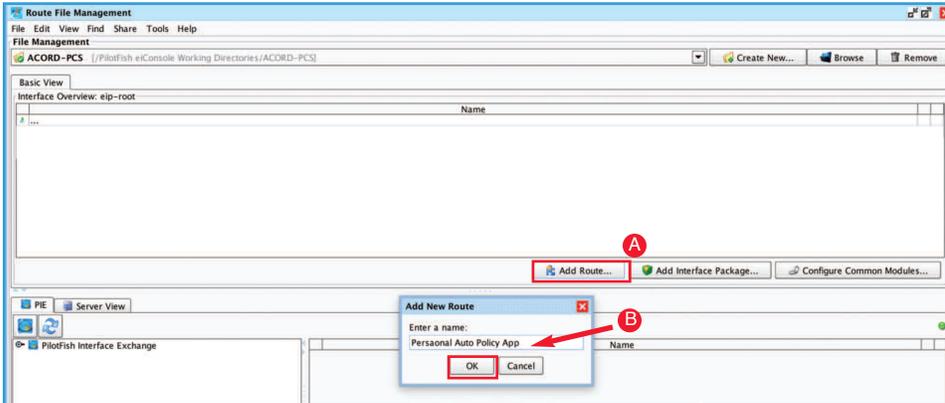


eiconsole

for Insurance The IDE for Graphically Configuring Interfaces

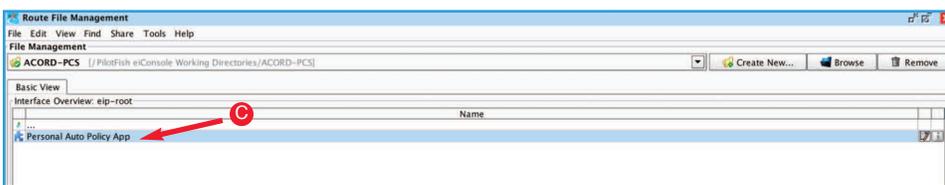
10 Easy Steps to Building an Interface

Follow the PilotFish Graphical Automated Interface Assembly Line to Quickly Configure, Test and Deploy an ACORD AL3 Interface



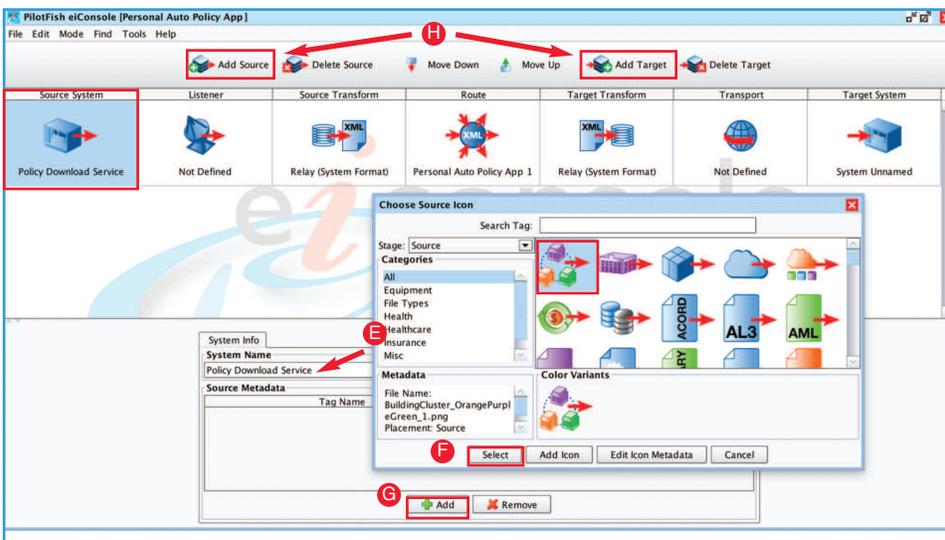
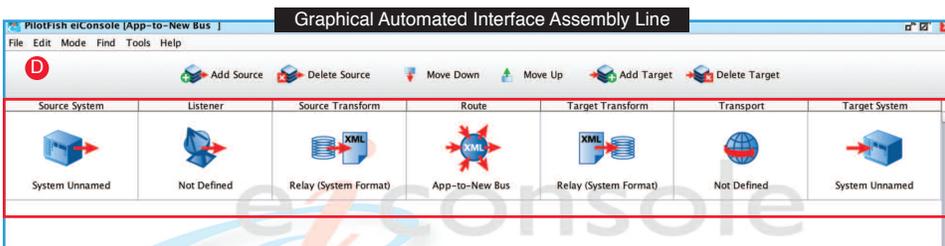
Step 1: Create a new Route/Interface

When you open the eiConsole, the first screen you see is the Route File Management screen. This is where the interface configuration files are managed. PilotFish configurations are divided into two levels. A single connection between a source and target is a route and a collection of routes working together is referred to as an interface. To create a new Route, click **Add Route** (A) and name it "Personal Auto Policy App" (B) and click **OK**.



Step 2: Build the Route

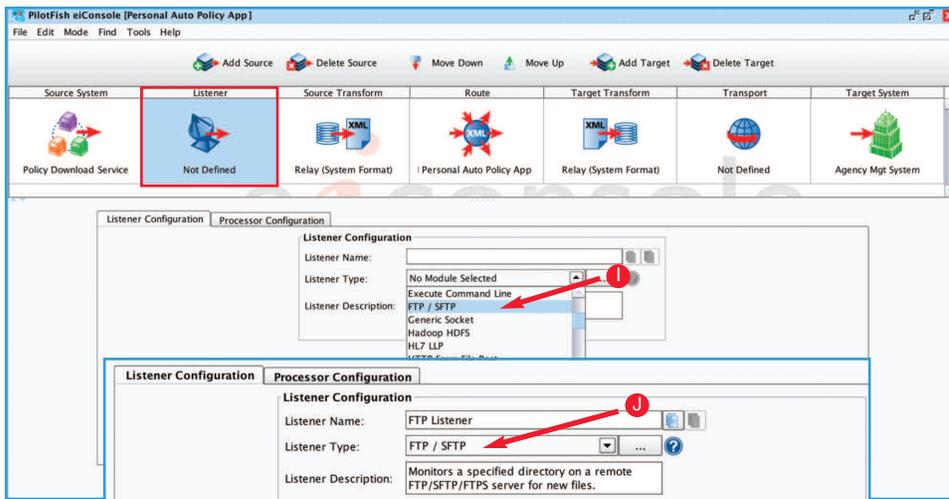
Double click your **Route** (C) to open the eiConsole's main route grid window. PilotFish routes are built in an assembly line fashion. The **Graphical Automated Interface Assembly Line** (D) consists of 7 stages which are laid out in the grid at the top of the screen. These stages handle processing the flow of data from the source to the target systems. Regardless of the type of integration being done, the process is always the same. There is also no limit to the number of source and target systems that can be linked in this way.



Step 3: Identify Your Source and Target System and Select Representative Icons

Select the **Source System** stage to name your source system. For reference, name your Source and Target Systems based on what they are supposed to represent. Next to the **System Name** field type in "Policy Download Service" (E) and select Choose Source Icon.

When the **Choose Source Icon** pop-up opens, select a representative icon from a library of hundreds of icons (or add your own). Click **Select** (F) to make your choice. Type in any metadata you would like to include and click **Add** (G). Follow the same process for the Target System. If you would like to add more Source or Target Systems, click the **Add Source** or **Add Target** (H) buttons above the grid and follow the previous steps to name the systems and to select the appropriate icons.

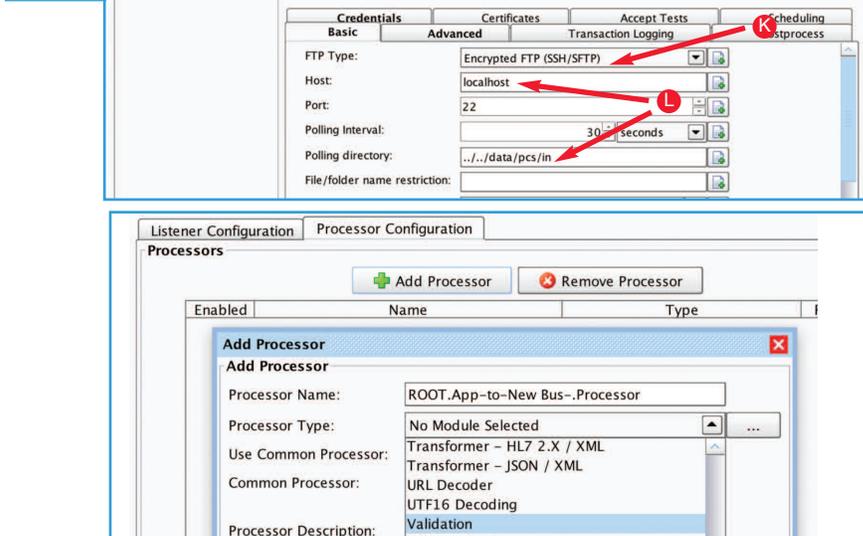


Step 4: Choose the Listener and Any Processors Required

Next, you need to establish connectivity with the Source System. PilotFish retrieves data from the Source System using a component called a Listener. This Listener communicates with the Source System at highly configurable intervals, retrieves the data and starts a PilotFish transaction. Select the **Listener** stage to open the Listener configuration panel. PilotFish comes pre-bundled with several dozen Listeners, capable of handling virtually any connectivity option that you might need and is easily extended using our Open API.

For this interface, select the **FTP/SFTP** (I) Listener. After the panel opens, change the default Listener name to "FTP Listener" (J). For the FTP Type select **Encrypted (SSH/SFTP)** (K). You'll also need to provide connectivity information (host name and the polling directory) to search for files (L).

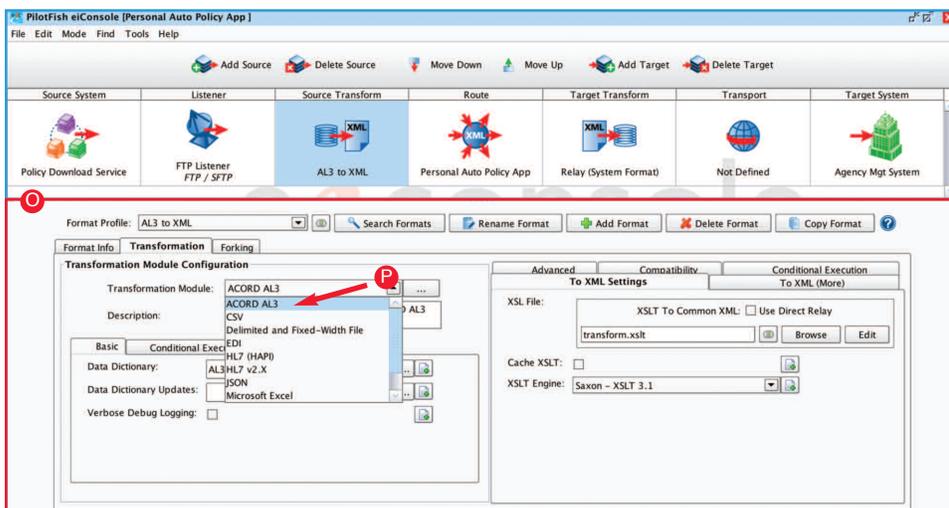
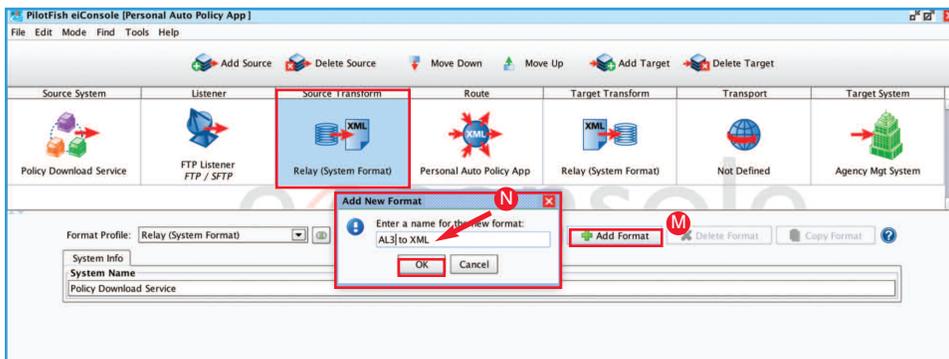
After specifying a Listener, Processors can be queued up to perform data manipulation. Processors are accessed by clicking the **Processor Configuration** tab. Processors perform general work over the data stream, either directly after it is received or immediately before it is sent. You can use a Processor to add decryption, perform authentication or Validation. Choose from nearly 100 processors or again add your own by using our Open API. For this interface, no Processors are needed – so we move on to the next stage.

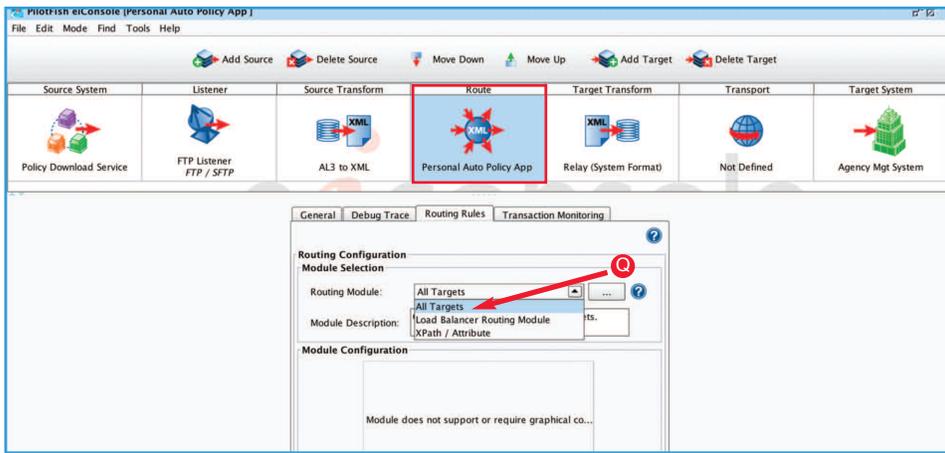


Step 5: Transform the Source Data to a Common Standard

For data transformation, PilotFish works primarily with XML as it represents an easy-to-transform, common standard for working with data. Once data is received by the application, it goes through this conversion process in two steps. First, it goes through an automated Transformer to convert the raw content to XML (if required). The eiConsole includes pre-built modules that can be set up through a simple graphical configuration. They take a wide range of common data types and generate an XML representation of them.

First, you'll need to transform the AL3 message to XML. Select the **Source Transform** stage and click the **Add Format** (M) button. In the pop up panel, name your format "AL3 to XML" (N) and click **OK**. The **Transformation Module** and **XSLT Configuration** panel (O) open. 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. Select PilotFish's ACORD AL3 Transformer (P) from the drop down. (Other Transformers convert CSV, Fixed Width / Delimited and Fixed Width Files, JSON, x12 EDI, XLS / XLSX, ACORD, Name Value and more.)

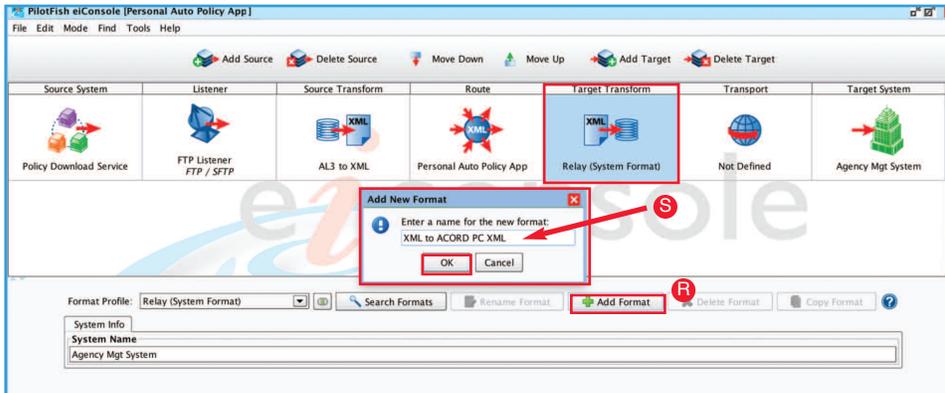




Step 6: Configure the Routing Module

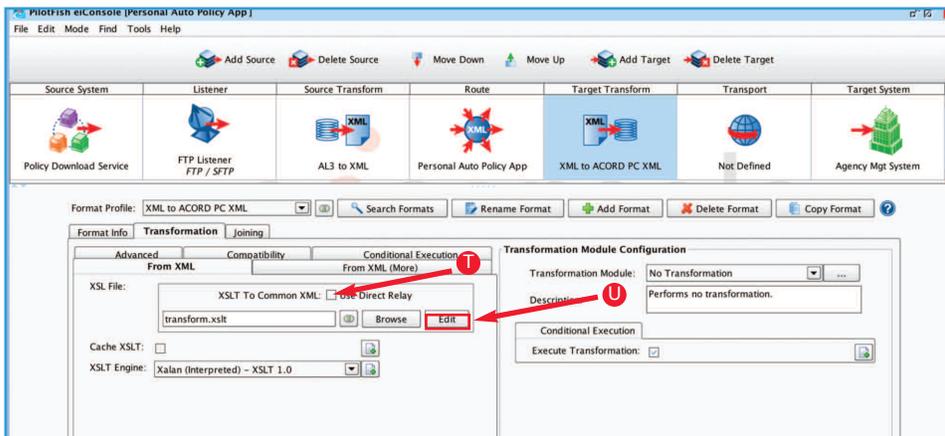
Select the **Route** stage. This is where you can maintain general metadata describing the Route, specify routing rules and configure Transaction Monitoring. When the panel appears, select the **Routing Rules** tab. This enables you to route messages to the appropriate target or targets based on the content of the message. From the drop-down, select **All Targets** (Q).

*Note: The **Transaction Monitoring** tab lets you customize the error notification system used by the interface when in production. This proactive alerting supplements the traditional, passive logging and audit trail supported and configured in the eiPlatform runtime.*

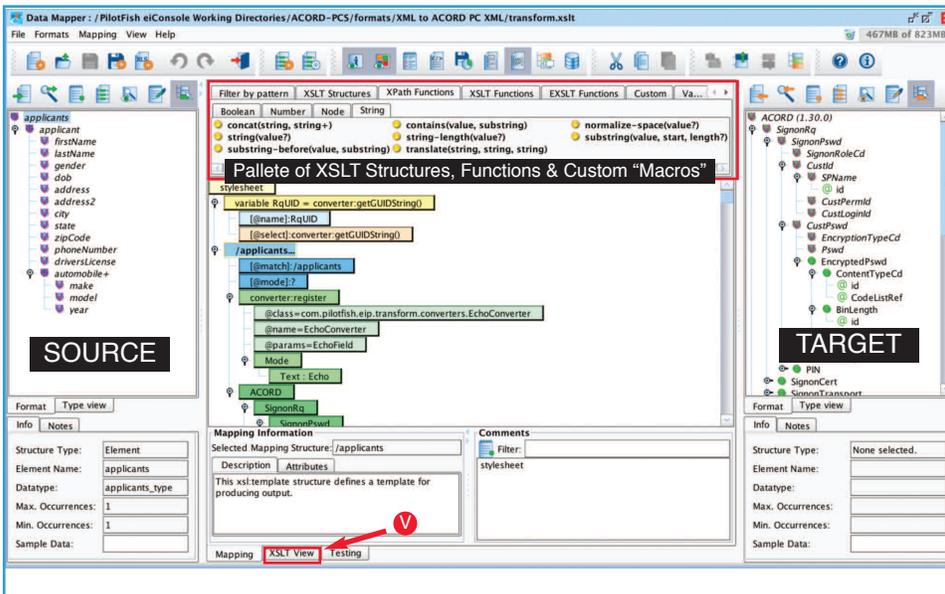


Step 7: Transform the Data for the Target

Next, select the **Target Transform** stage. This is where you will convert your new XML representation of XML into an ACORD PC XML. You'll use the same two-step process you used for the Source Transform stage, only in reverse. First, click the **Add Format** (R) button and then enter "XML to ACORD PC XML" (S) in the dialog and click **OK**. This opens the transformation panel.



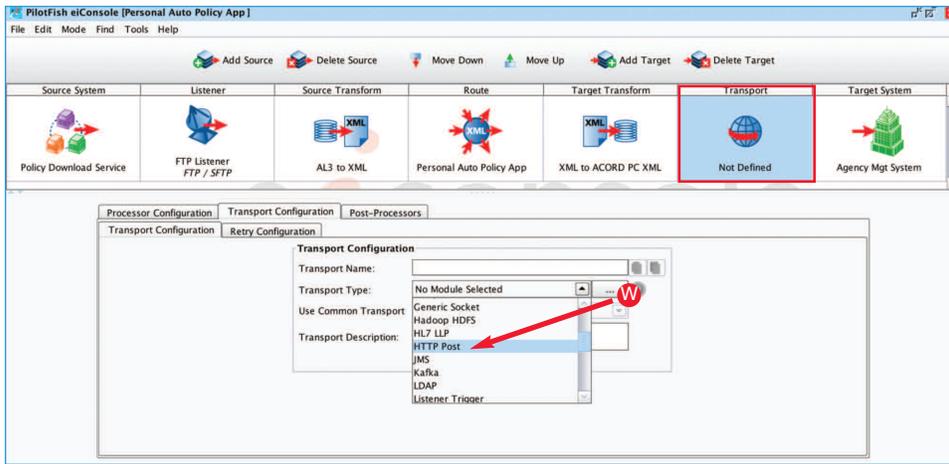
In the Transformation Module Configuration panel (on the left hand side in the XSLT Configuration panel), uncheck the **Use Direct Relay checkbox** (T). This enables the XSLT Transformation, which you can configure by clicking the **Edit** (U) button, to open the Data Mapper.



The Data Mapper is a graphical editor within the eiConsole used to generate the XSLT transformations that transform any data format to any other. The pane on the left shows the **Source** format, the pane on the right shows the **Target** format and the pane in the middle represents the relationship between the Source and the Target. Choose a Format Reader to load in Source and Target data. Over 2 dozen are built-in. Relating Source formats to Target formats is accomplished by simply dragging the corresponding fields from the Source and the Target to the map in the middle.

A "computationally complete" **Palette of XSLT Structures, Functions and Custom "Macros"** can be used to speed the development of high quality transformations. Selecting the **XSLT View** (V) tab lets users work in XSLT. While it's possible to build an entire mapping from scratch, it's often faster to drag & drop source values onto a sample target message or transaction.

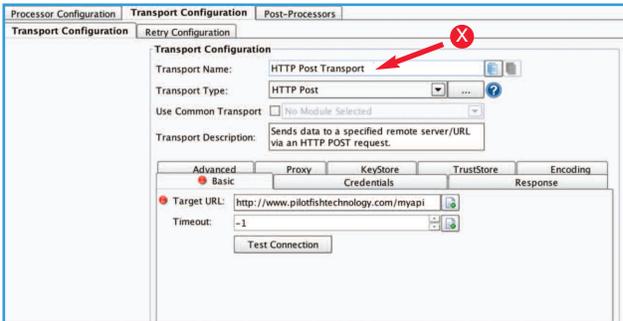
Users can also automatically create a skeleton for a transformation with sample source and target files and import vendor-specific transaction samples to allow easier Data Mapping with non-standard formats.



Step 8: Configure the Transport

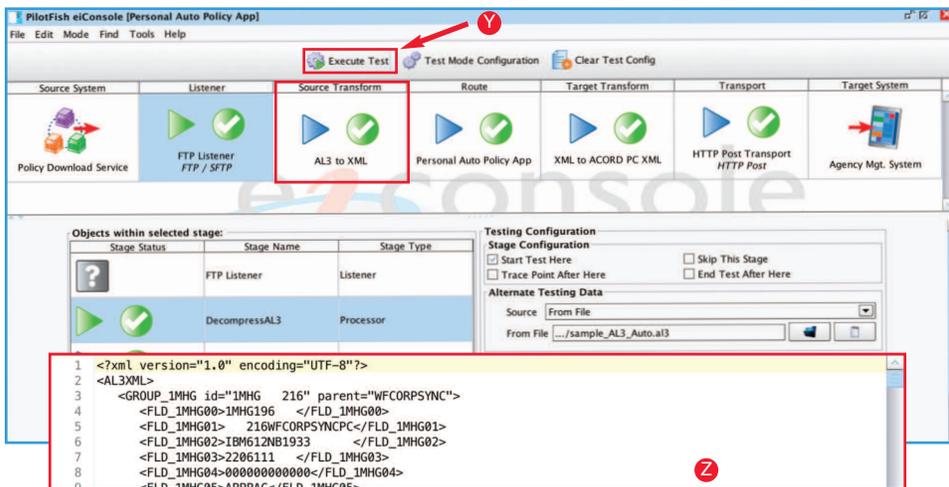
Click the Transport Stage. In the Transport Configuration panel, select the **"HTTP Post"** (W) Transport from the drop-down list. When the configuration panel opens, change the default Transport Name by typing in "New Business Connection" (X). In the Basic tab, paste in the link or type in the new Target URL. Then fill out any additional settings, such as connection timeouts, authentication settings, etc.

Note: There are several dozen Transports built into the eiConsole. These can easily be extended with our Open API (just as in the Listener stage). Processors can also be configured at this stage to perform operations that affect all of the outgoing data and may be layered in any order.



Step 9: Test Your Interface End-to-End

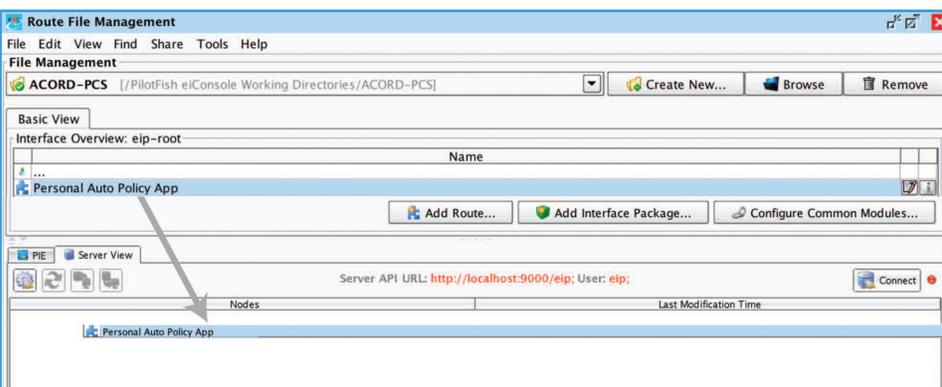
Next you'll want to test your interface. From the Route menu, select Testing Mode. We'll start with the Processor at the Listener stage to begin the testing and provide sample input data. Click the **Execute Test** (Y) icon and the blue question marks turn to green check marks. If a stage failed, a red X would replace the question mark.



You can click any of the stages in the **Stage Output Viewer**. You can view the output of each stage as the data undergoes the transformation and delivery process. For example, you can see the data in ACORD AL3 format (Z) transformed to generic XML, then to the ACORD format. Failed stages provide detailed error messages so that these can be quickly corrected and retested.

Step 10: Deploy Your Interface

Once an interface has been tested from end-to-end, the final step is deployment to an eiPlatform runtime environment. The completed interface is saved as a set of discrete, easily shared, easily managed configuration files. Promotion of an interface can be achieved by simply copying the configuration files to your source control management system or through drag & drop from your Route File Management panel into the Server view panel (as depicted on the left).



That's it. In 10 easy steps we've illustrated how by using the PilotFish graphical automated interface assembly line, you can quickly configure, test and deploy an ACORD AL3 interface.

No matter what the integration requirements, data formats or connectivity required and without any coding or scripting – PilotFish can make your systems interoperable, now. It's architected for reuse, too. Customers who leverage PilotFish for reuse have reported they can tweak an interface and implement a new customer in under 30 minutes. Amazing, right? Call us today to learn more.

