TIBCO Foresight® Studio®

Automator

Software Release 8.2
April 2015
Updated: September 2017

Two-second advantage®

Important Information

General Contact Information

TIBCO Software Inc.
3303 Hillview Avenue
Palo Alto, CA 94304 USA
Tel: +1 650 846 1000
Fax: +1 650 846 1005

Technical Support

E-mail: support@tibco.com
Web: https://support.tibco.com
(Note: Entry to this site requires a username and password. If you do not have one, you can request one. You must have a valid maintenance or support contract to use this site.)

Contents

1 Introduction 1

System Requirements 1
Intended Audience 1
Installing Automator 1
On Windows 1
On UNIX 1
Big Picture 2
Example Workflows 3

2 Using Automator 5

Automator Filenames 5
Starting Automator 5
Starting Automator from within Studio 6
Starting Automator from a Command Line 6
Starting Automator as a Service 13
Automator Display 14
Stopping Automator 14
Changing a Workflow 15
Running Automator on UNIX 16
Setting Environment Variables 16
Sample UNIX Automator Script 16
Log Files 17
Turning on logging 17
Log Contents 17
Deleting Unwanted Logs 19
Using a Log File 21
Problem Report 23

3 Appendix A – PGN and Track File Generation 25

4 Appendix B – Return Codes and Troubleshooting 27

Seeing Return Codes 27
Virus Checking and Foresight Products 28

1 Introduction

System Requirements

Please see **TIB_foresightstudio_8.2.0_readme.txt** for information about system requirements.

Intended Audience

This document is intended for technical staff who are implementing workflows that include TIBCO Foresight products.

Before using Automator, you should have a general familiarity with TIBCO Foresight® Instream® and with the EDI data flow through your company.

Installing Automator

On Windows

The TIBCO Foresight® Studio® installation program automatically installs Automator in a ForesightAutomator directory.

The TIBCO Foresight® Transaction Insight® installation program on Windows lets you choose whether to install Automator.

On UNIX

Foresight offers the **TIB_foresightstudio_***version***_automator_***platform***.tar.gz** archive to install Automator on UNIX.

The accompanying **TIB_foresightstudio_***version***_automator_installation_***platform***.tar.gz** contains directions.

Big Picture

Automator executes a workflow that moves files and starts programs to process them. These Automator scripts are in a workflow language using XML syntax. By default, they are in:

Foresight\Systems*systemname***\Workflows\1.0***workflowName*

Workflows are edited using the workflow editor module of Foresight Studio®, which is described in **ForesightStudio.pdf**.

When you start Automator:

1. It starts a new log file if logging is enabled (see page 16).
2. It checks each component in the workflow to see if its trigger condition (timed, file available, or multiple files available) has been met.
3. If the component's trigger condition has been met, the component will then attempt to execute the tasks in the Task section.
4. If all the tasks that were executed were successful, files may be moved to other components based on file extension and workflow layout. Files not moved will be moved to the components Out directory. If any of the executed tasks failed, all the resulting files will be moved to the component's Error directory.

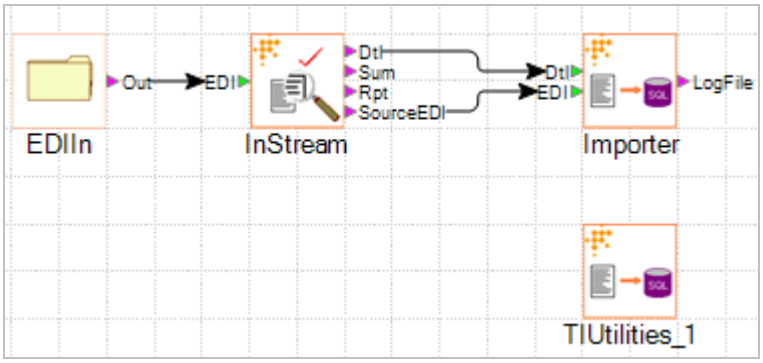
Additionally, Automator creates a track file (.trk) for every file dropped into or created during the execution of a Foresight Studio workflow. For more information on track files, see **Trk_Files.pdf**.

Example Workflows

Automator starts workflows like the following examples. With Foresight Studio, you can define workflows that will suit your own needs.

Validate and Feed EDI Data into Transaction Insight®

This simplified workflow moves EDI files from a directory into Instream® validation. The EDI file and the detail results file from validation are processed by Importer (put into Transaction Insight database). TIUtilities runs on a schedule to update the summary information for Transaction Insight.



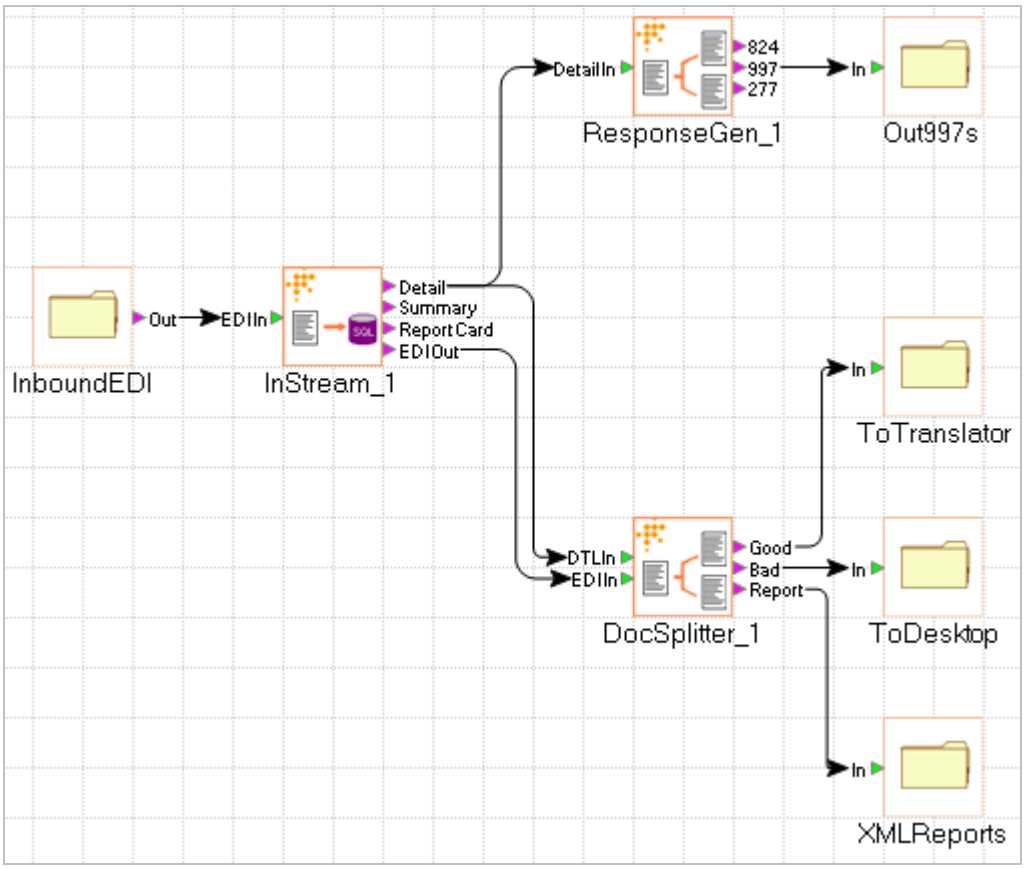
Validate, run Response Generator and DocSplitter

This workflow moves EDI files from a directory into Instream validation.

The detail results file from validation is processed by Response Generator and the resulting 997s go into a directory.

Meanwhile, the EDI file and the detail results file from validation are processed by DocSplitter. After this step:

- Good EDI is put in a directory to await the translator.
- Bad EDI is put in a directory for the support staff to examine with TIBCO Foresight® HIPAA Validator® Desktop.
- The DocSplitter XML reports go into another directory.


DetailIn
824
997
277
In
ResponseGen_1
Out997s
Out
EDIIn
Detail
Summary
Report Card
EDIOut
InboundEDI
InStream_1
In
ToTranslator
DTLIn
EDIIn
Good
Bad
Report
In
DocSplitter_1
ToDesktop
In
XMLReports

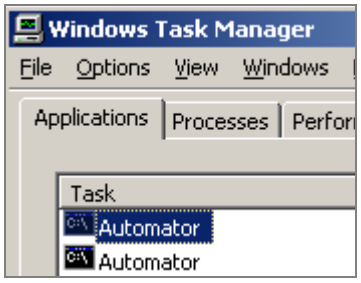
2 Using Automator

Automator Filenames

Not allowed:	% or #
Not recommended:	Any special characters except period and underscore

Starting Automator

Each running instance of Automator runs one and only one workflow. To run multiple workflows, you will need to run more than one instance of Automator.



Things that may cause problems:

- Running the same workflow with multiple copies of Automator.
- Starting multiple workflows within one second of each other.

Starting Automator from within Foresight Studio

Enter Foresight Studio, select the workflow, and use these toolbar buttons:

	Toolbar button
Starting	
Stopping	

Foresight Studio runs the last Automator that was installed on that machine, regardless of the version of Foresight Studio. If the wrong version of Automator runs, check the FSROOT and FSAUTOROOT environment variables on the machine where you are starting Automator.

You can change these environment variable settings. The location of environment variables varies by operating system. Check under something similar to **Control Panel | System | Advanced System Settings | Environment Variables | System Variables**:

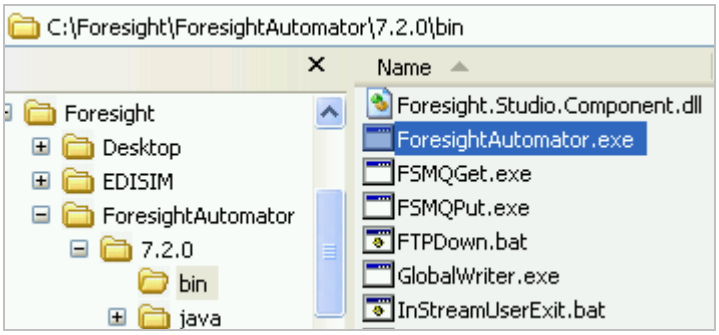
System variables

Variable	Value
FSAUTOROOT	ForesightAutomator\7.14.0\Bin
FSMAINROOT	C:\Foresight
FSROOT	C:\Foresight

Starting Automator from a Command Line

This method lets you start Automator for a workflow that is deployed on a machine that does not have Foresight Studio installed.

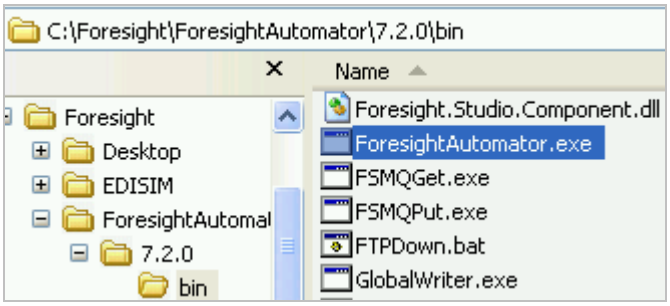
The default location for Automator executable **Automator.exe** is **Foresight\ForesightAutomator** *version***bin**.



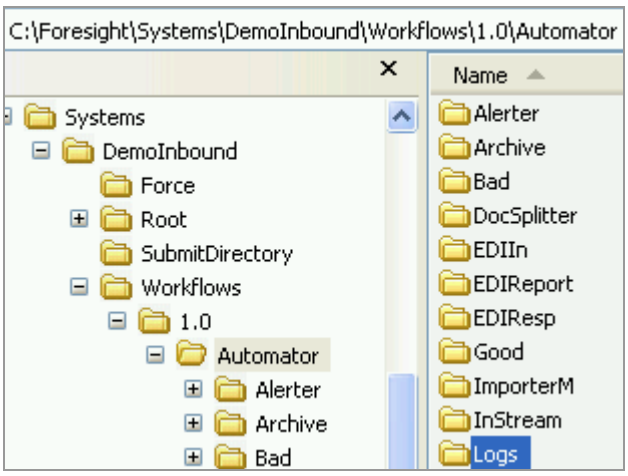
You can start Automator from the command line. The format is:

ForesightAutomator.exe -a"*AutomatorRoot***" -b***timeout* **-d -f"***ForesightRoot***" -i***number* **-l"***path***" -L***levels* **-n -m***number* **-o***PGNfilepath* **-p"***portnumber***"-s***SystemName* **-t***KillTime* **-w***WorkflowPath* **-z**

Where:

Parameter	Description
-a*"AutomatorRoot"*	Required. Path to ForesightAutomator.exe, relative to the "-f" directory. This is not a full path. Example: `-a"ForesightAutomator\7.2.0\bin"` 
-b*timeout*	Optional. Number of seconds until Automator shuts down. Example: `-b600` Causes Automator to shut down in 10 minutes.
-d	Display Automator window.
-f*"ForesightRoot"*	Required. Foresight high-level directory. Example: `-f"C:\Foresight"` See ① on page 12.
-i*number*	Specifies the number of seconds between log file resets. Default is to start a new log file whenever the workflow starts. Example: `-i600` creates a new log file every 10 minutes and `-i86400` will give you a new one every 24 hours.

Parameter	Description

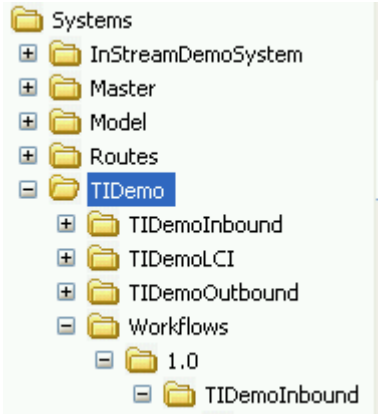
-l*"path"*	Turn on logging (see page 16).
	Example (no path and filename specified):
	`-l`
	The log will go under the workflow:
	 
	
	If you don't specify a log path and file, a new logfile is created after the interval specified with -i or when Automator starts, and the old log is left behind.
	
	Example (path and filename specified):
	`-l"c:\temp\DemoInboundW1.log"`
	You must include a filename if you include a directory.
	If you specify the log's path/filename, the file will be overwritten after the interval specified with -i or when Automator starts.

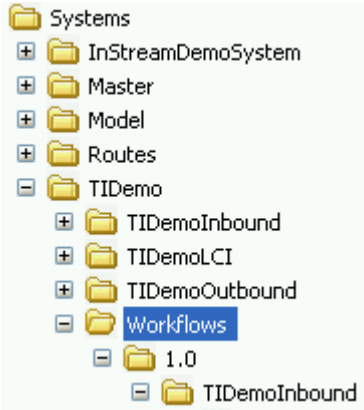
Parameter	Description

-L*levels*	Log file contents. *levels* are any combination of these:
	`a` AUTOMATOR ERROR
	`c` COMPONENT ERROR
	`x` COMPONENT EXECUTION
	`r` COMPONENT EXECUTION RESULT
	`f` FILE MOVEMENT
	`i` INFORMATIONAL
	`d` DEBUGGING
	`1` equivalent to options `aci`
	`2` equivalent to options `acxri`
	`3` equivalent to options `acxrfid`
	Examples:
	`-Lacxr`
	`-Lacxrfid`
	`-L2`
	See Log Files on page 17 for details.

-m*number*	Optional.
	Specifies the number of times Automator should try to copy or move a file within the workflow (e.g., from component to component).
	The default retry count is 5.
	Examples:
	`-m29`
	`-m110`

-n	Optional. Turns off both Automator track (.trk) and TIBCO Foresight® Operational Monitor PGN file creation. These files are used by Foresight Studio workflows that require ID values to be propagated across components. **If your workflow does not use these ID values**, you can use this switch to prevent the generation of these files. **Note:** If you use TIBCO Foresight® Archive and Retrieval System and/or Foresight® Operational Monitor you must create track files. Do not use the -n option. Refer to **Trk_files.pdf** for information about .trk files.
-o "*PGNfilepath*"	Optional. Tells Automator to create Operational Monitor PGN files containing event information and put them in the directory specified by *PGNfilepath*. When running a workflow from within Foresight Studio, the "eyeball" toolbar button turns on this feature and puts files in the OpMonLoggingDir under Foresight's Systems directory.

Parameter	Description
-p"*portnumber*"	Default is 6000. Port that Automator is listening on that machine (each Automator instance running must be listening on a different port). If you are running simultaneous instances of Automator, use the -p parameter to give each its own port. A "Bind" error means that two instances of Automator are trying to use the same port. At present, this will not cause any actual functionality loss since this is for future use. Example: `-p"6001"`
-s*SystemName*	Required. System being used. Example: `-s"TIDemo"`  See ② on page 12.
-t*killTime*	(default=one hour) How many seconds to wait until Automator kills a component of the workflow. Example: You used -t3600. Automator has not heard from Instream, a component in the workflow, that it completed validating a file within 3600 seconds of starting the validation. Automator therefore kills the instance of Instream that is validating that file and moves all associated files to its Error directory. It then starts another instance of Instream to validate the next file.

Parameter	Description
- w*WorkflowPath*	Required. Workflow path, relative to –s parameter. Example: `-w"Workflows\1.0\TIDemoInbound"`  See ③ on page 12.
-z	Optional. Automator checks for locked files by default. Use **-z** to turn off automatic lock-checking.

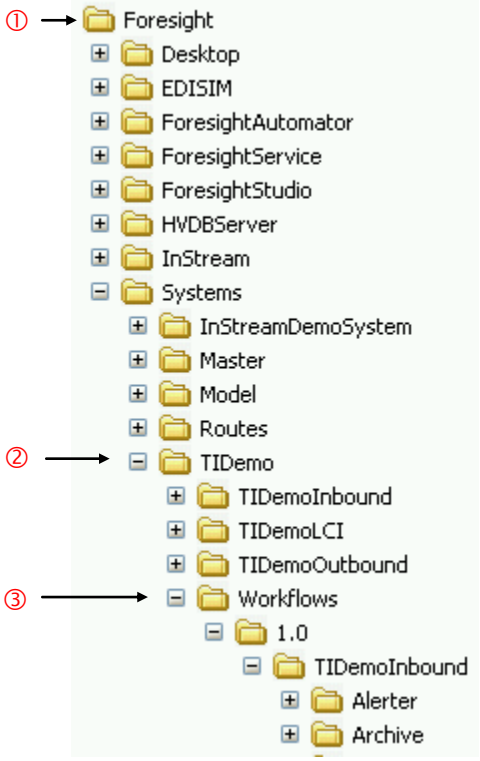
Command Line Examples

```
ForesightAutomator.exe -d -l -p”6000” -f"C:\Foresight"
-s”TIDemo” -a”ForesightAutomator\7.2.0\bin”
-w”Workflows\1.0\TIDemoInbound”
```

```
ForesightAutomator.exe d
-l"C:\ForesightGlobalsTest\Systems\InStreamDemoSystem\
AutomatorLogs\1.0\InStreamDemoWorkflow 2007-09-18 1319.txt" -L3 -
f"C:\ForesightGlobalsTest"
-a"ForesightAutomator\6.2.0\Bin"
-s"InStreamDemoSystem"
-w"Workflows\1.0\InStreamDemoWorkflow"
```

Automator puts together these pieces to find the workflow directory. It looks in:

*ForesightRoot***Systems***SystemName**WorkflowPath*\

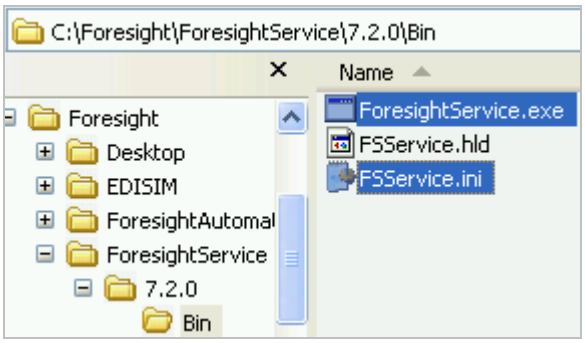


Starting Automator as a Service

You can run Automator as a service so that it is not running under a particular user. To do this:

1. Open **Control Panel | Administrative Tools | Services**.
2. Double-click on **ForesightService** and click **Start**.
3. To have this happen automatically on boot up, change the Startup type to Automatic.

This runs a program called ForesightService.exe, which runs command lines in FSService.ini:



Sample FSService.ini:

```
[Servers]

Utility=C:\Foresight\ForesightAutomator\7.2.0\bin\
ForesightAutomator.exe -l -aForesightAutomator\7.2.0
-fC:\Foresight -sKaverUtility -wAutomator

Inbound=C:\Foresight\ForesightAutomator\7.2.0\bin\
ForesightAutomator.exe -l -aForesightAutomator\7.2.0
-fC:\Foresight -sKaverInbound -wAutomator

Outbound=C:\Foresight\ForesightAutomator\7.2.0\bin\
ForesightAutomator.exe -l -aForesightAutomator\7.2.0
-fC:\Foresight -sKaverOutbound –wAutomator
```

If a path contains spaces, surround it with quotation marks:

```
Outbound=”C:\Foresight\Foresight Automator\7.2.0\bin\”
ForesightAutomator.exe -l ”–aForesight Automator\7.2.0”
-fC:\Foresight -sKaverOutbound –wAutomator
```

Quotation marks can enclose the parameter or not. Both of these are correct:

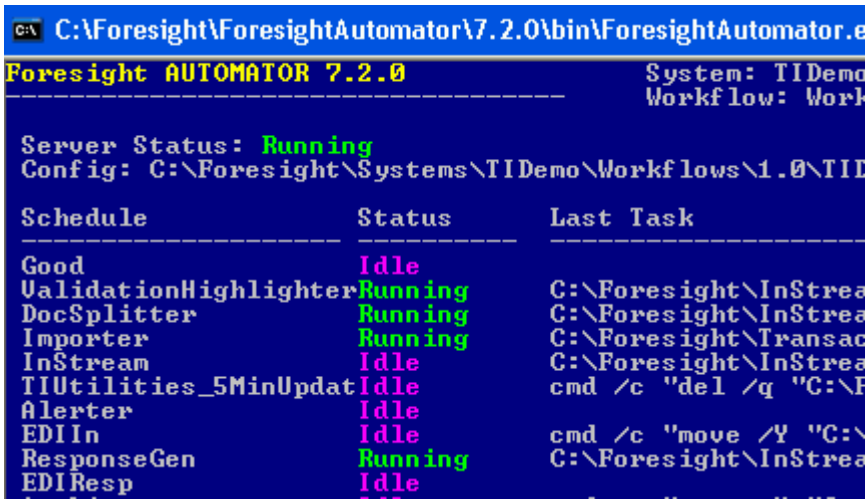
```
–a”Foresight Automator\7.9.0”

”–aForesight Automator\7.9.0”
```

Automator Display

Automator can display a window that shows workflow information:

- When starting Automator from the command line, use the **d** parameter.
- When starting Automator from within Foresight Studio, the component information automatically displays.
- It does not display a window if you are running it through ForesightService.

This window shows information about each component, like this:



The Status column displays:

Idle in pink	Last time the task ran, it was successful.
Idle in yellow	Last time the task ran, it was not successful. Check the log.
Running	Currently running.
Dead in red	Configuration or other fatal error in a running Automator component. Check the Automator log.

Stopping Automator

To stop Automator, close the Automator command window or click the ■ button on Foresight Studio's toolbar.

If you are running it through ForesightService, stop the service through **Control Panel | Administrative Tools | Services**.

The processes that it started should then finish their current task and shut down.

On AIX, you can stop Automator with **Ctrl-C** or the kill command:

```
ps -a
kill -9 pid
```

Changing a Workflow

Automator runs workflows that consist of one or more XML files. By default, they are in:

Foresight\Systems*system***\Workflows***version**workflow*

The recommended way to change a workflow is through Foresight Studio (see **ForesightStudio.pdf**).

However, if you do not have Foresight Studio, you can edit an XML file directly and make changes. See Appendix H of **TIB_transactioninsight_*n.n*_installation.pdf**, or call TIBCO Foresight Technical Support.

You can change a workflow while Automator is running it. However, for changes to actually take effect, you will need to stop and restart Automator.

Running Automator on UNIX

Setting Environment Variables

Some workflow programs might require environment variables. You can do this in the profile so they are always defined, or set them in the Automator startup script or in a custom script created to run TIUtilities.

```
export LIBPATH=/home/ti/TransactionInsight/3.3.0/bin
export FSTIUTILITIESINI=/home/ti/TransactionInsight/3.3.0/bin
```

If you define LIBPATH in the Automator startup script, include all directories that need it, separated by colons. This example points to the bin directories of Automator, Instream, and TI. These directories contain programs run by the script:

```
export LIBPATH=/home/ti/foresight/ForesightAutomator/7.1.0/bin:/home/ti/
foresight/instream/bin:/home/ti/foresight/TransactionInsight/3.3.0
/bin:$LIBPATH
```

Sample UNIX Automator Script

```
#!/bin/sh

#
# Change the following script variables to affect ForesightAutomator behavior:
#
# 'Display=-d' - turns on ForesightAutomator's status display.
# 'Logging=-l' - turns on ForesightAutomator's logging functionality.
# 'LogFile=<logfile>' - specifies the log file to be written to.
# 'LogLevel=<loglevel>' - specifies ForesightAutomator's logging level. Valid
values are -L1, -L2, and -L3.
#
Display=-d
LogFile=
Logging=-l $LogFile
LogLevel=-L3

export FSINSTREAMINI=/home/ti/foresight/instream/bin
export FSMAINROOT=/home/ti/foresight
export FSSYSTEM=INBOUND837I
export FSAUTOROOT=/home/ti/foresight/ForesightAutomator/7.1.0/bin
export LIBPATH=/home/ti/foresight/ForesightAutomator/7.1.0/bin:/home/ti/
foresight/instream/bin:/home/ti/foresight/TransactionInsight/3.3.0/bin:$LIBPATH
export CLASSPATH=:/home/ti/foresight/instream/java/lib:/home/ti/foresight/
ForesightAutomator/7.1.0/java/lib:$CLASSPATH
export FSTIUTILITIESINI=/home/ti/foresight/TransactionInsight/3.3.0/bin
export FSIMPORTERINI=/home/ti/foresight/TransactionInsight/3.3.0/bin/

/home/ti/foresight/ForesightAutomator/7.1.0/bin/ForesightAutomator $Display
$Logging $LogLevel -wWorkflows/1.0/UNIXDemoSimple -sINBOUND837I
-f/home/ti/foresight -a/home/ti/foresight/ForesightAutomator/7.1.0/bin
```

Please see **Variables_for_Workflows.pdf** for information about the variables set here.

Log Files

Log files contain information about steps taken by Automator.

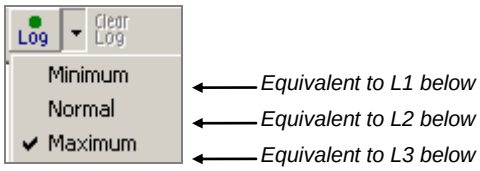
Turning on logging

Automator runs from…	To start logging	To see the log	To set logging level
Command line	Use **l** parameter. This is a lowercase "el."	Log locations can be specified on the command line with the **l** parameter (see page 6). Default location is under the workflow. Example: **Foresight\Systems***system***** Workflows\ *version* \ *workflow* \ Logs	Use the **L** parameter.
Within Foresight Studio	Click the  toolbar button until the spot turns dark green before starting Automator. The Log button with the red dot turns off logging.	Each time Automator starts, it creates a new log file under **Foresight\Systems***system***\ AutomatorLogs**version. Some information is also available on the Log tab at the bottom of Foresight Studio when the workflow is open.	To set logging level, click the arrow next to the log button: 
ForesightService	Use **l** parameter in the command line within FSService.ini.	Same as command line.	Use **L** parameter in the command line within FSService.ini.

Log Contents

You can set logging content with:

- The Automator command line parameter L (see page 9)
- The arrow next to the toolbar log button from within Foresight Studio:



Command line logging content parameters

Use one or more of the options listed below with the -L command parameter. Options are case-insensitive, but we recommend using lower-case for clarity's sake.

A message will only be displayed in the log if its particular event type is selected on the command line. -L[1 | 2 | 3] or -L[a | c | x | r | f | i | d]. Example: `-L1` or `-Lacx`.

Command line L option	Event type	Description
a	AUTOMATOR ERROR	An error indicating a problem within Automator itself. Examples: Automator cannot load a component, or runs out of memory during execution.
c	COMPONENT ERROR	An error indicating a problem within a component itself. Example: Attempting to sue a directory that doesn't exist.
x	COMPONENT EXECUTION	A message indicating that a task within a component was executed.
r	COMPONENT EXECUTION RESULT	A message showing the success or failure of task execution within a component.
f	FILE MOVEMENT	A message indicating that a file was moved by Automator.
i	INFORMATIONAL	A status message issued by Automator upon startup, workflow loading and startup, and shutdown.
d	DEBUGGING	A message useful to support and development staff detailing the internal workings of Automator.
1	(minimal logging)	Equivalent to options **aci**. Included for backwards compatibility.
2	(normal logging)	Equivalent to options **acxri**. Included for backwards compatibility.
3	(maximum logging)	Equivalent to options **acxrfid**. Included for backwards compatibility.

The top of the log shows the options being used, and each entry shows its event type.

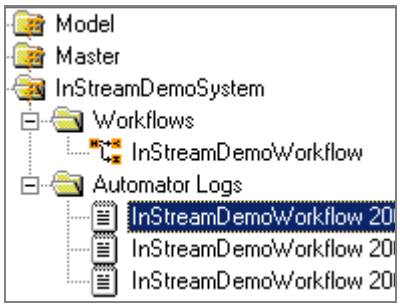
```
***************************************************************
***                Foresight    A U T O M A T O R
***
***************************************************************

8.2.0 x64
(C) 2004-2015 Tibco Inc. All Rights Reserved.

Start of log: C:\TIBCO\Systems\TIDemo\AutomatorLogs\1.0\c47ed9ce-
dffa-42d8-beaf-650403d98975 2012-10-24 1420.txt

Logging Event Mask: AUTOMATOR ERROR, COMPONENT ERROR, COMPONENT
EXECUTION, COMPONENT EXECUTION RESULT, FILE MOVEMENT,
INFORMATIONAL, DEBUGGING

          Time: Wed Oct 24 14:20:16 2012

  >0< Log Sequence Number -- >8876< Current Thread

---------------------------------------------------------------
ForesightAutomator run started at 14:20:16 10/24/12.
Message Event: INFORMATIONAL
---------------------------------------------------------------
```

Deleting Unwanted Logs

Each time Automator starts, it creates a new log file. It continuously records its activities in the log, regardless of whether files are going through the workflow at the moment. Since logs get large, check the log directory frequently and delete unwanted logs.

From within Foresight Studio

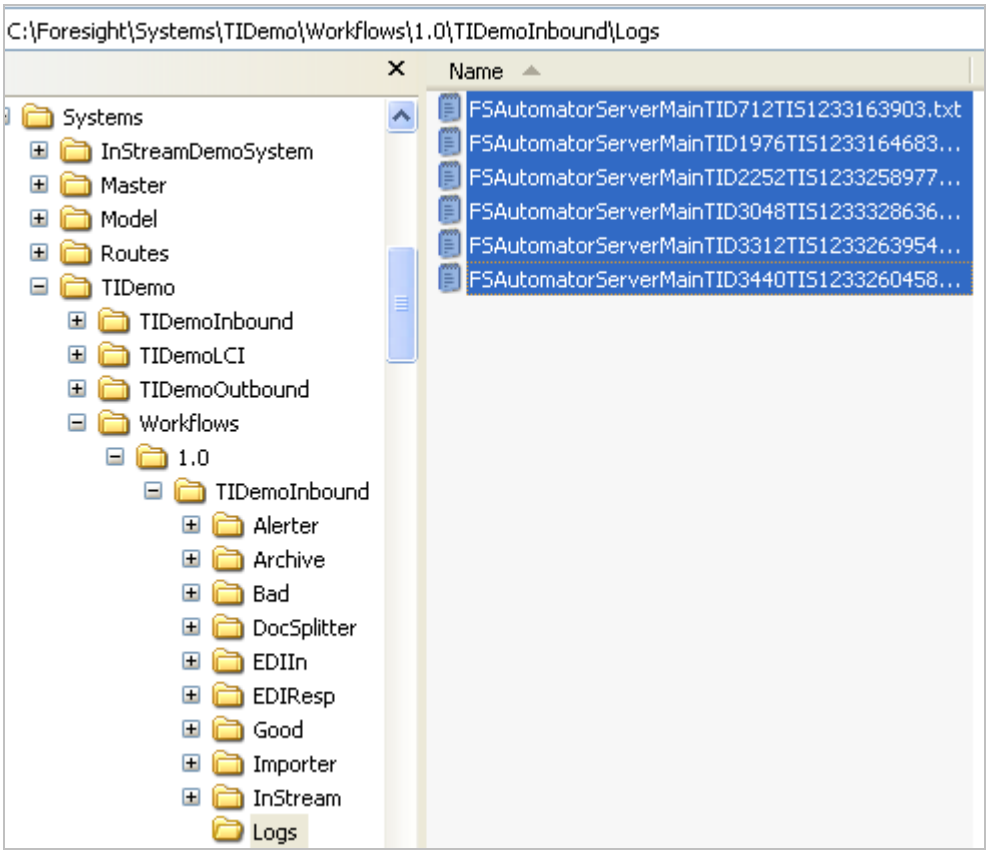
You can delete a log file by clicking on its name and pressing the **Delete** key:



The **Clear Log** toolbar button in Foresight Studio clears the information on the Automator Log tab at the bottom of Foresight Studio. It does not delete log files.

From Windows Explorer

You can find the files and delete them:



Using a Log File

Within a log file, you can search for workflow component names, including directories. Search for "Schedule" to find your workflow components. An opening and closing arrow surrounds each set of information:

```
-->
One activity
Results
<--
```

Components, or "schedules," will have information like this.

At this time …

… the timer triggered

… to check the InboundEDI directory

… and move all files with extension "edi" to the Process directory

***RC:1** means failure*
*See **Return Codes, etc.**, below.*

Details about the error

More details about the error

At this time …

… this component ran

… and validated a file (that had arrived previously – not from the failed copy attempt above)

… successfully
*See **Return Codes, etc.**, below.*

```
Thu Mar 24 08:27:37 2008
--> Current Thread:   1640
Timer trigger at: Thu Mar 24 08:27:37 2008
<--
Thu Mar 24 08:27:37 2008
--> Current Thread:   1640
Schedule: InboundEDI
Running: cmd /c "move /Y
"C:\Foresight\Systems\Tutorial\InboundEDI\*.edi"
"C:\Foresight\Systems\Tutorial\ForesightAutomator\
7.2.0\TutorialWorkflow837I\InboundEDI\Process\""
 GLE:18 RC:1 CP:1
 StdOut:
 StdError:
A duplicate file name exists, or the file cannot be
found.
<--
Thu Mar 24 08:27:37 2008
--> Current Thread:   1640
Possible failure in Schedule InboundEDI. Expected
a return code of 0, but got 1.  CP result: 1
<--
Thu Mar 24 08:27:40 2008
--> Current Thread:   1916
Schedule: InStream_1
Running: C:\Foresight\HVInStream\4.4.1\Bin\HVInStre
am.exe -i"C:\Foresight\Systems\Tutorial\ForesightAuto
mator\7.2.0\TutorialWorkflow837I\InStream_1\Process\
Tutorial837IA.edi" -o"C:\Foresight\Systems\Tutorial\
ForesightAutomator\7.2.0\TutorialWorkflow837I\InStrea
m_1\Process\Tutorial837IA" -gPDSA837I -s"C:\Foresight\
Systems\Tutorial\ValidatorProfiles\Usual.apf"  -r -a
 GLE:2 RC:100 CP:1
```

Checking inbound directory

Running validation

Return Codes, etc.

Each schedule will include a line that summarizes the success of the activity just above it. In the example above, this is:

```
GLE:18 RC:1 CP:1

StdOut:

StdError: A duplicate file name exists, or the file cannot be
found.
```

Where:

GLE — **G**et **L**ast **E**rror (the last error code returned). This value is not cleared for each schedule. These are Windows error codes. Seeing a value for GLE doesn't always mean there was an error, but if there was a problem at the Windows level, this value may help you figure out the problem. See http://msdn.microsoft.com/en-us/library/ms681382(v=VS.85).aspx for details.

Some common ones:

0 Success
2 File not found
3 Path not found
5 Access is denied

RC — **R**eturn Code.

For directories, 0 = success.

For Foresight programs, 100 = success. The manual for each program lists its other return codes.

CP — **C**reate **P**rocess. 1=success.

StdOut — Console output.

StdError — Details about the error.

Problem Report

You can get a problem report like this each time a component does not return the expected return code:

```
*********************************************************
  Foresight Automator Problem Report
*********************************************************
Schedule: InStream
File: 837file.edi
Return Code: 110
Expected Return Code: 100
Create Process Flag: 1
Time: Wed Jul 05 15:11:58 2008
```

To get a problem report for a component:

1. Edit the component's XML file in the workflow.

2. Add a property called **ProblemReportDir** like this:

   ```
   <Property>
      <Id>ProblemReportDir</Id>
      <Value>directory</Value>
   </Property>
   ```

   directory is the directory path that is to receive the problem report.

   Example:

   ```
   <Value>c:\reports</Value>
   ```

3. Repeat for each component that is to generate a problem report. The problem report filename will have this format:

   ProblemReport_*component*_*EDIfilename*_*UniqueNumber*.**txt**

   Example: ProblemReport_InStream_837file.edi_1152126718.txt

Example problem report contents:

This section of an XML file will generate a report in c:\reports whenever the component does not return a success code:

```
<Property>
   <Id>ProblemReportDir</Id>
   <Value>c:\reports</Value>
</Property>
<Task>
   <Id>HVInStream</Id>
   <Type>Executable</Type>
   <SuccessCode>100</SuccessCode>
```

3 Appendix A – PGN and Track File Generation

The Foresight Archive and Retrieval System and Foresight Operational Monitor components rely on PGN and/or Track files for critical information.

IF you run Automator with…	PGN files generated?	Track files generated?
no added parameters	No	Yes
the **-o** parameter	Yes	Yes
the **-n** parameter	No	No
If you are using…	**These files must be generated**	**Use this parameter in the Automator command line**
Foresight Operational Monitor only	PGN	-o
Foresight Archive and Retrieval System only	Track	None needed
Foresight Operational Monitor and Foresight Archive and Retrieval System	PGN and Track	-o
Neither Foresight Operational Monitor nor Foresight Archive and Retrieval System	None	-n

Please see **TRK_Files.pdf** for specifics on which versions of Automator create Track files under which circumstances.

4 Appendix B – Return Codes and Troubleshooting

Return Code	Meaning
None	

Troubleshooting information	Notes
From Foresight Studio	Log has the workflow name and is under the system. Example: `C:\Foresight\Systems\InStreamDemoSystem\AutomatorLogs\1.0`
From Command Line or ForesightService	As specified by -l command line parameter. -l without a path puts it under the workflow. Example: `C:\Foresight\Systems\InStreamDemoSystem\Workflows\1.0\InStreamDemoWorkflow\Logs`

Seeing Return Codes

To display return codes when you run a script, put this line similar to this in the script right after running the program:

UNIX echo "return code = " $?

Windows @echo [Return Code = %ERRORLEVEL%]

This returns something like: `[Return Code=100]`

Virus Checking and Foresight Products

Exclude all TIBCO Foresight workflow subdirectories from virus checking.