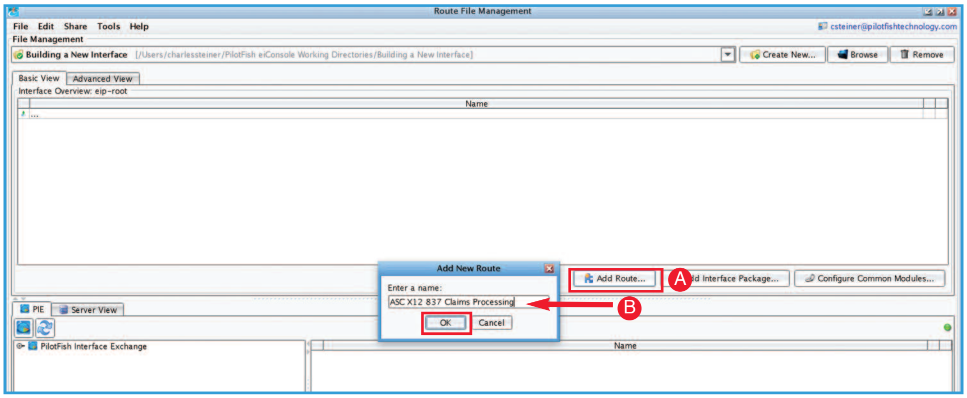
eiconsole for X12

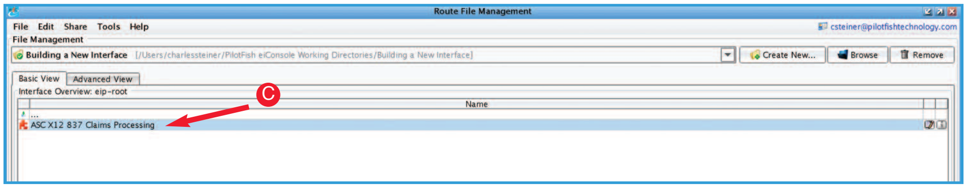
10 Easy Steps to Building an EDI X12 Interface

Featuring the Graphical Automated Interface Assembly Line – *Clone, Tweak, Test and Go!*

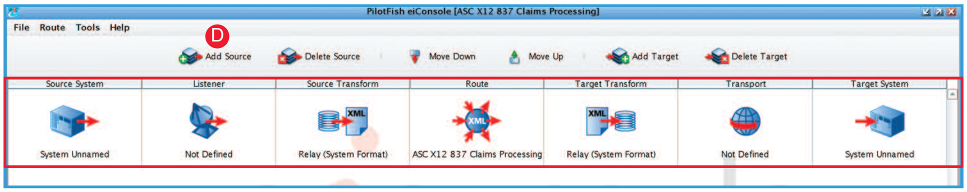

Route File Management
File Edit Share Tools Help
csteiner@pilotfishtechnology.com
File Management
Building a New Interface [/Users/charlessteiner/PilotFish eiConsole Working Directories/Building a New Interface]
Create New...
Browse
Remove
Basic View
Advanced View
Interface Overview: eip-root
Name
Add New Route
Enter a name:
ASC X12 837 Claims Processing
OK
Cancel
Add Route...
A
Add Interface Package...
Configure Common Modules...
B
PIE
Server View
PilotFish Interface Exchange
Name

Step 1: Create a new Route/Interface

When you open the eiConsole, you will see 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), name it (e.g., "X12 837 Claims Processing" (B) and click **OK**.

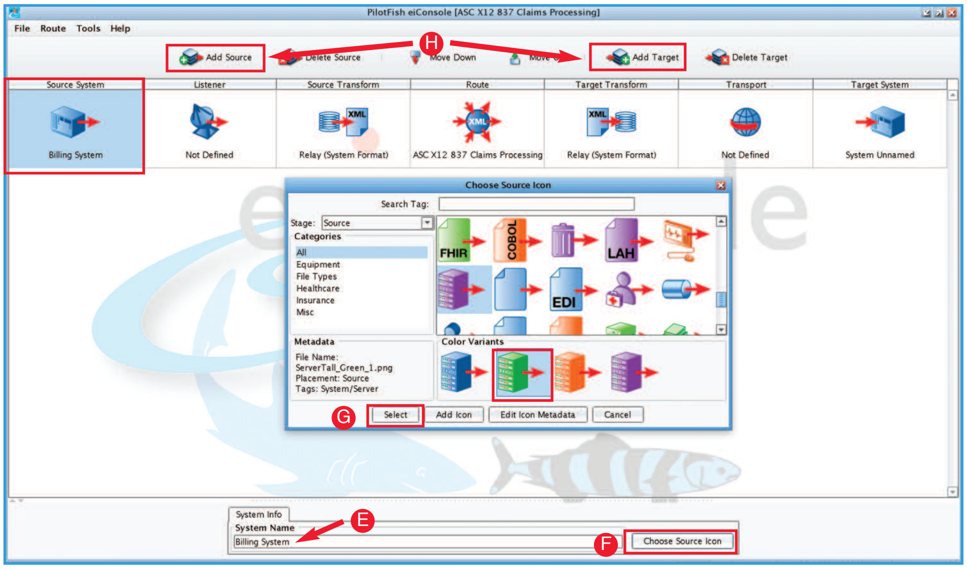

Route File Management
File Edit Share Tools Help
csteiner@pilotfishtechnology.com
File Management
Building a New Interface [/Users/charlessteiner/PilotFish eiConsole Working Directories/Building a New Interface]
Create New...
Browse
Remove
Basic View
Advanced View
Interface Overview: eip-root
C
Name
ASC X12 837 Claims Processing

Graphical Automated Interface Assembly Line



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 system(s). Regardless of the type of integration, the process is always the same. There is no limit to the number of source and target systems that can be linked in this manner.



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 "Billing System"(E) then click the **Choose Source Icon** (F) button.

When the **Choose Source Icon** pop-up opens, select a representative icon from a library of hundreds of icons. Then click **Select** (G) to make your choice. Select the Target Stage and follow the same process.

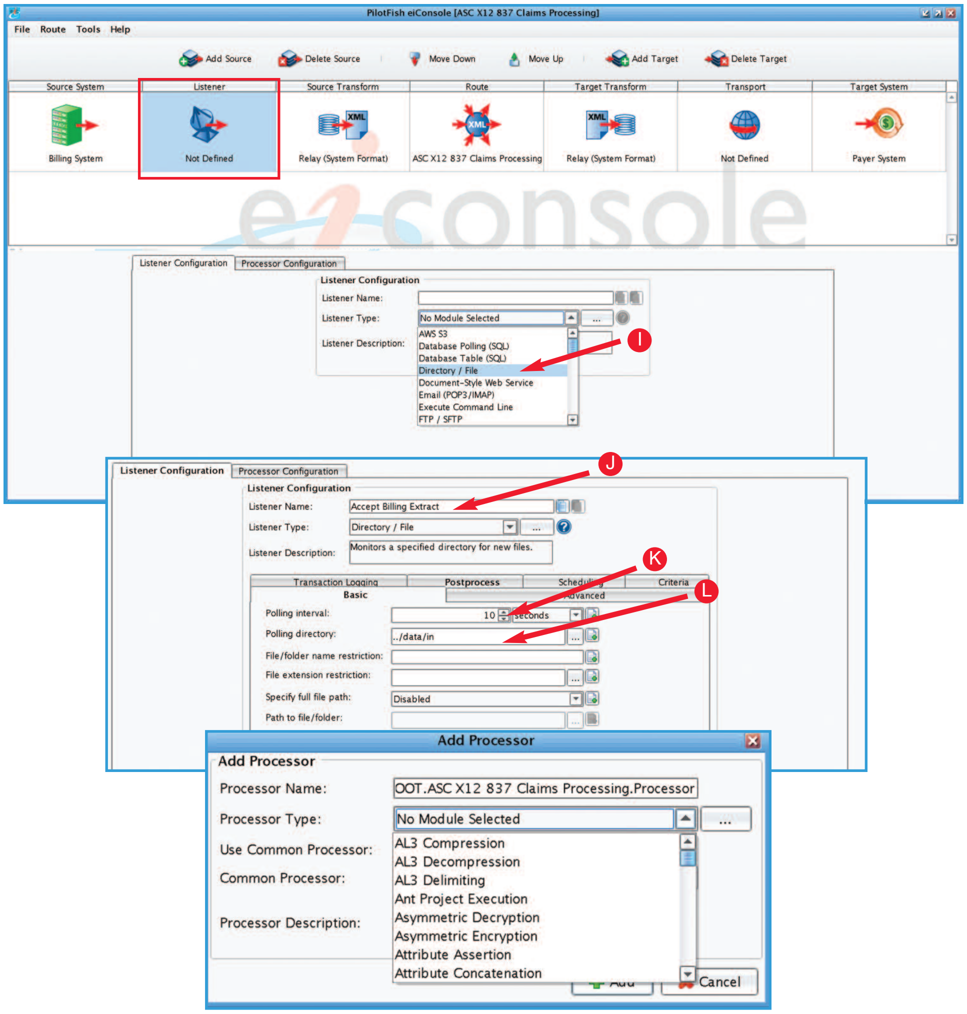
If you would like to add more Source or Target Systems, click the **Add Source** or **Add Target** (H) buttons above the grid. Follow the previous steps to name the systems and to select the appropriate icons.

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 either by polling at scheduled intervals, or receiving data realtime. 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.

For example, select the **Directory/File** (I) Listener. After the panel opens, change the default Listener name to “Accept Billing Extract” (J) and set the polling at 10 seconds (K). Next, click the ellipsis (…) button to select the polling directory. You’ll need to paste in the path (L) where the files you want to read in are located.

After adding a Listener, Processors can be configured to perform data manipulation on any message/files received. Processors are general purpose “widgets” that can perform a variety of tasks. You can use a Processor to add decryption, perform authentication or validation. Scroll to choose from nearly 100 processors or add your own using our open API. For this interface, no Processors are needed, so move on to the next stage.



Listener Configuration

Listener Name:

Listener Type: No Module Selected

Listener Description:

AWS S3
Database Polling (SQL)
Database Table (SQL)
Directory / File
Document-Style Web Service
Email (POP3/IMAP)
Execute Command Line
FTP / SFTP

Listener Configuration

Listener Name: Accept Billing Extract

Listener Type: Directory / File

Listener Description: Monitors a specified directory for new files.

Polling interval: 10 seconds

Polling directory: ../data/in

File/folder name restriction:

File extension restriction:

Specify full file path: Disabled

Path to file/folder:

Add Processor

Processor Name: OOT.ASC X12 837 Claims Processing.Processor

Processor Type: No Module Selected

Use Common Processor:

Common Processor:

Processor Description:

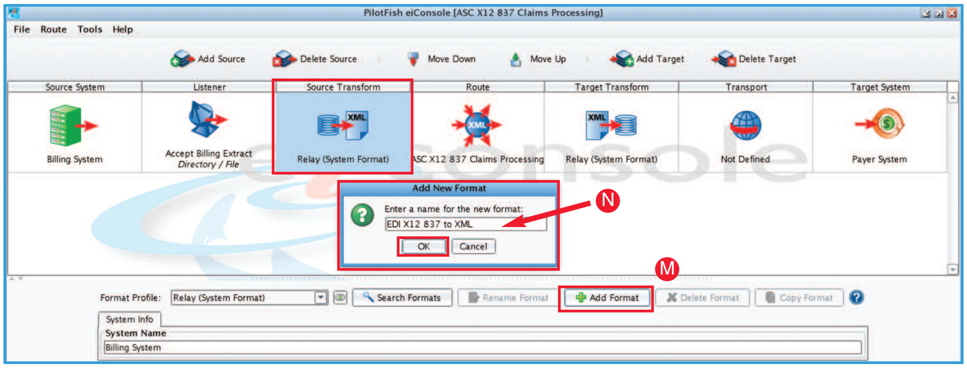
AL3 Compression
AL3 Decompression
AL3 Delimiting
Ant Project Execution
Asymmetric Decryption
Asymmetric Encryption
Attribute Assertion
Attribute Concatenation

Step 5: Transform the Source Data to a Common Standard

PilotFish data transformation is a two-step process. First, an automated, syntax conversion is performed to translate the inbound data into a parsed, XML representation. The eiConsole includes a number of modules that can be readily configured to handle a wide range of common structured data formats.

Transformation Modules are used to parse data from non-XML formats into an XML representation. Once parsed, the eiConsole’s Data Mapper, which generates XSLT, is used to configure logical mapping of that format onto another.

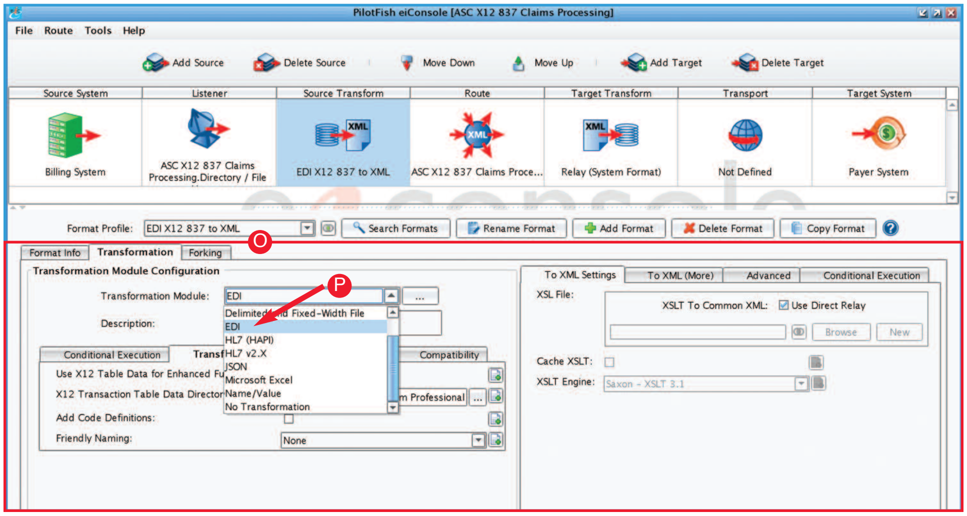
In this example, you’ll need to convert the X12 data to XML. Select the **Source Transform** stage and click the **Add Format** (M) button in the bottom panel. In the pop up panel, name your format “EDI X12 837 to XML” (N) and click **OK**. The **Transformation Module** and **XSLT Configuration** panel (O) open. Select PilotFish's **EDI** Transformation Module (P), which can consume any EDI X12 transaction based on X12's published schema and table data. Note that other transformation modules exist to parse a wide variety of formats, including JSON, flat files, and CSVs.



Enter a name for the new format: EDI X12 837 to XML

Format Profile: Relay (System Format)

System Name: Billing System



Format Profile: EDI X12 837 to XML

Transformation Module Configuration

Transformation Module: EDI

Description:

Delimited and Fixed-Width File
EDI
HL7 (HAPI)
HL7 v2.X
JSON
Microsoft Excel
Name/Value
No Transformation

Use X12 Table Data for Enhanced Fu...

X12 Transaction Table Data Director... ...m Professional

Add Code Definitions: □

Friendly Naming: None

XSL File:

XSLT To Common XML: ☑ Use Direct Relay

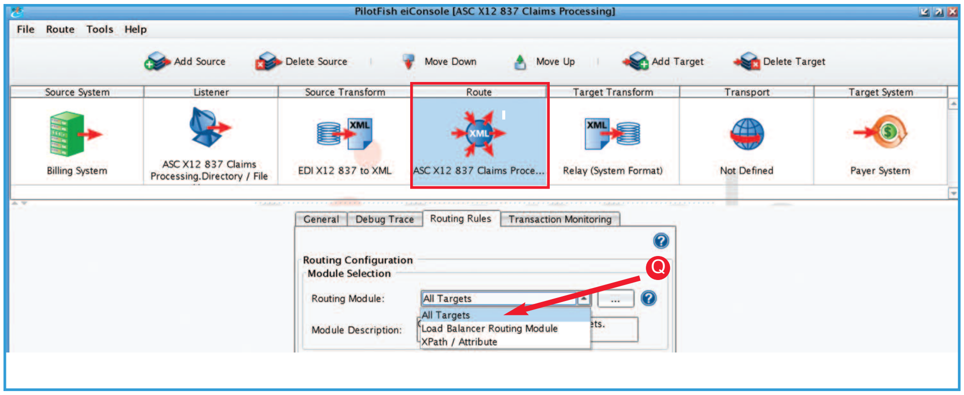
Cache XSLT: □

XSLT Engine: Saxon - XSLT 3.1

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 or filter messages to the appropriate target or targets based on the content of the message. From the drop-down, select **All Targets** (Q).

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



PilotFish eiConsole [ASC X12 837 Claims Processing]

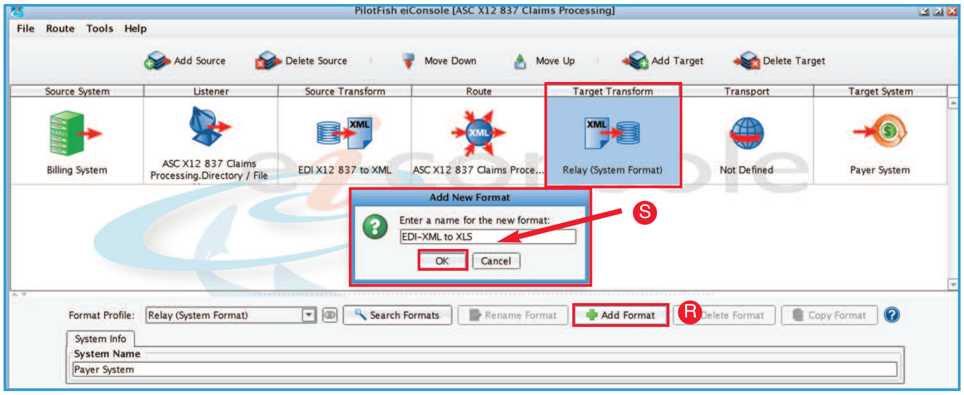
Routing Configuration – Module Selection

Routing Module: All Targets

Routing Module options: All Targets / Load Balancer Routing Module / XPath / Attribute

Step 7: Transform the Data for the Target

Next, select the **Target Transform** stage. In this example, this is where you will convert your new XML representation of X12 into an Excel spreadsheet. 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 "EDI-XML-to-XLS"(S) in the dialog and click **OK**. This opens the transformation panel down below.



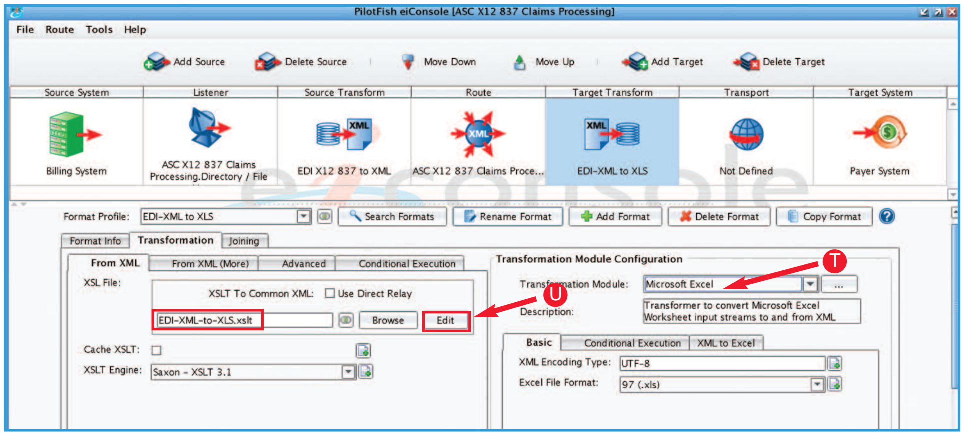
PilotFish eiConsole [ASC X12 837 Claims Processing]

Add New Format – Enter a name for the new format: EDI-XML to XLS

Format Profile: Relay (System Format)

System Info – System Name: Payer System

In the Transformation Module Configuration panel, select the **Microsoft Excel** (T) Transformation Module from the dropdown. This automatically converts an XML representation of Excel into a proper Excel spreadsheet. To create that new XML, go over to the left hand side, to the XSLT Configuration panel and uncheck the **Use Direct Relay** checkbox. This will allow you to author a logical mapping, which you can configure by clicking the **Edit** (U) button to open the Data Mapper.



PilotFish eiConsole [ASC X12 837 Claims Processing]

Format Profile: EDI-XML to XLS

XSLT To Common XML: □ Use Direct Relay

XSL File: EDI-XML-to-XLS.xslt

Cache XSLT: □

XSLT Engine: Saxon – XSLT 3.1

Transformation Module Configuration

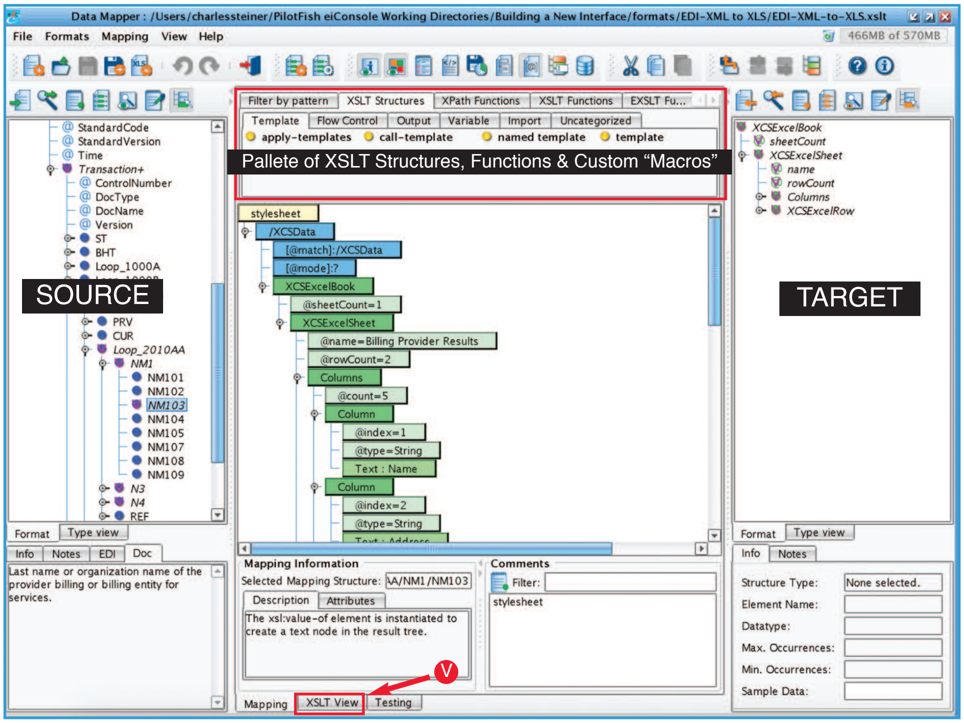
Transformation Module: Microsoft Excel

Description: Transformer to convert Microsoft Excel Worksheet input streams to and from XML

XML Encoding Type: UTF-8

Excel File Format: 97 (.xls)

The Data Mapper generates the XSLT that transforms any data format to any other. The tree on the left represents the **Source** format, and the one on the right the **Target** format. The panel in between is where we configure the relationship between the two via drag-and-drop mappings and additional logic. The palette above the center panel includes a library of useful functions for performing additional manipulations, including conditional logic, looping, and table-based lookups. Selecting the **XSLT View** (V) tab lets users work in XSLT, with changes made immediately available in the graphical view.



Data Mapper : /Users/charlessteiner/PilotFish eiConsole Working Directories/Building a New Interface/formats/EDI-XML to XLS/EDI-XML-to-XLS.xslt

Pallete of XSLT Structures, Functions & Custom "Macros"

SOURCE

TARGET

Doc: Last name or organization name of the provider billing or billing entity for services.

Mapping Information – Selected Mapping Structure: A/NM1/NM103

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

Comments – stylesheet

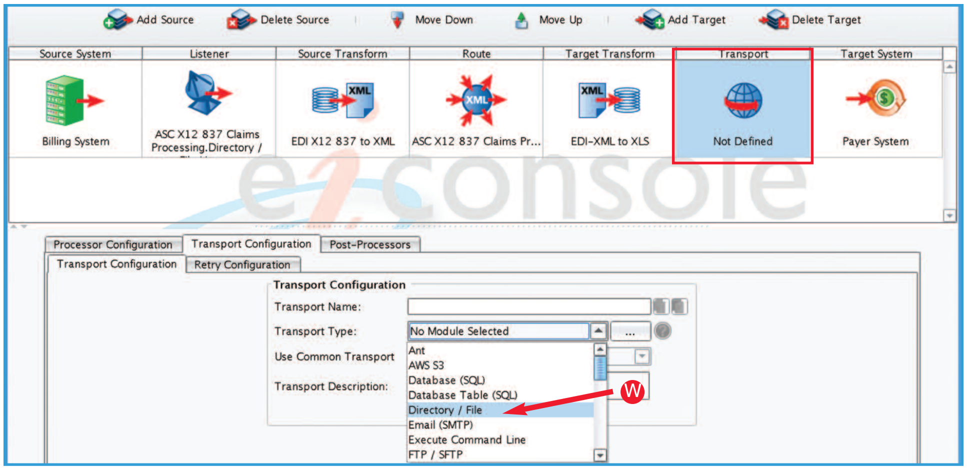
Structure Type: None selected.

While it's possible to start from scratch, users can also automatically create a a baseline for mapping by importing vendor-specific transaction samples, allowing easier data mapping to the specific requirements of the endpoint system.

Step 8: Configure the Transport

The **Transport Stage** is responsible for transmission of the message or file to its endpoint, or to another route in the interface flow. Like the Listener, a wide variety of communication protocols, both batch-oriented and real-time are supported. Real-time processes may be synchronous or asynchronous, where real-time responses can be configured to be handled by an associated route.

In this example, click the Transport Stage. In the Transport Configuration panel, select the **"Directory/File"** (W) Transport from the drop-down list. When the configuration panel opens, change the default Transport Name to "Spreadsheet Archive"(X). In the Basic tab, click the ellipsis (Y) button and paste in the link to your new "out" folder. Then fill in the Target file name to "Spreadsheet"(Z) and the Target file ext. to "xlsx" (aa). As in the Listener stage, processors may be used for preprocessing or cleanup operations.



Transport Configuration

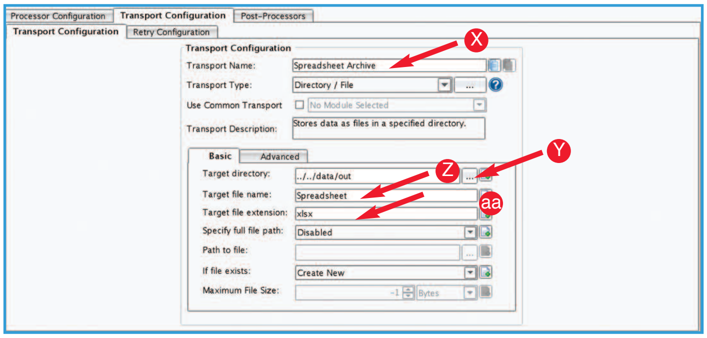
Transport Name:

Transport Type: No Module Selected

Use Common Transport

Transport Description:

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



Transport Configuration

Transport Name: Spreadsheet Archive

Transport Type: Directory / File

Use Common Transport □ No Module Selected

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

Target directory: ../../data/out

Target file name: Spreadsheet

Target file extension: xlsx

Specify full file path: Disabled

Path to file:

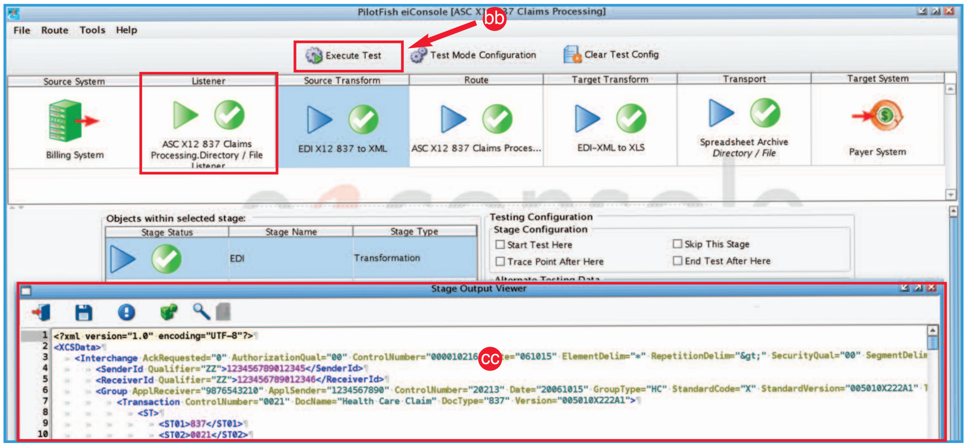
If file exists: Create New

Maximum File Size: -1 Bytes

Step 9: Test Your Interface End-to-End

The eiConsole includes a built-in, step-by-step unit testing capability. No compilation or deployment of the interface is required. Testing mode allows the interface developer to test any portion of their configured "assembly line", inspecting its function, performance, and output.

From the Route menu, select Testing Mode. You can start and stop your test at any stage. Select the **Listener** stage to begin the testing and provide sample input data. Click the **Execute Test** (bb) 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 and in the **Stage Output Viewer** (cc) view the output of each stage as the data undergoes the transformation and delivery process. Failed stages provide detailed error messages so that these can be quickly corrected and retested.



Stage Status	Stage Name	Stage Type
✓	EDI	Transformation

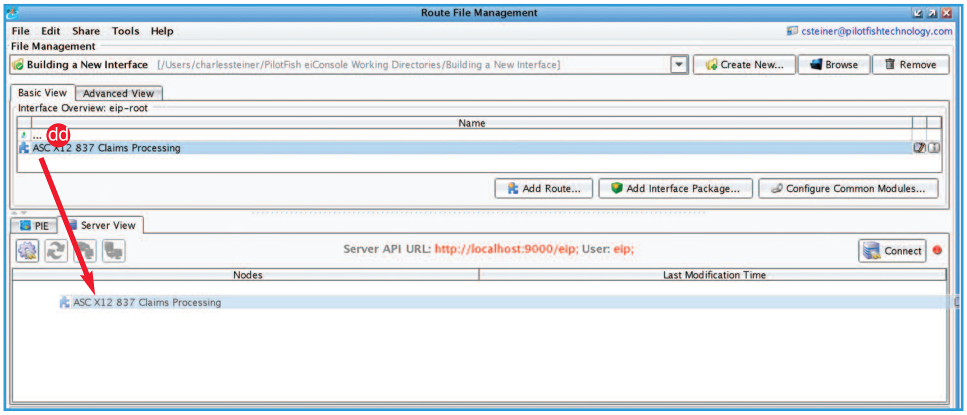
□ Start Test Here □ Skip This Stage
□ Trace Point After Here □ End Test After Here

```
<?xml version="1.0" encoding="UTF-8"?>
<XCSData>
  <Interchange AckRequested="0" AuthorizationQual="00" ControlNumber="000010216 te="061015" ElementDelim="*" RepetitionDelim="&gt;" SecurityQual="00" SegmentDelim
    <SenderId Qualifier="ZZ">123456789012345</SenderId>
    <ReceiverId Qualifier="ZZ">123456789012346</ReceiverId>
    <Group ApplReceiver="9876543210" ApplSender="1234567890" ControlNumber="20213" Date="20061015" GroupType="HC" StandardCode="X" StandardVersion="005010X222A1" T
      <Transaction ControlNumber="0021" DocName="Health Care Claim" DocType="837" Version="005010X222A1">
        <ST>
          <ST01>837</ST01>
          <ST02>0021</ST02>
```

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 managed configuration files. Promotion of an interface can be managed through your preferred source control system, simple file copy, a deployment API, or simple drag and drop (dd).

That's it – EDI X12 interface configuration, testing, and deployment in 10 easy steps.



Building a New Interface [/Users/charlessteiner/PilotFish eiConsole Working Directories/Building a New Interface]

Interface Overview: eip-root

ASC X12 837 Claims Processing

Server API URL: http://localhost:9000/eip; User: eip;

ASC X12 837 Claims Processing

With PilotFish, you can handle virtually any integration requirement, any communication protocol, and any data format with the automated interface assembly line. The PilotFish integration suite solves interoperability challenges without any coding or scripting – and without headaches. PilotFish can make your systems interoperable, now.

Call us today to learn more.

813.864.8662 I www.Healthcare.PilotFishTechnology.com

APPLIED
PilotFish
Healthcare Integration™