TIBCO Foresight™ Studio

Using Foresight™ Studio

Release 7.16.0
January 2012

TIBCO provides the two-second advantage™

TIBCO®
The Power of Now®

Important Information

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README.TXT FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Please see Licensing_Foresight_Products.pdf for licensing details.

Contents

1 Overview

Document Purpose

This document describes how to use TIBCO Foresight™ Studio to set up workflows, and create configuration files for TIBCO Foresight™ Instream® and TIBCO Foresight™ Transaction Insight®.

The workflows are in XML, and can be copied to another Windows system and run by a Windows version of Automator. They can also be copied to a UNIX system, where a UNIX version of the Automator will be able to execute them in production.

A basic knowledge of Instream® is assumed.

System Requirements

Please see **System_requirements.pdf** in Foresight™ Studio's Doc directory.

Opening Foresight Studio

From the Start menu

Choose **Start** | **Programs** | **Foresight** | **ForesightStudio** | **Foresight Studio** *version*

From Windows Explorer

Double-click **Foresight\ForesightStudio\**<*version*>\ **Foresight.Studio.IDE.exe**

Demo System

Studio ships with these demo systems:

InstreamDemoSystem 2 workflows:

- One runs Instream, Docsplitter, and Response Generator.
- One runs Gen275 to create a 275 transaction with attachment in the BIN02.

Routes Routes EDI 834, 835, and 837 data to directories under Foresight\Systems\Routes.

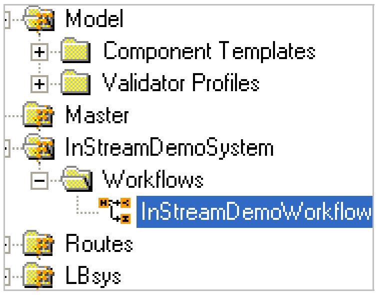
To try out the **InstreamDemoSystem** system:

1. Copy Tutorial837IA.edi

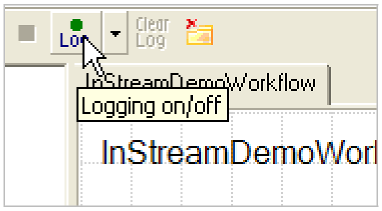
   From:
   Foresight\Systems\InStreamDemoSystem\DemoSystemFolders\TestDocuments

   To:
   Foresight\Systems\InStreamDemoSystem\DemoSystemFolders\ ToWorkflow\Inbound837I

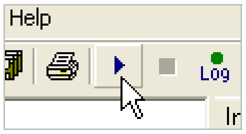
2. Return to Studio and open the workflow by double-clicking on **InStreamDemoSystem | Workflows | InStreamDemoWorkflow**.



3. Click the log toolbar button Log until it turns green:



4. Click the Start Automator toolbar button.



5. After tasks in the Automator window return to Idle, close the window.

6. Output from the workflow should appear under **Foresight\Systems\InStreamDemoSystem\DemoSystemFolders\FromWorkflow**.

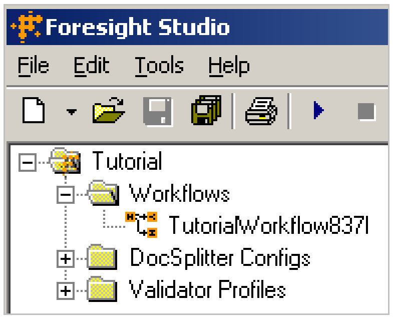
Running the Gen275 workflow

Please see **275atForesight.pdf** for directions.

Navigation

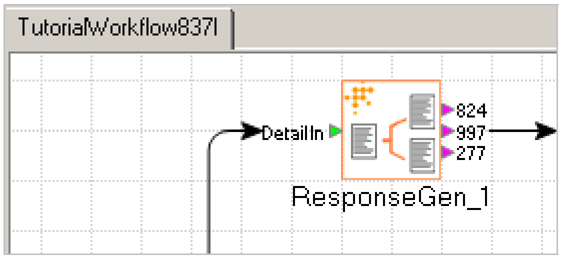
Project Navigation Pane (top left)

A list of all systems and other files. Clicking on one updates the rest of the panes accordingly. You can expend or collapse an item in this pane by right-clicking on it or by using the ⊞ and ⊟ in front of the item.



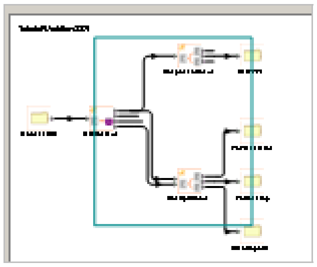
Work Area (center)

Your area to create workflows, edit files, etc. You can see an edit menu by right-clicking anywhere in the work area.



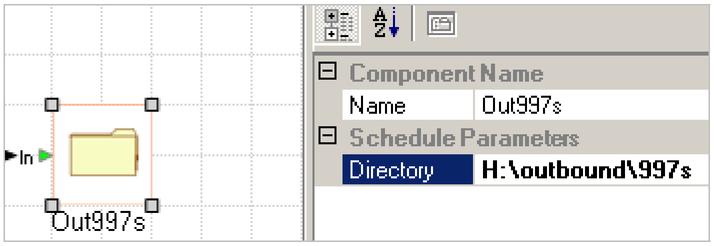
Workflow Overview (right)

When a workflow is selected in the project navigation pane, this area shows a miniature view of the entire workflow. You can zoom in on a workflow area by dragging over it with your mouse in this pane. The main work area then zooms in on the selected area. To zoom out, drag over a larger area.



Component Properties (top right)

Properties of the selected workflow component.



⊟ Component Name

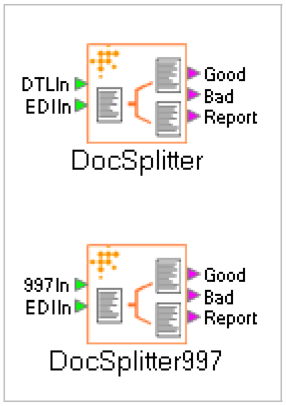
Name	Out997s

⊟ Schedule Parameters

Directory	H:\outbound\997s

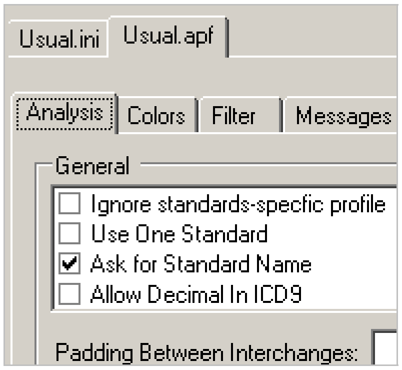
Palette (Bottom left)

Tools and objects you can drag to the work area. Notice the tabs: Custom (if applicable) I/O, Standard, Foresight™ Transaction Insight® (TI), and Validator. Each tab contains different components. Each tab contains different components.



Tabs (top center)

Other documents and workflows. You can close an open tab by right-clicking on it.



General

☐ Ignore standards-specfic profile
☐ Use One Standard
☑ Ask for Standard Name
☐ Allow Decimal In ICD9

Padding Between Interchanges:

Status bar (bottom border)

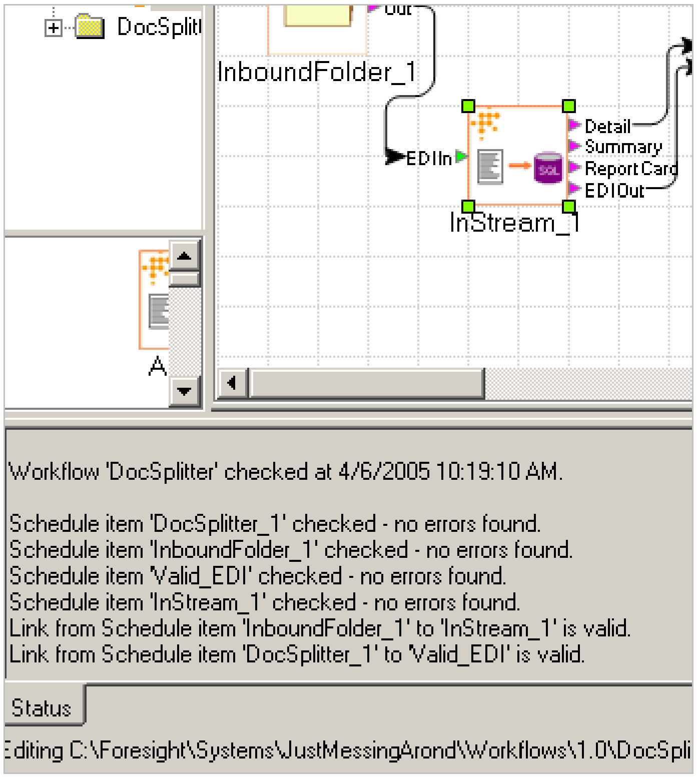
Information about the current document.

Editing C:\Foresight\Systems\Tutorial\ForesightAutomator\2.2.0\TutorialWorkflow837I

Status pane (bottom center)

Some views display a bottom pane with tabs that provide history and other information specific to the current view. Adjust the size of the pane by dragging the splitter bar at the top of this pane. A scroll bar at the right lets you see additional text.

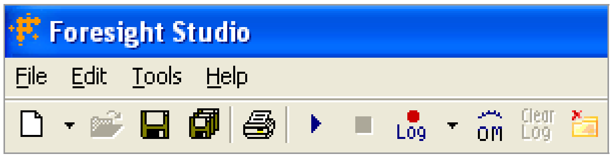
In this example pane, the workflow author checks the loaded schedule to see if the entries make sense semantically: are required values used, do links match up, etc.



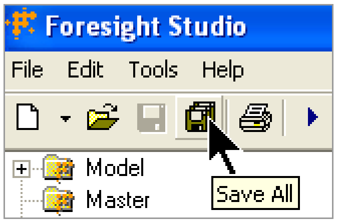
```
Workflow 'DocSplitter' checked at 4/6/2005 10:19:10 AM.

Schedule item 'DocSplitter_1' checked - no errors found.
Schedule item 'InboundFolder_1' checked - no errors found.
Schedule item 'Valid_EDI' checked - no errors found.
Schedule item 'InStream_1' checked - no errors found.
Link from Schedule item 'InboundFolder_1' to 'InStream_1' is valid.
Link from Schedule item 'DocSplitter_1' to 'Valid_EDI' is valid.
```

Editing C:\Foresight\Systems\JustMessingArond\Workflows\1.0\DocSpli

Toolbar



Hover over a button for help:

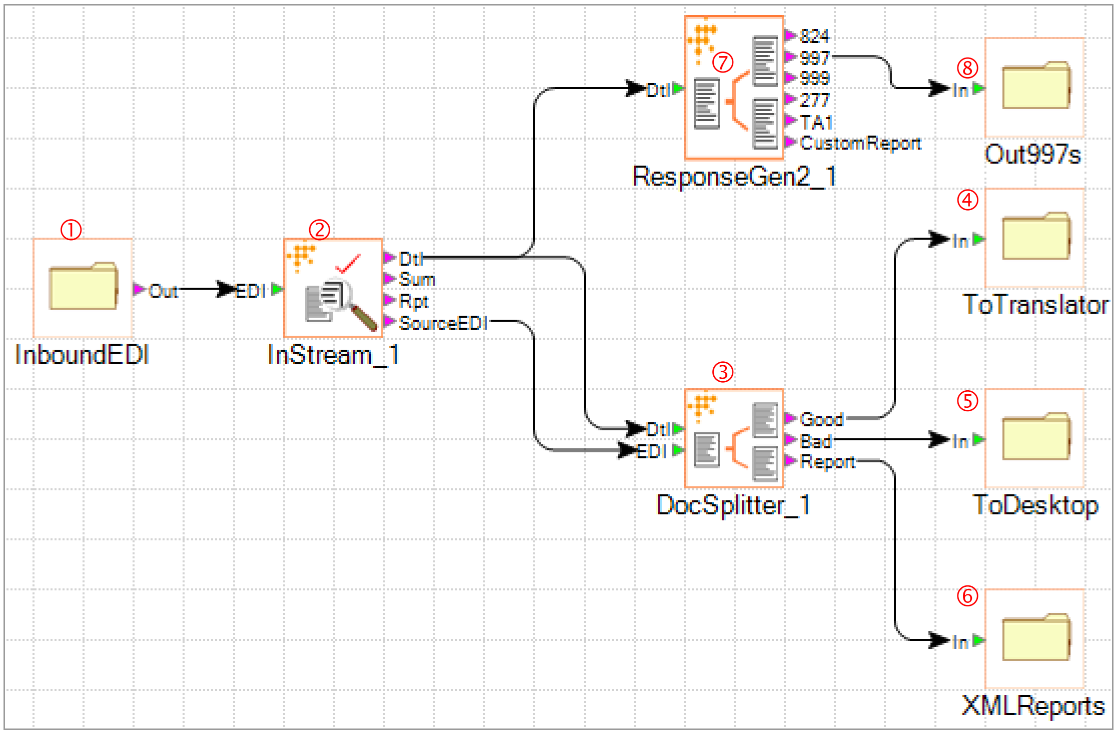


🗋 ▾	New item. Use the arrow to choose the type of item.
📂	Open item. Whatever item you have selected in the left pane will open.
💾	Save the current item.
💾	Save all changed items.
🖨	Print current item.
▶	Start the current workflow.
■	Stop the current workflow.
● Log ▾	To log the current workflow, click until the dot turns green. Choose the logging level with the arrow.
OM OM	Turn on (eye open) or off (eye closed) TIBCO Foresight™ Operational Monitor. Necessary when running a workflow containing certain components such as ISFileIDInsert and MetaUpdater and **ArchiverNoFilter**.
Clear Log	Clears the log tab at the bottom of the screen.
📂	Clears leftover files in the current workflow, after asking for confirmation.

2 Tutorial

Tutorial Workflow

This tutorial guides you through all major steps in setting up this workflow:



The workflow steps include:

① Move 837I EDI files from an inbound directory.

② Validate them with Instream, using a specific profile.

③ Split good from bad claims.

④ Put good claims in a directory to go to the translator.

⑤ Put bad claims in a directory to be viewed by a support person using Desktop.

⑥ Put XML reports in a separate directory.

⑦ At the same time, feed the validation detail results file into Response Generator and generate 997s.

⑧ Put the 997s in a directory where they will be sent out.

Step 1 - Checking Paths to the Executables

The Model system contains information that is copied to each workflow when it is created. This includes the paths to Instream, Docsplitter, and other executables in the workflow.

Therefore, customizing these paths is a one-time-only step if you do it in the Model system before creating any workflows.

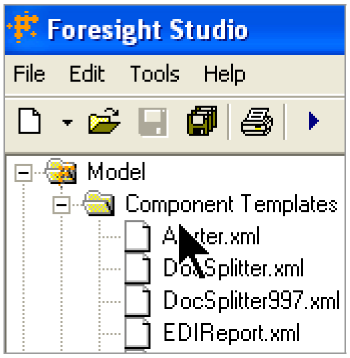
The paths to Instream, Docsplitter, and Response Generator will already be customized if they are installed before Studio. For other components, check the paths as described below.

To confirm that the paths to the Docsplitter, Instream, and Response Generator executables are accurate:

1. Find out the path of the executables used in your workflow.

2. Open Studio with **Start | Programs | Foresight | Foresight Studio | Foresight.Studio.IDE**

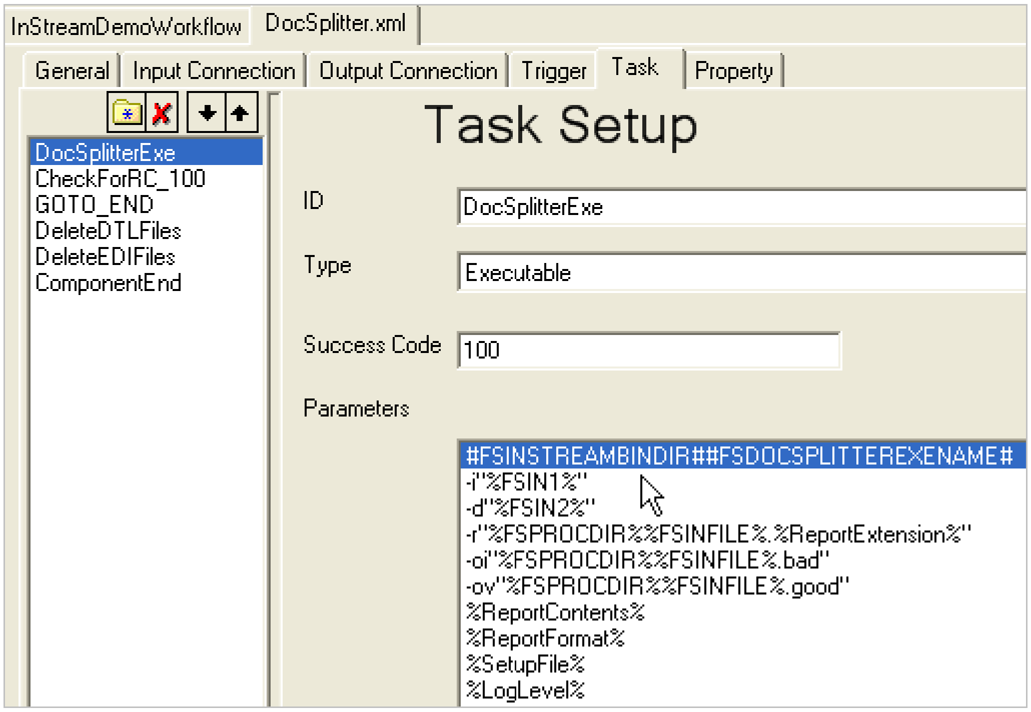
3. In Studio, double-click **Model** at the top left, and then on **Component Templates**.



4. Double-click **DocSplitter.xml**, the template for Docsplitter.

5. Choose the **Task** tab at the top.

6. Confirm the accuracy of the path on the top line. To see the value of the variables, look in WorkflowGlobals.xml in Foresight's Systems directory or choose **File | Edit Globals**.



Task Setup

DocSplitterExe
CheckForRC_100
GOTO_END
DeleteDTLFiles
DeleteEDIFiles
ComponentEnd

ID: DocSplitterExe

Type: Executable

Success Code: 100

Parameters:

```
#FSINSTREAMBINDIR##FSDOCSPLITTEREXENAME#
-i"%FSIN1%"
-d"%FSIN2%"
-r"%FSPROCDIR%%FSINFILE%.%ReportExtension%"
-oi"%FSPROCDIR%%FSINFILE%.bad"
-ov"%FSPROCDIR%%FSINFILE%.good"
%ReportContents%
%ReportFormat%
%SetupFile%
%LogLevel%
```

These variables will make it easier to move workflows to other systems. You can simply copy the workflow files and then edit the paths in WorkflowGlobals.xml.

7. Save.

8. Double-click **InStream.xml** in the left pane, choose its **Task** tab, and select **HVInStream** to the left of Name. Confirm the accuracy of the path.

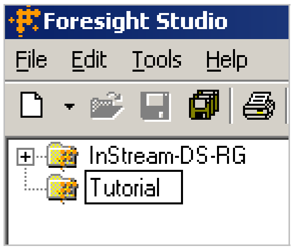
9. Do the same for **ResponseGen.xml**.

10. Click the 💾 (Save All) toolbar button.

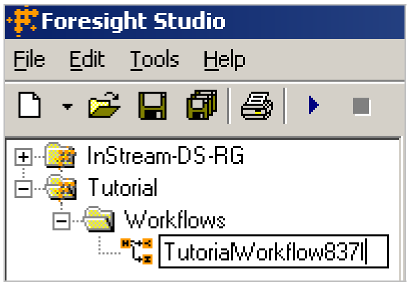
If you are using other executables in workflows, customize the paths in their component templates as well.

Step 2 - Creating a System and Workflow

1. Choose **File | New | System** and type the system's name: **Tutorial**.



2. With Tutorial selected, choose **File | New | Workflow** and type the workflow's name: **TutorialWorkflow837I**. Do not use spaces or other special characters in workflow names.

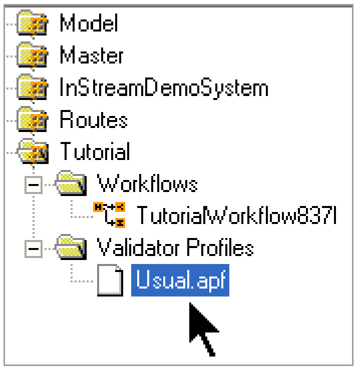


3. Save All.

Step 3 - Creating configurations

Creating a Validation Profile

1. Choose **File | New | Validator Profile** and type the name **Usual.apf** in the left pane.



Model
Master
InStreamDemoSystem
Routes
Tutorial
Workflows
TutorialWorkflow837l
Validator Profiles
Usual.apf

2. Unselect **Check Trailing Blanks**.

   We do not wish to flag them as errors.

3. View the choices under the Filter, Messages, and Instream Results tabs.

4. Save All.

Please see Validator Profiles on page 27 for details.

Creating a Docsplitter Configuration

1. Choose **File | New | DocSplitter Config** and type the name **Usual.ini** in the left pane.

2. Select **Final Tree** click **Save All**.

Notice the two rows of tabs at the top.

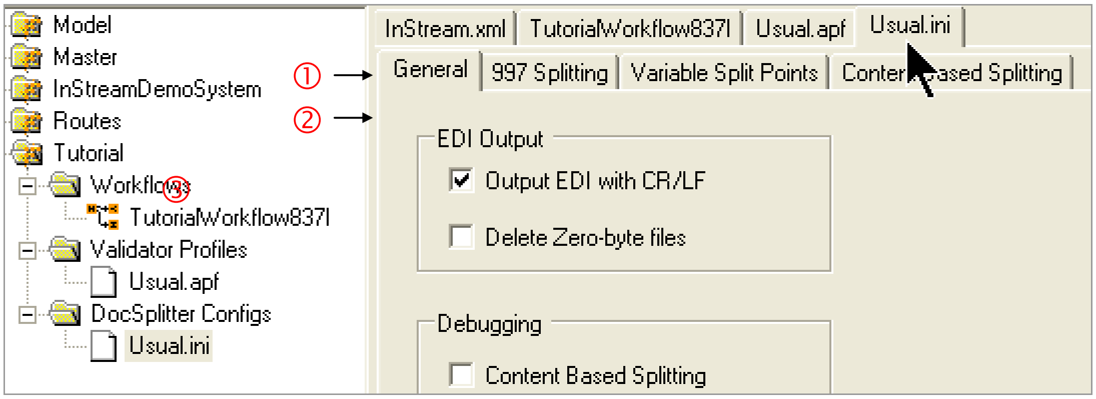
- The top row contains a tab for each document currently being edited.

  In the graphic below, the Docsplitter configuration **Usual.ini** is selected (see ① below).

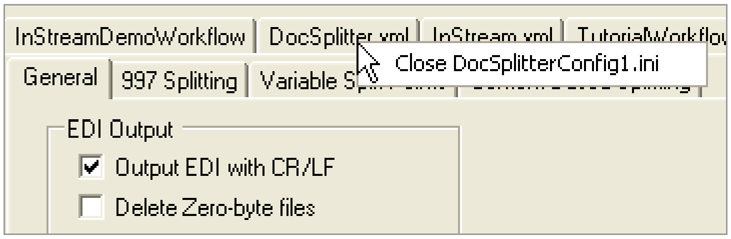
- The bottom row contains tabs specific to the selected document

  In the graphic below, these are tabs for Usual.ini (see ② below).

You can click a tab to change the display in the work area below, or you can double-click a system, workflow, APF file, or INI file in the left pane (see ③ below).



To close tabs at the top, right-click them and choose **Close**:



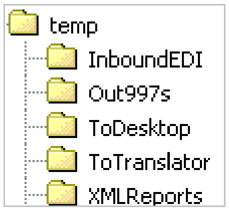
Step 4 - Setting up Directories

The input and output directories for your workflow can be anywhere accessible to your computer. In this tutorial, we set them up under **C:\temp**.

Open File Explorer (right-click **Start** and choose **Explore**) and create these directories:

Directory	**Contents of Directory**
InboundEDI	Directory to hold incoming EDI; files must have extension ".edi"
Out997s	Directory to hold 997 output from Response Generator
ToDesktop	Directory to hold bad EDI files created by Docsplitter
ToTranslator	Directory to hold good EDI file created by Docsplitter
XMLReports	Directory to hold XML output from Docsplitter

The directory structure now looks like this:



```
temp
  InboundEDI
  Out997s
  ToDesktop
  ToTranslator
  XMLReports
```

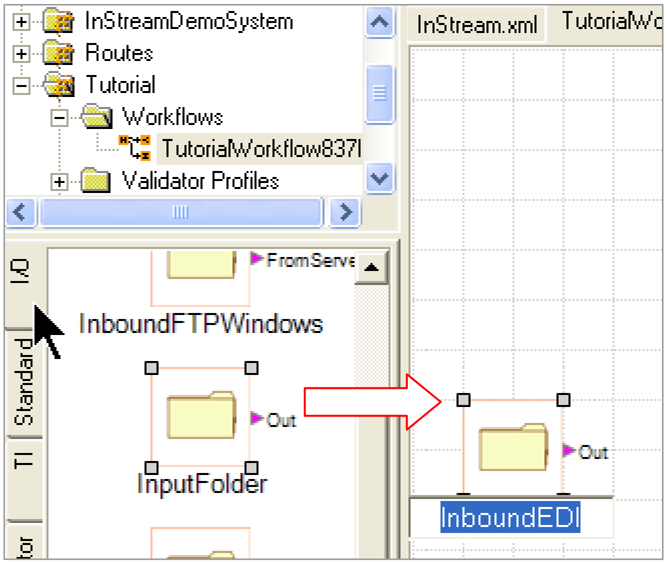
Step 5 - Setting up the workflow

Sending inbound EDI into Instream Validation

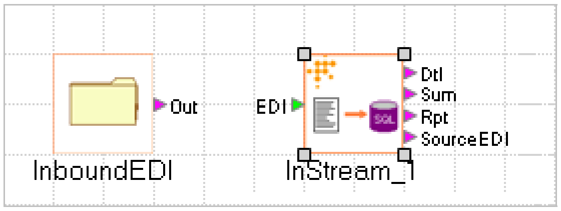
1. Return to Studio and double-click **TutorialWorkflow837I** in the top left pane.

2. Select the **I/O** tab at the bottom left and drag **InputFolder** to the work area. This will be where the inbound EDI starts.

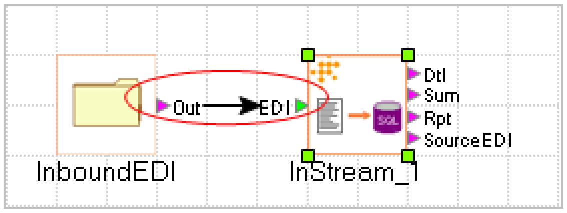
3. Double-click slowly on the component's name in the work area and change it to **InboundEDI** (with no spaces in name).



4. From InboundEDI, the EDI goes to Instream validation. Click the Validator tab at the bottom left, drag Instream to the work area, and put it to the right of InboundEDI:



5. To show that the data will go from **InboundEDI** to **Instream**, drag a line from InboundEDI's **Out** triangle to the **EDI** triangle for **Instream**.



To do this, rest your mouse cursor on the small purple triangle in front of the word **Out**. When it turns into a hand, drag it toward **EDI** for **Instream**.

6. Save.

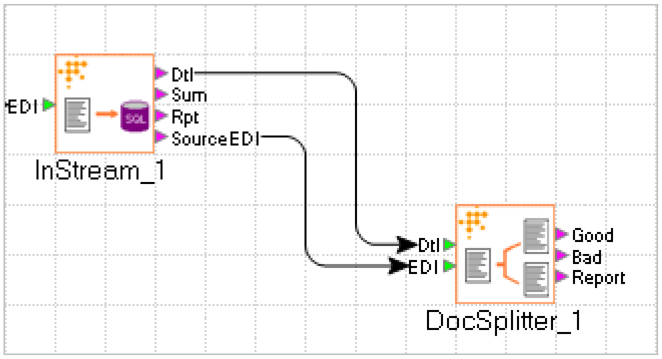
Sending Data to Docsplitter

After validation, the detail file and EDI go to Docsplitter, where good data is split from bad.

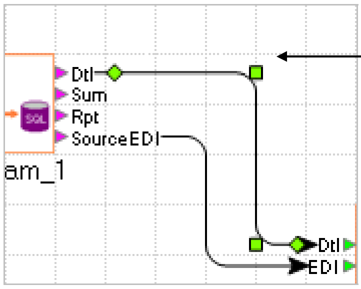
1. Drag **DocSplitter** (not DocSplitter997) to the work area and put it below and to the right of **Instream**.

2. In the work area, you can see **Dtl** (the validation detail results file) coming out of **Instream**. Connect it to DocSplitter's **Dtl**.

3. Likewise, connect **Instream**'s **SourceEDI** to DocSplitter's **EDI**.



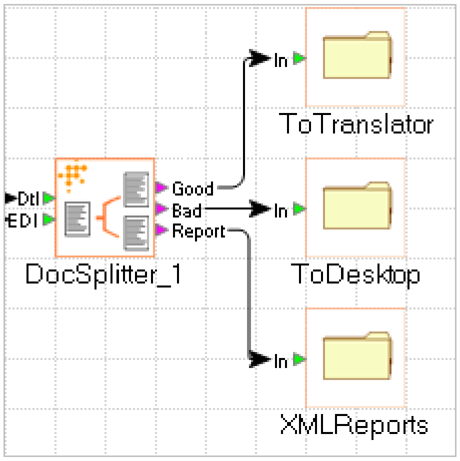
4. If you want to tidy up overlapping lines, click a line that you want to move, and then drag one of the green handles that appear at the corners.



Drag a green handle

5. DocSplitter output goes to three directories. To show this, click the **I/O** tab at the bottom left, drag three **OutputFolder**s to the work area, customize their names, and connect them to DocSplitter output as shown below.

6. Save.



Generating Responses

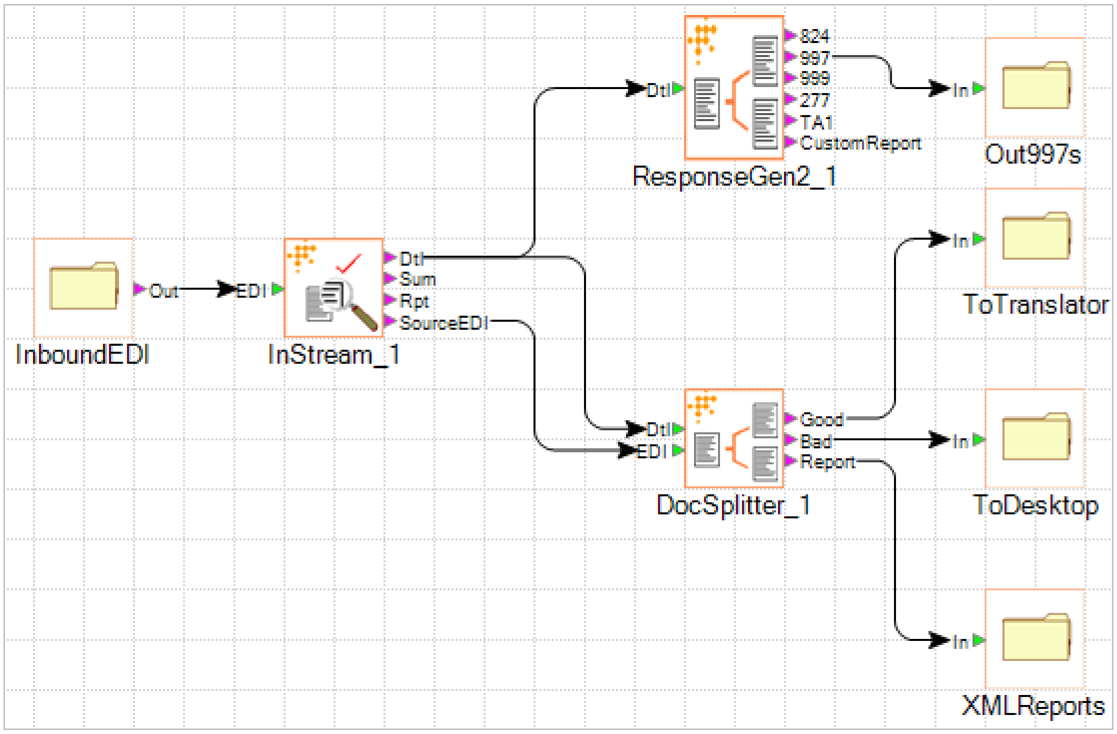
After validation, the detail file and EDI also go to Response Generator, where 997s are generated.

1. Click the **Validator** tab, drag **ResponseGen** to the work area, and put it above and to the right of **Instream**.

2. Connect **Instream**'s **Dtl** output to Response Generator's **Dtl** input.

3. Add an **OutputFolder** and label it **Out997s**. Connect it to ResponseGen's 997 output as shown below.

4. Save.



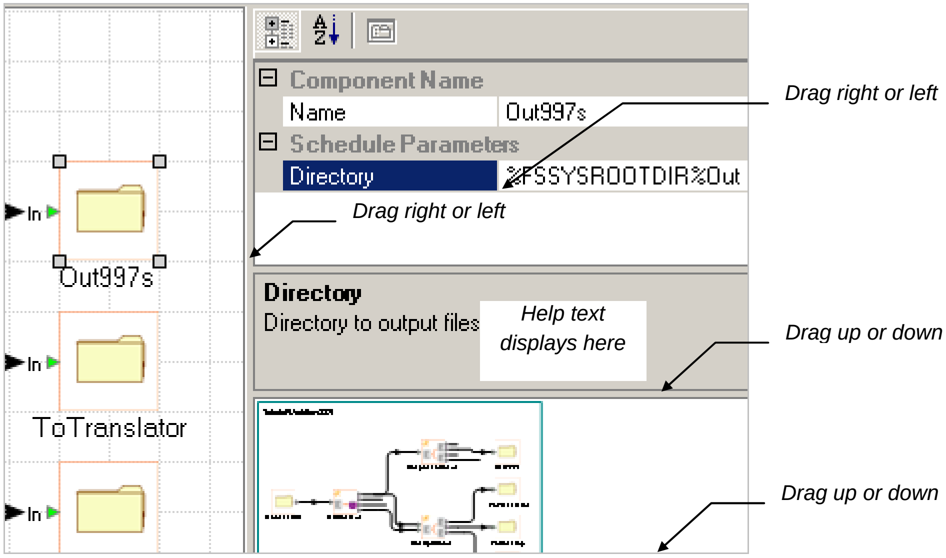
Customizing Workflow Directories

We now specify where each inbound and outbound directory is located and what file extensions they will contain.

1. Click the **out997s** directory in the workflow and notice that the right pane displays a Name and a Directory field for you to update.

2. Click these two buttons a few times to see how they change the appearance of the Schedule Parameters area:

3. Drag the vertical and horizontal splitters to arrange this pane like this:



⊟ Component Name	
Name	Out997s
⊟ Schedule Parameters	
Directory	%FSSYSROOTDIR%Out

Directory
Directory to output files

Help text displays here

Drag right or left

Drag right or left

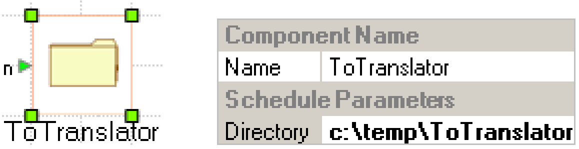
Drag up or down

Drag up or down

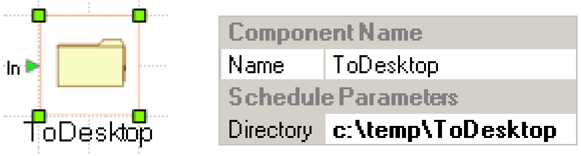
4. Leave Name set to **Out997s**.

5. For the Directory field, type **c:\temp\out997s**, the directory path to the 997s directory that you created previously.

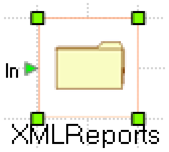
6. Likewise, specify the directories for **ToTranslator**, **ToDesktop**, **XMLReports,** and **InboundEDI** as follows:

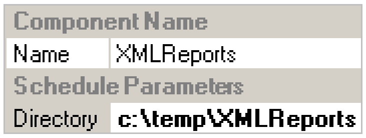


Component Name	
Name	ToTranslator
Schedule Parameters	
Directory	**c:\temp\ToTranslator**

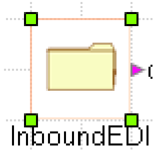


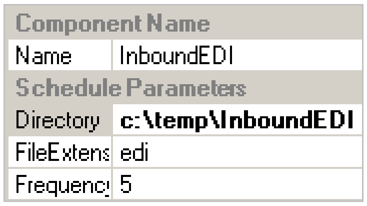
Component Name	
Name	ToDesktop
Schedule Parameters	
Directory	**c:\temp\ToDesktop**





Component Name	
Name	XMLReports
Schedule Parameters	
Directory	**c:\temp\XMLReports**





Component Name	
Name	InboundEDI
Schedule Parameters	
Directory	**c:\temp\InboundEDI**
FileExtens	edi
Frequenc	5

This directory will be checked every 5 seconds for files with extension **edi**.

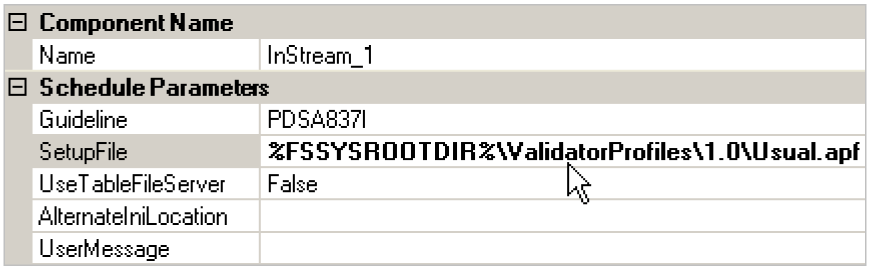
7. Save all.

Customizing Workflow Programs

We now specify the parameters to use with each executable. The selections we will make correspond to command line parameters. For details, see page 43.

1. Click **Instream** in the workflow and fill out the **SetupFile** field in the right pane with the path and filename of the validation profile that you created during this tutorial: **Usual.apf**.

   Do not use quotes around the path.

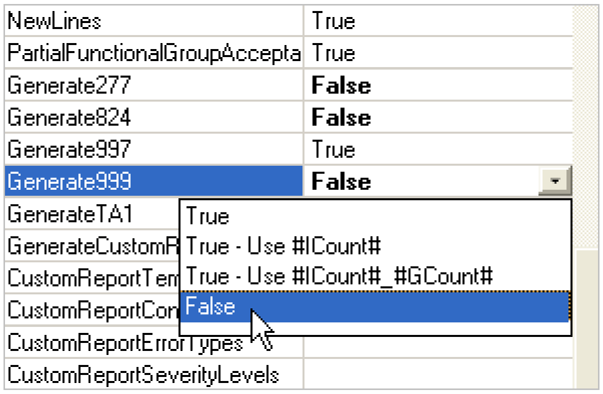
⊟ **Component Name**	
Name	InStream_1
⊟ **Schedule Parameters**	
Guideline	PDSA837I
SetupFile	**%FSSYSROOTDIR%\ValidatorProfiles\1.0\Usual.apf**
UseTableFileServer	False
AlternateIniLocation	
UserMessage	

   You could have put in the literal path C:\Foresight\Systems\Tutorial\ValidatorProfiles\1.0\Usual.apf but the variable FSSYSROOTDIR makes the workflow more portable.

   This field corresponds to Instream's **-s** command line parameter.

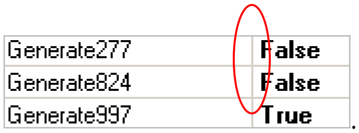
   Be sure that the guideline says **PDSA837I**.

2. Click **ResponseGen** and set all the Generates to false except for Generate997.

NewLines	True
PartialFunctionalGroupAccepta	True
Generate277	**False**
Generate824	**False**
Generate997	True
Generate999	**False**
GenerateTA1	
GenerateCustomR	
CustomReportTem	
CustomReportCon	
CustomReportErrorTypes	
CustomReportSeverityLevels	

   You may have to use the scroll bar and drag the splitter circled below to the right to see the whole parameter:

Generate277	**False**
Generate824	**False**
Generate997	**True**

   .

3. Click **DocSplitter**. Under SetupFile, specify the path to **Usual.ini**, the DocSplitter configuration file that you created during this tutorial:

   ```
   %FSSYSROOTDIR%DocSplitterConfigs\1.0\Usual.ini
   ```

4. Save all.

Step 6 - Testing the Workflow

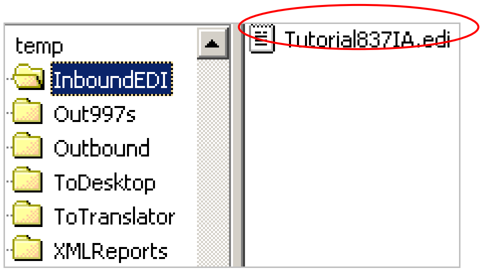
Running the Workflow

1. Copy the EDI file **Tutorial837IA.edi** from:

   Foresight\ForesightStudio\<*version*>\DemoFiles

   to **C:\temp\InboundEDI**.

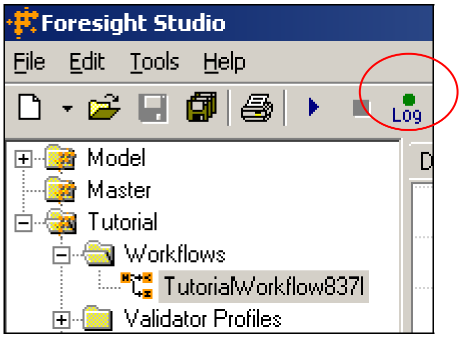
   This is an EDI file with 5 good and 5 bad claims



2. In Studio, click **Save All** in case there are unsaved changes.

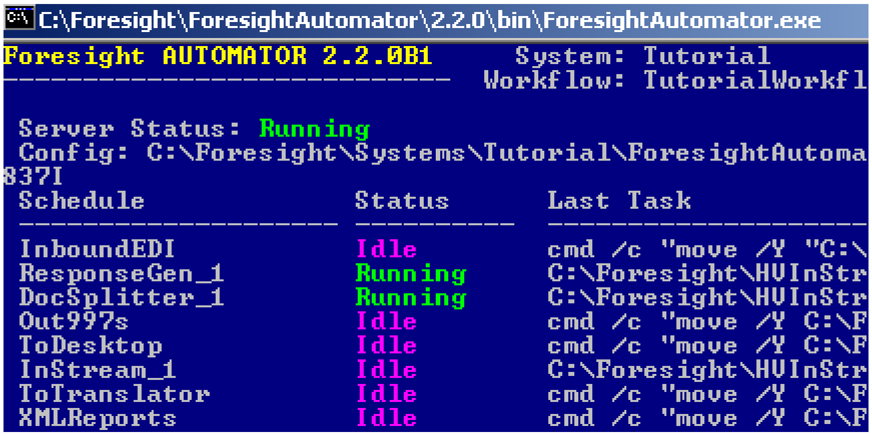
3. Be sure that TutorialWorkflow837I is open (it is showing in the work area). If not, double click it in the top left pane.

4. Turn on logging by clicking the Log toolbar button until it turns dark green.



5. Start Automator by pressing the ▶ toolbar button. This will execute the workflow.

Automator displays a window that shows its progress. After it finishes, all tasks except InboundEDI will return to **Idle**. You can review the steps it took by looking under the **Last Task** column.



```
C:\Foresight\ForesightAutomator\2.2.0\bin\ForesightAutomator.exe
Foresight AUTOMATOR 2.2.0B1      System: Tutorial
----------------------------   Workflow: TutorialWorkfl

 Server Status: Running
 Config: C:\Foresight\Systems\Tutorial\ForesightAutoma
8371
 Schedule                    Status      Last Task
 --------------------        ----------  --------------------
 InboundEDI                  Idle        cmd /c "move /Y "C:\
 ResponseGen_1               Running     C:\Foresight\HVInStr
 DocSplitter_1               Running     C:\Foresight\HVInStr
 Out997s                     Idle        cmd /c "move /Y C:\F
 ToDesktop                   Idle        cmd /c "move /Y C:\F
 InStream_1                  Idle        C:\Foresight\HVInStr
 ToTranslator                Idle        cmd /c "move /Y C:\F
 XMLReports                  Idle        cmd /c "move /Y C:\F
```

For details about the Automator display, see **Automator.pdf**.

6. Stop Automator by pressing the ■ toolbar button or closing the Automator window.

Looking for Output

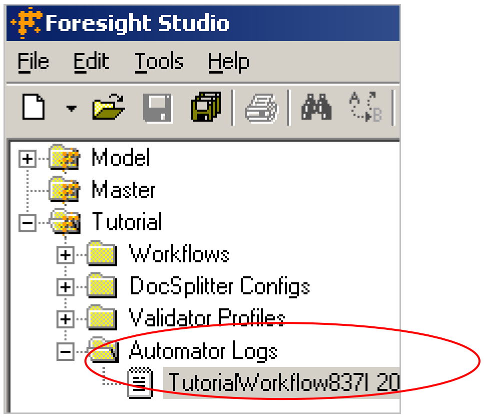
Look for the output under **C:\temp**:

Out997s	A 997 created by Response Generator.
ToDesktop	An invalid EDI file created by Docsplitter.
ToTranslator	A valid EDI file created by Docsplitter
XMLReports	A validation XML report.

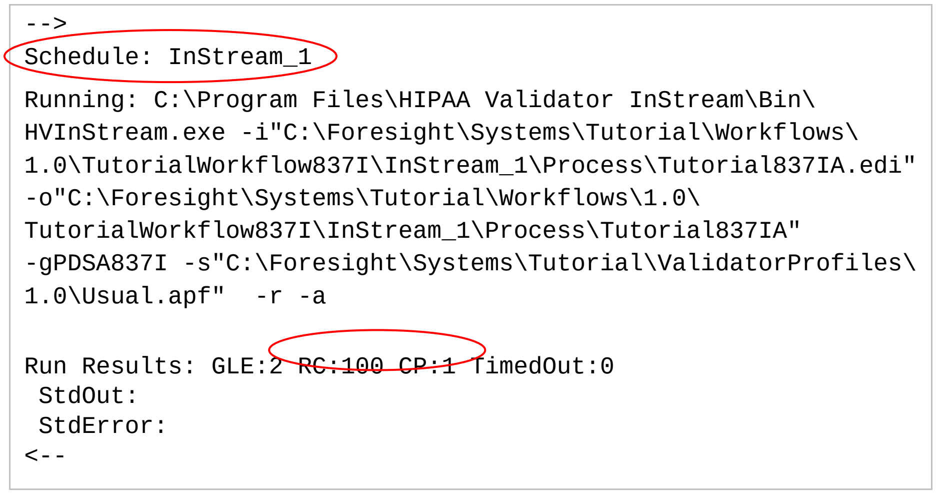
If the expected output is not there, some detective work is in order. See the next section on log files.

Checking the Log File

1. In the upper left pane, open the log file under the Tutorial system:



2. Use *Ctrl*-**f** and search for **HVInStream**. If validation succeeded, you will see RC: 100 (Return Code 100 – success for Instream) and CP:1 (Create Process 1).



```
-->
Schedule: InStream_1
Running: C:\Program Files\HIPAA Validator InStream\Bin\
HVInStream.exe -i"C:\Foresight\Systems\Tutorial\Workflows\
1.0\TutorialWorkflow837I\InStream_1\Process\Tutorial837IA.edi"
-o"C:\Foresight\Systems\Tutorial\Workflows\1.0\
TutorialWorkflow837I\InStream_1\Process\Tutorial837IA"
-gPDSA837I -s"C:\Foresight\Systems\Tutorial\ValidatorProfiles\
1.0\Usual.apf"  -r -a

Run Results: GLE:2 RC:100 CP:1 TimedOut:0
 StdOut:
 StdError:
<--
```

You can also open the log in Notepad or another text editor. It is under **Foresight\Systems\Tutorial\AutomatorLogs\1.0**.

When run from Studio, Automator generates a detailed log. This is to help you debug your workflows. When running a workflow outside of Studio, you have your choice of three levels of logging.

For more information about logging, see **Automator.pdf** in your Foresight Studio **Doc** directory.

Changing the Workflow while Automator is Running

If you change the workflow while Automator is running, you will need to stop and restart Automator to make the changes take effect. Automator only reads the workflow XML files when it starts.

Congratulations! You have just used the major features of Foresight Studio. Please review the rest of this manual for more details.

3 Configurations

Validator Profiles

A validation profile contains configuration information for Instream. Each time you validate, a profile is read to determine what is to be checked and what is to be written to output files.

Validation profiles are commonly called “APF files” because they have file extension APF.

Using an Instream Validation Profile

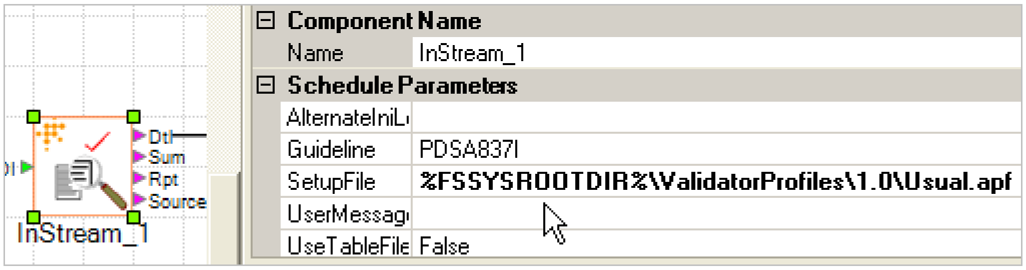
Instream validation profiles have file extension APF and are located in the **Foresight** directory under

Systems\\<*system*>\\ValidatorProfiles\\<*version*>

To use a profile in a workflow:

1. Click **Instream** in the workflow.

2. Type the APF file’s path and filename in the SetupFile field to the right:



⊟ **Component Name**	
Name	InStream_1
⊟ **Schedule Parameters**	
AlternateIniL	
Guideline	PDSA837I
SetupFile	**%FSSYSROOTDIR%\ValidatorProfiles\1.0\Usual.apf**
UserMessag	
UseTableFile	False

Importing an Instream Validation Profile

To import an existing Instream validation profile:

1. Click the system where it will be used.

2. Choose **File | Import | Validator Profiles**.

3. Find the APF file and click **Open**.

4. To copy it to any other system, click the profile and use **File | Copy To**.

   The original and copy are separate. Changing one does not affect the other.

To rename the copy, click slowly on its name and then type the new name. Use file type APF and be sure that the new name does not exist in the Master system.

Creating an Instream Validation Profile

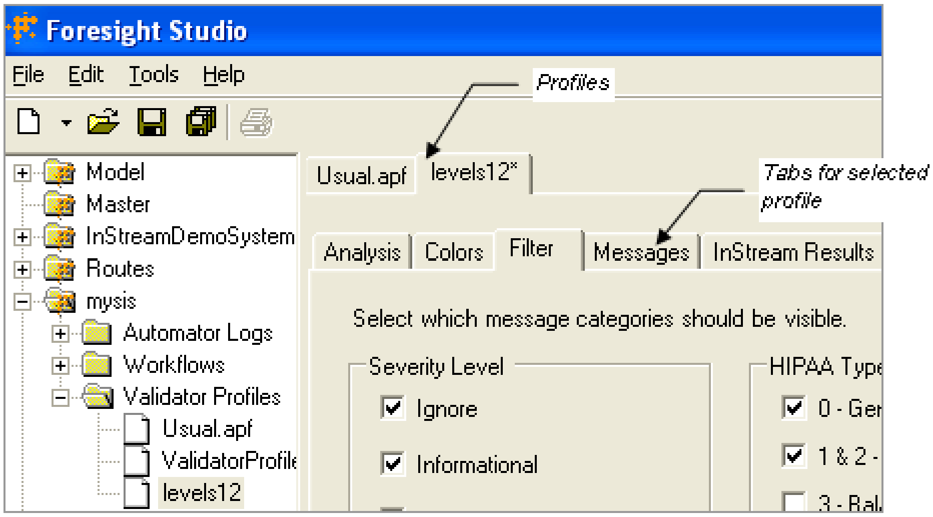
You can set up your own profiles from within Studio:

1. Click the system where the new validation profile will be used.

2. Choose **File | New | Validator Profile**.

3. In the left pane, type a name for the profile. Use file extension **apf**.

The work area will display a tab for the new profile. Each profile has its own tabs for Analysis, Colors, Filter, Messages, and InStream Results.



Select which message categories should be visible.

Analysis Tab

The **Analysis** tab contains basic validation options.

The options are explained in detail in the **Analyzer Options Section** of **APF.pdf**, which you can find in Studio's **Doc** directory.

Colors Tab

The **Colors** tab specifies which colors are used to display each error type.

Since Instream output does not use color, this tab affects Desktop only. Although Desktop cannot be part of a Studio workflow, you can customize colors from within Studio and use the resulting profile in Desktop.

Using the Colors Tab	
To do this …	**Do this …**
Change colors for a particular type	Click the color next to the type.
Reset colors to the default	Click **Default Colors**.
Set all colors to black	Click **No Colors**.

Default Colors for Diagnostic Types		
Type	**Color**	**Meaning**
Ignore	Light gray	Messages you consider unimportant.
Informational	Dark gray	Identifies segments, cancellations, etc.
Warning	Olive	Minor errors, such as leading zeros in numerics or assumptions made due to ambiguity.
Error	Blue	Errors such as wrong codes, violation of rules, etc.
Fatal	Red	Error that stops the validation from continuing.
User #1	Dark green	For your use.
User #2	Light green	For your use.

Filters Tab

Each message has a severity and a HIPAA type. This tab lets you designate the severities or types of validation messages that you want to display.

The options are explained in detail in the **Warning Allow** and **Types Allow** sections of **APF.pdf**, which you can find in Studio's **Doc** directory.

Messages Tab

This tab describes the characteristics of individual validation diagnostic messages. You can change the severity, HIPAA type, and other characteristics of the message.

The options are explained in detail in the **Warning Levels Section** of **APF.pdf**, which you can find in Studio's **Doc** directory.

InStream Results Tab

This tab lets you select which records will appear in Instream's detail results file.

The options are explained in detail in the **Detail Record Output Section** of **APF.pdf**, which you can find in Studio's **Doc** directory.

Docsplitter Configurations

Docsplitter has defaults that determine where and how to split the EDI, what output files to create, and what is to be in them. You can create one or more configurations that change Docsplitter behavior.

Using a Docsplitter Configuration

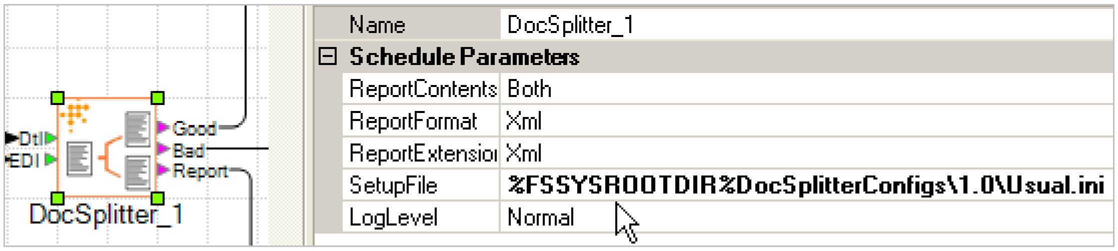
Docsplitter configs have file extension INI and are located in the **Foresight** directory under:

Systems\<*system*>\DocSplitterConfigs\<*version*>

To use a Docsplitter configuration:

1. Click the Docsplitter component in the workflow.

2. Type the path and filename in the **SetupFile** field to the right.



Name	DocSplitter_1
⊟ **Schedule Parameters**	
ReportContents	Both
ReportFormat	Xml
ReportExtensio	Xml
SetupFile	**%FSSYSROOTDIR%DocSplitterConfigs\1.0\Usual.ini**
LogLevel	Normal

Creating a Docsplitter Configuration

To create a Docsplitter configuration from within Studio:

1. Click the system where the new Docsplitter config will be used.

2. Choose **File | New | DocSplitter Config**.

3. In the left pane, type a name for the configuration. Use file extension **ini**.

4. In the work area, customize the settings under these tabs:

The options are explained in the **Docsplitter Setup File** section of **DocumentSplitterTechnicalManual.pdf** in Studio's **Doc** directory.

General Tab

The **General** tab corresponds to the [Options] and [Debugging] sections of the Docsplitter setup file.

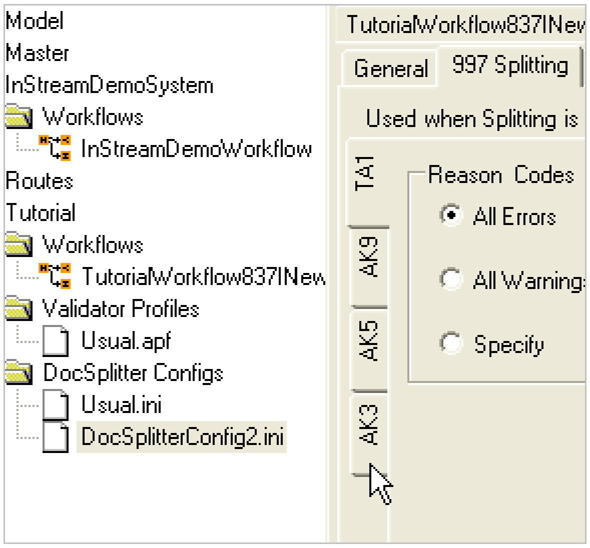
Option / field in Studio	Corresponding line in setup file	Effect
Output EDI with CR/LF	OutputEDIWithCRLF	Do you want a CR/LF to follow each segment terminator?
Delete Zero-byte files	RemoveZeroByteEDIFiles	Do you want to remove all empty files created by DocSplitter?
Content Based Splitting	Content	Should output messages include each value that caused a split, its location, and what output file will get it?
Variable Split Points	VariableSplitPoints	Do you want to display which data is splitting to each file for your variable split points?
Final Tree	TreeFinal	Do you want to display a hierarchical chart that shows what files received data from each split level? Turning this option on in a workflow can cause Automator to lock up and is not recommended.

Option / field in Studio	Corresponding line in setup file	Effect
Loop Levels	ProcessNode	Do you want to display loop levels as they are processed?
Detail File Record IDs	ProcessRecord	Do you want to display detail file record IDs as they are processed?
Progress Status Checkpoints	ProgressStatus	Do you want to display checkpoint messages as they are processed?
Control Number Status	ControlNumberStatus	Displays the control numbers of ISA, GS and ST segments after this many segments have been processed.

997 Splitting Tab

The **997 Splitting** tab lets you specify which error codes are considered warnings rather than errors. These settings are used with the DocSplitter997 component, which uses a 997 for input, rather than a validation results file.

Notice the vertical tabs.



Option / field in Studio	Corresponding line in setup file	Effect
TA1 tab	[TA1_Allow]	Select the TA105 codes that are to be treated as warnings when doing 997-based splitting.
AK9 tab	[AK9_Allow]	Select the AK905-AK909 codes that are to be treated as warnings when doing 997-based splitting.

Option / field in Studio	Corresponding line in setup file	Effect
AK5 tab	[AK5_Allow]	Select the AK502 codes that are to be treated as warnings when doing 997-based splitting.
AK3 tab	[AK3_Allow]	Select the AK304 codes that are to be treated as warnings when doing 997-based splitting.

Each tab has these choices:

All Errors	All error codes are treated as errors
All Warnings	All error codes are treated as warnings
Specify	Lets you select individual codes

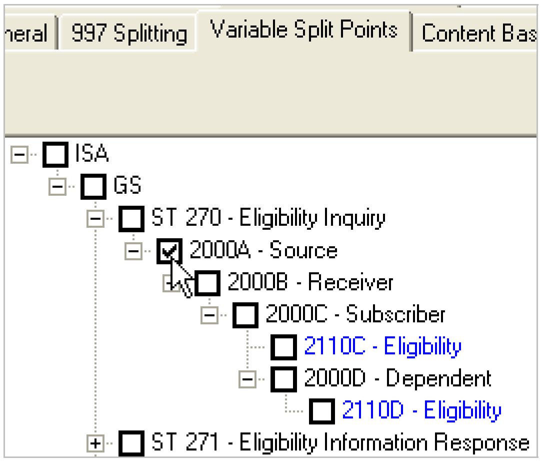
The options are explained in details in the **Docsplitter Setup File** and **Appendix E** sections of **DocumentSplitterTechnicalManual.pdf**, which you can find in Instream's **Doc** directory.

Variable Split Points Tab

This tab corresponds to the [Split Point] section of the Docsplitter setup file. It lets you set customized split points.

Open each document by clicking on the plus sign in front of it. Default split points are blue. To change the split points, select the desired split points.

In this example, errors below 2000A will cause a split at the 2000A loop:



```
⊟ ☐ ISA
  ⊟ ☐ GS
    ⊟ ☐ ST 270 - Eligibility Inquiry
      ⊟ ☑ 2000A - Source
        ⊟ ☐ 2000B - Receiver
          ⊟ ☐ 2000C - Subscriber
              ☐ 2110C - Eligibility
            ⊟ ☐ 2000D - Dependent
                ☐ 2110D - Eligibility
    ⊞ ☐ ST 271 - Eligibility Information Response
```

Variable split points are explained in details in **Appendix C** of **DocumentSplitterTechnicalManual.pdf**, which you can find in Studio's **Doc** directory.

Content Based Splitting Tab

This tab corresponds to the [Content Splitting Options] section of the Docsplitter setup file. It lets you set customized split points.

Option or field in Studio	**Corresponds to . . .**	**Effect**
Enable Content Based Splitting	The Docsplitter command line **s** parameter pointing to a setup file with a splitting map	Activates the options on this tab.
Maximum splits per file	**SeparateAtSplitPoint** in Docsplitter setup file	Not available with content based splitting. Type the maximum number of split data units to put in a single file.
Merge Valid Output	**MergeValidOutput** in Docsplitter setup file	When using content-based splitting, do you want all valid EDI data to go to one file?
Merge Invalid Output	**MergeInValidOutput** in Docsplitter setup file	When using content-based splitting, do you want all invalid EDI data to go to one file?
Validation Splitting	**ValidationSplitting** in Docsplitter setup file	When using content-based splitting, do you want to split valid from invalid data also?
Filename End, Content	Lines in **[Content Splitting Map]** section in Docsplitter setup file	Which content causes a split and what file name ending shall it split to? Type the end of the filename and the content that should go to that file. Drag the right side of the column headers to adjust the width.

Content based splitting is explained in **Appendix F** of **DocumentSplitterTechnicalManual.pdf**, which you can find in Studio's Doc directory.

Finding your Docsplitter Configuration File

Docsplitter configurations are specific to the system under which they were defined. Look for the files under **Foresight\Systems\**<*system name*>\ **DocSplitterConfigs**\<*version*> .

Envelope-Based Partner Automation Config

Partner automation is described in **InStreamTPAutomation.pdf** in Instream's Doc directory. You will need to be familiar with partner automation before setting up a lookup file.

The lookup file lets Instream validation automatically select guideline, profile, output path based on one or more values in the ISA or GS. During validation, Instream consults the lookup table at each GS segment

Lookup files are commonly called "CSV files" because they have file extension CSV.

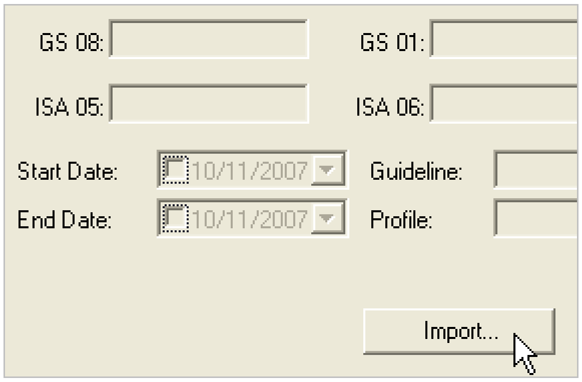
Scope Partner automation lookup files are specific to the system under which they were defined. Look for the files under **Foresight\Systems\<***system name***>\PartnerAutomationConfigs\<***version***>** .

Importing a lookup file

1. Click the system where the new partner automation lookup file will be used.

2. Choose **File | New | Envelope-Based Partner Automation Config**.

3. At the bottom, click **Import**:



GS 08:
GS 01:
ISA 05:
ISA 06:
Start Date: 10/11/2007
Guideline:
End Date: 10/11/2007
Profile:

4. Open the CSV file containing the partner automation.

   This file must be a lookup file for "envelope-based' partner automation, like SamplePartnerAutomation.csv in Instream's **Bin** directory.

   See Setting Up a Lookup Table for Envelope-Based Partner Automation in InStreamTPAutomation.pdf.

5. Rename the CSV file in the left pane, if you wish.

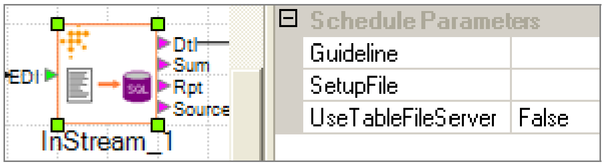
6. Save All.

Creating a lookup file from within Studio

1. Click the system where the new partner automation lookup file will be used.
2. Choose **File | New | Envelope-Based Partner Automation Config**.
3. In the left pane, type a name for the lookup file. Use file extension **csv**.
4. In the right pane, click **Add New**.
5. Fill in the ISA and GS fields that uniquely identify the partners that are to use a specific profile and/or guideline. Do not include trailing spaces.
6. Type the name of the guideline, or use the ... button to browse to it.
7. Type the name of the validation profile to use for this partner, or use the ... button to browse to it.
8. Save All.

Using the lookup table in a workflow

- In Instream's **$dir.ini** (Windows) or **fsdir.ini** (UNIX), remove the colon before **PARTNERAUTOMATION** and have the line point to the lookup table.
- In the workflow, the Instream component should not have a Guideline (which disables partner automation). It should not have a SetupFile if you want to use the ones from the lookup file:



Schedule Parameters	
Guideline	
SetupFile	
UseTableFileServer	False

Content-Sensitive Partner Automation Config

Partner automation is described in **InStreamTPAutomation.pdf** in Instream's Doc directory. You will need to be familiar with content-based partner automation before setting up a lookup file.

The lookup file lets Instream validation automatically select guideline and/or profile based on one or two values from the transaction set.

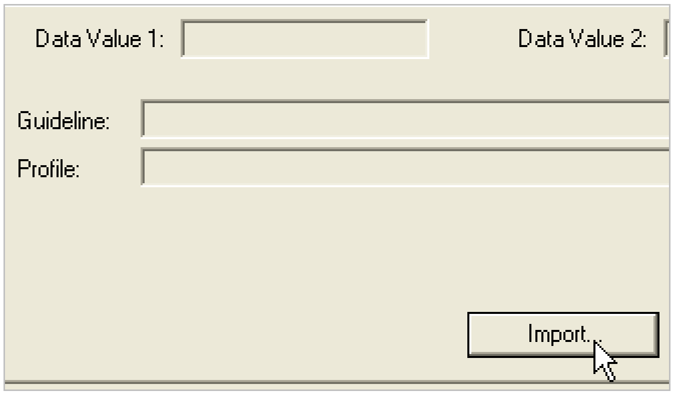
Scope Content-sensitive partner automation lookup files are specific to the system under which they were defined. Look for the files under **Foresight\Systems\ <***system name***>\PartnerAutomationConfigs\<***version***>** .

Importing a lookup file

1. Click the system where the new content-sensitive partner automation lookup file will be used.

2. Choose **File | New |Content-Sensitive Partner Automation Config**.

3. At the bottom, click **Import**:



4. Open the CSV file containing the partner automation specifications.

   This file must be a lookup file for "content-based' partner automation, like MyCBpartnerAutomation.csv in Instream's DemoData\CBTPAdemo directory.

   See Setting up your lookup table for content-based TPA in InStreamTPAutomation.pdf.

5. Rename the CSV file in the left pane, if you wish.

6. Save All.

Creating a lookup file from within Studio

1. Click the system where the new partner automation lookup file will be used.

2. Choose **File | New | Content-Sensitive Partner Automation Config**.

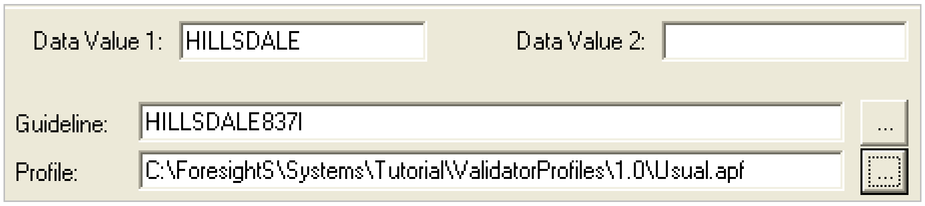
3. In the left pane, type a name for the lookup file. Use file extension **csv**.

4. In the right pane, click **Add New**.

5. Fill in one or both data value fields that uniquely identify the data that is to use a specific profile and/or guideline.

6. Type the name of the guideline, or use the … button to browse to it.

7. Type the name of the validation profile to use for this partner, or use the … button to browse to it.



Data Value 1: HILLSDALE

Data Value 2:

Guideline: HILLSDALE837I

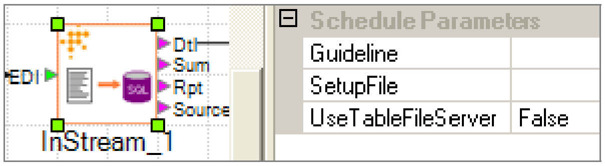
Profile: C:\ForesightS\Systems\Tutorial\ValidatorProfiles\1.0\Usual.apf

8. Save All.

Using the lookup table in a workflow

- In Instream’s **$dir.ini** (Windows) or **fsdir.ini** (UNIX), remove the colon before **CBPARTNERAUTOMATION** and have the line point to the lookup table.

- In the workflow, the Instream component should not have a Guideline (which disables partner automation). It should not have a SetupFile if you want to use the ones from the lookup file:



⊟ Schedule Parameters	
Guideline	
SetupFile	
UseTableFileServer	False

Partner Router Config

TPARouter is described in **TPARouter.pdf** in Instream's Doc directory. This file explains how to set up your configuration file to automatically route documents based on characteristics of the file.

STC Error Codes Editor

The STC Error Codes Editor lets you import or create a CSV file that lets Response Generator replace values in the 277CA STC segment with ones of your choosing. For details, please see Appendix G of ResponseGeneratorTechnicalManual.pdf.

Using an STC Error Codes Override File

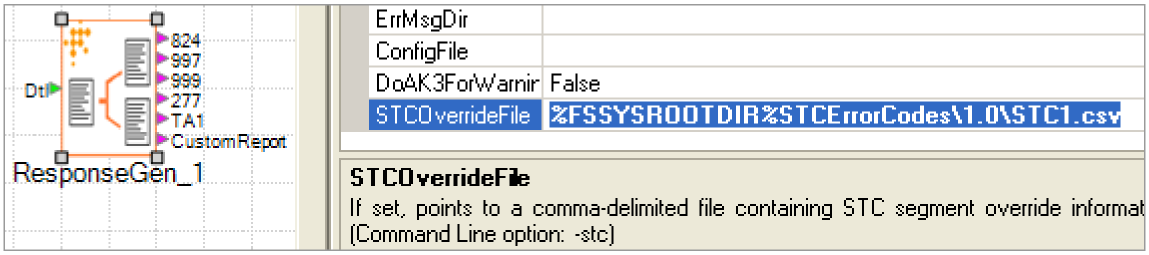
Instream validation profiles have file extension **csv** and are located in the **Foresight** directory under

Systems\\<*system*>\\ STCErrorCodes \\<*version*>

To use a profile in a workflow:

1. Click **ResponseGenerator** in the workflow.

2. Type the CSV file's path and filename in the STCOverrideFile parameter to the right:



ErrMsgDir	
ConfigFile	
DoAK3ForWarnir	False
STCOverrideFile	%FSSYSROOTDIR%STCErrorCodes\1.0\STC1.csv

STCOverrideFile
If set, points to a comma-delimited file containing STC segment override informat
(Command Line option: -stc)

Importing an STC Error Codes Override File

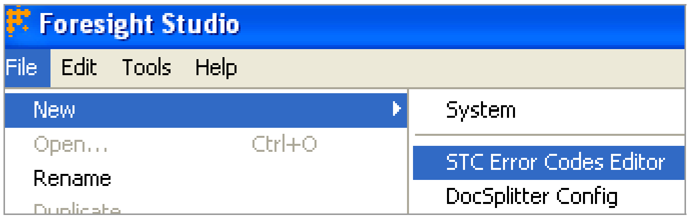
To import an existing STC override file:

1. Click the system where it will be used.
2. Choose **File | New | VSTC Error Codes Editor**.
3. Click the **Import** button at the bottom.
4. Find the CSV file and click **Open**.

Creating or Editing an STC Error Codes Override File

You can set up your own STC override files from within Studio:

1. Click a system where the new override file will be used.
2. Choose **File | New | STC Error Codes Editor**.

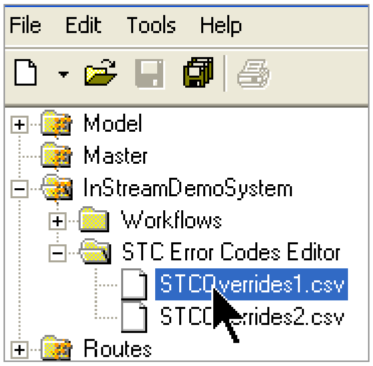


3. In the left pane, type a name for the file. Use file extension **csv**.
4. Click **Add New** at the bottom right.
5. Fill in any pertinent fields at the bottom, referring to Appendix G in ResponseGeneratorTechnicalManual.pdf for guidance.
6. Click **Add New** to insert the information.
7. Repeat for the next error number that you want to override.
8. When finished, **Save**.

Copying an STC Error Codes Override File

Once an STC override file has been added to one system, you can copy it to any other system:

1. Click the profile:



2. Use **File | Copy To**.

   The original and copy are separate. Changing one does not affect the other.

3. To rename the copy, click slowly on its name and then type the new name. Use file type **csv** and be sure that the new name does not exist in the Master system.

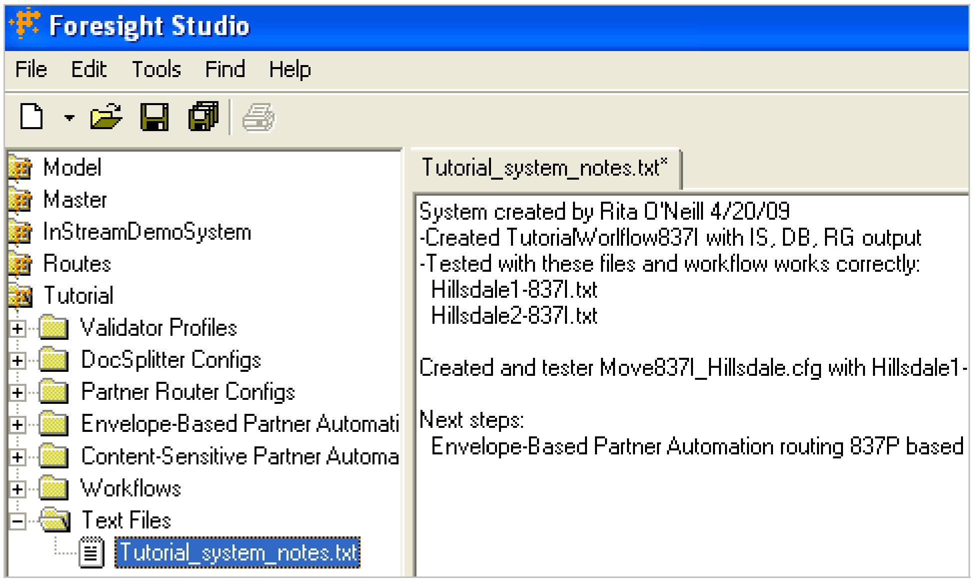
Text File

To create a text file from within Studio:

1. Choose **File | New | Text File**.

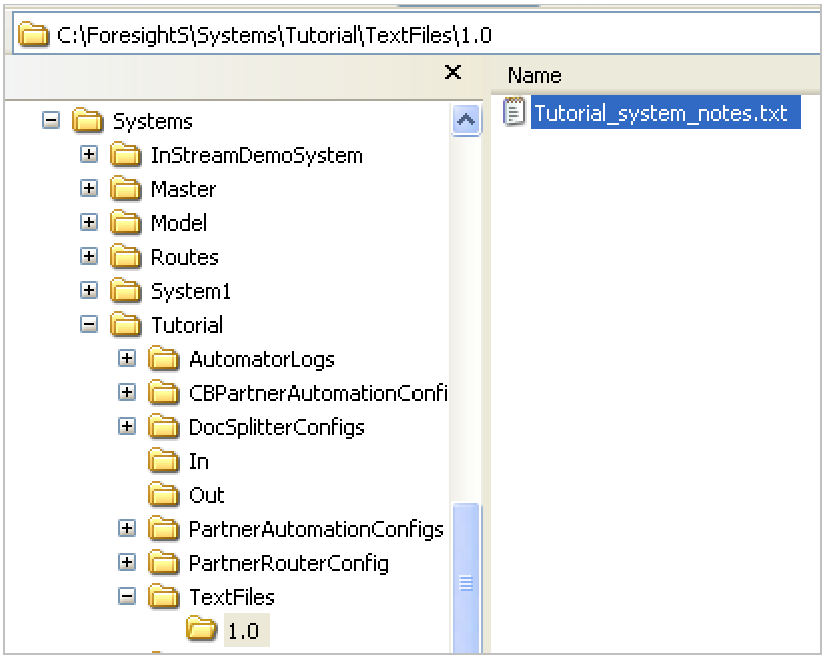
2. Rename it in the left pane, if desired.

3. Type the text and save.



```
System created by Rita O'Neill 4/20/09
-Created TutorialWorlflow837I with IS, DB, RG output
-Tested with these files and workflow works correctly:
  Hillsdale1-837I.txt
  Hillsdale2-837I.txt

Created and tester Move837I_Hillsdale.cfg with Hillsdale1-

Next steps:
  Envelope-Based Partner Automation routing 837P based
```

You can open it though Studio, or by locating it in its directory under the system's **TextFiles\***version* directory:



4 Workflows

Getting Started with Workflows

Please complete the tutorial on page 9 before reading this section.

Paths to Executable Components

Important: Before creating a workflow, customize the paths to the executables as described in Step 1 - Checking Paths to the Executables on page 11.

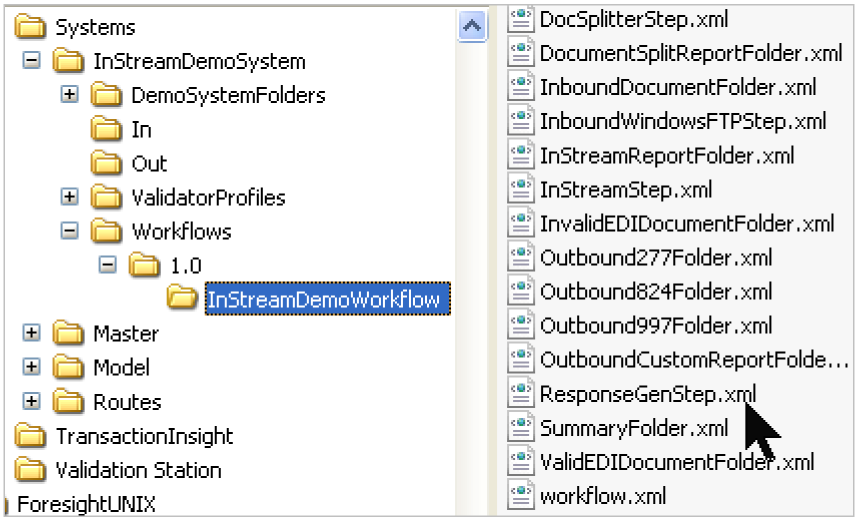
Using variables in paths makes your workflows easier to move to other systems. See Variables_for_Workflows.pdf.

If you created a workflow before customizing the paths

You can edit the XML file for each executable component with Notepad and correct the paths. This has to be done for each workflow that was created before the Model system paths were customized.

By default, the XML files are in

Foresight\Systems\<*system*> |Workflows\<*version*>\<*workflow*>



```
Systems
  InStreamDemoSystem
    DemoSystemFolders
    In
    Out
    ValidatorProfiles
    Workflows
      1.0
        InStreamDemoWorkflow
  Master
  Model
  Routes
TransactionInsight
Validation Station
ForesightUNIX
```

```
DocSplitterStep.xml
DocumentSplitReportFolder.xml
InboundDocumentFolder.xml
InboundWindowsFTPStep.xml
InStreamReportFolder.xml
InStreamStep.xml
InvalidEDIDocumentFolder.xml
Outbound277Folder.xml
Outbound824Folder.xml
Outbound997Folder.xml
OutboundCustomReportFolde...
ResponseGenStep.xml
SummaryFolder.xml
ValidEDIDocumentFolder.xml
workflow.xml
```

File Movement

This section describes file movement behind the scenes.

Every component in your workflow has **In**, **Process**, **Out**, and **Error** directories that operate as follows:

1. A file arrives in the **In** directory.

2. The component moves it to **Process**, takes action, and creates output files in the Process directory.

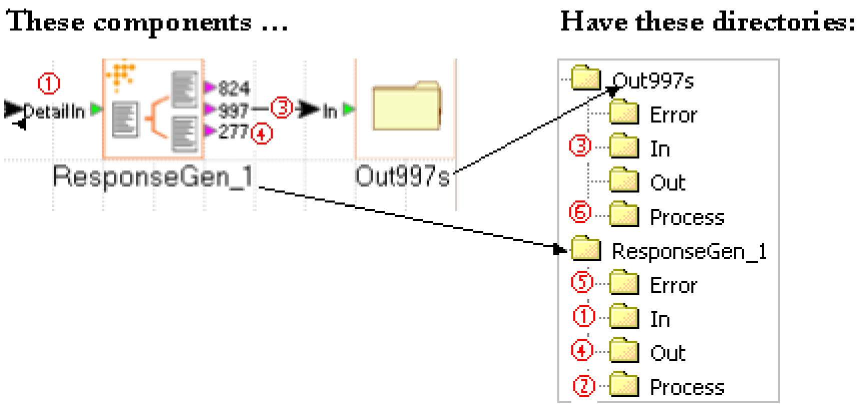
3. From there they are moved:

   a. Those that have a destination in the workflow are moved to the **In** directory of the next component in the workflow.

   b. Those with no destination go to the current component's **Out** directory if the component is successful. Some components delete leftover files from the Process directory. To see if a component does this, look at its Task tab under Component Templates.

   c. If the component returns an error, all files in the current component's **Process** directory go to the **Errors** directory. See Large File Copying on page 46.

Example



The general flow is:

① A validation detail results file arrives in ResponseGen_1's **In** directory.

② The file is moved to the **Process** directory, where Response Generator uses it to create 824, 997, and 277 files in the Process directory.

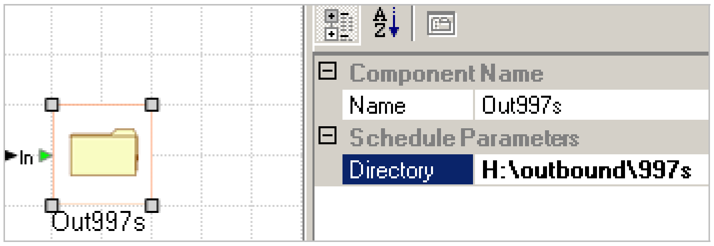
③ Because the workflow has a line connecting the 997 output to a directory called Out997s, the 997 is moved to Out997's **In** directory.

④ If Response Generator creates 824s and 277s, they are moved to ResponseGen_1's **Out** directory, since the workflow has no lines defining where they go.

⑤ If Response Generator cannot successfully complete its tasks, all files in Process go to ResponseGen_1's **Errors** directory.

⑥ The Out997s component finds the file in its **In** directory and moves it to **Process**.

It executes a **Move** command to move it to the directory specified for Out997s. In the example below, the properties pane for Out997s shows that this is **H:\outbound\997s**.



⊟ Component Name

Name	Out997s

⊟ Schedule Parameters

Directory	H:\outbound\997s

Large File Copying

You will not want your workflow to act on an incoming file until it has completely arrived. During the file transfer from one workflow component to another, this should not happen.

However, the program that copies files into your workflow may not lock each file until it is completely copied and ready for the workflow to act on it.

If you find incomplete files being processed by the workflow, you have some options:

- You can copy large files (especially if copied from another machine) to a temporary extension that the workflow does not process. Then, when finished, change it to the extension needed by the workflow. A rename command will do the trick.
- You can copy large files that come from another machine into a separate directory on the workflow machine. When completely copied, use a move or rename command to put them into the directory where they will be processed.

Error Handling

Please review File Movement above.

Setting up Success Codes

Each task in each component has a Success Code field where you can specify a code that means the task completed successfully.

You can customize this:

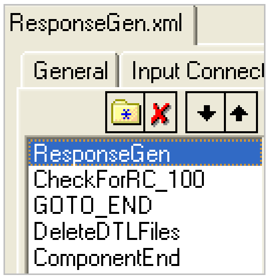
- From within Studio. Choose **Model** system **|** **Component Templates** **|** *component* **|** **Task** tab. Select a task and look at the Success Code field.
- In the workflow component's XML file with the <SuccessCode> tag.

Example

1. From Studio, open the Model system and choose **Component Templates |** **ResponseGen.xml | Task** tab.

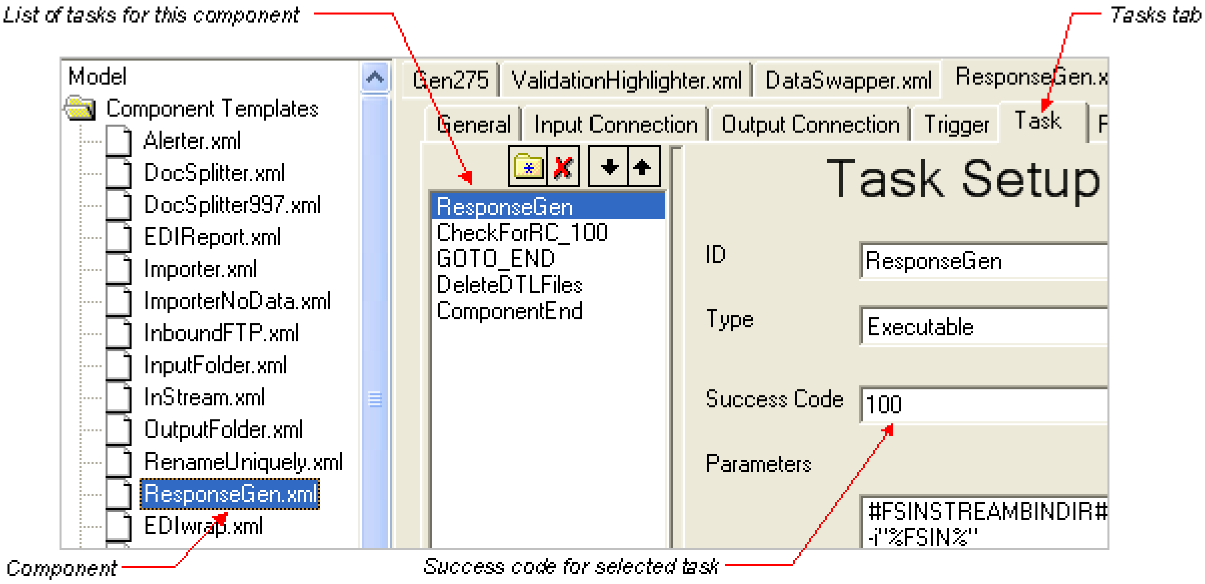
   This defines the behavior of the ResponseGen component.

2. The task list contains these tasks plus the ComponentEnd marker.



Currently, the ResponseGen task is selected and the rest of the screen shows information about it.

Since RespGen.exe sends 100 if it is able to complete its assignment, the Success Code field should also contain 100.



ID: ResponseGen

Type: Executable

Success Code: 100

Parameters: #FSINSTREAMBINDIR# -i"%FSIN%"

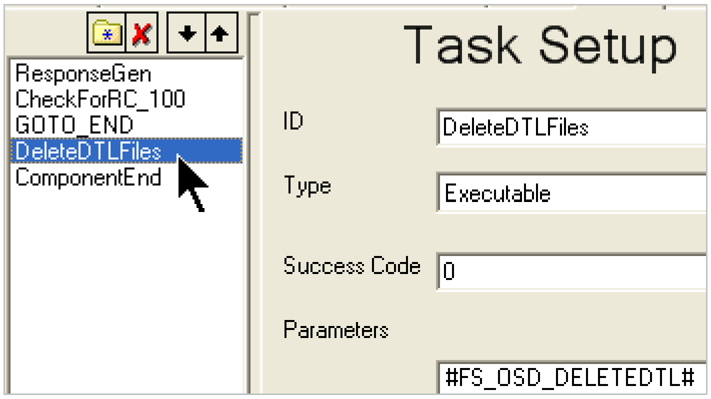
List of tasks for this component

Tasks tab

Component

Success code for selected task

3. Choose the **DeleteDTLFiles** task.



ID: DeleteDTLFiles

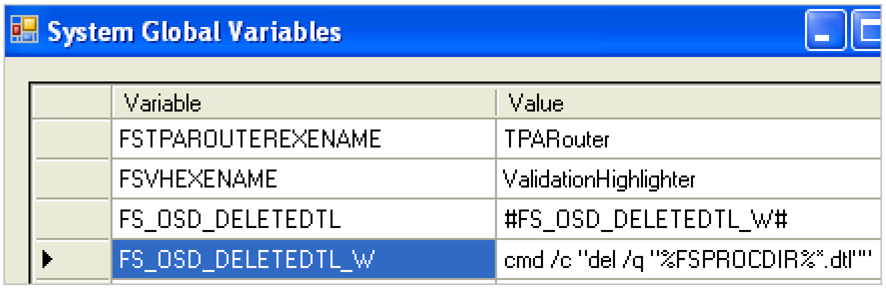
Type: Executable

Success Code: 0

Parameters: #FS_OSD_DELETEDTL#

This contains a workflow global variable. If the task succeeds, it sends a code of 0; therefore, its Success Code field should contain 0.

To see what the parameter means, choose **File | Edit Globals**. We can see that it is executing a DOS **del** command to delete the DTL file from the Component's Process directory:



Variable	Value
FSTPAROUTEREXENAME	TPARouter
FSVHEXENAME	ValidationHighlighter
FS_OSD_DELETEDTL	#FS_OSD_DELETEDTL_W#
FS_OSD_DELETEDTL_W	cmd /c "del /q "%FSPROCDIR%*.dtl""

What is a Task Error?

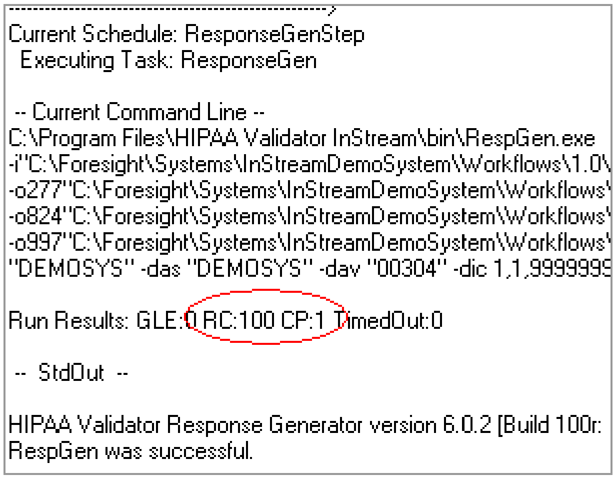
For each task in a component, both of these must be true for the task to be successful:

- The operating system create process code is 1.
- The task sends a return code that matches the Success Code for the task in the component template (see Setting up Success Codes on page 46).

Seeing Task Success in the Log

This section of a log shows RespGen.exe sending a return code of 100 and having a create process of 1.

Since a return code of 100 indicates success for Respgen.exe, this task succeeded.

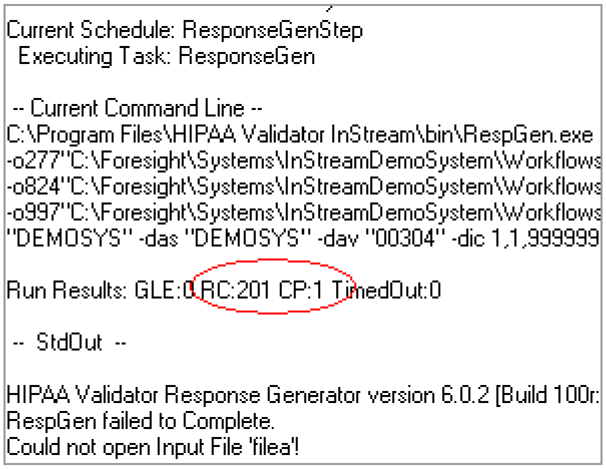


```
------------------------------------------------------>
Current Schedule: ResponseGenStep
  Executing Task: ResponseGen

 -- Current Command Line --
C:\Program Files\HIPAA Validator InStream\bin\RespGen.exe
-i"C:\Foresight\Systems\InStreamDemoSystem\Workflows\1.0\
-o277"C:\Foresight\Systems\InStreamDemoSystem\Workflows\
-o824"C:\Foresight\Systems\InStreamDemoSystem\Workflows\
-o997"C:\Foresight\Systems\InStreamDemoSystem\Workflows\
"DEMOSYS" -das "DEMOSYS" -dav "00304" -dic 1,1,9999999

Run Results: GLE:0 RC:100 CP:1 TimedOut:0

 --  StdOut  --

HIPAA Validator Response Generator version 6.0.2 [Build 100r:
RespGen was successful.
```

Seeing Task Errors in the Log

This section of a log shows RespGen.exe sending a return code of 201 and having a create process of 1.

Since a return code of 100 indicates success for Respgen.exe, this task failed. Even though it had a CP of 1, its return code causes it to be treated as an error.



```
Current Schedule: ResponseGenStep
 Executing Task: ResponseGen

 -- Current Command Line --
C:\Program Files\HIPAA Validator InStream\bin\RespGen.exe
-o277"C:\Foresight\Systems\InStreamDemoSystem\Workflows
-o824"C:\Foresight\Systems\InStreamDemoSystem\Workflows
-o997"C:\Foresight\Systems\InStreamDemoSystem\Workflows
"DEMOSYS" -das "DEMOSYS" -dav "00304" -dic 1,1,999999

Run Results: GLE:0 RC:201 CP:1 TimedOut:0

 --  StdOut  --

HIPAA Validator Response Generator version 6.0.2 [Build 100r:
RespGen failed to Complete.
Could not open Input File 'filea'!
```

Error Handling Behavior

Typical Error Handling Behavior

When a task error occurs (as described in What is a Task Error? above):

1. The workflow continues to the next task for the component.

2. When all tasks for the component are finished, there is a check made to see if any task had an error.

   If so, all files in the component's **Process** directory move to its **Errors** directory.

Changing Error Handling Behavior

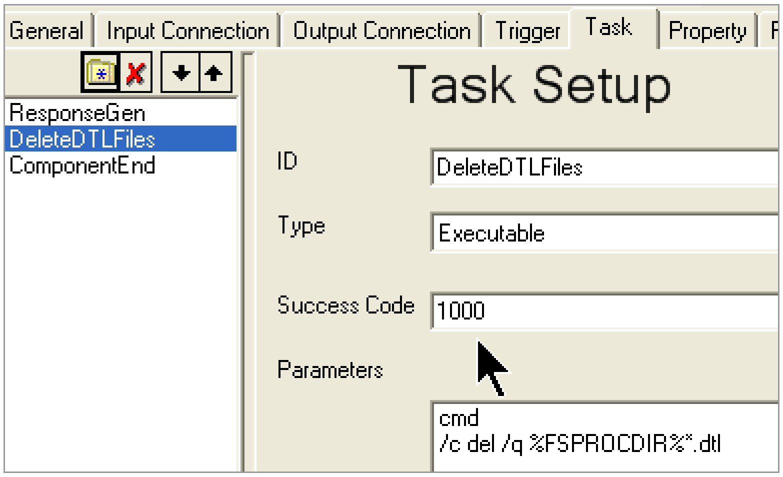
You can change error handling behavior by changing the success code for a task to one of these. These affect error handling for a task, not for the whole component.

1000 This disables error handling by having the workflow ignore the return code and create process.

1001 This disables error handling based on the return code. However, if create process is 0, then it does perform the normal error handling.

Example

Error handling is disabled for this DeleteDTLFiles task:



Task Setup

ID: DeleteDTLFiles

Type: Executable

Success Code: 1000

Parameters:

```
cmd
/c del /q %FSPROCDIR%*.dtl
```

File Extensions

This is a list of file extensions created by Studio components. They are case sensitive in UNIX but not in Windows.

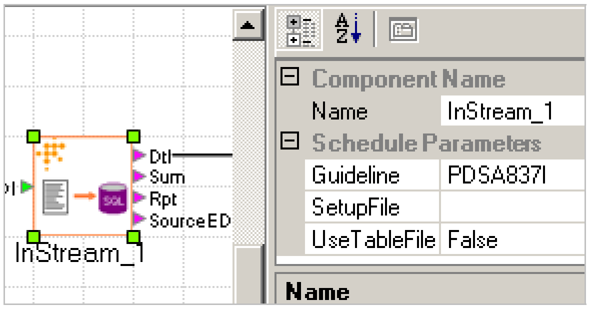
extension	Created by …
275	Generate275
277	ResponseGen - 277
824	ResponseGen - 824
997	ResponseGen - 997
999	ResponseGen - 999
audit_good audit_bad	Docsplitter-Dataswapper
bad	Docsplitter - bad data
dlm	Docsplitter - delimited report
dtl	Instream validation - detail results file
edi	EDIShuffler EDIWrap FFTranslator InboundFTP
ff	EDITranslator FFShuffler

extension	Created by …
good	Docsplitter - good data
html	EDIReport ValidationHighlighter
log	Importer
rpt	Instream validation - report card
sum	Instream validation - summary results file
swp	DataSwapper - output EDI
TA1	ResponseGen - TA1 output
trk	Track files - provide data to other Foresight components and are visible only if default behavior is overridden.
txt	ResponsGen - custom report output Dataswapper - report
xml	Docsplitter - XML report

Component Properties

When you click a component in the workflow, its properties display in the top right corner.

In this example, we can tell that InStream_1 is selected because it has green handles at each corner. Notice the properties pane to the right.



⊟ Component Name

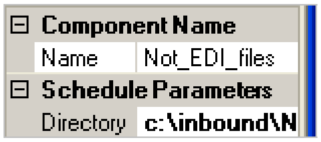
Name	InStream_1

⊟ Schedule Parameters

Guideline	PDSA837I
SetupFile	
UseTableFile	False

Name

To see the end of a long Schedule Parameter, like this:



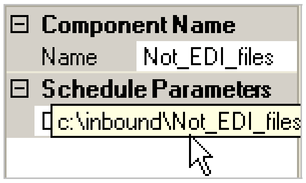
⊟ Component Name

Name	Not_EDI_files

⊟ Schedule Parameters

Directory	c:\inbound\N

… rest your mouse cursor on it to see a tooltip:



⊟ **Component Name**

Name Not_EDI_files

⊟ **Schedule Parameters**

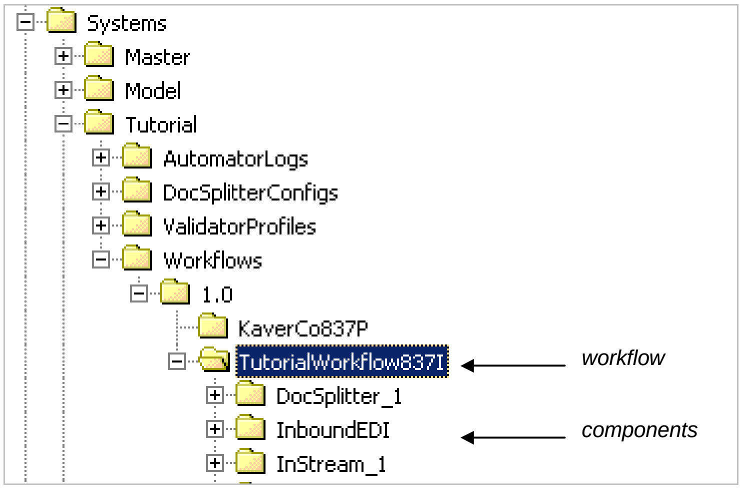
D c:\inbound\Not_EDI_files

The buttons above the list are:

Alphabetize the parameter list (component's Name field is inserted into list)

Organize the list into logical categories (component's Name field is above the list)

The component's Name field determines the name displayed on the workflow and the names of the directories under the workflow.



```
⊟ Systems
   ⊞ Master
   ⊞ Model
   ⊟ Tutorial
      ⊞ AutomatorLogs
      ⊞ DocSplitterConfigs
      ⊞ ValidatorProfiles
      ⊟ Workflows
         ⊟ 1.0
            KaverCo837P
            ⊟ TutorialWorkflow837I
               ⊞ DocSplitter_1
               ⊞ InboundEDI
               ⊞ InStream_1
```

You can change the name from within Studio by typing a new one in the component's Name field and then saving.

Most components have other fields that you can change. For details, see the properties table under each component:

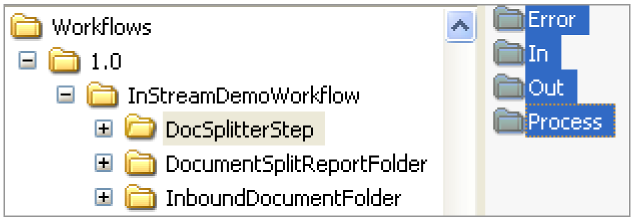
Workflow Maintenance

Checking Temporary Directories

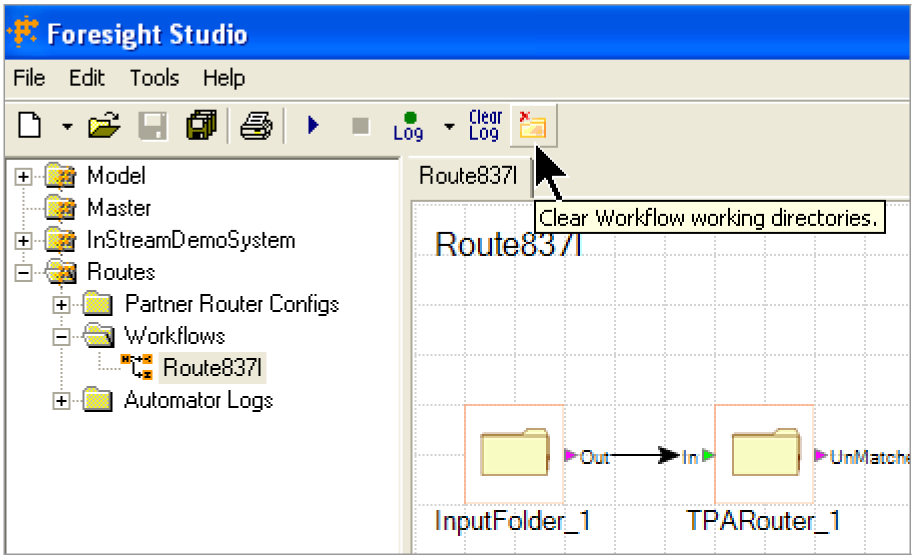
Check each component's **Out** and **Errors** directories frequently. These are in **Foresight\Systems\**<*system*>**\Workflows\**<*version*>\\<*workflow*>.

Clearing Temporary Directories

To delete all files from a workflow's Error, In, Out, and Process directories:



… open the workflow in Studio and press the toolbar button for clearing workflow directories:



Clear Workflow working directories.

Tracking down Malfunctions

If you suspect a malfunction, check the workflow's log file. For details, see **Automator.pdf** in Foresight Studio's **Doc** directory.

Location of Programs

If the log file shows an incorrect path to an executable, you can adjust its path as follows:

1. **Correct the model template**. These are copied each time you create a workflow:

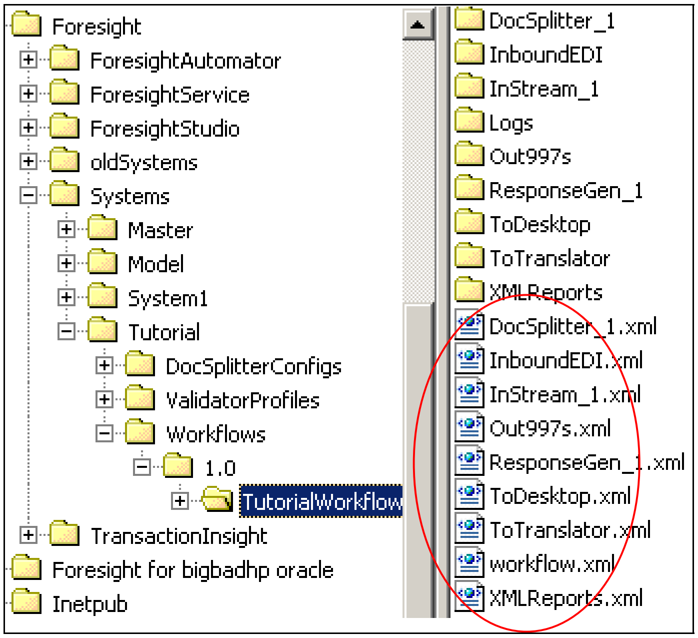
   a. At the top left, open Model | Component Templates | <*component*> | Task tab.

   b. Correct the path under Parameters and save.

2. **Correct the paths in each workflow that use the component:**

   a. Close Studio.

   b. Edit the **xml** files in each workflow's directory. This example workflow called TutorialWorkflow has nine **xml** files:

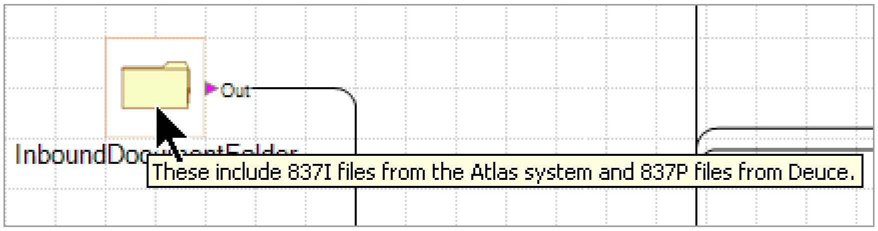


Use a text editor to edit the **xml** file of any component that has the wrong path. Correct the paths within the **Task** tags.

3. Close all **xml** files. Open Studio and test the workflow.

Annotations

You can add your own annotation to any component in a workflow:

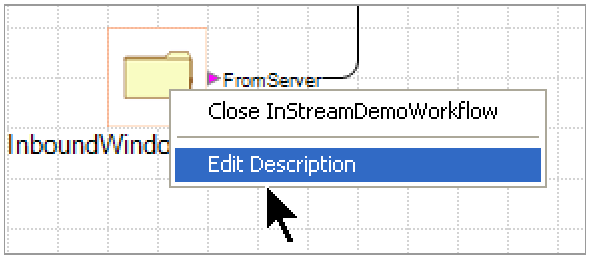


These include 837I files from the Atlas system and 837P files from Deuce.

To add an annotation:

1. Click the component so that its green handles appear.

2. Right-click on it and choose **Edit Description**.

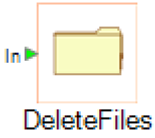


3. Type the description and **Save**.

4. Rest your mouse cursor on the component to see the annotation.

Components under I/O Tab

DeleteFiles

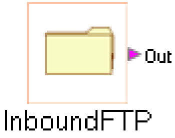


Trigger Arrival of a file in DeleteFiles' **In** directory.

What it does Deletes all files that are routed to it.

Properties	
Field	**Description**
None.	

InboundFTP



Trigger A timer periodically checks the **FTP** site.

What it does Moves a file from an FTP site to the **In** directory of the next component in the workflow. It can run on either Windows or UNIX systems and will retrieve one file with a specific name.

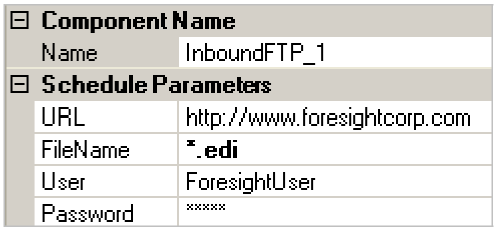
Configuring Set the parameters in Studio's component properties pane. Install **Curl.exe** as described below.

Properties	
Field	**Description**

URL	Type the URL of the FTP site. Examples:
	`216.12.134.170`
	`ftp.abc.downloads.gov`

FileName	Type the filename to retrieve. This can be an actual filename or it can include wildcards. The file will be saved to the same name on your local machine. The file is not deleted from the FTP site. This is transferred in ASCII mode.
User	Type the username to the FTP site.
Password	Type the password to the FTP site.

Example:



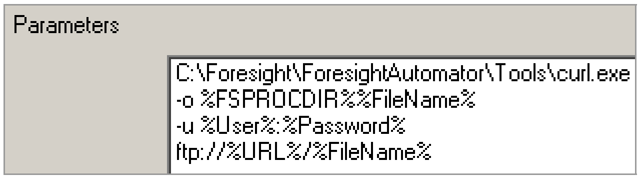
Component Name	
Name	InboundFTP_1
Schedule Parameters	
URL	http://www.foresightcorp.com
FileName	***.edi**
User	ForesightUser
Password	*****

Install Curl

InboundFTP uses **Curl.exe**, version 7 or later. This is a free command line FTP program that you can download from several Internet sites including http://curl.netmirror.org/.

We recommend that you place it in the **Foresight\ForesightAutomator\Tools** directory.

If it placed elsewhere, specify its path on the Tasks tab of the InboundFTP component template (see Overview of Component Templates on page 135):

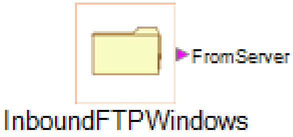


Parameters

```
C:\Foresight\ForesightAutomator\Tools\curl.exe
-o %FSPROCDIR%%FileName%
-u %User%:%Password%
ftp://%URL%/%FileName%
```

.

InboundFTPWindows



Trigger A timer periodically checks the **FTP** site.

What it does (Windows only) It moves one or more files from an FTP site to the **In** directory of the next component in the workflow.

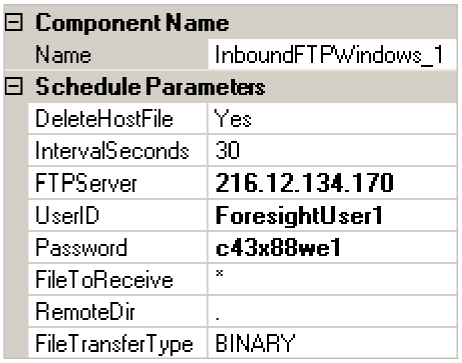
Configuring Set the parameters in Studio's component properties pane.

Properties	
Field	**Description**
DeleteHostFile	Choose **Yes** or **No**. If you choose **Yes**, the file is deleted after downloading.
IntervalSeconds	Set the schedule for checking the FTP site, in seconds. Make this long enough that the previous FTP checking is finished before the next one starts.

FTPServer	Type the URL of the FTP site. Examples:
	`216.12.134.170`
	`ftp.abc.downloads.gov`

UserID	Type the username to the FTP site.
Password	Type the password to the FTP site.
FileToReceive	Type the filename to retrieve. If your FTP site permits, this can contain wildcards. The files will be saved to the same names on your local machine.
RemoteDir	By default, this is a dot (.), meaning the login directory.
FileTransferType	Select BINARY or ASCII.

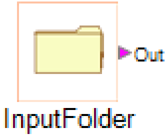
Example:



⊟ Component Name	
Name	InboundFTPWindows_1
⊟ Schedule Parameters	
DeleteHostFile	Yes
IntervalSeconds	30
FTPServer	**216.12.134.170**
UserID	**ForesightUser1**
Password	**c43x88we1**
FileToReceive	*
RemoteDir	.
FileTransferType	BINARY

InputFolder

(This component used to be called InboundFolder.)



Trigger A timer periodically checks the directory specified in InputFolder's properties pane.

What it does Uses the DOS move command to pick up files with the specified extension from the directory specified in properties.

Details for the curious:
The DOS Move command moves files with the extension specified in properties from the directory specified in properties to its own **Process** directory.

From there, Automator moves them to the **In** directory of the next component in the workflow.

Configuring Set these parameters in Studio's component properties pane.

Properties	
Field	**Description**
Directory	Type the directory path. Do not surround the path with quotation marks. Be sure that the directory exists and permissions are set to allow access. Variables like %FSSYSROOTDIR% are described in Variables_For_Workflows.pdf.
FileExtension	Type the extension of the files that are to be moved to the next component, or use a wildcard. Other files remain unmoved.
Frequency	Set the schedule for checking the directory, in seconds.

MQGet

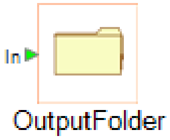
This component was removed from Foresight Studio.

MQPut

This component was removed from Foresight Studio.

OutputFolder

(This component used to be called OutboundFolder.)



Trigger Arrival of a file in OutboundFolder's **In** directory.

What it does Uses the Studio (Automator) Move command to move all files from its **In** directory to the directory specified in the component properties pane.

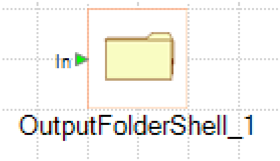
If you prefer to use the DOS Move command to move your files, use OutputFolderShell on pageT 61 instead.

If you would like to override default Studio functionality and retain Track (.trk) files, use OutputFolderTrk on page 61 instead. See Appendix A.

Configuring Set the final directory in Studio's component properties pane.

Properties	
Field	**Description**
Directory	Type the directory path where the files are to be placed. Do not use quotes around the directory path. Be sure that the directory exists and has the proper permissions. Variables like %FSSYSROOTDIR% are described in Variables_For_Workflows.pdf.

OutputFolderShell

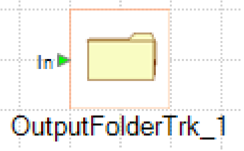


Trigger Arrival of a file in OutboundFolder's **In** directory.

What it does Uses the DOS Move command to move all files from its **In** directory to the directory specified in the component properties pane.

Configuring See OutputFolder on page 60.

OutputFolderTrk



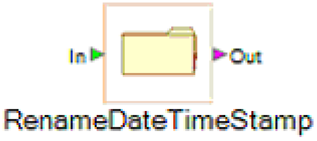
Trigger Arrival of a file in OutboundFolder's **In** directory.

What it does Uses the Studio (Automator) Move command to move all files from its **In** directory to the directory specified in the component properties pane, leaving track files in place.

Note: By default, track files are generated, utilized, and deleted by the system, leaving no trace behind. Undeleted track files can alter the behavior of the Foresight™ Operational Monitor component. See Appendix A.

Configuring See OutputFolder on page 60.

RenameDateTimeStamp



Trigger Arrival of a file in RenameDateTimeStamp's **In** directory.

What it does Appends the date and time to the filename.

Example:
Before: `file.edi`
After: `file_20070125135623756.edi`

The current resolution of the timestamp is 1/1000 of a second. With one workflow running, it is not possible to get the same timestamp and overwrite an existing file. If you are running workflows through multiple Automators, you could end up with the same timestamp but this would be unlikely.

Configuring Set FileExtension in Studio's component properties pane.

Properties	
Field	**Description**
FileExtension	Type the file extension of the files that are to be renamed. This can be: - A literal extension such as EDI or TXT; these are not case-sensitive - An asterisk for all files

ReportMailer

Trigger Arrival of a file in ReportMailer's **In** directory.

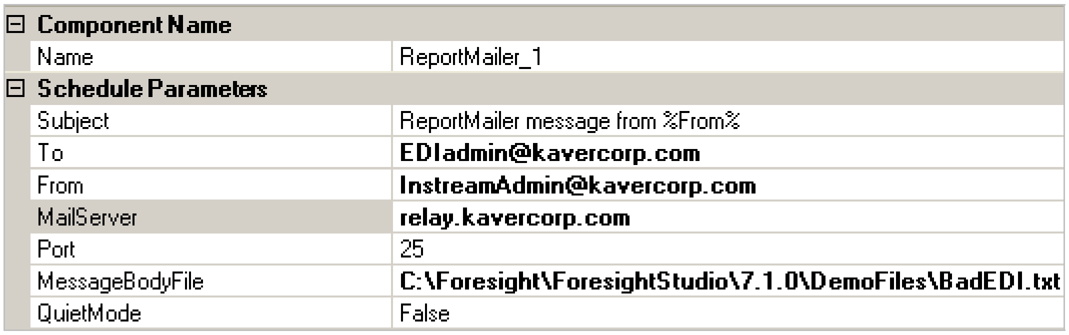
What it does Creates and sends an e-mail message to an address that you specify. The file coming into ReportMailer is attached to this e-mail.

Hint: Be aware of the file sizes being sent to ReportMailer, since they will become attachments to an e-mail.

Configuring Create a text file containing the message text.

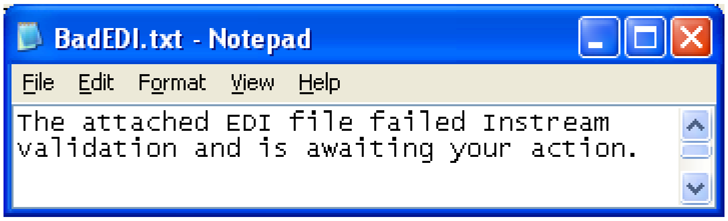
Properties	
Field	**Description**
Subject	Subject of the e-mail.
To	Send to this e-mail address. Separate multiple addresses with semi-colons.
From	e-mail address to put in the From field.
MailServer	SMTP server to use to send the e-mail.
Port	Port number for the SMTP request; usually 25.
MessageBodyFile	Path to file containing the text for the e-mail. If omitted, this default message is sent: `This message was created with Foresight ReportMailer, and may contain attached files.`
QuietMode	Do not actually send the e-mail. Useful when testing.

Example:



⊟ **Component Name**	
Name	ReportMailer_1
⊟ **Schedule Parameters**	
Subject	ReportMailer message from %From%
To	**EDIadmin@kavercorp.com**
From	**InstreamAdmin@kavercorp.com**
MailServer	**relay.kavercorp.com**
Port	25
MessageBodyFile	**C:\Foresight\ForesightStudio\7.1.0\DemoFiles\BadEDI.txt**
QuietMode	False

BadEDI.txt might contain something like this:

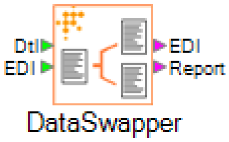


```
The attached EDI file failed Instream
validation and is awaiting your action.
```

Components under Validator Tab

DataSwapper

For more information about DataSwapper, see DataSwapper.pdf.



(This program used to be called Data eXchange)F

Trigger

Arrival of both an EDI file and a detail results file in its **In** directory. The extensions must be DTL and EDI (case sensitive in UNIX but not Windows). It matches them by the first part of their filename (before the dot). Unmatched files remain unprocessed in DataSwapper's In directory.

What it does

Runs DataSwapper (DataSwapper.exe) to replace specified data in the EDI file. The data to be replaced is identified by business rules in the guideline, which is identified in the Instream validation detail results file.

Files that are not sent to other locations are deleted after the component runs.

Input

EDI file and Instream validation detail results file. File extensions must be **edi** and **dtl**.

Output

New EDI file with replaced data.
Audit report showing what was replaced.

The output EDI file will have a file type of "swp" appended to the original name.

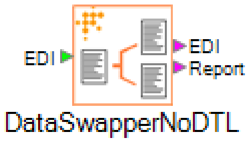
Example input:	File.edi
Example output	File.swp

Configuring Point to the DataSwapper setup file with the SetupFile parameter below.

Properties	
Field	**Description**
SetupFile	Path and filename to the setup file as described in DataSwapper.pdf. Do not use enclosing quotation marks. This parameter used to be called SubstitutionINIFile
InsertSegmentsAfter	If True, place inserted segments after the segment containing a GenerateFSUID or InsertSegment business rule. If False, the insertion goes before the segment that contains the business rule. This is equivalent to DataSwapper's **a** command line parameter.

DataSwapperNoDTL

For more information about DataSwapper, see DataSwapper.pdf.



Trigger Arrival of an EDI file in its **In** directory.

What it does Runs DataSwapper (DataSwapper.exe) to replace specified data in the EDI file. The data to be replaced is identified by business rules in the guideline, which you specify in the workflow parameters for DataSwapperNoDTL.

Input EDI file.

Output New EDI file with replaced data.
Audit report showing what was replaced.

The output EDI file will have a file type of "swp" appended to the original name.

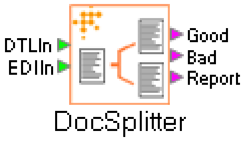
Example Input:	File.edi
Example Output:	File.swp

Configuring Point to the DataSwapper setup file with the SetupFile parameter below.

Properties	
Field	**Description**
SetupFile	Path and filename to the setup file as described in DataSwapper.pdf. Do not use enclosing quotation marks. This parameter used to be called SubstitutionINIFile.
Guideline	Mandatory. Type the name of the guideline to be used for splitting; omit the file extension ".STD." This is equivalent to DataSwapper's **g** command line parameter.

DocSplitter

For more information about DocSplitter, see DocumentSplitterTechnicalManual.pdf.



Trigger Arrival of both an EDI file and a detail results file in its **In** directory. It matches them by the first part of their filename (before a dot). Unmatched files remain unprocessed in Docsplitter's In directory.

What it does Runs Docsplitter (DocSplitter.exe) by using a validation detail results file to identify the location of errors for splitting.

Input EDI file and Instream validation detail results file. File extensions must be **edi** and **dtl**.

Output Valid EDI file(s)
Invalid EDI file(s)
XML or delimited report

If using content-based splitting, output may be multiple valid files and multiple invalid files.

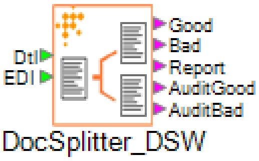
Configuring Define under **DocSplitter Configs** in Studio.

Set the parameters in Studio's component properties pane, including pointing to the INI file with SetupFile.

Properties	
Field	**Description**
ReportContents	Choose the contents of Docsplitter's XML or delimited report. This is equivalent to Docsplitter's **cv**, **ci**, and **cb** command line parameters.
ReportFormat	Choose the format of Docsplitter's report. It should match ReportExtension. This is equivalent to Docsplitter's **f** command line parameters.
ReportExtension	Choose the file extension for Docsplitter's report. It should match ReportFormat.
SetupFile	Type the path and filename of the Docsplitter INI file. Do not use quotes around the path. This can be any Docsplitter INI file. If it was defined in Studio, it will be in this directory: **Foresight\Systems\<*system*>\DocSplitterConfigs** This is equivalent to Docsplitter's **s** command line parameter.
LogLevel	Choose **Normal** or **Debug** log levels for the Docsplitter report file. Debug is much slower. This is equivalent to Docsplitter's **l** command line parameter.

DocSplitter_DSW

For more information about DocSplitter, see DocumentSplitterTechnicalManual.pdf.



Trigger Same as the DocSplitter component.

What it does Runs Docsplitter (DocSplitter.exe) by using a validation detail results file to identify the location of errors for splitting. Then swaps data in the valid and/or invalid output files. See Appendix H: Split-and-Swap in DocumentSplitterTechnicalManual.pdf.

Input Same as the DocSplitter component.

Output

Valid EDI file(s)
Invalid EDI file(s)
XML or delimited report
Dataswapper audit files

If using content-based splitting, output may be multiple valid files and multiple invalid files.

Configuring

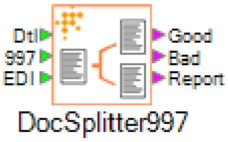
Same as the DocSplitter component. You will also need a Dataswapper configuration file, which is described in Appendix H: Split-and-Swap in DocumentSplitterTechnicalManual.pdf.

Set the parameters in Studio's component properties pane, including pointing to the INI file with SetupFile.

Properties	
Field	**Description**
ReportContents	Same as the DocSplitter component.
ReportFormat	Same as the DocSplitter component.
ReportExtension	Same as the DocSplitter component.
SetupFile	Same as the DocSplitter component.
LogLevel	Same as the DocSplitter component.
SwapSettings	Choose which Docsplitter output files should have their data swapped: valid, invalid, both, or none. This is equivalent to Docsplitter's **wv**, **wi**, and **wb** command line parameters
DSWSetupFile	Type the path and filename of the Dataswapper INI file. Do not use quotes around the path. This is equivalent to Docsplitter's **z** command line parameter.

DocSplitter997

For more information about DocSplitter 997-based splitting, see Appendix E of DocumentSplitterTechnicalManual.pdf.



Trigger Arrival of both an EDI file for splitting and a 997 file in the **In** directory. These must have file extensions **edi** and **997**. It matches them by the first part of the filename (before the dot).

What it does Runs Docsplitter (DocSplitter.exe) by using a 997 to identify the error locations.

Input 997 and another EDI file.

Output A valid and an invalid EDI file; an XML or delimited report.

Configuring Define INI files under DocSplitter Configs in Studio.

Set the parameters in Studio's component properties pane, including pointing to the INI file with the SetupFile parameter.

Properties	
Field	**Description**
ReportContents	Choose the contents of Docsplitter's XML or delimited report. This is equivalent to Docsplitter's **cv**, **ci**, and **cb** command line parameters.
ReportFormat	Choose the format of Docsplitter's report. It should match ReportExtension. This is equivalent to Docsplitter's **f** command line parameters.
ReportExtension	Choose the file extension for Docsplitter's report. It should match ReportFormat.
SetupFile	Type the path and filename of the Docsplitter INI file. Do not use quotes around the path. This can be any Docsplitter INI file. If it was defined in Studio, it will be in this directory: **Foresight\Systems\<*system*>\DocSplitterConfigs** This is equivalent to Docsplitter's **s** command line parameter.
LogLevel	Choose the log level for Docsplitter's report file. This is equivalent to Docsplitter's **l** command line parameter.

DSIdentSplit

Retrieves documents that have been archived. For more information, see Archiver.pdf.

DTL
EDI
TXT
Good
Bad
Report
DSIdentSplit

Trigger

Arrival of these in its In directory:

- **EDI file** with file type EDI

- **Detail results file** with file type DTL

- **Text file** of identifiers with type TXT

It matches them by the first part of their filename (before a dot). Unmatched files remain unprocessed in DSIdentSplit’s **In** directory.

What it does

Runs Docsplitter (DocSplitter.exe) to create a good and a bad EDI file containing only the documents listed in the TXT file.

Input

- EDI file with file extension **edi**. This is the original EDI file validated by Instream.

- Instream validation detail results file for the EDI file, with file extension **dtl**. This file must contain IDENT records with RuleID of I and a unique FSUID for each document.

- Text file containing a list of Foresight identification numbers (FSUID) of the documents that should be selected for output. File extension must be **txt**.

Output

- Good EDI file containing all valid selected documents

- Bad EDI file containing all invalid selected documents XML or delimited report

- Report in XML or delimited format

Configuring

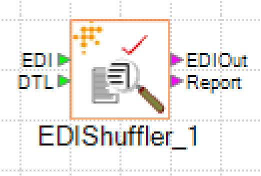
Set the parameters in Studio’s component properties pane, including pointing to the INI file with SetupFile.

Properties	
Field	**Description**
ReportFormat	Choose the format of Docsplitter’s report. It should match ReportExtension. This is equivalent to Docsplitter’s **f** command line parameters.
ReportExtension	Choose the file extension for Docsplitter’s report. It should match ReportFormat.

Properties	
Field	**Description**
SetupFile	Type the path and filename of the file containing the FSUIDs to be selected. Do not use quotes around the path. This is a plain text file containing one FSUID per line.
LogLevel	Choose **Normal** or **Debug** log levels for the Docsplitter report file. Debug is much slower. This is equivalent to Docsplitter's **l** command line parameter.

EDIShuffler

For details, see Shuffler.pdf.



Trigger

Arrival of both of these in its In directory:

- EDI file with file type edi

- Detail results file with file type dtl

It matches them by the first part of their filename (before a dot). Unmatched files remain unprocessed in EDIShuffler's In directory.

What it does

For CMS users. Runs Shuffler.exe to add CTR and STC segments to the EDI.

Input

- EDI file that has been validated by Instream.

- Instream validation detail results file for the EDI file.

Output

- EDI file containing an STC segment below any segment that generates an error number listed in the STCDefinitionTable.

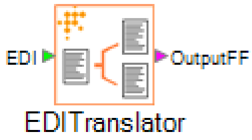
- Flat file containing CMS – Error Report Record (specific to TIBCO Foresight™ Instream® MAC Adapter customers). This file, which has the extension .err, is placed in the output folder.

Configuring Set the parameters in Studio's component properties pane.

Properties	
Field	**Description**
CreateCRT	Choose True or False to determine whether a CRT record is to be placed before each ISA. This record contains a unique number for each ISA.
STCDefinitionTable	Type the path and filename of the STC override table, as described in Appendix G of ResponseGeneratorTechnicalManual.pdf. This file contains the error numbers that will generate an STC segment, and information about what that STC segment should include. Do not use quotes around the path.

EDI

For information about TIBCO Foresight™ Instream® Translator, see Translator.pdf.



Trigger Arrival of an EDI file.

What it does Runs Translator.exe to convert an EDI file into a flat file.

Input EDI file with any file extension.

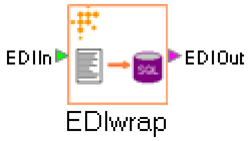
Output Flat file with the same name as the input file and file extension ff.

Configuring Point to the translation map in Studio's component properties pane.

Properties	
Field	**Description**
TranslationMap	Type the path and filename of the file containing the translation map. Do not use quotes around the path. This is an XML file created by Translation Tool.exe.

EDIWrap

For more information about EDIWrap, see EDIwrap.pdf.



Trigger The arrival of an EDI file in EDIWrap's **In** directory.

What it does Runs **EDIWrap.exe**, which can remove and/or add line breaks, and change the segment terminator. See **EDIWrap.pdf** in Instream's **Doc** directory.

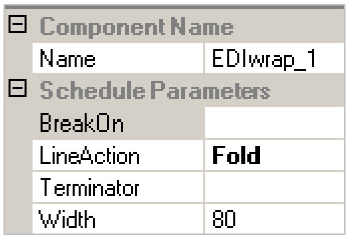
Input Arrival of an EDI file with file extension EDI. This is case sensitive in UNIX but not in Windows.

Output Reformatted EDI file.

Configuring Set the parameters in Studio's component properties pane.

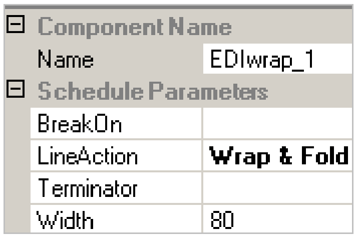
Properties	
Field	**Description**
LineAction	**Fold** - Insert a CR/LF after reaching the width specified in width (if width is not specified, records will be folded after the 80th character). **Wrap and Fold** – Remove all CR/LF and then chop at the specified width. New segments do not cause a line break. **Unwrap** – Breaks each segment into a new line by inserting CR/LF after each segment terminator. Terminator must be specified. The segment terminator must be the same for all interchanges in the file. **None** – Line breaks are not changed. This is used when you are specifying a terminator.
Terminator	Replace the CR/LF segment terminator with another one. *xx* specifies which character or hex code is to be used as the segment terminator in the output file. If *xx* is a single character, EDIWRAP assumes it is the segment terminator; If *xx* is two characters, EDIWRAP assumes it is the 2-digit hex ASCII representation of the segment terminator. The input file is assumed to have segments separated only by CR/LF.
BreakOn	If you want to include a line break before certain segments, type their tags here, separated by commas. Examples: ISA ISA,GS
Width	The output record length (default: 80). Each ISA will start on a new line, and the previous incomplete line will be padded with blanks.
NoEndPadding	Choose whether fixed-width output should pad the last record to the specified width.

Example: Break lines longer than 80 bytes:



⊟ Component Name	
Name	EDIwrap_1
⊟ Schedule Parameters	
BreakOn	
LineAction	**Fold**
Terminator	
Width	80

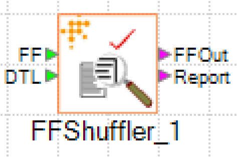
Example: Remove all CR/LF and break into 80-byte lines:



⊟ Component Name	
Name	EDIwrap_1
⊟ Schedule Parameters	
BreakOn	
LineAction	**Wrap & Fold**
Terminator	
Width	80

FFShuffler

For details, see Shuffler.pdf.



Trigger

Arrival of both of these in its In directory:

- **flat file** with file type **ff**

- **Detail results file** with file type **dtl**

It matches them by the first part of their filename (before a dot). Unmatched files remain unprocessed in FFShuffler's **In** directory.

What it does

For CMS users. Runs Shuffler.exe to add CTR and STC records to the flat file.

Input

- Flat file that has been validated by Instream.

- Instream validation detail results file for the EDI file.

Output

- Flat file containing an STC record below any record that generates an error number listed in the STCDefinitionTable.

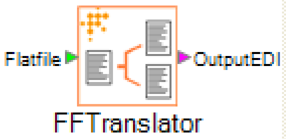
- Flat file containing CMS – Error Report Record (specific to Instream® MAC Adapter customers). This file, which has the extension .err, is placed in the output folder.

Configuring Set the parameters in Studio's component properties pane.

Properties	
Field	**Description**
CreateCTR	Choose True or False to determine whether a CTR record is to be placed iteration of the outermost loop. Each time this record appears, it contains a unique number.
STCDefinitionTable	Type the path and filename of the STC override table, as described in Appendix G of ResponseGeneratorTechnicalManual.pdf. This file contains the error numbers that will generate an STC segment. Do not use quotes around the path. This is a plain text file containing one error number per line.

FFTranslator

For information about Instream® Translator, see Translator.pdf.



Trigger Arrival of a flat file.

What it does Runs Translator.exe to convert a flat file into an EDI file.

Input Flat file with any file extension.

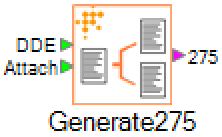
Output EDI file with same name as the flat file, and file extension edi.

Configuring Point to the translation map in Studio's component properties pane.

Properties	
Field	**Description**
TranslationMap	Type the path and filename of the file containing the translation map. Do not use quotes around the path. This is an XML file created by Translation Tool.exe.

Generate275

For more information, see 275atForesight.pdf.



Trigger The arrival of a DDE and matching Attach file in Generate275's **In** directory. Files must have the same filename and extensions must be DDE and Attach.

What it does Creates a 275 transaction with the attachment in its BIN02 element.

Input DDE file and Attach files.

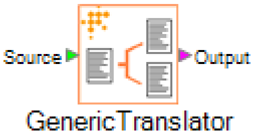
Output File containing one interchange, with one 275 containing one BIN02, containing the attachment.

Configuring Set this parameter in Studio's component properties pane.

Properties	
Field	**Description**
Stylesheet	This points to the stylesheet used for formatting the CDA header information included in the BIN02.

GenericTranslator

For more information about translation, see Translator.pdf.



Trigger The arrival of a data file in GenericTranslator's **In** directory.

What it does Runs Foresight's Translator (Translator.exe) to translate any format of data that can be handled by Translator.

Input A data file.

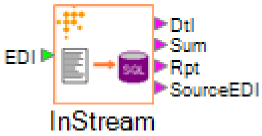
Output A translated data file.

Configuring Set the parameters in Studio's component properties pane.

Properties	
Field	**Description**
OutboundExt	Type the extension to be used for files created by Translator. The outbound extension must be different from the inbound extension and it must not be a wildcard. Otherwise there will be a conflict and the output file may be deleted. This is equivalent to Translator's **o** command line parameter.
TranslationMap	Type the filename and extension of the map. Do not include a path. The map must be in Instream's Database directory along with the source and target guidelines/schemas. This is equivalent to Translator's **t** command line parameter.

Instream

For more information about Instream validation, see InstreamValidationTechnicalManual.pdf.



Trigger The arrival of an EDI file in Instream's **In** directory.

What it does Changes the file extension to **edi** and then runs Instream validation (HVInStream.exe) to validate the EDI data.

Input EDI.

Output Summary results file
Detail results file
Report card (formatted summary of results).

It also forwards the EDI file that it validated.

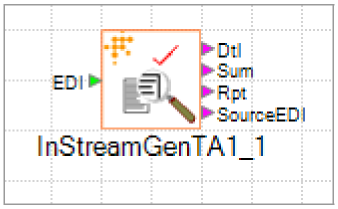
Configuring Set the parameters in Studio's component properties pane.

You can define your own profile (APF file) using Validator Profile within Studio, and then point to it with **SetupFile** under the component properties.

Properties	
Field	**Description**
Guideline	Type the name of a guideline to use for validating. Do not include the file extension STD. This is equivalent to Instream's **g** command line parameter.
SetupFile	Type the path and filename of the validation profile (APF file). Do not use quotes around the path. This can be any APF file. If it was defined in Studio or imported into Studio, it will be in this directory: **Foresight\Systems\<*system*>\ValidatorProfiles\ <*version*>** If omitted, $fsdeflt.apf in Instream's Bin directory will be used. This is equivalent to Instream's **s** command line parameter.
UseTableFileServer	Choose whether to use Instream's table file server. See Appendix B in InstreamValidationTechnicalManual.pdf. This is equivalent to Instream's **f** command line parameter.
AlternateIniLocation	**c**onfiguration file location - the high-level directory containing $dir.ini (Windows) or fsdir.ini (UNIX). This is equivalent to Instream validation's **-c** command-line parameter.
UserMessage	Free-form text. Lets you insert whatever text you'd like in a GEN record with number 15078. This is for your own use. Transaction Insight® can display it. This is equivalent to Instream validation's **-u** command-line parameter.

InstreamGenTA1

For more information about Instream validation, see InstreamValidationTechnicalManual.pdf.



Trigger The arrival of an EDI file in Instream's **In** directory.

What it does If TPA fails due to a missing GS01 or GS08, a TA1 file is created and stored along with other output in the user-specified directory.

Input EDI

Output All output is stored in the specified directory.

Summary results file
Detail results file
Report card (formatted summary of results).

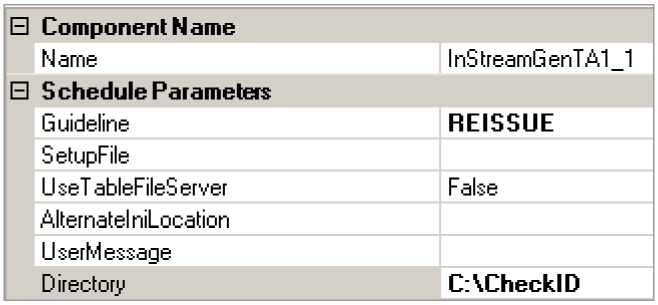
It also forwards the EDI file that it validated.

Configuring Set the parameters in Studio's component properties pane.

You can define your own profile (APF file) using Validator Profile within Studio, and then point to it with **SetupFile** under the component properties.

Properties	
Field	**Description**
Guideline	Type the name of a guideline to use for validating. Do not include the file extension STD. This is equivalent to Instream's **g** command line parameter.
SetupFile	Type the path and filename of the validation profile (APF file). Do not use quotes around the path. This can be any APF file. If it was defined in Studio or imported into Studio, it will be in this directory: **Foresight\Systems\<*system*>\ValidatorProfiles\ <*version*>** If omitted, $fsdeflt.apf in Instream's Bin directory will be used. This is equivalent to Instream's **s** command line parameter.
UseTableFileServer	Choose whether to use Instream's table file server. See Appendix B in **Instream**ValidationTechnicalManual.pdf. This is equivalent to Instream's **f** command line parameter.
AlternateIniLocation	**c**onfiguration file location - the high-level directory containing $dir.ini (Windows) or fsdir.ini (UNIX). This is equivalent to Instream's **-c** command line parameter.
UserMessage	Free-form text. Lets you insert whatever text you'd like in a GEN record with number 15078. This is for your own use. Transaction Insight can display it. This is equivalent to Instream's **-u** command line parameter.
Directory	The directory in which output should be stored.

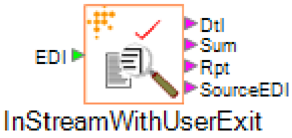
Example:



⊟ Component Name	
Name	InStreamGenTA1_1
⊟ Schedule Parameters	
Guideline	**REISSUE**
SetupFile	
UseTableFileServer	False
AlternateIniLocation	
UserMessage	
Directory	**C:\CheckID**

InStreamWithUserExit

For more information about User Exits, see BusinessRules.pdf.



Trigger The arrival of an EDI file in Instream's **In** directory.

What it does Changes the file extension to **edi** and then runs Instream validation (HVInStream.exe) to validate the EDI data. The guideline contains UserExit or UserExitWithWait business rules.

Input EDI.

Output Summary results file
Detail results file
Report card (formatted summary of results).

It also forwards the EDI file that it validated.

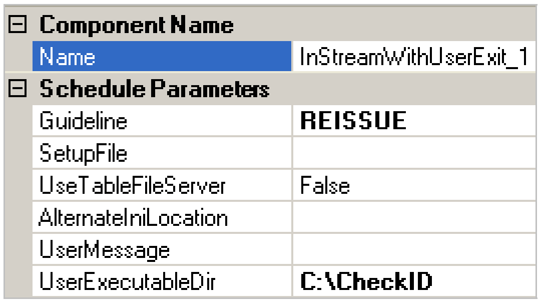
Configuring Set the parameters in Studio's component properties pane.

You can define your own profile (APF file) using Validator Profile within Studio, and then point to it with **SetupFile** under the component properties.

Properties	
Field	**Description**
Guideline	Type the name of a guideline to use for validating. Do not include the file extension STD. The guideline contains UserExit or UserExitWithWait business rules. This is equivalent to Instream's **g** command line parameter.
SetupFile	Type the path and filename of the validation profile (APF file). Do not use quotes around the path. This can be any APF file. If it was defined in Studio or imported into Studio, it will be in this directory: **Foresight\Systems\<*system*>\ValidatorProfiles\ <*version*>** If omitted, $fsdeflt.apf in Instream's Bin directory will be used. This is equivalent to Instream's **s** command line parameter.
UseTableFileServer	Choose whether to use Instream's table file server. See Appendix B in **Instream**ValidationTechnicalManual.pdf. This is equivalent to Instream's **f** command line parameter.

AlternateIniLocation	configuration file location - the high-level directory containing $dir.ini (Windows) or fsdir.ini (UNIX). This is equivalent to Instream validation's -c command-line parameter.
UserMessage	Free-form text. Lets you insert whatever text you'd like in a GEN record with number 15078. This is for your own use. Transaction Insight can display it. This is equivalent to Instream validation's -u command-line parameter.
UserExecutableDir	The directory containing the program called by the guideline's UserExit or UserExitWithWait business rule.

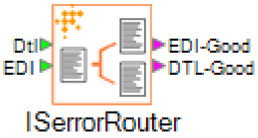
Example:



⊟ **Component Name**	
Name	InStreamWithUserExit_1
⊟ **Schedule Parameters**	
Guideline	**REISSUE**
SetupFile	
UseTableFileServer	False
AlternateIniLocation	
UserMessage	
UserExecutableDir	**C:\CheckID**

ISerrorRouter

For more information about ISErrors, see ISErrors.pdf.



See also Example that Routes on Errors in the EDI on page 130.

Trigger

The arrival of a detail file and EDI file in ISerrorRouter's **In** directory. The part of the filename before the first dot must be the same. Extensions must be **dtl** and **edi**. Unmatched files remain unprocessed in ISerrorRouter's In directory.

What it does

Runs ISErrors.exe to route files based on whether their validation detail results files report errors.

Input

EDI file and corresponding Instream validation detail results file.

Output

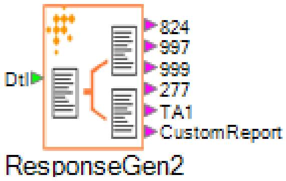
Route good EDI and detail files to the next component in the workflow. Place bad EDI and detail files in a directory specified under parameters.

Configuring Set the parameters in Studio's component properties pane. Create a setup file as described in ISErrors.pdf. Do not use ShowAllDTLRecords=1 in the setup file.

Properties	
Field	**Description**
BadDirectory	Specify where bad EDI and detail files go.
SetupFile	Type the path and filename of an ISErrors setup file. This is equivalent to ISErrors' **-s** command line parameter.

ResponseGen

For more information about Response Generator, see ResponseGeneratorTechnicalManual.pdf.



Trigger The arrival of a detail file in ResponseGen's **In** directory.

What it does Runs Response Generator (RespGen.exe) to create response EDI documents.

Input Instream validation detail results file.

Output 997, 999, 824, and 277 files, and custom reports.

Configuring Set the parameters in Studio's component properties pane.

ResponseGen Properties	
Field	**Description**
DeleteZeroLength Results	Choose whether to delete Response Generator output files that are empty. If set to true, empty files are deleted. The default is false. This is equivalent to Response Generator's **nz** command line parameter.

ResponseGen Properties

Field	Description
StartingST02	Type the starting transaction set control number for the outbound EDI file. The default is 1. - For X12, this is the starting ST02 number. - For EDIFACT, this is the starting UNH01 number. This is equivalent to Response Generator’s **c** command line parameter.
StartingGS06	Starting Group Control Number for the first outbound GS in each interchange. The default is 1. - For X12, this is the starting GS06 number. - For EDIFACT, this is the starting UNG05 number. This is equivalent to Response Generator’s **dac** command line parameter.
ApplicationReceiverID	Type the Application Receiver ID (GS03) for the outbound GS. The default is the first Application Sender ID (GS02) in the inbound file. This is equivalent to Response Generator’s **dar** command line parameter.
ApplicationSenderID	Type the Application Sender ID (GS02) for the outbound GS. The default is the first Application Receiver ID (GS03) in the inbound file. This is equivalent to Response Generator’s **das** command line parameter.
ApplicationVersion	Type the Application Version (GS08) for the outbound GS. This is equivalent to Response Generator’s **dav** command line parameter.
InterchangeControl NumberMode	How should Response Generator choose the starting interchange control number? **Number** Use the starting number in InterchangeControl NumberStart **File** Use the starting number in a file (specified with InterchangeControl NumberFile) **None** Use RespGen default ICN generation. (Default) This is equivalent to Response Generator’s **dic** command line parameter.
InterchangeControl NumberFile	Specifies the file where control number is to be saved between runs, when InterchangeControlNumberMode is set to File. This is equivalent to Response Generator’s **dic** command line parameter.

ResponseGen Properties	
Field	**Description**
InterchangeControl NumberStart	Type the starting interchange control number. The default is 1. This is equivalent to Response Generator’s **dic** command line parameter.
InterchangeControl NumberMin	Type the minimum interchange control number. The default is 1. This is equivalent to Response Generator’s **dic** command line parameter.
InterchangeControl NumberMax	Type the maximum interchange control number. The default is 999999999. This is equivalent to Response Generator’s **dic** command line parameter.
InterchangeReceiverID	Type the Interchange Receiver ID to use in the responses’ ISA07 and ISA08. The default is the value in the originating interchange. This is equivalent to Response Generator’s **dir** command line parameter.
InterchangeSenderID	Type the Interchange Sender ID to use in the responses’ ISA05 and ISA06. The default is the value in the originating interchange. This is equivalent to Response Generator’s **dis** command line parameter.
InterchangeUsage Indicator	Choose the Interchange Usage Indicator (ISA15) for the responses’ ISA: - Production (default) - Testing - Input (which uses the value from the input file's ISA15) This is equivalent to Response Generator’s **diu** command line parameter.
EDIFACTVersionNumber	Specify a UNH02.02 version number value to be used instead of the existing value. For EDIFACT CONTRL responses only. This is equivalent to Response Generator’s **dev** command line parameter.
EDIFACTRelease Number	Specify a UNH02.03 release number value to be used instead of the existing value. For EDIFACT CONTRL responses only. This is equivalent to Response Generator’s **der** command line parameter.
EDIFACTCtrlAC	Specify a UNH02.04 controlling agency code value to be used instead of the existing value. For EDIFACT CONTRL responses only. This is equivalent to Response Generator’s **dec** command line parameter.

ResponseGen Properties	
Field	**Description**
EDIFACTAssocAC	Specify a UNH02.05 association assigned code value e to be used instead of the existing value. For EDIFACT CONTRL responses only. This is equivalent to Response Generator's **dea** command line parameter.
GenerateUNA	Choose whether to generate a starting UNA segment when requesting EDIFACT CONTRL responses. If set to false (the default), Response Generator generates a UNA if any delimiters are different from the defaults. This is equivalent to Response Generator's **fedifact_gen_una** command line parameter.
EDIFACTNoUCM	If set to true, Response Generator suppresses UCM segments in CONTRL response messages. Default is false. This is equivalent to Response Generator's **fedifact_no_ucm** command line parameter.
EDIFACTDoICOnly	If set to true, the EDIFACT CONTRL document generates a response for the status of the Interchange only. Default is false. This is equivalent to Response Generator's **fedifact_ic_only** command line parameter.
CRContactNumber	Sets the value for the user-defined ContactPhone custom report template variable. This is equivalent to Response Generator's **dcc** command line parameter.
CROriginalFilename	Sets the value for the OriginalFname custom report template variable. This is equivalent to Response Generator's **dco** command line parameter.
CRControlNumber	Sets the value for the UserGSControlNum custom report template variable. This user-specified number should be thought of as a GS-level trace number; it is not the same as the GS06 number. This is equivalent to Response Generator's **dcg** command line parameter.
Do864Wrapper	Choose whether to create an 864 wrapper around a custom report. If set to true, all enveloping flags are valid. Default is false. This is equivalent to Response Generator's **fdo864wrapper** command line parameter.

ResponseGen Properties	
Field	**Description**
864PurposeCode	Type the 864 purpose code. This is equivalent to Response Generator’s **dcp** command line parameter.
864Description	Type the 864 description field. This is equivalent to Response Generator’s **dcd** command line parameter.
RejectErrorSeverity	Choose the error severity that will result in a rejected transaction set. 0 = Ignore 1 = Information 2 = Warning 3 = Error (Default) 4 = Serious Error 5 = User Level #1 6 = User Level #2 This is equivalent to Response Generator’s **er** command line parameter.
WarnErrorSeverity	Choose the highest severity that results in the transaction set being accepted with errors. 0 = Ignore 1 = Information 2 = Warning (Default) 3 = Error 4 = Serious Error 5 = User Level #1 6 = User Level #2 This is equivalent to Response Generator’s **ew** command line parameter.
GenerateInterchange AndFunctionalGroup	If set to false, interchange and functional group envelopes are not generated. Default is true. This is equivalent to Response Generator’s **ge** command line parameter.
GenTA1	Choose a setting for generating TA1 segments. **Based on contents** means based on the contents of the ISA14. Default is Never. This is equivalent to Response Generator’s **TA1** command line parameter.

ResponseGen Properties	
Field	**Description**
SegmentDelimiter	Type the segment delimiter. Default is ~. This is equivalent to Response Generator's **-l** command line parameter.
ElementDelimiter	Type the element delimiter. Default is *. This is equivalent to Response Generator's **-l** command line parameter.
SubelementDelimiter	Type the subelement delimiter. Default is : . This is equivalent to Response Generator's **-l** command line parameter.
RepetitionSeparator	Type the element repetition separator. Default is ^. This is equivalent to Response Generator's **-l** command line parameter.
NewLines	Choose whether a new-line sequence is to follow each segment delimiter in the EDI output. Default is true. **False** is equivalent to Response Generator's **n** command line parameter.
PartialFunctional GroupAcceptance	Choose whether partial functional group acceptance is permitted when generating a 997 or 824 response. If set to true (the default) partial functional group acceptance is permitted, and a functional group is rejected only if all transaction sets are rejected. If set to false, partial functional group acceptance is not permitted. If any transaction is rejected, the entire functional group is rejected. **False** is equivalent to Response Generator's **np** command line parameter.
Generate277	Choose whether to generate a 277. Default is true. This is equivalent to Response Generator's **o277** command line parameter.
Generate824	Choose whether to generate an 824. Default is true. This is equivalent to Response Generator's **o824** command line parameter.

ResponseGen Properties	
Field	**Description**
Generate997	Choose whether to generate a 997: **True** (example output filename: *myfile*.997) (default) **True - Use #ICount#** (example output filename: *myfile_1*.997) **True - Use #ICount#_#GCount#** (example output filename: *myfile_1_1*.997) **False** This is equivalent to Response Generator's **o997** command line parameter.
Generate999	Choose whether to generate a 999. Choices are the same as Generate997. Default is false. This is equivalent to Response Generator's **o999** command line parameter.
GenerateTA1	Choose whether to generate a TA1. Choices are the same as Generate997. Default is false. This is equivalent to Response Generator's **oTA1** command line parameter.
GenerateCustomReport	Choose whether to generate a custom report. Choices are the same as Generate997. Default is false. This is equivalent to Response Generator's **otext** command line parameter.
Generate824x166	Optionally generate 824s to the 824x166 standard. Default is false. This is equivalent to the **fdo824x166** command line parameter.
Generate824x186	Optionally generate 824s to the 824x186 standard. Default is false. This is equivalent to the **fdo824x186** command line parameter.
Generate277x070	If true, the 277 response will conform to the 3070x070A Health Care Payer Unsolicited Claim Status. This is equivalent to Response Generator's **fdo277x070** command line parameter.
Generate277x167	If true, creates a 277 that conforms to the 4040x167 standard. This is equivalent to Response Generator's fdo277x167 command line parameter.

ResponseGen Properties	
Field	**Description**
GenerateHIPAA824	Choose whether to generate 824s to the -824x166 standard, if the source is less than 5010, or -824x186 standard, if the source is 5010. This is incompatible with Generate824, Generate824x166, and Generate824x186. This is equivalent to Response Generator's **fdo_hipaa_824** command line parameter.
GenerateNonErrata999	In cases where a HIPAA 999 will be generated, will generate a non-errata (005010x231) 999. This is equivalent to Response Generator's **fdo_999NE** command line parameter.
CustomReportTemplate	This component is only valid if custom report generation is enabled. Type the path and filename of the custom report template. This component is **required** to generate a custom report. This is equivalent to Response Generator's **tpl** command line parameter.
CustomReportContents	This component is only valid if custom report generation is enabled. Choose what claims are to go in the report. **All** – all claims go in the report. **Errors only** – claims with errors go in the report. (default) **Error free only -** claims with no errors go in the report. This is equivalent to Response Generator's **te** command line parameter.
CustomReportError Types	This component is only valid if custom report generation is enabled. Type the numbers of the error types to go in the report. Example: 3457 will put errors of types 3, 4, 5, and 7 in the report. If left blank, all error types go in the report. This is equivalent to Response Generator's **tel** command line parameter.
CustomReportSeverity Levels	This component is only valid if custom report generation is enabled. Type the severity numbers to go in the report. Example: 345 will put errors of severity 3, 4, and 5 in the report. If left blank, all severity levels go in the report.

ResponseGen Properties	
Field	**Description**
PartialTransactionSet Acceptance	Choose whether partial transaction set acceptance is permitted when generating a 997. If set to true, partial transaction set acceptance is permitted, and a transaction set is rejected only if all transaction sets are rejected. If set to false (the default), partial transaction set acceptance is not permitted. If any transaction is rejected, the entire transaction set is rejected. **True** is equivalent to Response Generator’s **pt** command line parameter.
Strict997	Choose whether strict 997 response processing is used. If set to true, 997s or 999s will only show error types 1 and 2 and 824s will only show application error types (usually types 3-7). If set to false (the default), if only one of these response types is generated (997, 999, or 824), ALL errors, regardless of type, are put in the one response type that was generated. **True** is equivalent to Response Generator’s **s** command line parameter.
DoAK901E	Choose whether to use E in the 997/999 AK901 to indicate Accepted with Errors when all AK501s generated for the functional group are set to A or E. If set to true, the 997/999 AK901 value will contain E if any AK501 value contains E and none contain R. If set to false (the default), the AK901 contains A under these circumstances. This is equivalent to Response Generator’s **fak901E** command line parameter.
Do277CA	Choose whether to generates a 277CA instead of a 277U (must be accompanied by -**o277**).
Verbosity	Choose the amount of output to go in the text report: - No Output - Errors Only (default) - Debug (this option generates a large amount of output) This is equivalent to Response Generator’s **v** command line parameter.

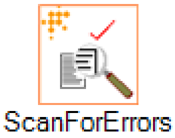
ResponseGen Properties	
Field	**Description**
ErrMsgDirOptions	Sets how ResponseGen locates its error message text files: - Default (use default location) - Specify (user sets the directory) - ReadFromIni (read from the **dir.ini** file). This is equivalent to Response Generator's **-cd** and **–u** command line parameters.
ErrMsgDir	Specifies where ResponseGen can find its error text files. **ErrMsgDirOptions** (above) must be set to **Specify**. This is equivalent to Response Generator's **-cd** command line parameters.
ConfigFile	Type the path and filename to the Response Generator config file. This is equivalent to Response Generator's **z** command line parameter.
DOAK3ForWarnings	Choose whether, for warnings level errors (severity of 2), Response Generator should insert AK3/AK4 in 997s and IK3/IK4 in 999s. The default is false. This is equivalent to Response Generator's **fak3** command line parameter.
ShowAllClaims	Choose whether to show information for all claims in all response documents. If set to false (the default), the responses show only claims with errors. This is equivalent to Response Generator's **fsa** command line parameter.
ShowAllClaims277	Choose whether to show information for all claims in 277 responses. If set to false (the default), the 277 shows only claims with errors. This is equivalent to Response Generator's **fsa277** command line parameter.
ShowAllClaims824	Choose whether to show information for all claims in 824 responses. If set to false (the default), the 824 shows only claims with errors. This is equivalent to Response Generator's **fsa824** command line parameter.
DoVersionPassthrough	If set to true, Response Generator passes the ISA and GS version numbers for the source document through unchanged. ST03/AK0103/AK0203s are generated based on the presence or absence of ST03 in the source document. Default is false. This is equivalent to Response Generator's **fver_pass** command line parameter.

ResponseGen Properties	
Field	**Description**
997GroupOnly	If set to true, 997s and 999s are generated with group information (AK1 and AK9 segments) only (no AK2/AK3/AK4/AK5). The default is false. This is equivalent to Response Generator's **f997_group_only** command line parameter.
UseHighest99x	If set to true, generates a 999 if the source document is a 5010 and a 997 if it is a 4010. This option overrides any previously specified output (such as -o997 or –o999). The default is false. This is equivalent to Response Generator's **fuse_highest_99x** command line parameter
LooseScrCheck	If set to true, Response Generator skips groups containing invalid source document types and continues processing. If set to false (the default), Response Generator terminates with an error. This is equivalent to Response Generator's **floose_scr_check** command line parameter.
DoGS06Passthrough	If set to true, outgoing response documents (864, 997, 824, and 277) will use the GS06 (Group Control Number) of the incoming document. The default is false. Note that, if you use an 864 wrapped around a custom report, this parameter causes the MIT01 (currently set to use GS06) to use the value in the incoming GS06. This is equivalent to Response Generator's **fdo_gs06_passthrough** command line parameter.
DoGS08Passthrough	If set to true, the outgoing response documents (864, 997, 824, 277, and 864) will use the GS08 (Version/Release/Industry Identifier Code) of the incoming document. The default is false. Note that, if you use an 864 wrapped around a custom report, this parameter causes the MIT01 (currently set to use GS08) to use the value in the incoming GS08. This is equivalent to Response Generator's **fdo_gs08_passthrough** command line parameter. This option is incompatible with the -dav (ApplicationVersion) option, and Response Generator fails if both are set.
DoLoopRollup	If set to true, 277U responses will have one loop per provider, even if the incoming document had multiple 2000A loops for a single provider. The default is false. This is equivalent to Response Generator's **fdo_loop_rollup** command line parameter.

ResponseGen Properties

Field	Description
STCOverrideFile	Type the path and filename of the STC override file, which is used to specify different and/or multiple STC codes for 277CA. Please see Appendix G in ResponseGeneratorTechnicalManual.pdf. This is equivalent to Response Generator's **stc** command line parameter.
RespGenOverrideFile	Type the path and filename of the RespGen override file, which is used for customization of code assignment for the AK9, IK3/4/5,TA1, CMS, and RED06 and code elements. Please see Appendix H in ResponseGeneratorTechnicalManual.pdf.

ScanForErrors



Trigger Timer.

What it does Checks the Error directories of each component in its workflow. If it finds a file there, it moves it to a directory called ReportedErrors under its current location and sends an e-mail.

Input None.

Output E-mail.

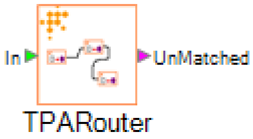
Configuring Set the parameters in Studio's component properties pane.

Properties

Field	Description
IntervalSeconds	Set the schedule for executing ScanForErrors, in seconds.
MailServer	SMTP server to use to send the e-mail.
Port	Port number for the SMTP request; usually 25.
To	Send to this e-mail address. Separate multiple addresses with semi-colons.
From	e-mail address to put in the From field.
QuietMode	Prevents e-mails from being sent. Useful during testing.

TPARouter

For more information about TPARouter, see TPARouter.pdf.



Trigger Arrival of a file.

What it does Uses TPARouter.exe to route files to various directories, based on characteristics of the file (filename, file extension and file size), values in the enveloping (EDI only), or whether the files contain X12 or EDIFACT EDI. It also can change the file extension if the file contains X12 EDI.

Files that match criteria in the configuration file will go to the directories specified in the configuration file. Unmatched files continue to the next workflow component.

You can create the configuration file from within Studio with **File | New | Partner Router Config**.

Input File.

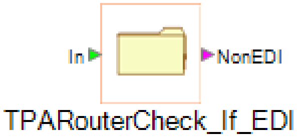
Output Files routed to specified directories or to the next component in the workflow. The directories for files matched in the configuration file need not be components in the workflow.

Configuring Set the parameters in Studio's component properties pane. Also, set up a TPARouter configuration file as described in TPARouter.pdf.

Properties	
Field	**Description**
TPARouterConfig	The path and filename of a TPARouter configuration file. Do not use quotes around the path. This corresponds to the TPARouter command-line -c parameter.
Debug	Writes detailed processing messages to the Automator log file.

TPARouterCheck_If_EDI

For more information, see TPARouter.pdf.



Trigger Arrival of a file.

What it does Uses TPARouter.exe to move files containing X12 EDI to a specific directory.

Files that do not contain X12 EDI continue to the next workflow component.

Input File.

Output EDI files are routed to a specified directory or to the next component in the workflow. The directory for the EDI files need not be a component in the workflow.

Configuring Set the directory in Studio's component properties pane.

Properties	
Field	**Description**
ValidEDI_Dir	The directory where the X12 EDI files should go. This corresponds to the TPARouter command-line -o parameter.

Components under Standard Tab

AddExtension

Trigger Arrival of a file.

What it does Adds a dot and file extension to the filename.

Examples	
Original Name	**Renamed to**
file1.txt	file1.txt.EDI
file2	file2.EDI
file3.txt.edi	file3.txt.edi.EDI

Input File.

Output Same file, different name.

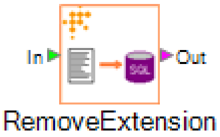
Configuring Set the parameters in Studio's component properties pane.

Properties	
Field	**Description**
FileExtension	This file extension will be added. The dot is automatically included; do not type it in the Schedule Parameters.

EDIReport

This component has been replaced by ValidationHighlighter and is only included for compatibility with previously created workflows.

RemoveExtension



Trigger Arrival of a file.

What it does Removes the last file extension.

Examples	
Original Name	**Renamed to**
file1.txt	file1
file2	file2
file3.edi	file3
file4.ediZ	file4
file5.Zedi	file5
file6.txt.edi	file6.txt
file7.edi.txt	file7.edi
file8.EDI	file8

Input File.

Output Same file without the last extension.

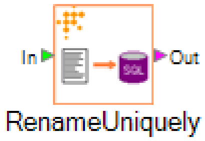
Configuring Set the parameters in Studio's component properties pane.

Properties	
Field	**Description**
none	

Rename

This component was removed in Foresight Studio 7.2 since it is not portable between operating systems. To change an extension, use RemoveExtension and then AddExtension.

RenameUniquely



Trigger Arrival of a file.

What it does Renames the file. The original filename will have a dot and unique number inserted before the extension: The number resets each time Automator starts.

```
file1.edi -> file1.0.txt
file1.edi -> file1.1.txt
file2.edi -> file2.2.txt
More.edi  -> More.3.txt
```

Input File.

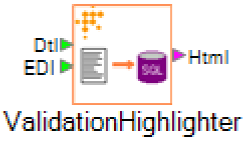
Output Same file, different name.

Configuring Set the parameters in Studio's component properties pane.

Properties	
Field	**Description**
FileExtension	Files with this extension will be renamed. Do not use a preceding dot. You can use wildcards.

ValidationHighlighter

For more information, see ValidationHighlighter.pdf.



Trigger Arrival of a file.

What it does Uses ValidationHighlighter.exe to create an HTML report of validation results.

Input Validation detail results file and EDI file.

Output HTML file containing EDI segments and corresponding messages.

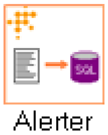
Configuring

Set the following in Studio's component properties pane. Also, set up a configuration file as described in ValidationHighlighter.pdf.

Properties	
Field	**Description**
OutputFragment	If true, output will be in fragment mode. This means it does not contain the html, head, and body tags. Fragment mode produces a file that can be included in an existing page. This corresponds to ValidationHighlighter's -f command-line parameter.
AlwaysShowEnvelope	If true, ISA/IEA, GS/GE, and ST/SE envelope information is included in the report. This corresponds to ValidationHighlighter's -showenv command-line parameter.
SeverityToOutput	Select the minimum severity for messages that are included in the report. Default is all messages. This corresponds to ValidationHighlighter's -s command-line parameter.
LibLinkPath	Type the path to the HTML help files. Include a trailing slash but not a filename. Default is `../LibraryLinks/` This corresponds to ValidationHighlighter's -liblinkpath command-line parameter.
StyleSheetPath	Type the path to a CSS style sheet for the html help, if you are using one. No default. This corresponds to ValidationHighlighter's -csspath command-line parameter.
IniFile	Type the path and filename to the ValidationHighlighter INI file.

Components under TI Tab

Alerter



Trigger A timer runs Alerter every 30 seconds or as defined by the Frequency parameter.

What it does Runs Alerter (AlertProcessor.exe) to detect situations under which Transaction Insight alerts should be sent out, and then sends out the e-mail alerts.

Input Information from the TI database.

Output E-mail.

Configuring Set up alerts on TI's Alerts page.

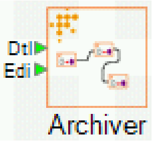
The TI database is identified in AlertProcessor.exe.config under the Foresight\TransactionInsight directory.

Set the Frequency parameter (listed below) in Studio's component properties pane.

The directory that contains AlertProcessor.exe also contains its configuration file **AlertProcessor.exe.config**.

Properties	
Field	**Description**
Frequency	Set the schedule for checking Alerter's **In** directory, in seconds.

Archiver



For more information, please see:

- Example Archive Workflow on page 131
- Sample workflows for TIBCO Foresight™ Archive and Retrieval System that are installed with Transaction Insight
- The Foresight™ Archive and Retrieval System documentation set.

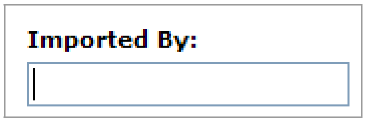
Trigger The arrival of a detail file and a data file in Importer's **In** directory. The part of the filename before the first dot must be the same. Extensions must be **dtl** and **edi**, regardless of the type of data in the data file. Unmatched files remain unprocessed in Archiver's In directory.

What it does Runs Archiver (Archiver.jar) to archive data files.

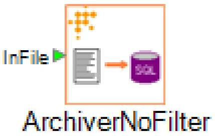
Input EDI, XML, and flat files, plus a validation DTL file for each. The DTL file must contain a GEN record with a FSUID (see FSUID.pdf).

Output Archived data.

Configuring Most configuring is done automatically by installation. To change logging levels, edit **ArchiverLogConfig.properties** in Archiver's Java\Bin directory.

Properties	
Field	**Description**
Description	For your use. Can be used to identify this instance of this Archive component. This is useful if you have several Archive components running. Archive's search page now has this field, where you can search for this text:  **Imported By:**
LoggingDir	Directory where log files will be created for this instance of the Archive component. Filename will be the name of the archived file with extension **log**.
ConfigDir	Directory containing Archiver properties files.

ArchiverNoFilter



For more information, please see:

- Example Archive Workflow on page 131
- Sample workflows for Archive that are installed with Transaction Insight
- The Archive documentation set.

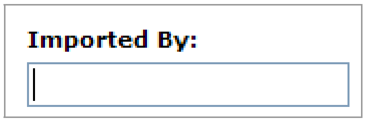
Trigger The arrival of a file.

What it does Runs Archiver (Archiver.jar) to archive the file. Unlike the Archive component, this file does not need an accompanying validation DTL file. In the Archive user interface, the user cannot search for this file directly but can see it as part of a fileset.

Input Any file.

Output Archived data.

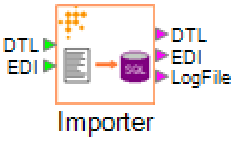
Configuring Most configuring is done automatically by installation. To change logging levels, edit **ArchiverLogConfig.properties** in Archiver's Java\Bin directory.

Properties	
Field	**Description**
Description	For your use. Can be used to identify this instance of this Archive component. This is useful if you have several Archive components running. Archive's search page now has this field, where you can search for this text:  Imported By:
LoggingDir	Directory where log files will be created for this instance of the Archive component. Filename will be the name of the archived file with extension **log**.
ConfigDir	Directory containing Archiver properties files.

Please see Example Archive Workflow on page 131.

Importer

For more information about Importer, see **Importer.pdf**.



Important

Check the path to Importer.exe under **Model | Component Templates | Importer.xml | Task** before using this component in a workflow.

Trigger

The arrival of a detail file and EDI file in Importer's **In** directory. The part of the filename before the first dot must be the same. Extensions must be **dtl** and **edi**. Unmatched files remain unprocessed in Importer's In directory.

What it does

Runs Importer (Importer.exe) to read files and put data and statistical information in the TI database.

Input

EDI and detail files.

Output

Database updates.

It can pass along the EDI file and validation DTL file that served as its input, as well as a log file.

The log file is identified by Importer.ini. Default is Importer.log in the directory containing Importer.exe.

Configuring

Set the parameters in Studio's component properties pane.

The directory that contains Importer.exe also contains its INI file, which identifies the database.

See page 127 for an example that uses Importer.

Importer Properties	
Field	**Description**
Direction	Choose whether this TI data will be flagged as inbound or outbound.
SaveGoodData	Choose whether to save good EDI documents as well as bad. This is equivalent to Importer's **savegooddata** command line parameter.

Importer Properties	
Field	**Description**
SaveGoodDataTypes	List the transactions for which good data is to be saved, separated by commas. Example: `837P,837I,834` This is equivalent to Importer's **savegooddatatypes** command line parameter.
SaveWarnings	Set to true if you want to store validation warnings in the database, as well as errors. This is equivalent to Importer's **savewarnings** command line parameter.
LogLevel	Choose how much information should be written to the log. This is equivalent to Importer's **v** command line parameter.
MaxFileSizeInMemory	You can set the maximum file size (in kilobytes) that will be loaded into memory. This is equivalent to Importer's M command line parameter.
TreatDataAsTest	Should data to be treated as test data regardless of the value in ISA15? This is equivalent to Importer's **t** command line parameter.
CommitFrequency	How many SQL statements should be executed before issuing a commit? This is equivalent to Importer's **cf** command line parameter. Default is 0, meaning commit when a transmission is complete.
ConnectString	If you want to use a connection string other than the one in Importer.ini, type it here. This is equivalent to Importer's **cs** command line parameter.
ConnectionAttempts	How many times should Importer try to connect to the database? This is equivalent to Importer's **ca** command line parameter. Default is 3.
SkipISAOnError	Set to true if you want to prevent interchanges from being imported if their ISA has an error.
InstanceName	Type a unique name for this Importer instance, 1 to 50 characters. This may be useful for your own purposes if you run multiple instances of Importer. This is equivalent to Importer's **n** command line parameter.

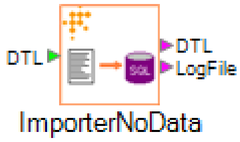
Importer Properties	
Field	**Description**
OverrideSenderLoc	You can force the new sender partner to be internal or external. Default lets Importer decide. This is equivalent to Importer's **sender_location** command line parameter.
OverrideRecvLoc	You can force a new receiver partner to be internal or external. Default lets Importer decide. This is equivalent to Importer's **receiver_location** command line parameter.

ImporterForwardsDTL

This component was removed from Foresight Studio because the Importer component now allows you to pass on the validation DTL file.

ImporterNoData

For more information about Importer, see Importer*n*_*n*.pdf.



Trigger Arrival of an Instream validation detail file in its **In** directory.

What it does Runs Importer (Importer.exe) to read detail results files and put data and statistical information in the TI database. The actual EDI file is not supplied.

Input Instream's validation detail results files.

Output Database changes and a log file. It also passes along the validation DTL file that served as its input.

Configuring Set the parameters in Studio's component properties pane.

The directory that contains Importer.exe also contains its INI file, which identifies the database.

ImporterNoData Properties
These are the same as the Importer Properties. Please see page 104.

ISFileIDInsert

DetailIn
DetailOut
SQL
ISFileIDInsert

Trigger Arrival of an Instream validation detail file in its **In** directory.

What it does Runs ISFileID.exe in Automator's Bin directory to insert a GEN record containing a FSUID. If this record already exists, the FSUID in it is replaced. See FSUID_and_AppDocs.pdf for details. This is primarily used for TI's Operational Monitoring.

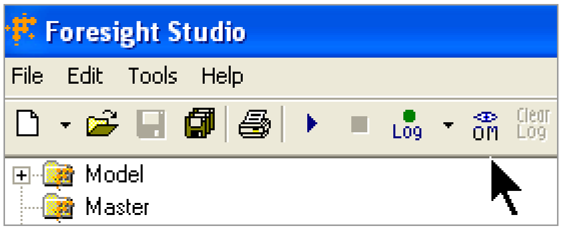
Input Instream's validation detail results files.

Output Same DTL file with the GEN record added or updated.

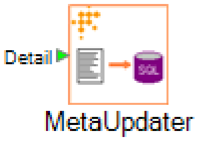
Configuring None needed.

For an example, see the TIDemoInboundOM workflow under the TIDemo system.

Before running a workflow that contains ISFileIDInsert, turn on Operational Monitoring by clicking this toolbar button until its eye opens:

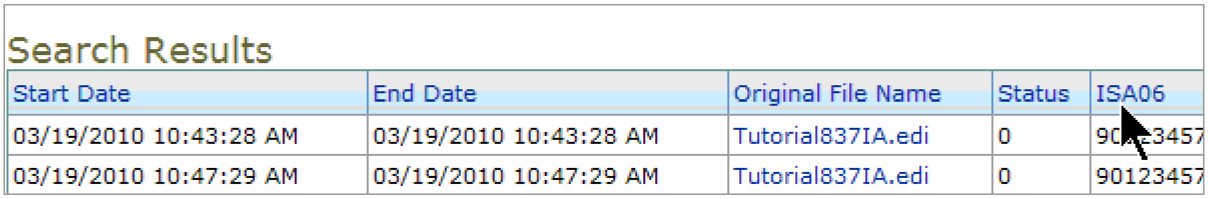


MetaUpdater



Trigger Arrival of a file.

What it does Primarily for demo purposes. Runs MetaUpdater, a specialized program that adds the ISA06 value to the search results for Operational Monitoring:



Search Results

Start Date	End Date	Original File Name	Status	ISA06
03/19/2010 10:43:28 AM	03/19/2010 10:43:28 AM	Tutorial837IA.edi	0	90123457
03/19/2010 10:47:29 AM	03/19/2010 10:47:29 AM	Tutorial837IA.edi	0	90123457

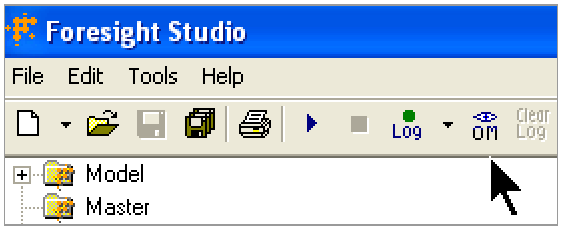
Input Validation detail results (DTL) file with any file extension.

Output Updates Operational Monitoring **pgn** file for this job and deletes the DTL file.

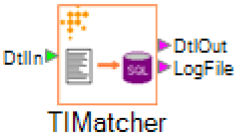
Configuring Set the parameter in the properties pane.

Properties	
Field	**Purpose**
OpManDirectory	Directory that contains the OpMon **pgn** files. Default is Foresight's Systems\OpMonLoggingDir directory.

Before running a workflow that contains MetaUpdater, turn on Operational Monitoring by clicking this toolbar button until its eye opens:



TIMatcher



Trigger Arrival of a file.

What it does Runs TIMatcher (TIMatcher.bat) to load matching information from this document into the TI database and match this document to others.

Input Instream validation detail results (DTL) file.

Output Entries in the TI database. Log file entries. Passes on the DTL file.

Configuring Set the parameters in the properties pane.

See TIMatching.pdf for details.

TIMatcher Properties	
Field	**Purpose**
loglevel	Select the logging level: **Error** = Errors only **Warnings** = Warnings and Errors **Info** = Info, Warnings and Errors This corresponds to the TIMatcher's **-v** command-line parameter.
config	Path and filename of a configuration file that specifies the other command line parameters. Generally you will either use -config or the other options but if you use both, any command line options take precedence over those in the config file. This corresponds to the TIMatcher's **-config** command-line parameter.
db	Connection string for the TI database: **For Oracle** *jdbc:oracle:thin:username/password@dbserver:port:instance* **For SqlServer** *jdbc:sqlserver://dbserver:port;databaseName=dbname;user=myuser; password=mypass;* This corresponds to the TIMatcher's **-db** command-line parameter.
window	Specify how many days back the matcher should look for data that matches this file. Default is 30 days. See -purge also. The window affects what data it purges. This corresponds to the TIMatcher's **-window** command-line parameter.
log	Type the path and filename to the log file. This corresponds to the TIMatcher's **-log** command-line parameter.
purge	Select True or False to purge old matching criteria or not. Each time TIMatcher runs, it stores the matching criteria in the database. Over time a lot of data can accumulate. The -purge option removes this data from the database. Only the criteria used to match the initial match is purged. The actual match information itself persists. The -window parameter is used to determine what data to purge. Anything older than the window is deleted. This corresponds to the TIMatcher's **-purge** command-line parameter.

TIUtilities

For more information about TIUtilities, see TIUtilities.pdf.

TIUtilities

Important Check the path to TIUtilities.exe under **Model | Component Templates | TIUtilities.xml | Task** before using this component in a workflow.

Trigger Timer.

What it does Runs TIUtilities (TIUtilities.exe) to do some or all of these for Transaction Insight:

1. Put files into the TI database
2. Assign tasks to teams
3. Populate summarized statistical tables
4. Delete expired data
5. Generate filter values and automatically assign them to partners

You will probably want to use two separate TIUtilities components in a workflow. One would run frequently and do 1-3 above. The second would run much less frequently and do 4 and 5.

Input **None.**

Output **None.**

Configuring Set the parameters in the properties pane.

See page 127 for an example that uses TIUtilities.

TIUtilities Properties

Field	Purpose
997GSfuture	Number of days after a given 997 to include in search for a GS match. This corresponds to the TIUtilities -997gsfuture command-line parameter.
997GSpast	Number of days prior to a given 997 to include in search for a GS match. This corresponds to the TIUtilities -997gspast command-line parameter.

TIUtilities Properties

Field	Purpose
997DaysPast	Number of days prior to start date to include in search for 997s. This corresponds to the TIUtilities -997dayspast command-line parameter.
997Date	Date to start search for 997s. *yyyymmdd* format This corresponds to the TIUtilities -997date command-line parameter.
MatchOn997	Match 997 responses to a GS. This corresponds to the TIUtilities -997 command-line parameter.
EnableLogToConsole	If **True**, logging information will go to the console as well as to a log file.
LogfileDirectory	The output directory for trace files. `C:\logs` This corresponds to the **td** command line parameter for TIUtilities.
LoggingLevel	Choose the amount of logging generated by TIUtilities: **None** **Errors Onl**y **Warnings** (includes errors) **Informational** (includes errors and warnings) **Verbose** (includes errors, warnings, and informational) This corresponds to the **tl** command line parameter for TIUtilities.
AssignTasks	If **True**, documents with errors will be assigned to teams and an e-mail sent to the team owner. This corresponds to the **t** command line parameter for TIUtilities. Requires: URL SMTPServer EMailSenderAddress
UpdateSummaryTables	If True, TIUtilities updates the statistical summary tables. This corresponds to the **s** command line parameter for TIUtilities.
SummarizeDocuments	If **True**, update the summary statistics on the Documents, Document Volumes, and Success Rate pages. This corresponds to the **sd** command line parameter for TIUtilities.

TIUtilities Properties	
Field	**Purpose**
NonTransFilterDay	Controls whether to update summary statistics by day that have no transaction filters applied. PastDates means to summarize days except for today. SummarizeAll means to summarize days including today. This corresponds to the TIUtilities sd_nonTransFilter option.
NonTransFilterWeek	Controls whether to update summary statistics by week that have no transaction filters applied. PastDates means to summarize weeks except for this week. SummarizeAll means to summarize weeks including this week. This corresponds to the TIUtilities sd_nonTransFilter option.
NonTransFilterMonth	Controls whether to update summary statistics by month that have no transaction filters applied. PastDates means to summarize months except for this month. SummarizeAll means to summarize months including this month. This corresponds to the TIUtilities sd_nonTransFilter option.
SingleTransFilterDay	Controls whether to update summary statistics by day that have a single transaction filter applied. PastDates means to summarize days except for today. SummarizeAll means to summarize days including today. This corresponds to the TIUtilities sd_singleTransFilter option.
SingleTransFilterWeek	Controls whether to update summary statistics by week that have a single transaction filter applied. PastDates means to summarize weeks except for this week. SummarizeAll means to summarize weeks including this week. This corresponds to the TIUtilities sd_singleTransFilter option.
SingleTransFilterMonth	Controls whether to update summary statistics by month that have a single transaction filter applied. PastDates means to summarize months except for this month. SummarizeAll means to summarize months including this month. This corresponds to the TIUtilities sd_singleTransFilter option.

TIUtilities Properties	
Field	**Purpose**
MultiTransFilterDay	Controls whether to update summary statistics by day that have multiple transaction filters applied. PastDates means to summarize days except for today. SummarizeAll means to summarize days including today. This corresponds to the TIUtilities sd_multiTransFilter option.
MultiTransFilterWeek	Controls whether to update summary statistics by week that have multiple transaction filters applied. PastDates means to summarize weeks except for this week. SummarizeAll means to summarize weeks including this week. This corresponds to the TIUtilities sd_multiTransFilter option.
MultiTransFilterMonth	Controls whether to update summary statistics by month that have multiple transaction filters applied. PastDates means to summarize months except for this month. SummarizeAll means to summarize months including this month. This corresponds to the TIUtilities sd_multiTransFilter option.
SummarizeErrors	If **True**, update the top error counts on the Errors page. This corresponds to the **se** command line parameter for TIUtilities.
SummarizeWarnings	If True, update the error and warning summary statistics on the Error page. This is equivalent to the TIUtilities **sw** command line parameter.
SummarizeTransmissions	If **True**, update the summary statistics at the top of the Transmissions page This corresponds to the **st** command line parameter for TIUtilities.
GeneratePartnerFilters	If **True**, generate dynamic partner filters. See the TIP Admin Guide for details. This corresponds to the **f** command line parameter for TIUtilities.
ExpireOldData	If **True**, delete expired data from the database. (Log in to the TI web portal and check the **Settings | Data Expiration** tab to see what will be deleted.) This corresponds to the **e** command line parameter for TIUtilities.
EMailSenderAddress	Enter the “from” address for mail sent by TI. This corresponds to the **from** command line parameter for TIUtilities.

TIUtilities Properties

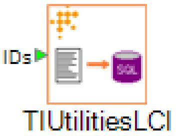
Field	Purpose
SMTPServer	SMTP server that TIUtilities will use to send e-mail notifications. Example: `mail.foresightcorp.com` This corresponds to the **smtp** command line parameter for TIUtilities.
DBConnectString	Type or paste (Ctrl-V) the connection string to access the TI database. Example: `DRIVER={SQL Server};SERVER=(local);DATABASE=TI231SQL;UID=U234A;PWD=W3342` This corresponds to the **db** command line parameter for TIUtilities.
DatabaseType	Select the TI database type: **Oracle** or **ODBC**. This corresponds to the **dbtype** command line parameter for TIUtilities.
URL	URL to access the TI web portal. `Example: http://KAVERTEST/TI231Demo//` This corresponds to the **url** command line parameter for TIUtilities.
IntervalSeconds	Set the schedule for executing TIUtilities, in seconds. Make it long enough that the previous execution will have completed.

TIUtilitiesLCI

For more information about TIUtilities, see TIUtilities.pdf.

For information about confidential identifiers, see Configuring Confidentiality in TransactionInsightInstallation*nn*.pdf.

Transaction Insight ships with a sample TIDemoLCI workflow.



Important Check the path to TIUtilities.exe under **Model | Component Templates | TIUtilities.xml | Task** before using this component in a workflow.

Trigger Timer.

What it does Runs TIUtilities (TIUtilities.exe) to load confidential identifiers into the Transaction Insight database.

Input IDs

Output **None.**

Configuring Set the parameters in the properties pane.

Properties	
Field	**Purpose**
EnableLogToConsole	If **True**, logging information will go to the console as well as to a log file.
LogfileDirectory	The output directory for trace files. `C:\logs` This corresponds to the **td** command line parameter for TIUtilities.
LoggingLevel	Choose the amount of logging generated by TIUtilities: **None** **Errors Onl**y **Warnings** (includes errors) **Informational** (includes errors and warnings) **Verbose** (includes errors, warnings, and informational) This corresponds to the **tl** command line parameter for TIUtilities.
DBConnectString	Type or paste (Ctrl-V) the connection string to access the TI database. Example: `DRIVER={SQL Server};SERVER=(local);DATABASE=TI231SQL;UID=U234A;PWD=W3342` This corresponds to the **db** command line parameter for TIUtilities.
DatabaseType	Select the TI database type: **Oracle** or **ODBC**. This corresponds to the **dbtype** command line parameter for TIUtilities.

Variables for Component Templates and Workflows

For details about global variables and WorkflowGlobals.xml, Automator “Built-in” Variables, and component properties, please see Variables_for_Workflows.pdf.

Creating your own Components

For details, see Component on page 135.

Copying and Moving

Click any workflow, configuration, profile, or text file and use:

- **File | Duplicate** to copy it to the same system.
- **File | Copy To** to copy it to another system.
- **File | Move To** to move it to another system.

Opening a Different Studio Project

To open a project that is not currently showing in Studio:

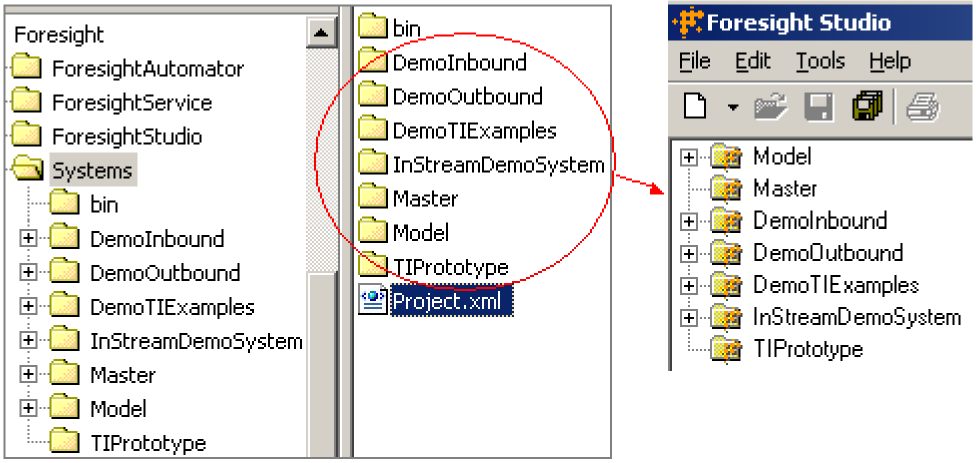
1. Open Studio.

2. Use **File | Open Project**.

3. Navigate to **Systems** directory that contains the other **Project.xml** file.

4. Select **Project.xml**.

The projects that were previously showing disappear and are replaced with the newly-opened project. This project will include all workflows, component templates and configurations that are under its Systems directory:



To see the original project, choose **File | Open Default Project**.

Opening the Default Project

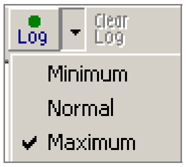
To open the project located under Studio's Systems directory:

1. Open Studio.
2. Select **File | Open Default Project**.

Logging

Logging Levels

You can set the detail of logging information by clicking the arrow next to the log toolbar button, which displays when a workflow is open in Studio:



See Log Files in **Automator.pdf** for more information.

Starting and Stopping Logging

To start logging, click the Log toolbar button until it turns green. ● Log

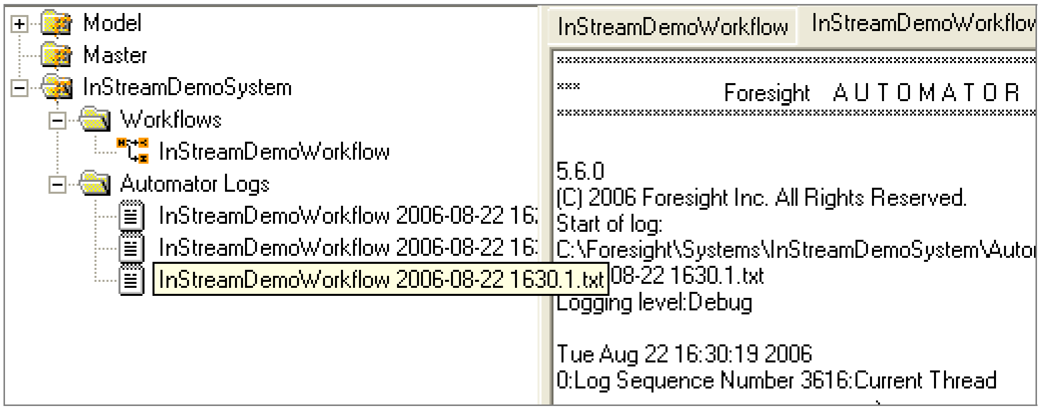
To stop logging, click the Log toolbar button until it turns red.

Viewing and Understanding the Log

To open a log from within Foresight Studio, double-click it under Automator Logs. The most recent log is at the bottom.

You have these ways to search the open log:

- Use the Find menu.
- Click anywhere in the log and use *Ctrl*-**f**.
- For "Find Next," use *Ctrl*-**g** or **F3**.

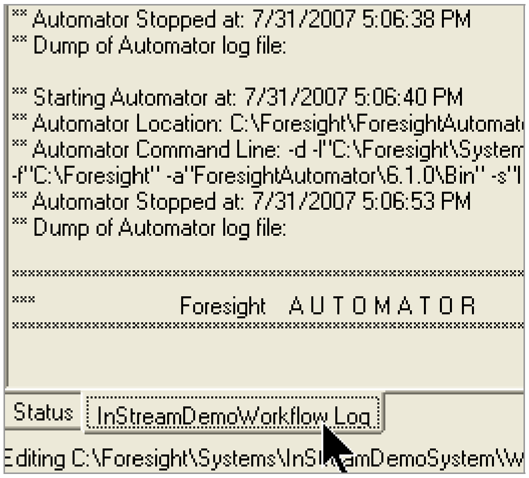

Model
Master
InStreamDemoSystem
Workflows
InStreamDemoWorkflow
Automator Logs
InStreamDemoWorkflow 2006-08-22 16:
InStreamDemoWorkflow 2006-08-22 16:
InStreamDemoWorkflow 2006-08-22 1630.1.txt
InStreamDemoWorkflow
InStreamDemoWorkflow

```
**************************************************************
***                     Foresight   A U T O M A T O R
**************************************************************

5.6.0
(C) 2006 Foresight Inc. All Rights Reserved.
Start of log:
C:\Foresight\Systems\InStreamDemoSystem\Auto
        08-22 1630.1.txt
Logging level:Debug

Tue Aug 22 16:30:19 2006
0:Log Sequence Number 3616:Current Thread
```

For more information about understanding logs, see Log Files in **Automator.pdf**.

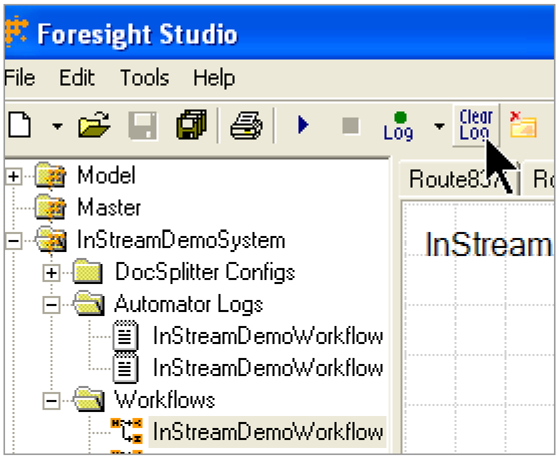
Clearing Logs

To clear the log tab at the bottom of the screen:



```
** Automator Stopped at: 7/31/2007 5:06:38 PM
** Dump of Automator log file:

** Starting Automator at: 7/31/2007 5:06:40 PM
** Automator Location: C:\Foresight\ForesightAutomat
** Automator Command Line: -d -l"C:\Foresight\System
-f"C:\Foresight" -a"ForesightAutomator\6.1.0\Bin" -s"l
** Automator Stopped at: 7/31/2007 5:06:53 PM
** Dump of Automator log file:

****************************************************************
***                    Foresight    A U T O M A T O R
****************************************************************
```

… use the Clear Log toolbar button:



5 Automator and Foresight Service

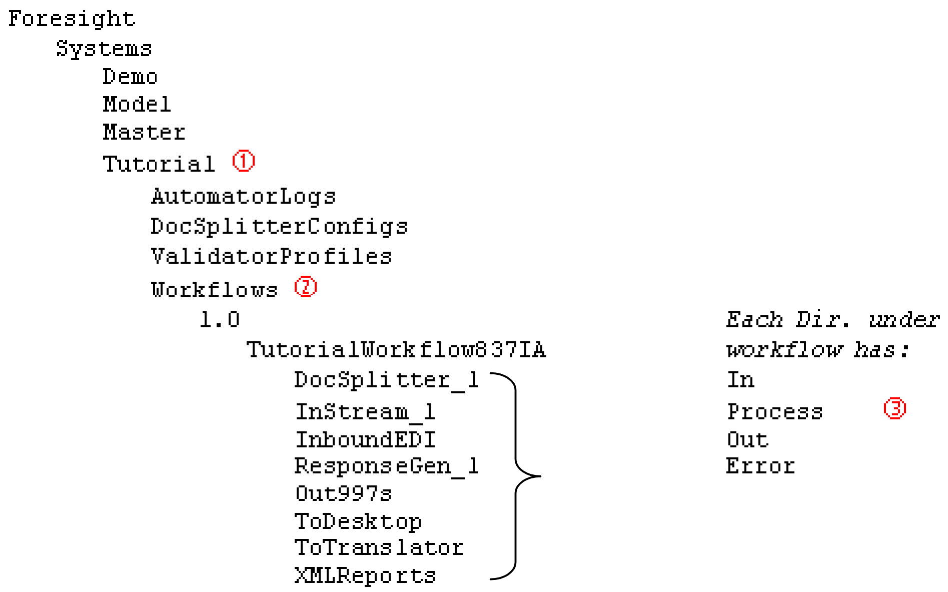
Automator is the program that executes a workflow defined in Studio. Please see **Automator.pdf** in the **ForesightAutomator\Doc** directory for details.

ForesightService runs Automator as a service, so that it does not run under any particular user and you do not need to start Automator from within Studio. Please see **ForesightService.pdf** in the ForesightStudio\Doc directory for details.

6 Important Directories and Systems

Directory Structure

The tutorial's directory structure is typical:



```
Foresight
    Systems
        Demo
        Model
        Master
        Tutorial ①
            AutomatorLogs
            DocSplitterConfigs
            ValidatorProfiles
            Workflows ②
                1.0                                     Each Dir. under
                    TutorialWorkflow837IA               workflow has:
                        DocSplitter_1  ⎫                In
                        InStream_1     ⎪                Process    ③
                        InboundEDI     ⎪                Out
                        ResponseGen_1  ⎬                Error
                        Out997s        ⎪
                        ToDesktop      ⎪
                        ToTranslator   ⎪
                        XMLReports     ⎭
```

① Each system has a directory structure under **Foresight\Systems**. It contains:

- XML files for internal use by Studio.
- Directories:

  AutomatorLogs – Contains log files detailing each step of the workflow (only if the workflow is started from within Studio. Otherwise, the log defaults to a Logs directory under the specific workflow.).

DocSplitterConfigs – Created by Studio for Docsplitter configuration files.

PartnerAutomationConfigs – Created by Studio for Partner automation lookup files.

PartnerRouterConfig – Created by Studio for TPARouter configuration files.

ValidatorProfiles – Created by Studio for Validator profiles.

Workflows – Directories used by Automator when moving files through the workflow.

② Under **Workflows**, open the version directory and the workflow directory to see a directory named for each component of the workflow. Once the workflow runs, each will contain **Error**, **In**, **Out**, and **Process** subdirectories.

③ Directories to temporarily hold files:

In Holds files awaiting processing by this component

Process Holds files while they are processed by this component

Out Holds files after processing by this component, until they are picked up by the next component

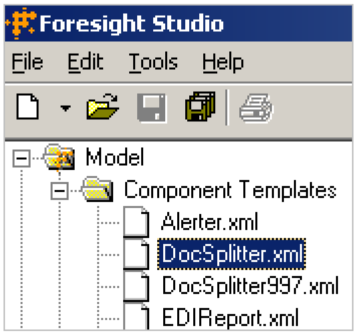
Error Holds files that did not process properly

Occasionally check these directories for the presence of stuck or unwanted files.

Model System for Setting Defaults

When you create a new system, defaults are copied from the Model system to your new system. From that point on, the new system is independent of the Model system. Changes to Model do not affect existing systems.

If your workflow cannot find an executable it uses, you may have to adjust its path from within Studio's Model system as described in Paths to Executable Components on page 43.



When you make a change to a component template, an XML file is updated in **Foresight\Systems\Model\ComponentTemplates\**<*version*>.

When you make a change within a workflow, XML files are updated in **Foresight\Systems\<*system*>\Workflows\<*version*>\<*workflow*>**.

As a last resort, you can edit these files directly with Notepad or another text editor. Use caution, exit Studio first, and back up the files before changing them.

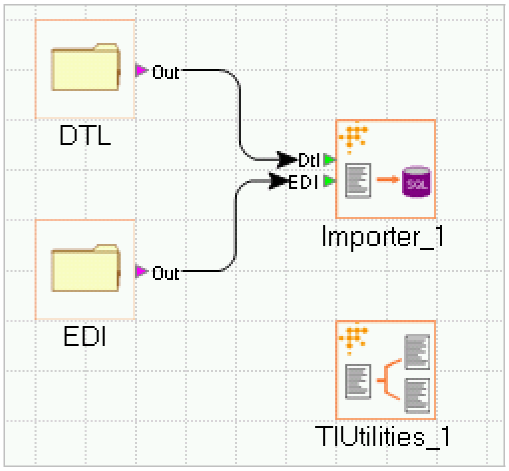
Master System for Shared Items

The Master system is a central repository of workflows, profiles, and error messages. Files in Master can be copied to and from all other systems.

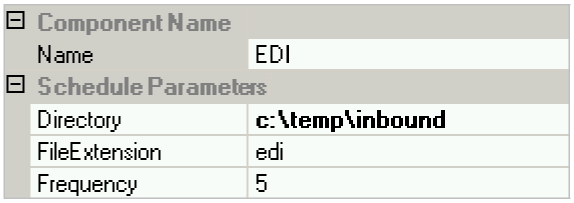
7 Example Workflows

Example TI Workflow

This is a simple Transaction Insight workflow:



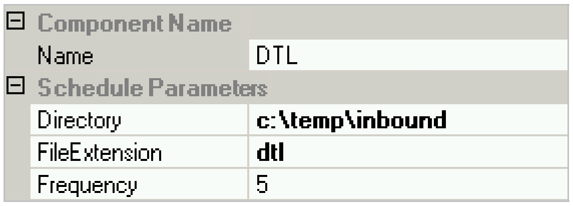
1. When the **EDI** folder finds files with the file extension **edi** in **c:\temp\Inbound**, it sends them to Importer.



⊟ Component Name	
Name	EDI
⊟ Schedule Parameters	
Directory	**c:\temp\inbound**
FileExtension	edi
Frequency	5

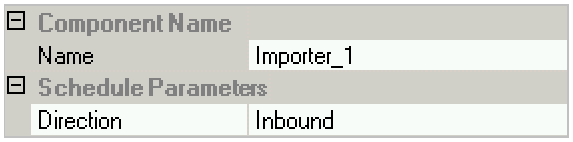
2. When the **DTL** folder finds files with the file extension **dtl** in **c:\temp\Inbound**, it sends them to Importer.

   This is the same directory used by EDI. These components are set up separately since they move files with different file extensions.



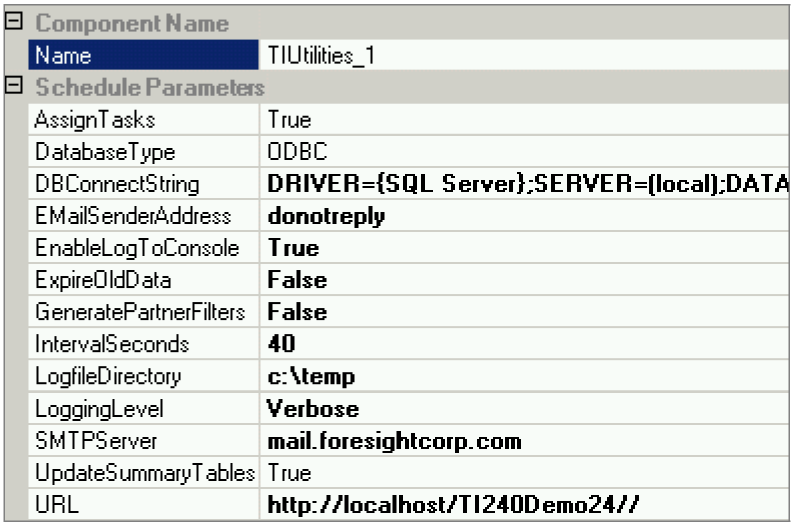
⊟ Component Name	
Name	DTL
⊟ Schedule Parameters	
Directory	**c:\temp\inbound**
FileExtension	**dtl**
Frequency	5

3. The Input folders funnel their EDI and DTL files to Importer. When Importer finds files with names that match up until the first dot, it loads information from them into the TI database



⊟ Component Name	
Name	Importer_1
⊟ Schedule Parameters	
Direction	Inbound

4. On a timer, TIUtilities assigns tasks and updates the database statistics for display in the TI website.

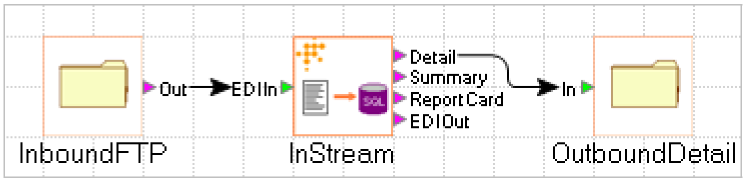


⊟ Component Name	
Name	TIUtilities_1
⊟ Schedule Parameters	
AssignTasks	True
DatabaseType	ODBC
DBConnectString	**DRIVER={SQL Server};SERVER=(local);DATA**
EMailSenderAddress	**donotreply**
EnableLogToConsole	**True**
ExpireOldData	**False**
GeneratePartnerFilters	**False**
IntervalSeconds	**40**
LogfileDirectory	**c:\temp**
LoggingLevel	**Verbose**
SMTPServer	**mail.foresightcorp.com**
UpdateSummaryTables	True
URL	**http://localhost/TI240Demo24//**

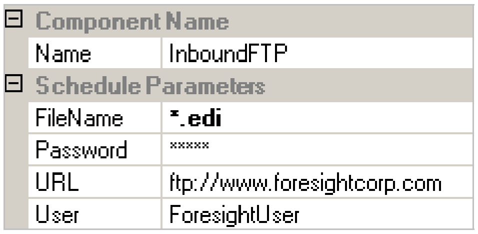
Example Inbound FTP Workflow

Input: EDI files at an FTP site.

This workflow uses FTP to move files to Instream validation and puts the results in c:\temp\output.

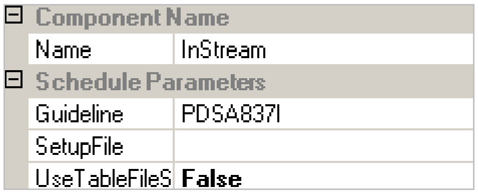


1. When **InboundFTP** finds files with the file extension **edi** at **ftp://www.foresightcorp.com**, uses the FTP username **ForesightUser** and the password to sends them to Instream.



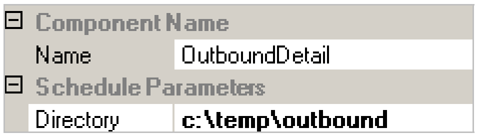
⊟ Component Name	
Name	InboundFTP
⊟ Schedule Parameters	
FileName	***.edi**
Password	*****
URL	ftp://www.foresightcorp.com
User	ForesightUser

2. Instream validates the files.



⊟ Component Name	
Name	InStream
⊟ Schedule Parameters	
Guideline	PDSA837I
SetupFile	
UseTableFileS	**False**

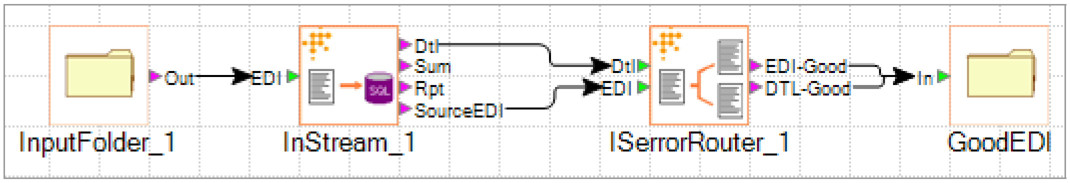
3. Detail results files go to **C:\temp\outbound**.



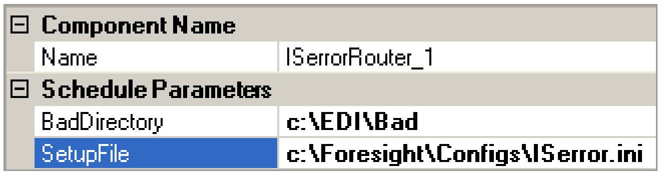
⊟ Component Name	
Name	OutboundDetail
⊟ Schedule Parameters	
Directory	**c:\temp\outbound**

Example that Routes on Errors in the EDI

Foresight Studio's ISerrorRouter component runs ISErrors.exe, as in this example Studio workflow:



The parameters for ISerrorRouter_1 might be:

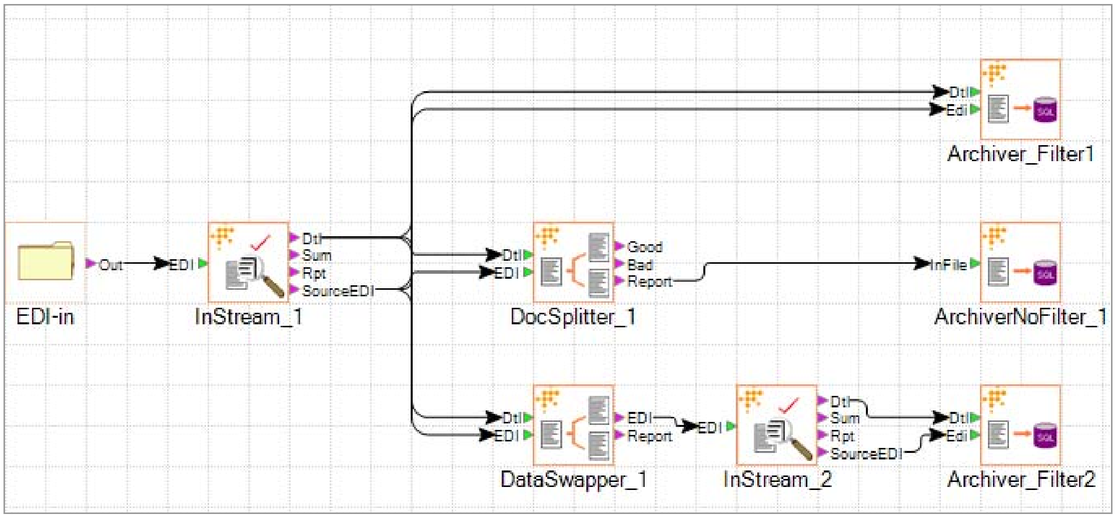


⊟ Component Name	
Name	ISerrorRouter_1
⊟ **Schedule Parameters**	
BadDirectory	**c:\EDI\Bad**
SetupFile	**c:\Foresight\Configs\ISerror.ini**

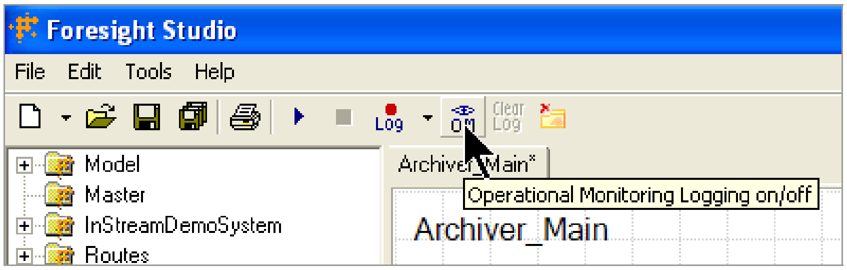
Please see ISErrors.pdf for details.

Example Archive Workflow

This workflow uses the Archiver and Archiver_Filter components.



To work as intended, the Operational Monitoring Logging toolbar button must be on when the workflow is running. The OM “eye” should be open:



Pertinent parts of the workflow:

- The OM “eye” is turned on when the workflow runs.
- There should be only one inbound folder so that all files can be tied together by a Job ID.

  This allows all files that originate from a single input file in the EDI-in folder to be considered a set. When an Archive portal user searches for one file in the set, Archive will list the entire set for them.
- The first Instream component validates the input file and inserts a unique ID number called a FSUID into its DTL file. The original EDI file and its DTL file go to all three branches of the workflow.
- The top branch uses the Archiver component to archive the EDI and DTL file. An Archive user can locate these files by using the Search pages.
- The middle branch splits the original EDI file and sends the report to the ArchiverNoFilter component. This component archives files that are not an EDI-DTL pair of files. The report is archived but it cannot be located with a search on Archive’s Search page.

- The bottom branch runs DataSwapper to modify the EDI, and sends that modified EDI for validation. At this point, the DTL file will be given a new FSUID, so it will no longer be tied by FSUID to the files archived via the top branch.

  The modified EDI and DTL files are then sent to the Archiver component and placed in the archive.

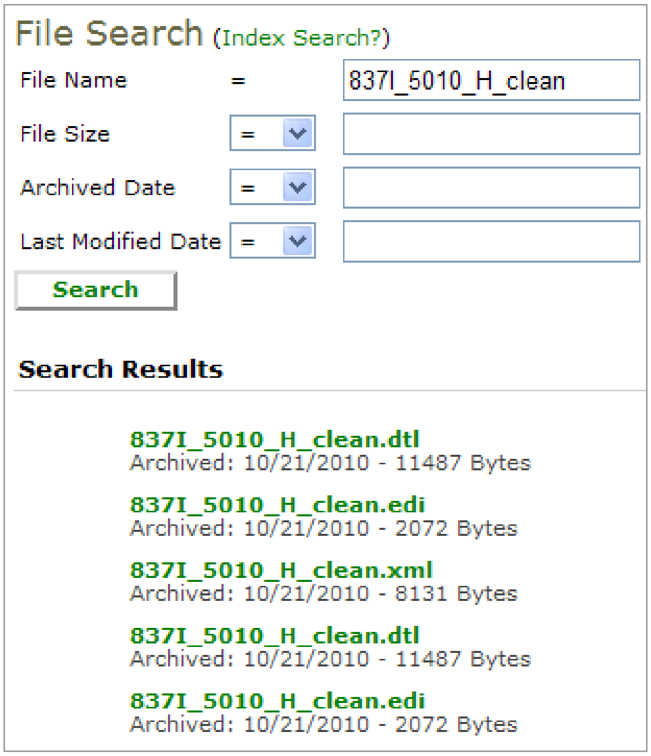
We now have five files in the archive:

1. The original EDI file
2. The DTL file from validating the original EDI file
3. The Docsplitter report
4. The EDI file modified by DataSwapper
5. The DTL file from validating this modified EDI file

An Archive portal user can search for 1, 2, 4, or 5.

Since 1-2 and 4-5 have different FSUIDs, and 3 has no FSUID, what ties them together so that all five are listed in Archive's Search results? That is handled by the "Job ID." The workflow assigns a unique job ID to the original input file. It links each file created by the workflow to that same ID and passes along this information to Archive.

In Archive, searching for the root filename, with no extension, returns all five files:



File Search (Index Search?)

File Name	=	837I_5010_H_clean
File Size	=	
Archived Date	=	
Last Modified Date	=	

Search Results

837I_5010_H_clean.dtl
Archived: 10/21/2010 - 11487 Bytes

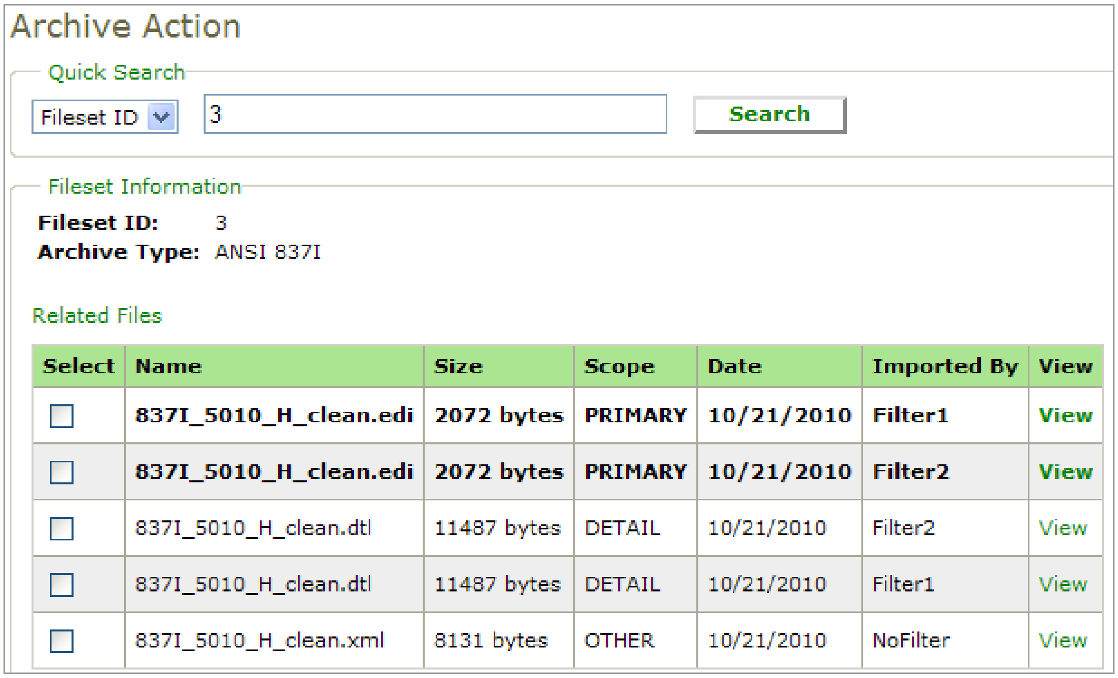
837I_5010_H_clean.edi
Archived: 10/21/2010 - 2072 Bytes

837I_5010_H_clean.xml
Archived: 10/21/2010 - 8131 Bytes

837I_5010_H_clean.dtl
Archived: 10/21/2010 - 11487 Bytes

837I_5010_H_clean.edi
Archived: 10/21/2010 - 2072 Bytes

If you click on a file, you can see the Fileset ID and the Description of the component that archived it:



3

Fileset Information

Fileset ID: 3
Archive Type: ANSI 837I

Related Files

Select	Name	Size	Scope	Date	Imported By	View
□	**837I_5010_H_clean.edi**	**2072 bytes**	**PRIMARY**	**10/21/2010**	**Filter1**	**View**
□	**837I_5010_H_clean.edi**	**2072 bytes**	**PRIMARY**	**10/21/2010**	**Filter2**	**View**
□	837I_5010_H_clean.dtl	11487 bytes	DETAIL	10/21/2010	Filter2	View
□	837I_5010_H_clean.dtl	11487 bytes	DETAIL	10/21/2010	Filter1	View
□	837I_5010_H_clean.xml	8131 bytes	OTHER	10/21/2010	NoFilter	View

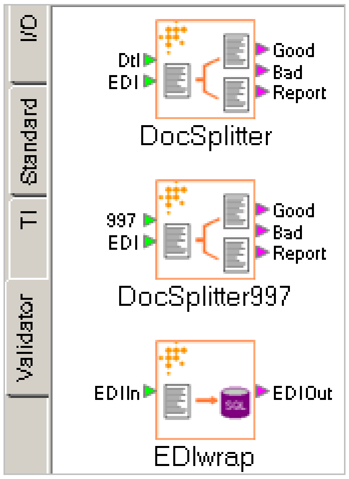
8 Component Templates

Overview of Component Templates

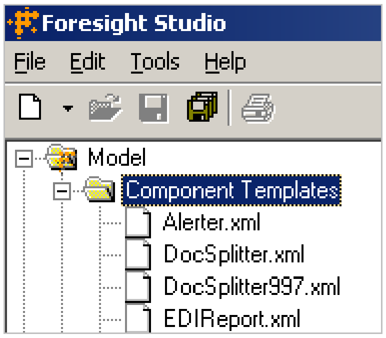
Please Note

This is an advanced feature for use by those familiar with Studio and Instream.

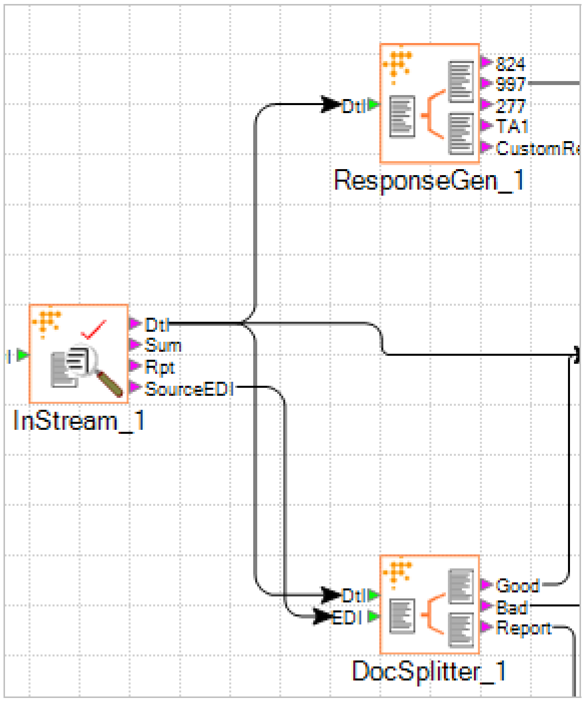
Component templates define the components that can go into your workflows:



They are located under the Model system:



You can change these templates. For example, you might need to customize the paths to your executables such as Instream, as described in Paths to Executable Components on page 43.



For more information, see:

- Changing a Component Template on page 137
- Creating a new Component Template on page 137
- Component Tabs on page 138
- Variables for Component Templates and Workflows on page 114
- Example - Creating a Component to Delete Files on page 158

Changing a Component Template

1. Open the **Model** system and double-click the template that you want to change.

2. Make your changes (see Component Tabs on page 138).

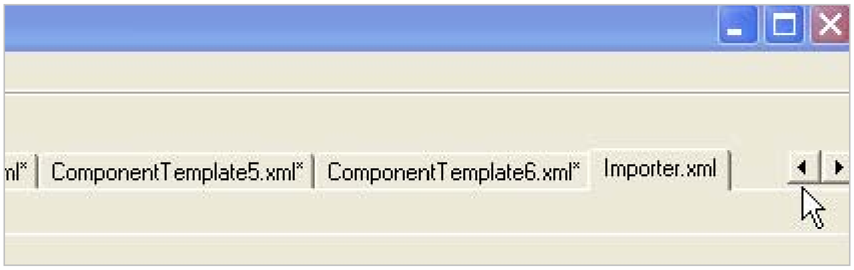
3. Save all.

4. Close and re-open Studio.

5. In each workflow that used the component, delete the component from the workflow and add it again.

6. Save all.

If you have many component templates open, you will have arrows at the top right to scroll through them:



Creating a new Component Template

1. Click the **Model** system and choose **File | New | Component Template**.

2. Name the component and include a file extension of **xml**. Filenames can include special characters except these:

   blanks

   * ! ? / \ > < |

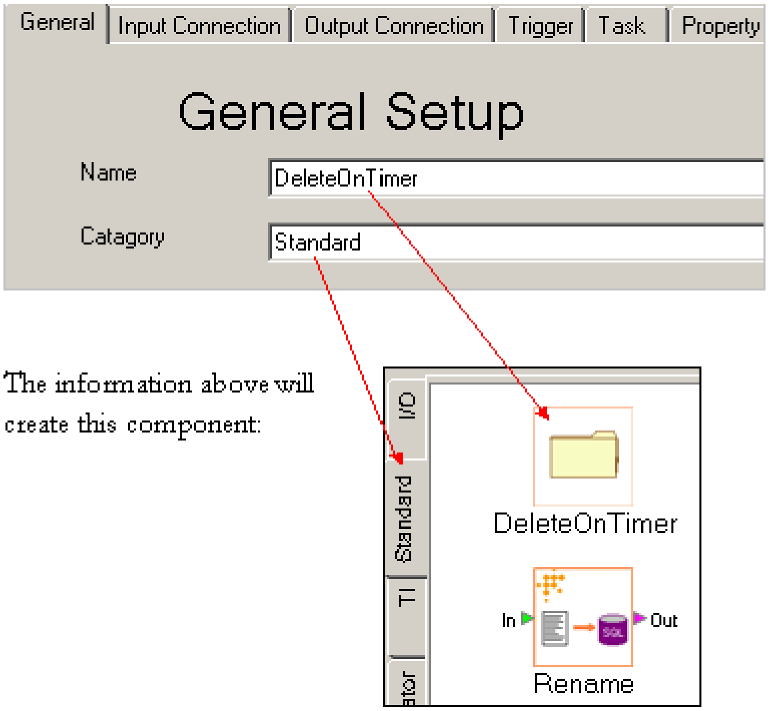
3. Choose a Category where it will reside when editing a workflow:

4. Go to each tab and enter the appropriate information (see Component Tabs on page 138).

5. Save all.

6. Open a workflow.

7. In the bottom left pane, click the tab that you specified under the Category field in General Setup, and notice the new component.



General Setup

Name: DeleteOnTimer

Catagory: Standard

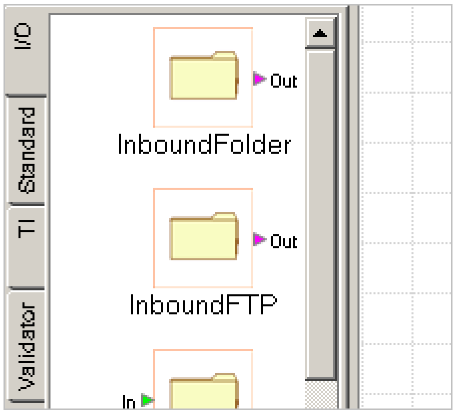
The information above will create this component:

Component Tabs

General Tab

Name Required. Type a name that will appear below the component in the workflow.

Category Select the tab under which it should appear in the bottom left pane when a workflow is open, or type the name of a new category. These will show up as a tab at the bottom left when a workflow is open.



If omitted, the component will appear under the Standard tab.

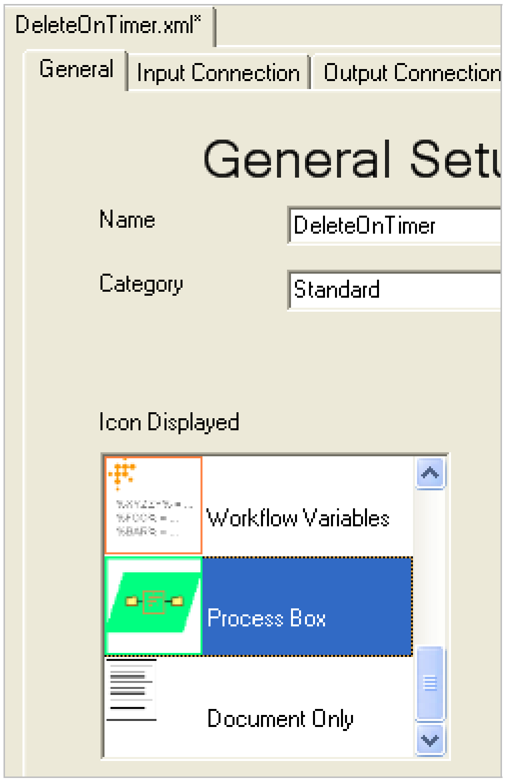
Icon Displayed

Select an icon.

Leave Tracking Files

Check only if you wish to retain track files. See Appendix A - Track Files.

Example 1 – Using an existing category



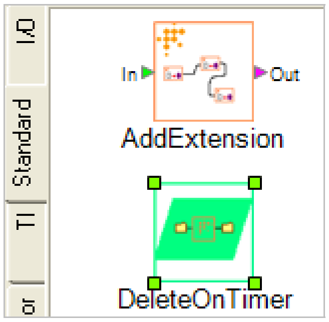
General Setu

Name DeleteOnTimer

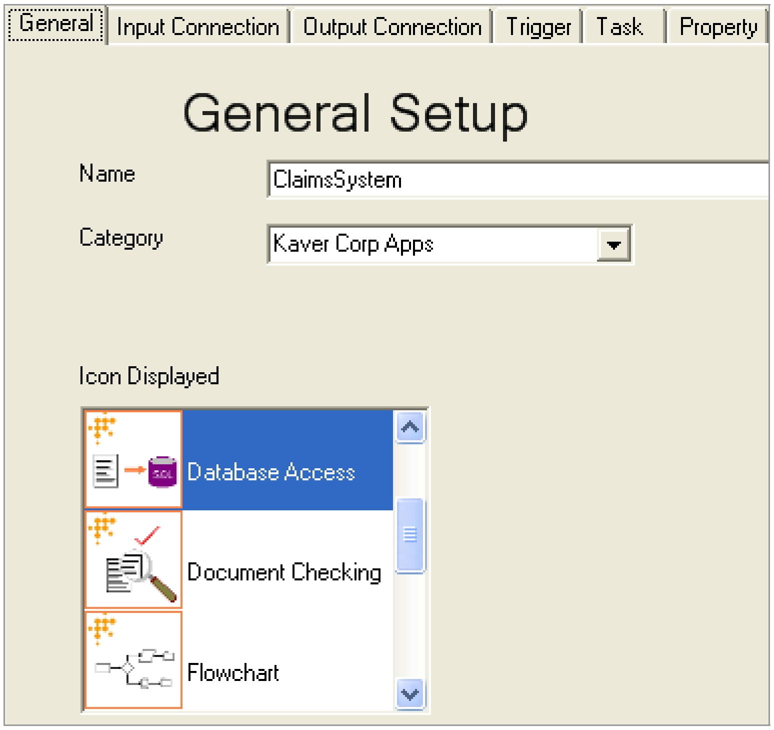
Category Standard

Icon Displayed Process Box

The information above will create this component under the Standard tab:



Example 2 – Creating a new category



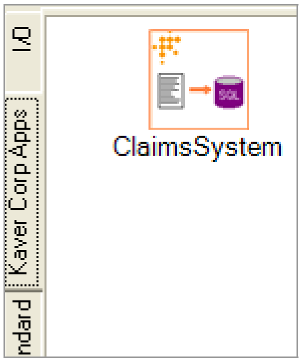
General Setup

Name ClaimsSystem

Category Kaver Corp Apps

Icon Displayed

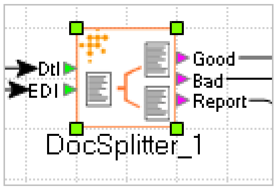
The information above will create a Kaver Corp Apps category and a ClaimsSystem component in it:



Input Connection Tab

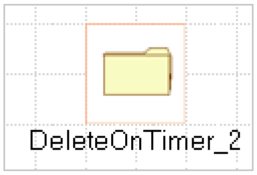
This tab sets up the input connections for the component you are defining.

This example has two input connections: **Dtl** and **EDI**:



Components with no input or output connections

If a component does not have an input connection, the Input Connection tab is not used.

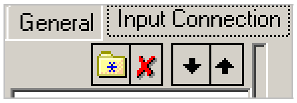


In that case, it would need a Trigger Type of SimpleTimer.

See Example - Creating a Component to Delete Files on page 158 for an example of a component with no input or output connections.

Creating an input connection

1. Click the 📁 button.



2. Fill out the fields and then save.

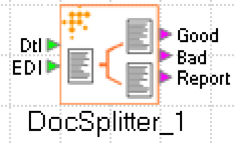
3. Repeat for each connection. Notice the connection names appearing along the left side.

4. If you want to reorganize the list of input connection names, use the ↓↑ keys.

5. Save.

Fields include:

Name Required. Type the name for the connection. This appears on the left side of the component in the workflow.



Expected File Type

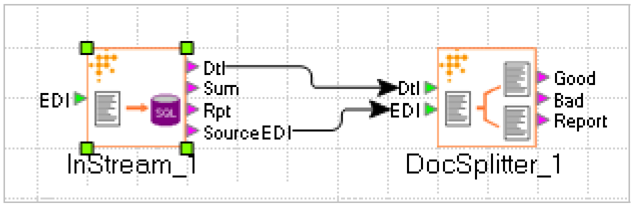
When defining a workflow, this can only be connected to another component's output connection if they both have the same Expected File Type. This helps to prevent misconnections when defining a workflow.

If you select **All types**, Studio does not check the file type before letting you draw a connection between workflow components.

In the example below, Docsplitter has two input connections:

Dtl has an Expected File Type of Detail. It connects to Instream's output connection **Dtl**, which has an Expected File Type of Detail.

EDI has an Expected File Type of EDI. It connects to Instream's output connection **SourceEDI**, which has an Expected File Type of EDI.



Parameters

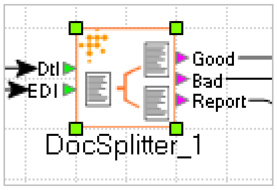
For use with future enhancements.

For an example, look at the InStream.xml component template.

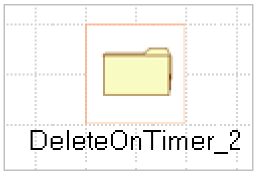
Output Connection Tab

This tab sets up the output connections for the component you are defining.

This example has output connections named **Good**, **Bad**, and **Report**:

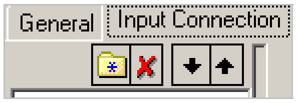


Some components have no output connections, so the Output Connection tab is not used.



To create an output connection:

1. Click the button.



2. Fill out the fields and then save.

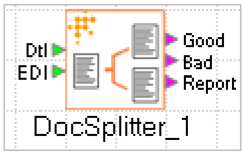
3. Repeat for each connection. Notice the connection names appearing along the left side.

4. If you want to reorganize the list of input connection names, use the ↓↑ keys.

5. Save.

Fields include:

Name Required. Type the name for the input connection. This appears on the component in the workflow:



Expected File Type

This must match the Input Connection Expected File Type for the next component. See the description for Expected File Type under Input Connection Tab on page 140.

Parameters Type the extension of the files to be sent. Do not include a dot.

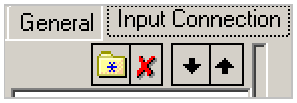
For an example, look at the **InStream.xml** component template.

Trigger Tab

This tab defines what activates the component.

To set up a trigger:

1. Click the 📁 button.



2. Fill out the fields.

3. If the component can be triggered in multiple ways, repeat for the next trigger.

4. If you want to reorganize the list of trigger names, use the ↓↑ keys.

5. If you want to delete a trigger name, click it and press ✗.

6. Save.

Fields include:

Name Type a name for the trigger.

Type **FileAvailable** triggers the component when a file arrives in the component's In directory

(**Foresight\Systems\<*system*>\Workflows\<*version*>\<*workflow*>\<*component*>\In**).

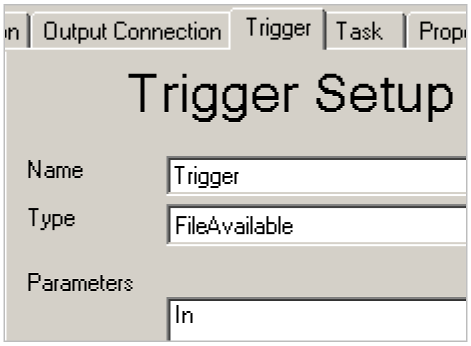
FileNameMatch triggers the component when multiple files arrive in the component's In directory with the extensions specified under Parameters. All must be there for activation to occur. In addition, the left part of the filename (up to the dot) must match. You can use two dots in filenames if you want to have the middle section for special purposes such as content-based splitting. Example: `Myfile.PartnerA.dtl`

SimpleTimer triggers the component every *n* seconds.

Parameters

FileAvailable

The Parameters area contains the directory where the component should look for incoming files. It will be in one of the component's directories in the workflow: In, Process, Out, or Error.



Trigger Setup

Name: Trigger

Type: FileAvailable

Parameters: In

FileNameMatch

The top line contains the directory where the component should look for incoming files. It will be in one of the component's directories in the workflow: In, Process, Out, or Error.

The second line is 2, 3, or 4 - the number of files that are to be matched up before they can be processed.

The remaining lines are the file extensions.

Example. If extensions are edi and dtl:

These match:

```
File1.edi
File1.dtl
```

These do not match:

```
File1.edi
File1_results.dtl
```

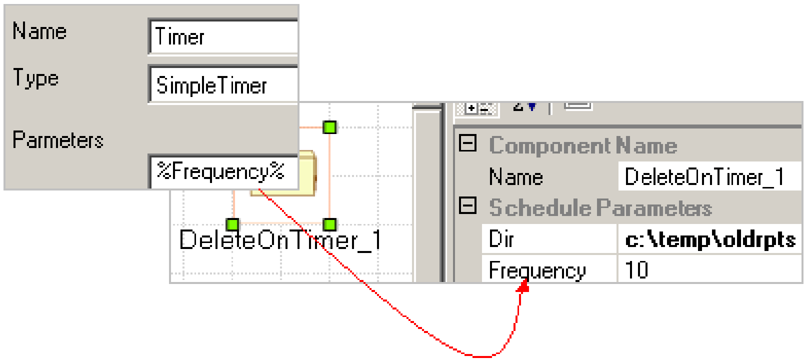
These do not match:

```
File1.edi
File1.A.dtl
```

SimpleTimer

Type the number of seconds before the component activates.

If you want the workflow developer to specify the number of seconds, use a variable that is defined under the Property tab.



Name Timer

Type SimpleTimer

Parmeters %Frequency%

DeleteOnTimer_1

⊟ Component Name

Name DeleteOnTimer_1

⊟ Schedule Parameters

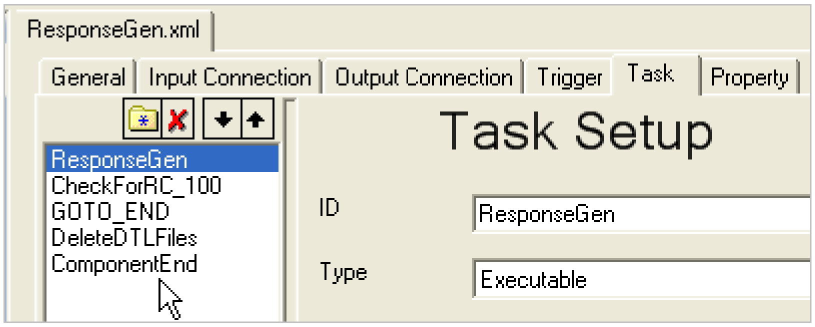
Dir c:\temp\oldrpts

Frequency 10

Task Tab

This tab defines what the component does when it activates.

Each component contains one or more tasks. This ResponseGen component has four tasks, plus the ComponentEnd marker.



ResponseGen.xml

ResponseGen
CheckForRC_100
GOTO_END
DeleteDTLFiles
ComponentEnd

Task Setup

ID ResponseGen

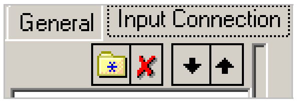
Type Executable

When you click a task, the rest of the Task tab updates to show information about it.

When the component activates, its tasks execute sequentially.

To set up a new task:

1. Click the 📁 button if the fields are not available.

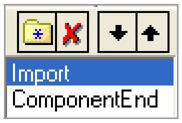


2. Fill out the fields as described in the table below.

3. Save.

4. Repeat for the other tasks in the component.

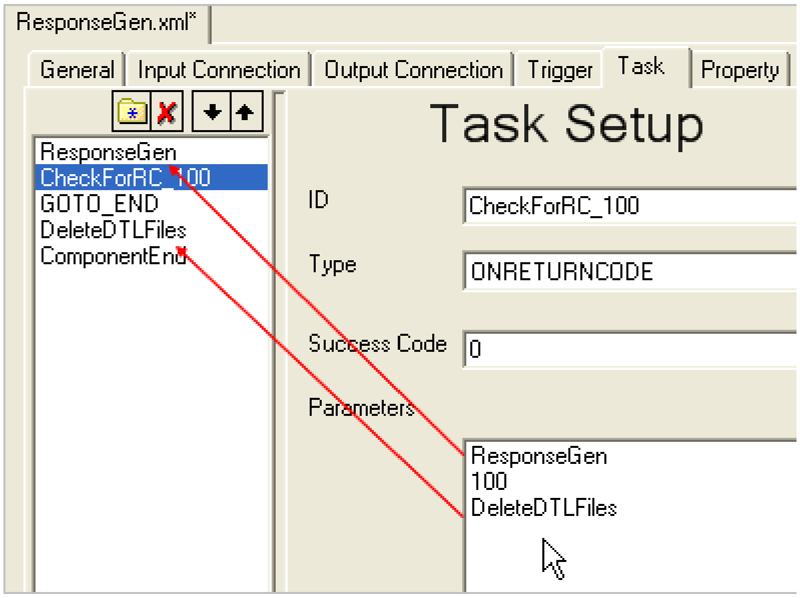
Fields on the Task tab

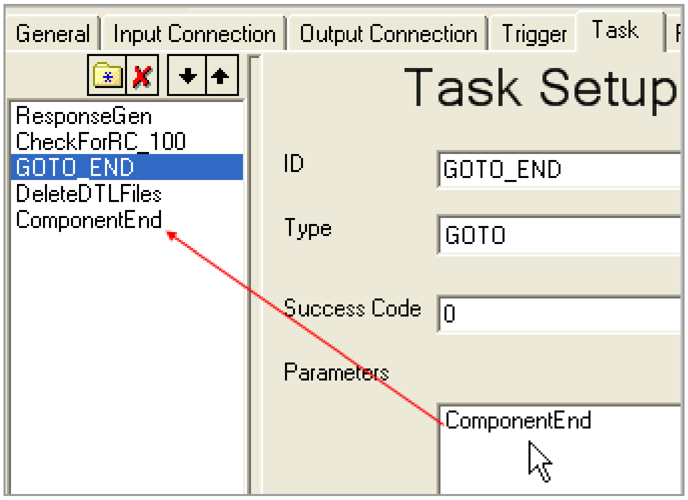
Type	Description
ID	Required. Type the name for the set of tasks. Parameter: The task name shows up in the task list:  Import ComponentEnd
Type	**Executable** Run an executable, batch file, or command prompt. Parameters: 1 The first line is the path and filename of the executable. 2 Each additional line is one command line parameter. You can include variables surrounded by %, as described in Variables_for_Workflows.pdf. See the Instream component's HVInStream task for an example.

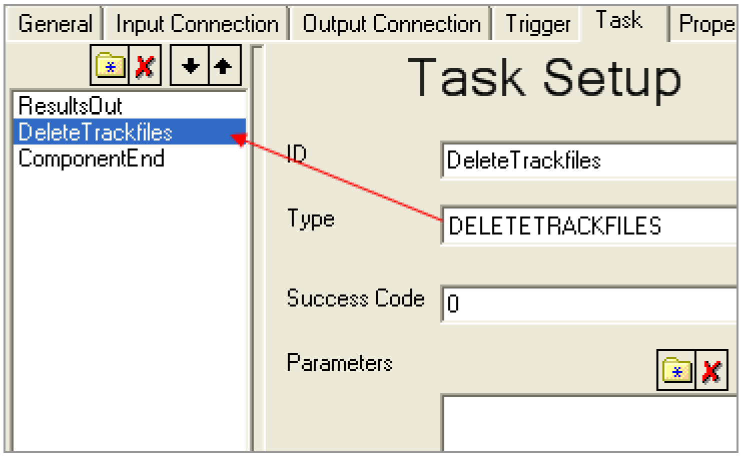
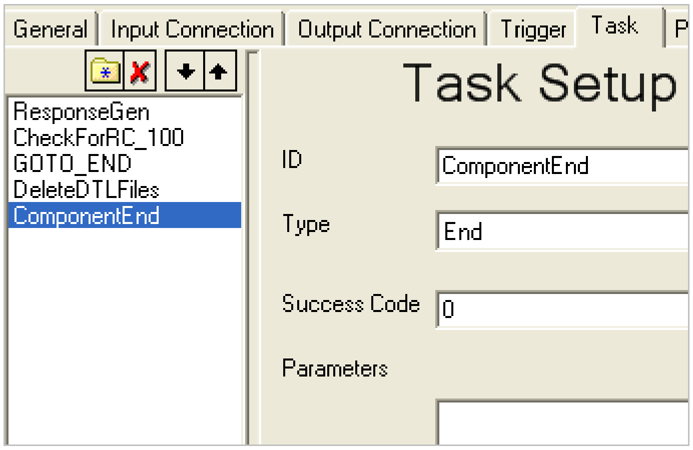
Type	Description

	Copy
	Copy a file or files from one directory to another.
	Notes:
	This will overwrite any files with the same name in the target directory.
	You can include variables surrounded by %, as described in Variables_for_Workflows.pdf.
	Any existing tracking files are copied as well, before their source file is copied.
	Parameters:
	1 The first parameter line is the directory from which one or more files are to be copied
	OR
	the specific file to be copied
	2 If a directory is specified, the second parameter line is the file extension to match (e.g. EDI, TXT). The * wildcard character is supported for “all”.
	If copying an individual file, this line is left blank or set to a dummy value.
	3 The third parameter line is the target directory.
	Example 1:
	Copy all files with a .edi extension from the specified directory to the target directory.
	`c:\PickupDir`
	`EDI`
	`%FSDPROCDIR%`
	Example 2:
	Copy a specific file from the specified directory to the target directory.
	`c:\PickupDir\filename.edi`
	`<blank>`
	`%FSDPROCDIR%`
	Example 3:
	Copy all files from the specified directory to the target directory.
	`c:\PickupDir`
	`*`
	`%FSDPROCDIR%`

Type	Description
	Move Move a file or files from one directory to another. Notes: This will overwrite any files with the same name in the target directory. You can include variables surrounded by %, as described in Variables_for_Workflows.pdf. Any existing tracking files are moved as well, before their source file is copied. Parameters: The parameters are the same as for **Copy**.
	Clean Delete all zero-byte files in the specified directory. Parameter: 1 The only parameter line is the directory. Examples: `%PROCDIR%` or `c:\docsplitter\out`

Type	Description
	ONRETURNCODE If the return code of the task matches a given value, branch to another task. Otherwise, execute the next task in the list. Parameters: 1 The first parameter line is the ID of the task to check. 2. The second parameter line is the return code number 3. The third parameter line is the ID of the task to go to. This example checks the ResponseGen task's return code. If it is 100, then processing goes to the DeleteDTLFiles task:  ResponseGen.xml* General Input Connection Output Connection Trigger Task Property ResponseGen CheckForRC_100 GOTO_END DeleteDTLFiles ComponentEnd Task Setup ID: CheckForRC_100 Type: ONRETURNCODE Success Code: 0 Parameters: ResponseGen 100 DeleteDTLFiles
	JAVA Load and run a class written in Java. This can be a user-written class. It must support an Automator Java Task interface. Parameters: 1 The first parameter line is classname,method. 2 The second parameter line is the class path 3 The third is the parameter string

Type	Description
	GOTO Unconditionally jump to another task. Parameters: The first parameter line is the task ID to go to. Example:  General Input Connection Output Connection Trigger Task Task Setup ResponseGen CheckForRC_100 GOTO_END DeleteDTLFiles ComponentEnd ID GOTO_END Type GOTO Success Code 0 Parameters ComponentEnd

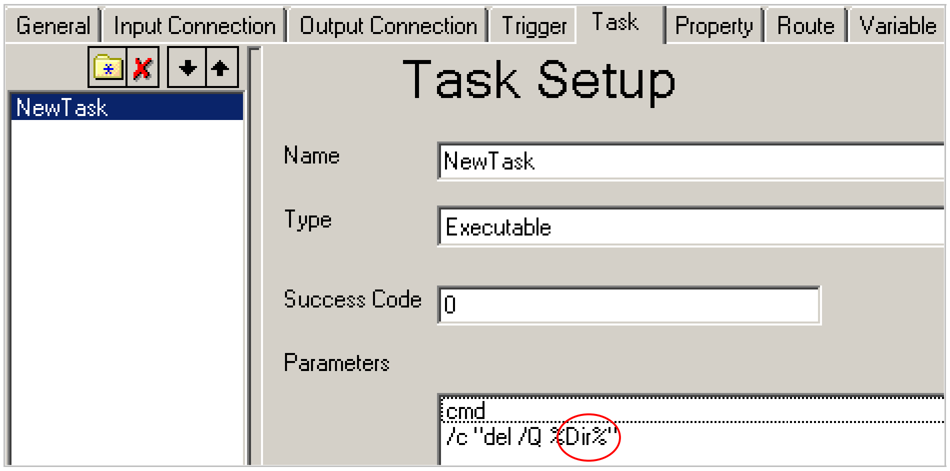
Type	Description
	DELETETRACKFILES If track files have been retained to this point, delete them. Parameters: None Example:  General Input Connection Output Connection Trigger Task Prope Task Setup ResultsOut DeleteTrackfiles ComponentEnd ID: DeleteTrackfiles Type: DELETETRACKFILES Success Code: 0 Parameters **End** Marks the end of processing for this component. Similar to an END statement in other programming languages. Parameters: None Example:  General Input Connection Output Connection Trigger Task P Task Setup ResponseGen CheckForRC_100 GOTO_END DeleteDTLFiles ComponentEnd ID: ComponentEnd Type: End Success Code: 0 Parameters
Success Code	Type the return code that the application uses for success. Typically, this is 0. Most Instream components use 100 for success. Success codes 1000 and 1001 override the normal error handling for this task. See Error Handling Behavior on page 43.
Parameters	*These are described under each type above.*

Task Tab Example: Simple Component to Delete Files

The component below does two things:

1. Opens a command window.

2. Deletes all files from the directory specified by the workflow, without displaying a confirmation message before each file is deleted.



Task Setup

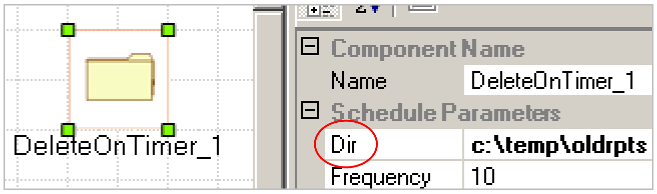
Name: NewTask

Type: Executable

Success Code: 0

Parameters:

```
cmd
/c "del /Q %Dir%"
```

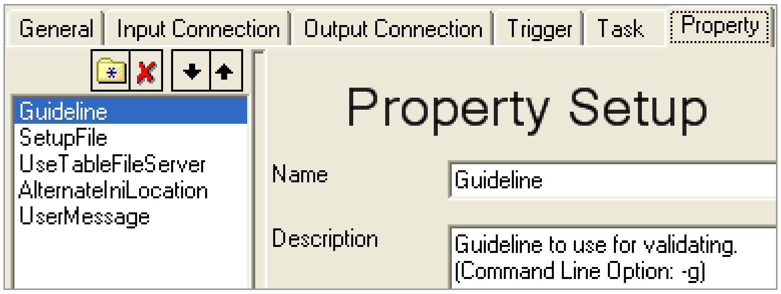


⊟ Component Name	
Name	DeleteOnTimer_1
⊟ Schedule Parameters	
Dir	**c:\temp\oldrpts**
Frequency	10

Property Tab

On the Property tab, you can set up variables to use in a component.

This example creates a variable called Guideline:



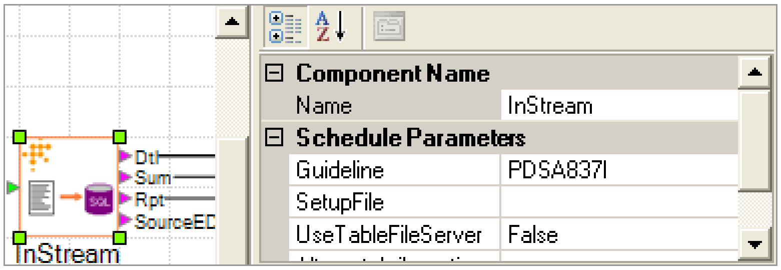
Guideline
SetupFile
UseTableFileServer
AlternateIniLocation
UserMessage

Property Setup

Name: Guideline

Description: Guideline to use for validating. (Command Line Option: -g)

The value can be different from one workflow to another. By default, properties show up under the workflow's Schedule Parameters when that component is selected. Here we can see the Guideline property:

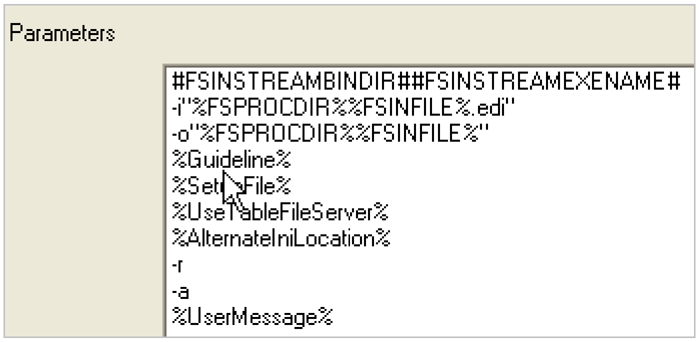


Component Name	
Name	InStream
Schedule Parameters	
Guideline	PDSA837I
SetupFile	
UseTableFileServer	False

If **Display in Grid** is enabled, the person authoring the workflow can see it., as in the example above - where the workflow author has filled in the value **PDSA837I**.

When a workflow runs this component, these values are substituted in for the variable. These variables are specific to this component.

Where are they used? Look in the parameters fields of the **Trigger** and **Task** tabs in the component template:

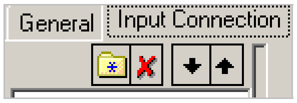


Parameters

```
#FSINSTREAMBINDIR##FSINSTREAMEXENAME#
-i"%FSPROCDIR%%FSINFILE%.edi"
-o"%FSPROCDIR%%FSINFILE%"
%Guideline%
%SetupFile%
%UseTableFileServer%
%AlternateIniLocation%
-r
-a
%UserMessage%
```

To set up variables under the Property tab:

1. Click the button if the fields are not available.



2. Fill out the fields (see below).

3. Save.

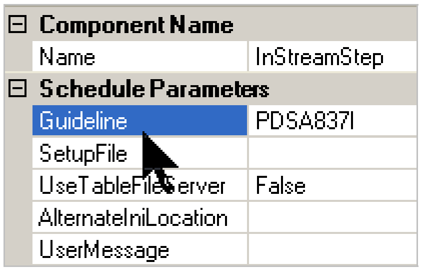
4. Repeat for additional properties.

Fields include:

Name Required. Type the name for the variable.

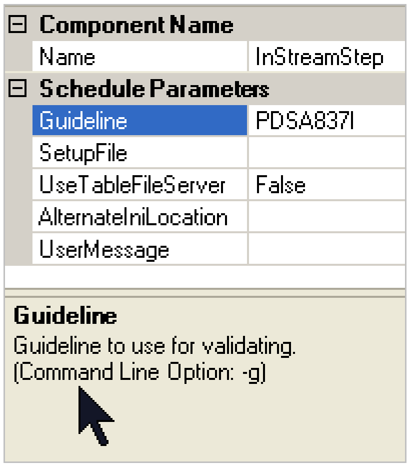
When using this variable on the **Trigger**, **Task**, or **Route** tabs, be sure that the spelling and capitalization matches exactly.

This name appears under schedule parameters:



⊟ **Component Name**	
Name	InStreamStep
⊟ **Schedule Parameters**	
Guideline	PDSA837I
SetupFile	
UseTableFileServer	False
AlternateIniLocation	
UserMessage	

Description Help text that displays when you click the schedule parameter.



⊟ **Component Name**	
Name	InStreamStep
⊟ **Schedule Parameters**	
Guideline	PDSA837I
SetupFile	
UseTableFileServer	False
AlternateIniLocation	
UserMessage	

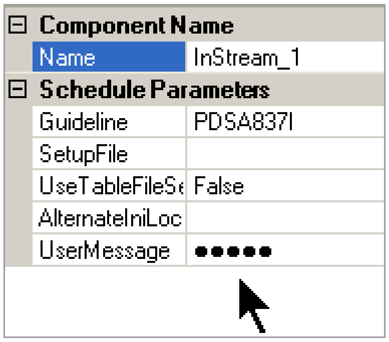
Guideline

Guideline to use for validating.
(Command Line Option: -g)

Required Must the workflow author supply a value for this property? If so, select Required.

Display as Password

If Display as Password is selected and a property's Edit Type is Text Entry, its value will be displayed in the workflow parameters as asterisks.



⊟ **Component Name**	
Name	InStream_1
⊟ **Schedule Parameters**	
Guideline	PDSA837I
SetupFile	
UseTableFileSe	False
AlternateIniLoc	
UserMessage	●●●●●

Display in Grid

If Display as Grid is selected, this property will display in the workflow's Component Properties and the workflow author will be able to change its value.

Not Used Value

If the workflow author supplies this value in the workflow, assume a null value.

Prepend Value

Characters or digits to appear before the user-supplied value.

Append Value

Characters or digits to appear after the user-supplied value.

Default Value

Value to be used if the user does not supply a value.

Edit Type **Text entry** - User types a value.

Selection - User selects a value from a drop-down list.

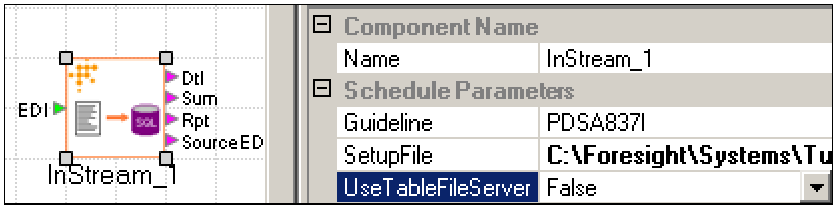
Selection Populates the drop-down list if Edit Type is Selection.

Display - Text displayed to user.

Value - Value actually used for that selection.

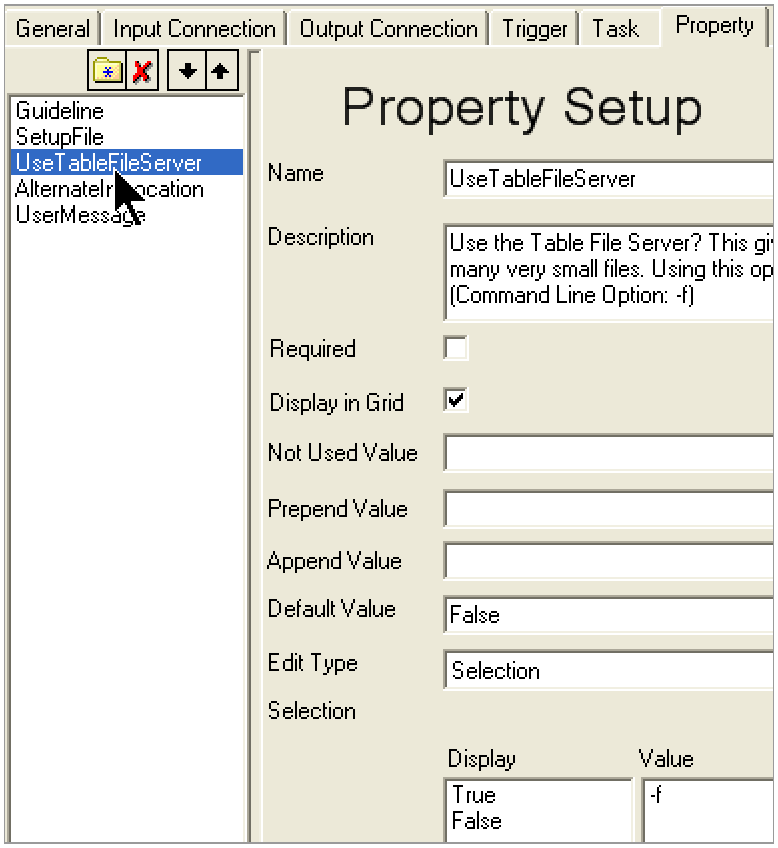
Example

Here are the schedule parameters for the Instream component in a workflow:



⊟ **Component Name**	
Name	InStream_1
⊟ **Schedule Parameters**	
Guideline	PDSA837I
SetupFile	**C:\Foresight\Systems\Tu**
UseTableFileServer	False

Under Component Templates, the Property tab is as follows. This example shows the setup for the UseTableFileServer variable. Since Edit Type is Selection, the schedule parameters will display a drop-down list. If the user selects **True** from the list, **`-f`** is added to the command line for Instream.



Guideline
SetupFile
UseTableFileServer
AlternateInvocation
UserMessage

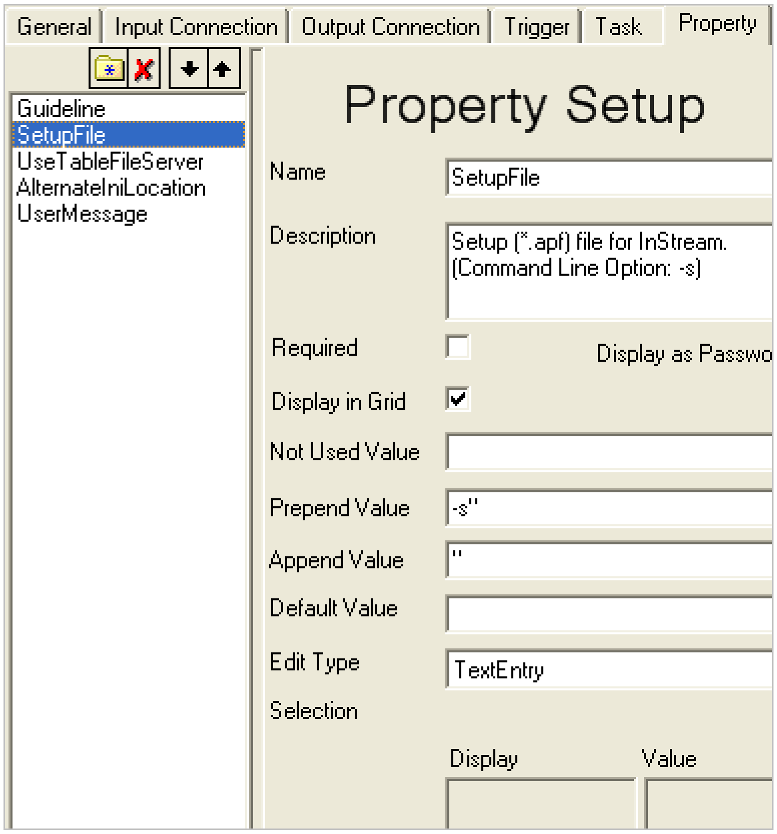
Property Setup

Name	UseTableFileServer
Description	Use the Table File Server? This gi many very small files. Using this op (Command Line Option: -f)
Required	□
Display in Grid	☑
Not Used Value	
Prepend Value	
Append Value	
Default Value	False
Edit Type	Selection

Selection

Display	Value
True	-f
False	

The next example shows the setup for the SetupFile variable. When the user types the path to the setup file, it will be preceded with **`-s“`** and followed by **`”`** when added to the command line for Instream.



Guideline
SetupFile
UseTableFileServer
AlternateIniLocation
UserMessage

Property Setup

Name: SetupFile

Description: Setup (*.apf) file for InStream. (Command Line Option: -s)

Required □ Display as Passwo

Display in Grid ☑

Not Used Value:

Prepend Value: -s"

Append Value: "

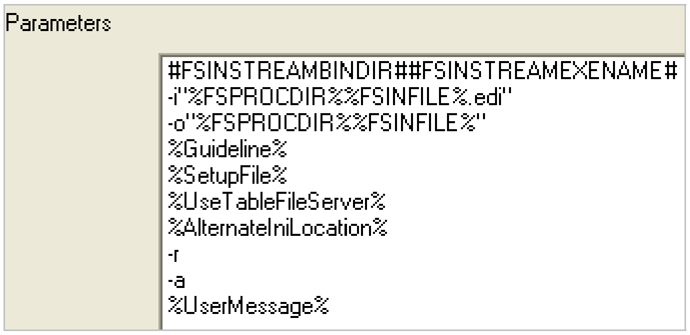
Default Value:

Edit Type: TextEntry

Selection

Display	Value

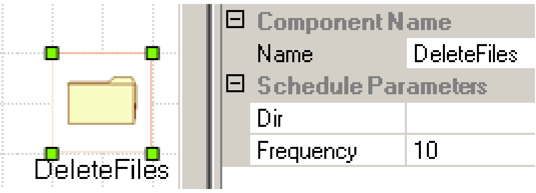
Here is how these variables are used on the Task tab for Instream:


Parameters

```
#FSINSTREAMBINDIR##FSINSTREAMEXENAME#
-i"%FSPROCDIR%%FSINFILE%.edi"
-o"%FSPROCDIR%%FSINFILE%"
%Guideline%
%SetupFile%
%UseTableFileServer%
%AlternateIniLocation%
-r
-a
%UserMessage%
```

Example - Creating a Component to Delete Files

The example in this section creates a component that deletes all files in a directory.



⊟ Component Name

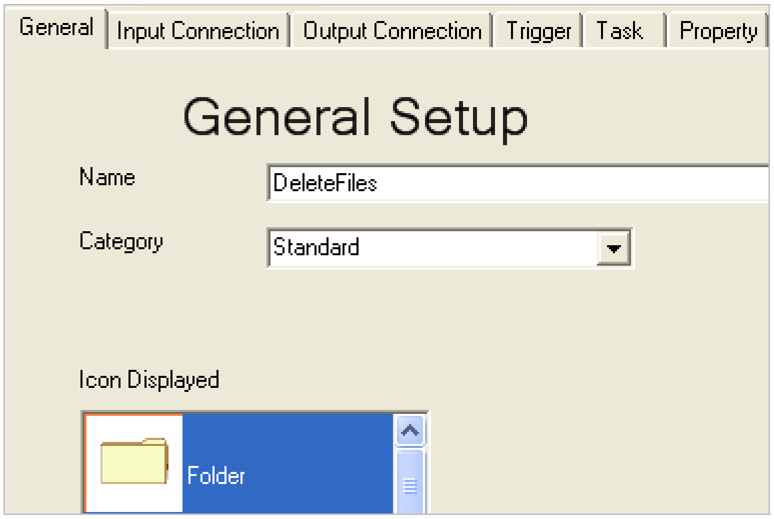
Name	DeleteFiles

⊟ Schedule Parameters

Dir	
Frequency	10

Example General Tab

Click the Model system and Choose **File | New | Component Template**. Fill out the General Tab as follows.



General Setup

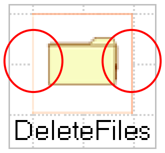
Name: DeleteFiles

Category: Standard

Icon Displayed: Folder

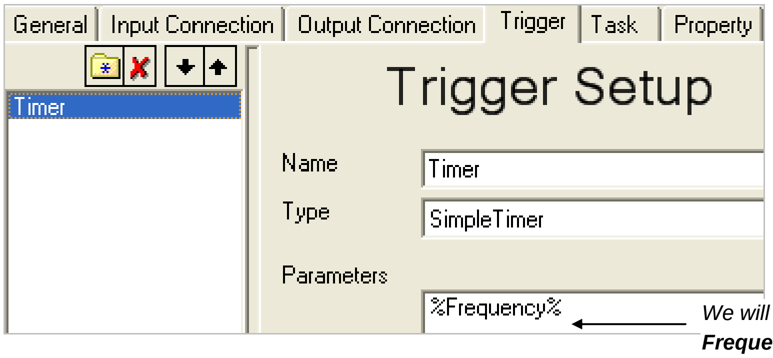
Example Input Connection and Output Connection Tabs

Skip Input Connection and Output Connection since it has none.



Example Trigger Tab

Set up a trigger called Timer.



Timer

Trigger Setup

Name: Timer

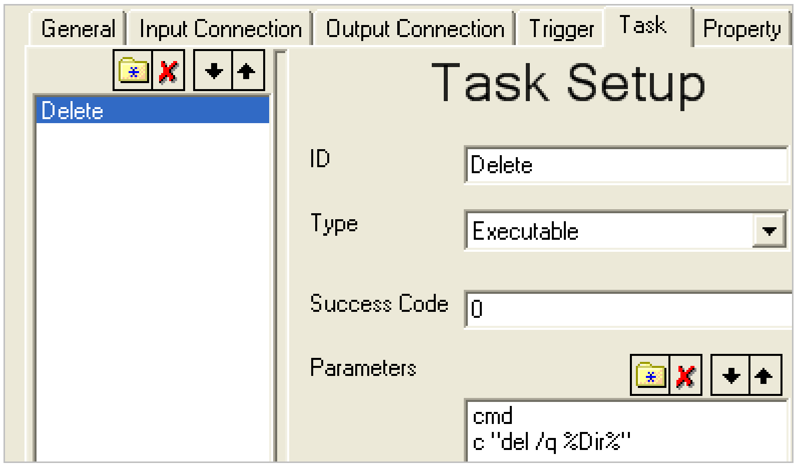
Type: SimpleTimer

Parameters: %Frequency%

*We will define the variable **Frequency** under the Property tab.*

Example Task Tab

Set up this task:



Delete

Task Setup

ID: Delete

Type: Executable

Success Code: 0

Parameters:

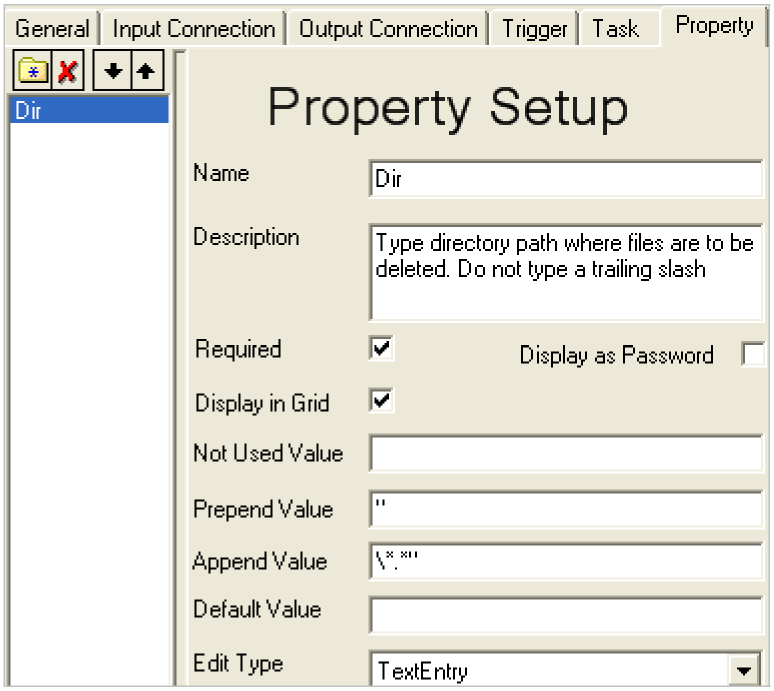
```
cmd
c "del /q %Dir%"
```

Under Parameters:

- The first line (**cmd**) opens a DOS shell.
- The next line shows how to set up the DOS **del** command. **/Q** prevents it from prompting before deleting each file. We will define the variable **Dir** under the Properties tab.

Example Property Tab

Set up the **Dir** property:



Property Setup

Name: Dir

Description: Type directory path where files are to be deleted. Do not type a trailing slash

Required ☑ Display as Password ☐

Display in Grid ☑

Not Used Value:

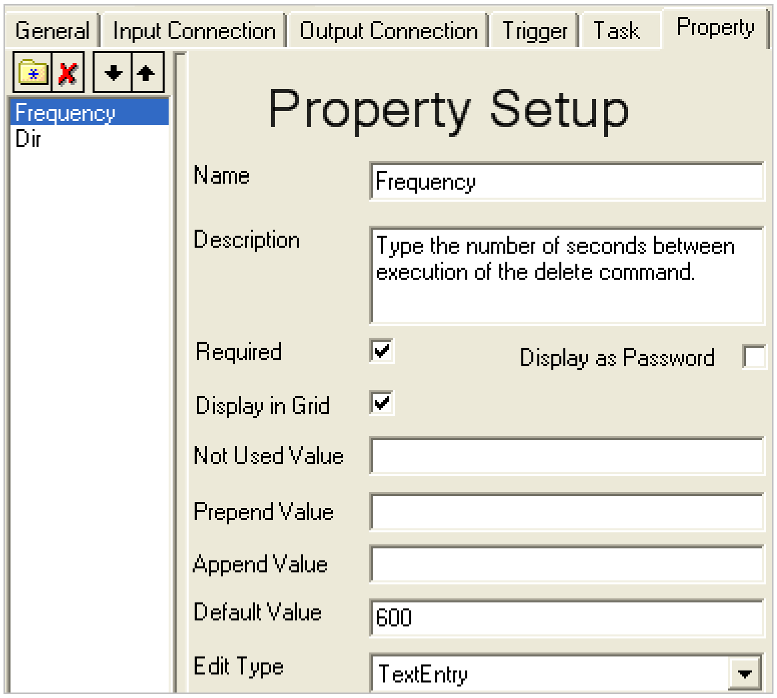
Prepend Value: "

Append Value: *.*"

Default Value:

Edit Type: TextEntry

Set up the **Frequency** properties:



Property Setup

Name: Frequency

Description: Type the number of seconds between execution of the delete command.

Required ☑ Display as Password ☐

Display in Grid ☑

Not Used Value:

Prepend Value:

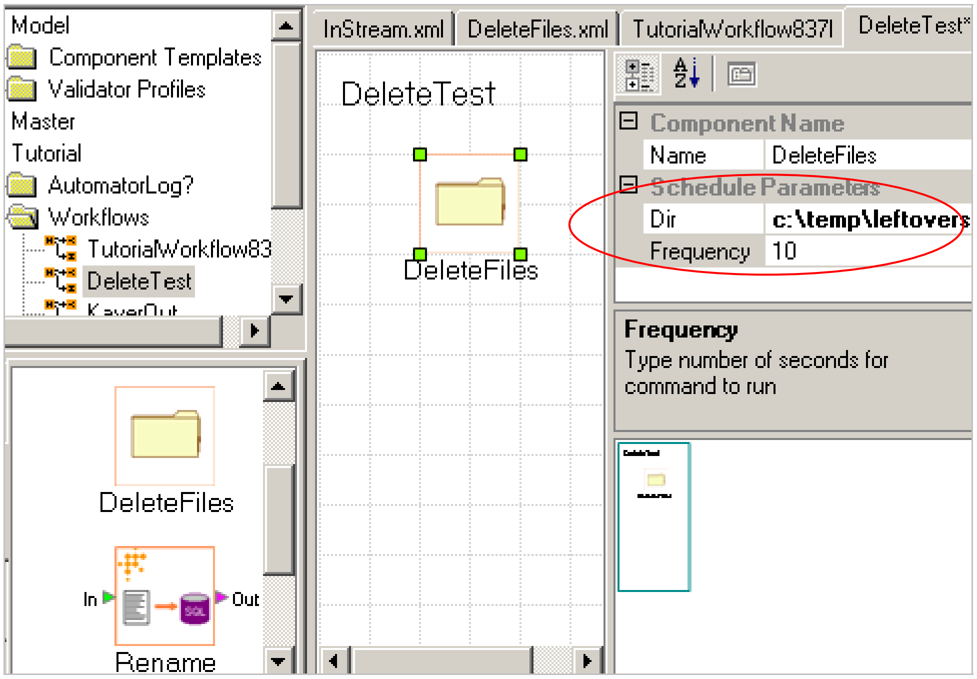
Append Value:

Default Value: 600

Edit Type: TextEntry

Save and Test New Component Template

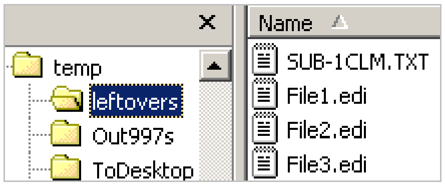
Create a simple workflow that uses this component. Add values for **Dir** and **Frequency** and save the workflow.



Name: DeleteFiles
Dir: c:\temp\leftovers
Frequency: 10

Frequency
Type number of seconds for command to run

Load up some files in the directory and then start the workflow.



9 Appendix A - Track Files

What is a Track File?

Track files are used by TIBCO Foresight to provide data to other Foresight components such as Foresight™ Operational Monitor, Archiver and Transaction Insight. By default, track files are generated, utilized, and deleted by the system, leaving no trace behind.

Advanced users or administrators may choose to use ForesightStudio to retain track files for other purposes. For example, you might wish to save the track files into a separate OutputFolder component and then move them elsewhere via a tool such as FTP or MQ.

Important

Note that undeleted track files can alter the behavior of the Operational Monitor component and may cause confusion if they are not managed properly. **We recommend the use or storage of track files only with the oversight of the TIBCO Foresight Professional Services Group.**

Generation of Track Files

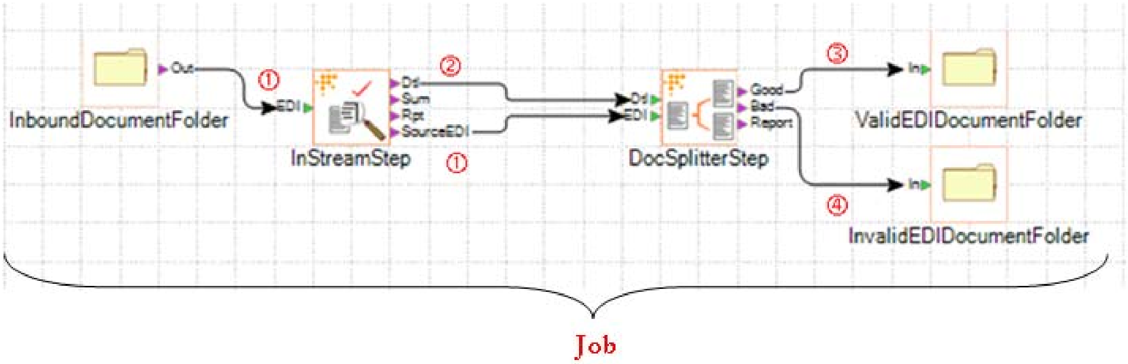
By default, the Automator component of Studio creates a track file (.trk) for every file dropped into or created during the execution of a ForesightStudio workflow. A track file logs the file's movement through the workflow and notes event such as:

- split points for the data in the file
- routing of the file itself
- additional files created from the data, such as detail files.

When a source EDI file is dropped into the InboundDocumentFolder, the dropped file is assigned a JobID. All track files created from this source file as it moves through the workflow are associated with this JobID and are also assigned a unique FileID.

The JobID is associated with the movement of the source EDI file; it is not associated with the workflow. Additional source EDI files dropped into the InboundDocumentFolder receive their own unique JobID, and track files generated are associated with that JobID.

The following graphic depicts a file moving through a workflow.



As the workflow is executing, Automator generates track files at the following points:

- A source EDI file is dropped into the **InboundDocumentFolder** and a JobID is created. All track files created as a result of this EDI source file will be associated with this JobID.
- The source EDI file generates track file ①. Because the source EDI file passes through **InstreamStep** unchanged, track file ① continues to log the process of the file as it proceeds to **DocsplitterStep**; no new track file is generated.
- Track file ② is generated to track the Detail file produced by Instream.
- Track files ③ and ④ are generated to track the files created by **DocSplitterStep**. File ③ tracks the Good data file and ④ tracks the Bad data file. Each of these contain the same JobID as the original file.

A second source EDI file dropped into the InboundDocumentFolder is assigned a different JobID, and track files generated from that file contain that JobID. Any subsequently dropped source EDI files would have their own JobIDs, and so on.

Track File Contents

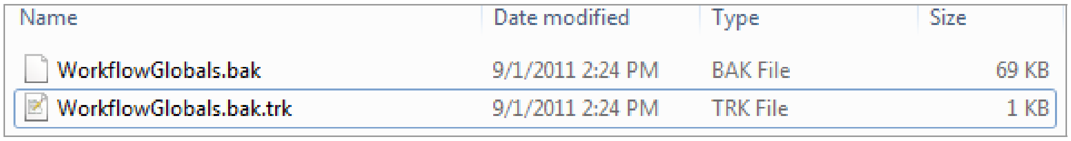
A track file contains two pieces of information:

JobID – Identifies the job being completed. All files in the same job will have the same JobID associated with them. For example: JobID=a1f51ca5-d4c7-11e0-bb8d-fbc3d6a05b3b.

FileID –Each file in a job has its own track file, which contains the ID of that file. For example: FileID=a2f51ca3-d4c3-11e0-bb9d-fbc3d6a05b46

Retaining Track Files

Track files appear in the designated Output folder with a .trk extension and an identical name to the file they are tracking, as shown here:

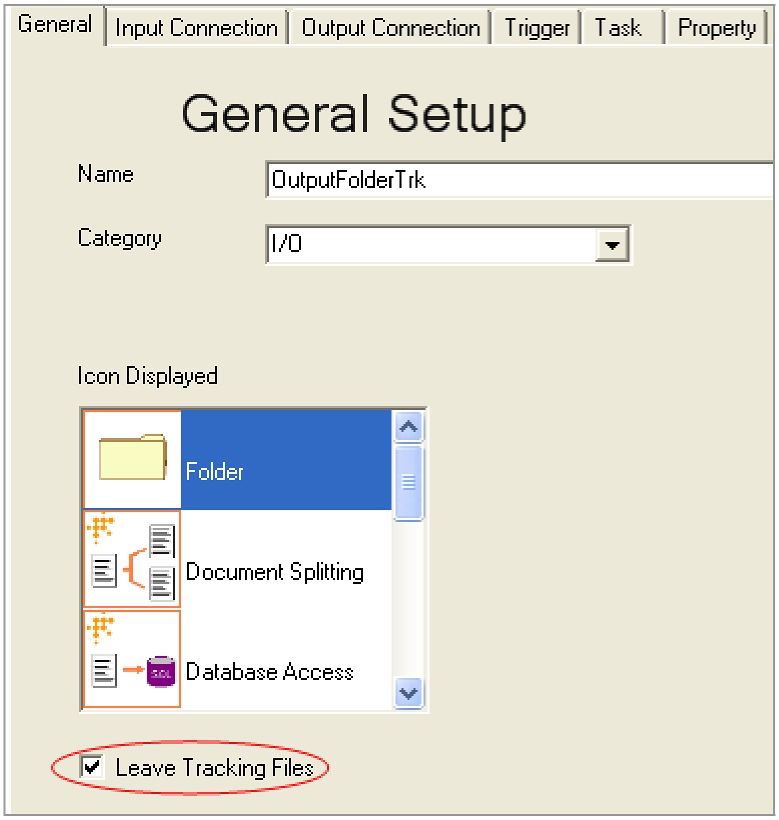


Name	Date modified	Type	Size
WorkflowGlobals.bak	9/1/2011 2:24 PM	BAK File	69 KB
WorkflowGlobals.bak.trk	9/1/2011 2:24 PM	TRK File	1 KB

To override default behavior and retain track files, do the following in ForesightStudio:

1. Access the Component Template for which you want to enable track files. For example: **Model | Component Templates | OutputFolderTrk.xml**.

2. Check **Leave Tracking Files**.



General Setup

Name: OutputFolderTrk

Category: I/O

Icon Displayed: Folder

Document Splitting

Database Access

☑ Leave Tracking Files

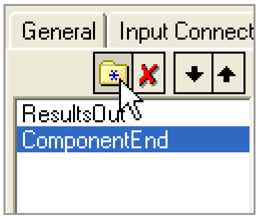
Deleting Track Files

When track files are retained, they should be used as needed and then moved or deleted as soon as possible.

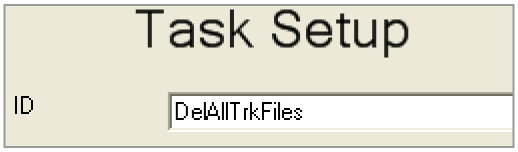
To delete track files as part of a task, do the following in ForesightStudio:

1. Under **Model | Component Templates** open the component and access the Task Tab.

2. Under the task list, create a new task and move it up or down as needed.



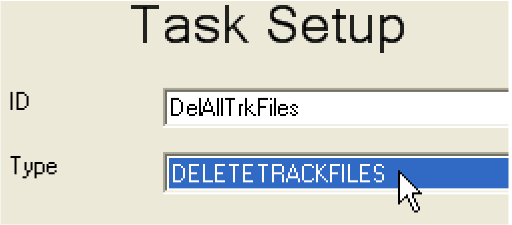
2. Select the Task ID. Change the ID to the wording of your choice.



Task Setup

ID DelAllTrkFiles

3. Using the dropdown list for the Type, select DELETETRACKFILES.

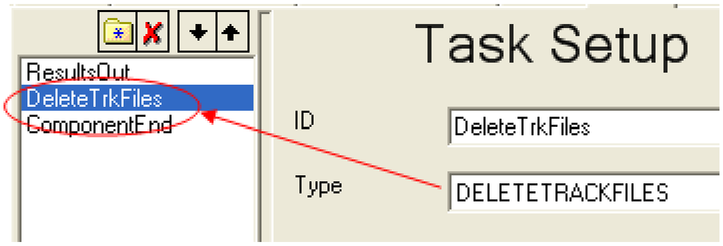


Task Setup

ID DelAllTrkFiles

Type DELETETRACKFILES

4. The new ID appears on the task list.



Task Setup

ID DeleteTrkFiles

Type DELETETRACKFILES

The component will now force the deletion of any remaining track files as part of this task.

10 Index