21</Age>
    </Person>
    <Person index="2">
        <First index="1">Thomas</First>
        <Last index="2">Smith</Last>
        <Age index="3">45</Age>
    </Person>
    <Person index="3">
        <First index="1">Sandy</First>
        <Last index="2">Ocean</Last>
        <Age index="3">37</Age>
    </Person>
</XCSData>
```

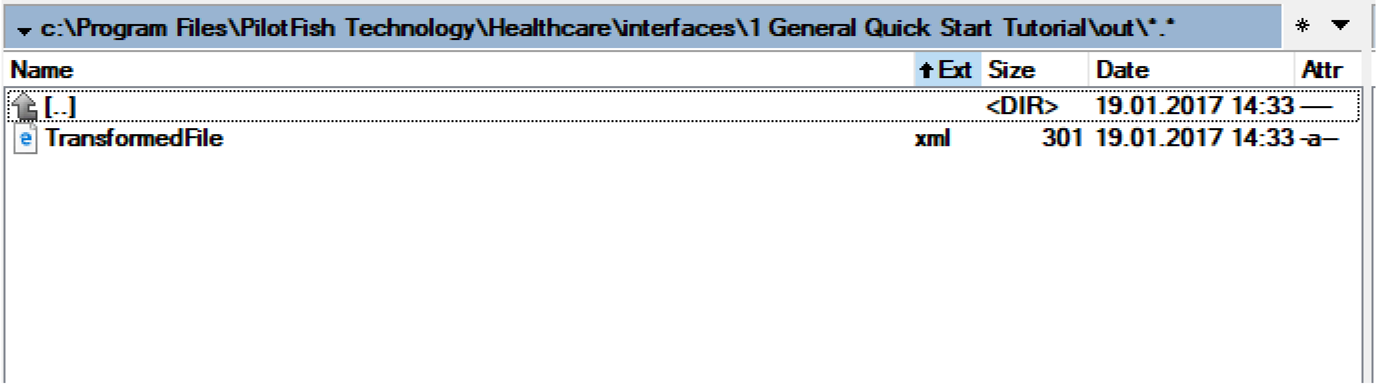
Here, you can see the data after it's been converted to XML. Choose the XML format.



Stage Status	Stage Name	Stage Type
✓	XSLT-Mapping.xslt	XSLT
✓	Joining	
✓	No Transformation	

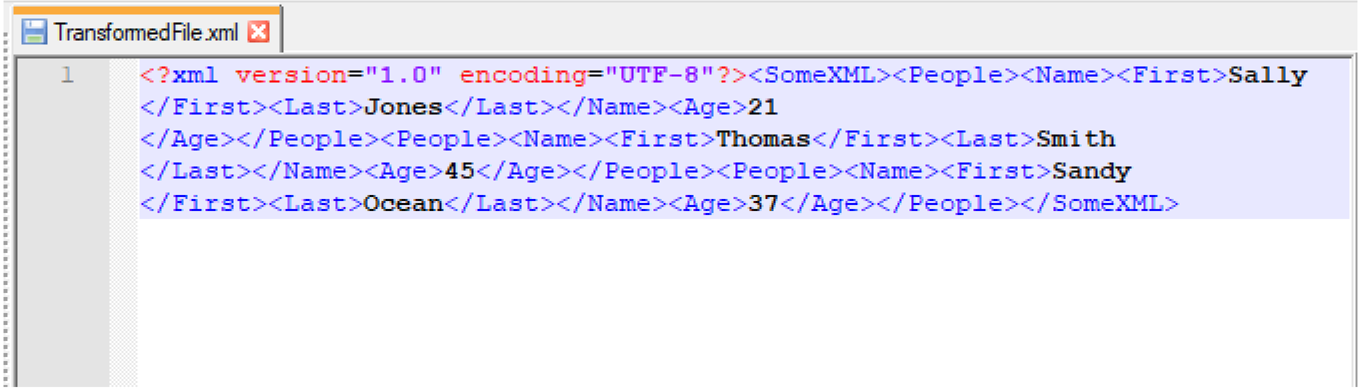
```
<?xml version="1.0" encoding="UTF-8"?>
<SomeXML>
    <People>
        <Name>
            <First>Sally</First>
            <Last>Jones</Last>
        </Name>
        <Age>21</Age>
    </People>
    <People>
        <Name>
            <First>Thomas</First>
            <Last>Smith</Last>
        </Name>
        <Age>45</Age>
    </People>
    <People>
        <Name>
            <First>Sandy</First>
            <Last>Ocean</Last>
        </Name>
        <Age>37</Age>
    </People>
</SomeXML>
```

The data was routed along towards our one Transport. If you click on the Target Transformation stage, you can double click the **XSLT** stage to see the transformed data. It will be in the SomeXML structure with People tag, Name sub-tag, with First and Last name tags as children.


c:\Program Files\PilotFish Technology\Healthcare\interfaces\1 General Quick Start Tutorial\out\*.*

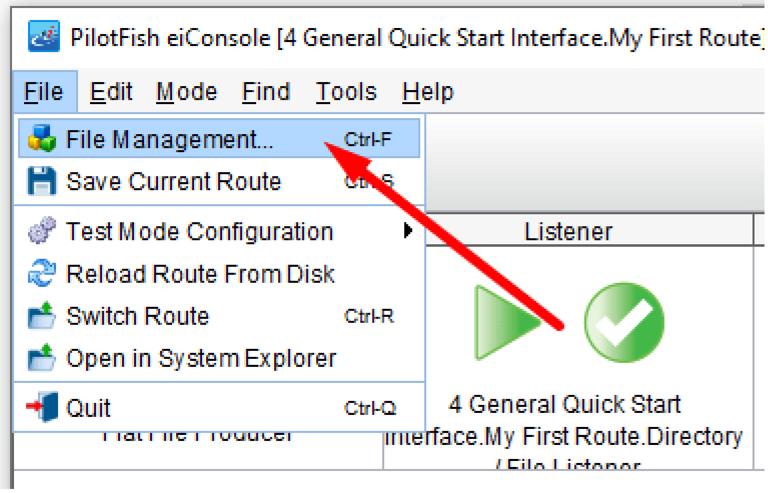
Name	↑Ext	Size	Date	Attr
[..]		<DIR>	19.01.2017 14:33	—
TransformedFile	xml	301	19.01.2017 14:33	-a—

Finally, the Transport deposits the **TransformedFile.xml** file in the selected Out folder.



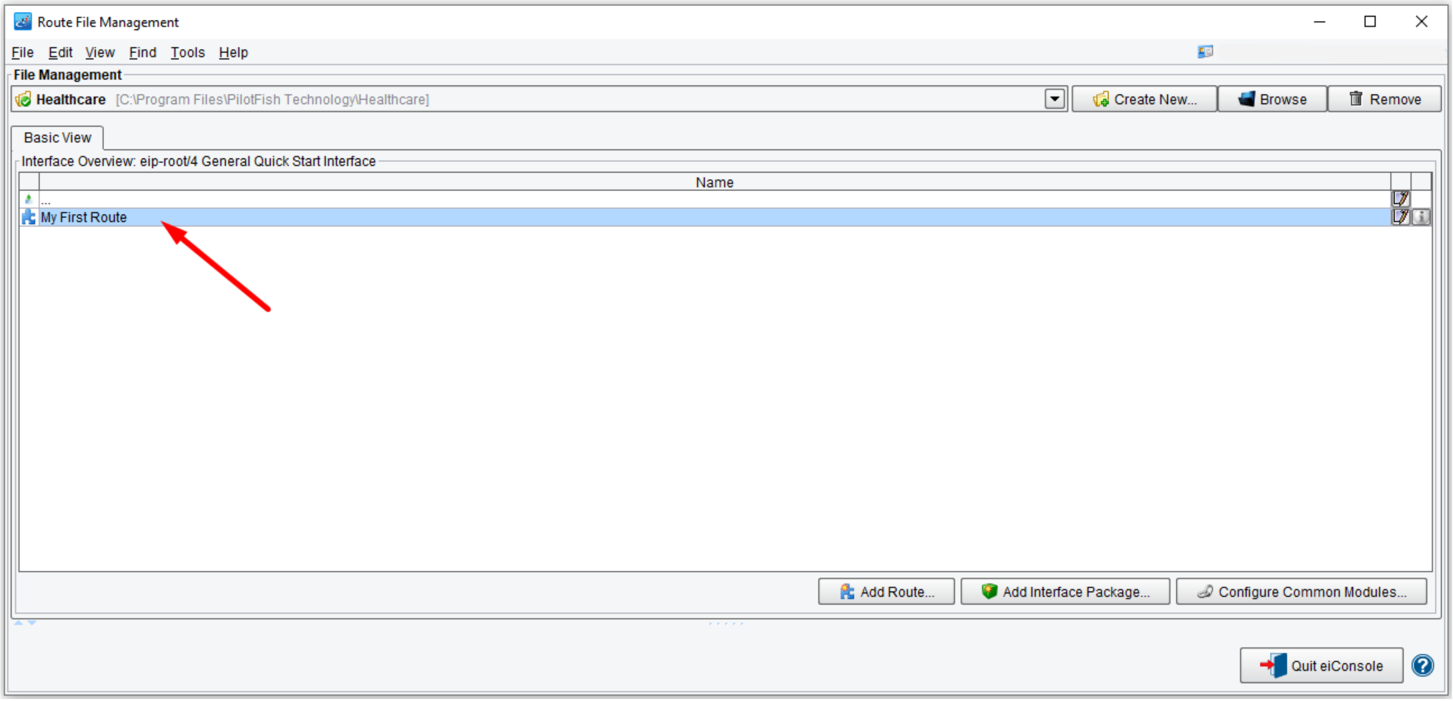
```
<?xml version="1.0" encoding="UTF-8"?><SomeXML><People><Name><First>Sally
</First><Last>Jones</Last></Name><Age>21
</Age></People><People><Name><First>Thomas</First><Last>Smith
</Last></Name><Age>45</Age></People><People><Name><First>Sandy
</First><Last>Ocean</Last></Name><Age>37</Age></People></SomeXML>
```

Double click the file to open. Notice the file named **TransformFile.xml** has been written to the disc.


PilotFish eiConsole [4 General Quick Start Interface.My First Route]
File
Edit
Mode
Find
Tools
Help
File Management... Ctrl-F
Save Current Route Ctrl-S
Test Mode Configuration
Reload Route From Disk
Switch Route Ctrl-R
Open in System Explorer
Quit Ctrl-Q
Listener
4 General Quick Start Interface.My First Route.Directory / File Listener
Flat File Producer

Lastly, go to the File menu and select **File Management** to return to the Route File Management window.

Deploying an Interface



Healthcare [C:\Program Files\PilotFish Technology\Healthcare]

Interface Overview: eip-root/4 General Quick Start Interface

When the Route File Management window opens, you'll see the icon next to **My-First-Route** is now blue indicating it is a fully configured interface. Typically, if you wanted to deploy this interface into production, you'd connect to your eiPlatform server and drag & drop the interface into the eiPlatform server window to deploy.

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

Now you have completed the Quick Start Tutorial and have learned the basic topology of the eiConsole you can move on to more complex tutorials:

- **General Getting Started Tutorial**

Or choose an industry-specific tutorial:

- **Healthcare Getting Started Tutorial**
- **HR-XML Getting Started Tutorial**
- **Insurance Getting Started Tutorials (LAH & PCS)**
- **OpenTravel Getting Started Tutorial**

The industry-specific tutorials will teach you how to leverage the many standards-specific features and components that make the eiConsole a powerful tool for interface building, management and maintenance for each industry. Or you can also browse through the topics in **General User Reference Level I-IV** for more advanced interface configuration documentation and tutorials.