

TIBCO Foresight[®] Transaction Insight[®]

Filters for Foresight Portal Applications

*Software Release 5.2
September 2017*

Two-second advantage[®]



Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HERE OF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, Two-Second Advantage, TIBCO Foresight Archive and Retrieval System, TIBCO Foresight EDISIM, TIBCO Foresight Instream, TIBCO Foresight Studio, TIBCO Foresight Test Asset Management Suite, and TIBCO Foresight Transaction Insight are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, Java EE, J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

The United States Postal Service holds the copyright in the USPS City State Zip Codes. (c) United States Postal Service 2017.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

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 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.

Copyright© 2010-2017 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential 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.)

1 Transaction Insight Filters

Skills Needed

This is an advanced task requiring the following skills:

- Creating guidelines
- Validating with TIBCO Foresight® Instream®
- Using databases
- Using and Administering TIBCO Foresight® Transaction Insight®.

Consider asking TIBCO Foresight Technical Support about having TIBCO Foresight create your filters.

Transaction Filter Overview

Note Please be sure of your needs before setting up transaction filters. They are extremely hard to delete. Contact TIBCO Foresight Technical Support for assistance.

A transaction filter is a set of values for a specific transaction set or message. It lets the user select certain documents for inclusion on statistics and search pages. If a document contains a certain value or values at a certain location, the document is included.

You must validate with a guideline that creates filter information and import the validation detail results file into Transaction Insight to make these filters work.

Transaction filters appear under the Partner Rankings, Category Compare, and Search once you have identified a particular standard and transaction set or message.

Examples

Search Transmissions page:

Transaction Filters

Transaction Filter =

- All
- SCAC code
- TCTR code

Partner Filters

Documents page:

	Partner Rankings	Compare By
	HIPAA/X12 5010 820 Inbound	Choose Category
	X12 4010 214 Inbound	Choose Category
	X12 5050 850 Inbound	Choose Category State Relationship management LB favorites Carrier Code
Download To Excel		

Partner Rankings page:

Transaction Filters (214)

Carrier Code

- All
- All
- SCAC code
- TCTR code

The example filters shown above provide a list of choices. You can also set up a transaction filter that lets the user type a value:

Transaction Filters (850)

Total_Price

Major Steps in Setting up Transaction Filters

Warning: A validation detail file with a ZT record will not import unless there is already a corresponding transaction filter defined in Common Administration. You must complete these steps before sending data with the custom ZT record to Transaction Insight.

Plan your filters ahead of time. It is important to set the filters up before you start using Transaction Insight on real data. Transaction filters only include data that comes in after:

- The filter is set up in Transaction Insight
- And you validate using the guideline that contains rules that create a ZT record in the validation detail results (DTL) file.

Steps

1. Adding Custom Records to the Guideline

Prepare a guideline that will create the ZT record. See page 3.

2. Setting up the Transaction Filter in Transaction Insight

In Common Administration, add the transaction set and filter. See page 9.

3. Testing the Transaction Filter

Test the transaction filter with some data. See page 14.

4. Updating the Transaction Filter

Update the transaction filter if necessary. See page 15.

1. Adding Custom Records to the Guideline

You will need an experienced TIBCO Foresight® EDISIM® Standards Editor user to write business rules that capture the values to be used for filtering. The values, which can be up to 100 characters long, go into a custom record that starts with ZT followed by two digits.

For overview information about creating custom records, see **BusinessRules.pdf**.

Custom Record Names

Considerations when naming the custom record:

- When you create the custom record in EDISIM®, have its name start with ZT followed by exactly two digits. The two digits are called the *index*.
- The index must be unique within the Standard-Transaction Set pair.

For example, you can define only one record ZT01 in an X12 820 guideline; it will apply to all versions of the X12 820. You can use the same ZT01 record name in an X12 837P guideline and it will apply to all versions of 837P. You can also use the same ZT01 in an EDIFACT guideline.

This filter will apply to all X12 820 transactions. You cannot have another filter listed here with the same index.

PARTNER FILTERS	TRANSACTION FILTERS	GROUP FILTERS
Standard: X12	Transaction Set: 820	
Index	Filter Name	Description
1	Credit/Debit Flag	BPR-03

Best Practice: It is best to group search-only filters with search-only filters and summarizable filters with summarizable filters. In the above example, because ZT01 is used for both the 820 and the 837P, best practice would be that they **both** be marked/not marked as “Is Search Only.”

- To see if an index has been used, look in the database’s ZRecord table. In this example, only index 2 and 1 have been used:

ZRecordID	ZRecordIndex	TransactionType	Name	Description	HasValues
1	2	214	Carrier Code	B10-03 Alpha Carrier Code	1
2	1	820	Credit-Debit	BPR-02	1

Please note: If a Version column exists in this table, it is ignored because the Z record index applies to all versions.

- During validation, Instream® adds another Z to the beginning of the name, so record ZT01 becomes ZZT01 in Instream’s detail results file, like this:

GEN	311213	1	0	start	d
SVALU	3	STST	1	ST*	820*0
ZZT01	4C				
STRUS	6	1000A	0	1	325

- These indexes are used for special purposes:

Index	Used for ...	Created by ...
98	TIBCO Foresight® Test Asset Management Suite (TAMs) only Submitter Identifier	The schema is created automatically. Validate with a guideline that creates ZZT98. To get this guideline, you can merge your company guideline with the corresponding FS-FILTERS guideline in Instream’s CommunityManagerGuidelines directory.
99	TAMs only Test Environment	The schema is created automatically. Validate with a guideline that creates ZZT99 records. To get this guideline, you can merge your company guideline with the corresponding FS-FILTERS guideline in Instream’s CommunityManagerGuidelines directory.

Example – HIPAA guideline

This example creates a ZZT01 record that captures the Credit-Debit Flag Code (BPR-03) in an 820.

820 Payment Order/Remittance	
Table 1	
0100 : ST	820 Header
0200 : BPR	Financial Information
01 : 305	Transaction Handling Code
02 : 782	Monetary Amount
03 : 478	Credit/Debit Flag Code
04 : 591	Payment Method Code

ISA*00*	*
GS*RA*901234572000	
ST*820*0131*005010	
BPR*C*100.00*	ACH
TRN*1*392039493020	

The first rule is on the ST segment. It defines a custom record ZT01 that will contain the contents of variable BPR03var. In this example, the data will have a length of 1.

What Rule to Run		
DefineCustomRec	<input checked="" type="checkbox"/>	<input type="checkbox"/> Look-Ahead Rule
Text	Parameter Name	Parameter Value
	ID	ZT01
	Flag	M
	VarInfo	BPR03var/1

The next rule is on the BPR-03 element. It captures the value in that element and places it in the variable **BPR03var**.

What Rule to Run		
SetVar	<input checked="" type="checkbox"/>	<input type="checkbox"/> Look-Ahead Rule
Text	Parameter Name	Parameter Value
	VarToAssign	BPR03var
	Value	Current_Element

The third rule, also on the BPR-03, writes out record ZZT01 during Instream validation.

What Rule to Run		
OutputCustomRec	<input checked="" type="checkbox"/>	<input type="checkbox"/> Look-Ahead Rule
Text	Parameter Name	Parameter Value
	ID	ZT01

Be sure the guideline conforms to the requirements outlined in **Data_Types_and_TI.pdf**.

Use Instream to validate with it and look for the ZZT01 record in the output. Note the additional Z added by Instream.

GEN	311213 1 0start d
SVALU	3 STST 1 ST*820*0
ZZT01	4C
STRUS	6 1000A 0 1 325

Where:

- ZZT01 The record that populates the filter
- 4 The line number
- C The value in the BPR-03

Example – EDIFACT guideline

This example creates a ZZT01 record that captures the item identifier (LIN-03-01) in an EDIFACT ORDERS message.

We can use the same record index as in the HIPAA example above since this is an ORDERS message and the other one was an 820.

1100 : Group 29: LIN-PIA-IMD-M...	Line Item
1110 : LIN	Line Item
01 : 1082	Line item identifier
02 : 1229	Action code
03 : C212	Item Number Identification
03.01 : 7140	Item identifier
03.02 : 7143	Item type identification code
03.03 : 1131	Code list identification code

```
DTM+2:20070919:102'
NAD+BS+1111111111111111::9
NAD+COP+TC11111111:ZZZ'
LIN+1+1+TCIN12345:SA'
IMD+P++:::120S SERVER'
```

The first rule is on the UNH segment. It defines a custom record ZT01 that will contain the contents of variable **ItemIDvar**. In this example, the data will have a length of 20, even though EDIFACT allows up to 35 characters there. Assume that our company only allows up to 20 characters for this element in incoming data.

What Rule to Run

DefineCustomRec Look-Ahead Rule

Text	Parameter Name	Parameter Value
	ID	ZT01
	Flag	M
	VarInfo	ItemIDvar/20

The next two rules are on the LIN-03-01 element. The first one captures the value in that element and places it in the variable **ItemIDvar** and the second one writes out the ZZT01 record into the DTL file.

What Rule to Run

SetVar Look-Ahead Rule

Text	Parameter Name	Parameter Value
	VarToAssign	ItemIDvar
	Value	Current_Element

What Rule to Run

OutputCustomRec Look-Ahead Rule

Text	Parameter Name	Parameter Value
	ID	ZT01

Next, be sure the guideline conforms to the EDIFACT requirements outlined in **Data_Types_and_TI.pdf**.

Use Instream to validate with it and look for the ZZT01 record in the output. Note the additional Z added by Instream.

```

ESEG          7NAD+S+TC111
STRUE         8|NAD|0|2|28
STRUS         8|LIN|0|1|28
ZZT01         8TCIN12345
STRUE         11|LIN|0|1|34
  
```

Where:

ZZT01 The record that populates the filter
 8 The line number
 CINT12345 The value in the LIN-03-01

Example – X12 guideline

This example creates a ZZT01 record that captures the total price per PO1 loop. This value is calculated by multiplying the Quantity * Unit Price.

PO1 (Loop)	Item Data
S 0100 : PO1	Baseline Item Data
E 01 : 350	Assigned Identification
E 02 : 380	Quantity
E 03 : 355	Unit or Basis for Measurement Code
E 04 : 212	Unit Price
E 05 : 639	Basis of Unit Price Code

PO1*3ALDKF*.500*HB*12
 3.90*LE*AB*CUST12345*
 BL*ASSEMBLY*ON*BRAND
 OR
 LABEL*AB*CUST12345*BL

The first rule is on the ST segment. It defines a custom record ZT01 that will contain the contents of variable **TotPriceVar**. In this example, the data will have a length of up to 15.

What Rule to Run

DefineCustomRec Look-Ahead Rule

Text	Parameter Name	Parameter Value
	ID	ZT01
	Flag	M
	VarInfo	TotPriceVar/15

The next rule, on the PO1-02 element, captures the quantity into variable **QtyVar**.

What Rule to Run

SetVar Look-Ahead Rule

Text	Parameter Name	Parameter Value
	VarToAssign	QtyVar
	Value	Current_Element

The other rules, on the PO1-04 element, capture the unit price, multiply it by the quantity, and output the result in custom record ZZT01.

What Rule to Run

SetVar Look-Ahead Rule

Text	Parameter Name	Parameter Value
	VarToAssign	UnitPriceVar
	Value	Current_Element

What Rule to Run

Numbers Look-Ahead Rule

Text	Parameter Name	Parameter Value
	ValueA	QtyVar
	Operator	*
	ValueB	UnitPriceVar
	ResultVar	TotPriceVar

What Rule to Run

OutputCustomRec Look-Ahead Rule

Text	Parameter Name	Parameter Value
	ID	ZT01

Be sure the guideline conforms to the requirements outlined in **Data_Types_and_TI.pdf**.

Use Instream to validate with it and look for the ZZT01 record in the output. Note the additional Z added by Instream.

STRUS	85	PO1	0	1	5981
ZZT01	8561.95000				
STRUS	91	CTP	0	1	6794

Where:

ZZT01 The record that populates the filter

8 The line number

561.95000 The calculated value. In the guideline, you could use additional rules to reformat the number

Merging and Using the Custom-Record Guideline

Review **Data_Types_and_TI.pdf** to be sure you meet the requirements for importing that data type.

If you are adding the filter to a HIPAA guideline, merge your guideline with the corresponding GuidelinePlus (listed in **ForesightHIPAAguidelinelist.pdf**).

Otherwise, merge with a PDSAGM guideline or another one that has DSR marks for the ISA, GS, and ST segments, or add DSR marks to the guideline yourself. See **Creating Guidelines for Instream** in **TIB_fsp_edisim_n.n_fseditor.pdf** in EDISIM's documentation.

For details about merging, see **GuideMerge.pdf**.

Copy the guideline to Instream's Database directory and use it for validating data being imported into Transaction Insight. This will give Transaction Insight access to the value in the ZT record.

2. Setting up the Transaction Filter in Transaction Insight

The Transaction Insight portal's administrator will need the following information from the guideline developer:

- The name of the custom record without the extra Z in front (in our example, this is ZT01).
- The type, version, and document type.

Examples:

X12 5010 820

HIPAA/X12 5010 837P

UN/EDIFACT D93A ORDERS

- If the transaction filter is to contain a drop-down list: The possible values for that element, and the description that corresponds to each value.

Adding the Transaction Set

Go to TI's **Admin | Settings | Transaction Sets** page and add a row for the transaction or message, if it isn't already there:

Standard	Version	Transaction Type	Message Version	Message Release
UN/EDIFACT	D.11A	FINPAY	D	11A
UN/EDIFACT	D.11A	INVOIC	D	11A
UN/EDIFACT	D.11A	ORDERS	D	11A
UN/EDIFACT	D.11A	PROTAP	D	11A
UN/EDIFACT	D.11A	WKGRRE	D	11A
HIPAA/X12	5052	837	005	052

See the Transaction Sets section in **TIB_transactioninsight_n.n_commonadmin.pdf** for details about these columns.

Here is our HIPAA 5010 820 from page 5:

Standard	Version	Transaction Type	Message Version	Message Release	Message Industry
HIPAA/X12	5010	820	005	010	X218

Our EDIFACT D.93A ORDERS from page 6:

Standard	Version	Transaction Type	Message Version	Message Release	Message Industry
UN/EDIFACT	D.93A	ORDERS	D	93A	

Our X12 5050 850 from page 7:

Standard	Version	Transaction Type	Message Version	Message Release	Message Industry
X12	5010	850	005	010	

Creating the Transaction Filter

The transaction set or message should now be available on the **Transaction Filters** tab in **Common Administration**.

Log in to Common Administration and:

1. Choose **Admin | Filters | Transaction Filters**.
2. Choose the standard and transaction set or message:

Standard:	UN/EDIFACT	Transaction Set:	ORDERS		
Index	Filter Name	Description	Has Pre-defined Values	Is Search Only	
1	Item_ID	LIN-03-01	No	No	

You will see existing filters for this type of transaction.

3. Click **Add new row**.

- Fill out the line:

Index

The numeric part of the custom record (Example: Choose 1 for ZT01; choose 12 for ZT12).

Filter Name

The label for the filter.

Description

For your own use.

Has Pre-defined Values

Select if the element has specific values – such as an element with code values. This creates a drop-down list that will appear in the filter. If not selected, the filter gets a text field where the user can type a value.

Is Search Only

Determines whether the data in this custom record is to be Search Only:

- selected This transaction filter will only be used on a Search page, and it will **not** be summarized for use on the Statistics pages. This saves database space and improves performance of other transaction filters on the Statistics pages.
- not selected This transaction filter is not limited to just the Search page. It will be used on the Search page **and** on the Statistics pages. Transaction Insight will store additional statistical data for this transaction filter in the database. The TIUtilities -SD parameter determines when the filters’ statistical information will be updated. Please see **TIUtilities.pdf**.

For example, here is one way to set up our EDIFACT LIN-03-01 filter that we set up on page 6. It does not have a drop-down containing specific values.

Index	Filter Name	Description	Has Pre-defined Values	Is Search Only
1	Item_ID	LIN-03-01	<input type="checkbox"/>	<input type="checkbox"/>

Edit

For existing filters. Click to change items in this row or to change values.

Delete

Immediately deletes the filter.

Here is one way to set up our HIPAA 820 BPR-03 filter that we started on page 5. It has only two code values.

Index	Filter Name	Description	Has Pre-defined Values	Is Search Only
1	Credit/Debit Flag	BPR-03	<input checked="" type="checkbox"/>	<input type="checkbox"/>

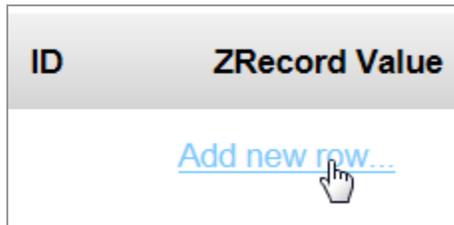
- Click **Save**.

If the save fails with a message “Sorry, new zrecord with index '*nn*' could not be created,” be sure that the database administrator has run the script for a custom record with that index.

Pre-Defined Values

If a filter has specific values that are to appear in a drop-down list, then you need to specify what they are. After choosing **Has Pre-defined values** and saving, add the values for the filter's drop-down list:

1. Click **Add New Row** at the bottom.



2. Fill out the row:

ZRecord Value

A value that could be in the EDI and be captured in the custom record.

Name

A label that will appear for this value in the transaction filter.

IsActive

Select so that the name will appear in the filter. This option lets you enter all possible values at once and then leave some inactive until a later date

Order

A digit indicating the order in which this value should be listed in the drop-down list.

Example for our Credit/Debit Flag filter:

ZRecord Value	Name	IsActive	Order
C	Credit	Yes	1

3. Click **Save**.
4. Continue adding rows until you have listed all values that you want to appear in your filter. Save after each row.

This example shows the Credit-Debit filter values for the custom record created on page 1. It will have a drop-down list containing the words **Credit** and **Debit**.

Index	Filter Name	Description	Has Pre-defined Values	Is Search Only
1	Credit/Debit Flag	BPR-03	Yes	No
Add new row...				
ZRecord Value	Name	IsActive	Order	
C	Credit	Yes	1	
D	Debit	Yes	2	

Checking Web.config

To place transaction filter information at the bottom of the document summary page, update this setting in Web.config:

```
<add key="ShowDocumentSummaryDetails" value="true"/>
```

TRANSACTION FILTER VALUES	
Transaction Filter Name	Transaction Filter Value
Total_Price	61.95000

3. Testing the Transaction Filter

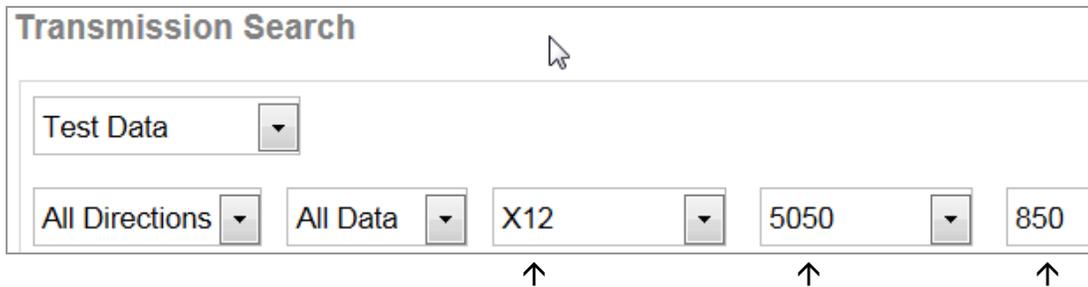
Validate some data with the merged guideline containing the custom record and send the detail file to Transaction Insight.

See if the filter is showing up:

1. Go to Transaction Insight's **Search | Transmissions** page.
2. Select a standard, version, and transaction or message.
3. See if the filter is in the drop-down lists.

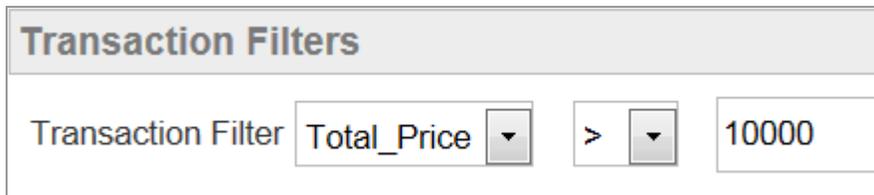
Example X12 filter

To see if our 850 filter is there, select these three fields:



The screenshot shows the 'Transmission Search' interface. It features a dropdown menu for 'Test Data'. Below it are two rows of filters. The first row contains 'All Directions', 'All Data', 'X12', '5050', and '850'. The second row contains 'All Directions', 'All Data', 'X12', '5050', and '850'. Arrows point up from the 'X12', '5050', and '850' filters.

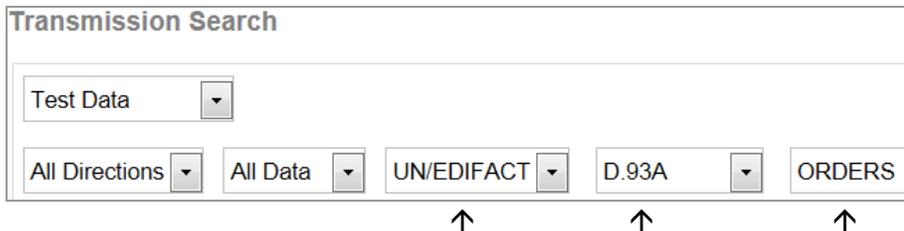
Look for the filter.



The screenshot shows the 'Transaction Filters' interface. It features a dropdown menu for 'Transaction Filter' with 'Total_Price' selected. To the right of the dropdown is a greater-than sign (>) and a text input field containing '10000'.

Example EDIFACT filter

Select these three fields:



The screenshot shows the 'Transmission Search' interface. It features a dropdown menu for 'Test Data'. Below it are two rows of filters. The first row contains 'All Directions', 'All Data', 'UN/EDIFACT', 'D.93A', and 'ORDERS'. The second row contains 'All Directions', 'All Data', 'UN/EDIFACT', 'D.93A', and 'ORDERS'. Arrows point up from the 'UN/EDIFACT', 'D.93A', and 'ORDERS' filters.

Look for the filter.



The screenshot shows the 'Transaction Filters' interface. It features a dropdown menu for 'Transaction Filter' with 'Item_ID' selected. To the right of the dropdown is an equals sign (=) and a text input field containing 'TCIN12345'.

The bottom of the Document Summary page should show the value in the filter:

TRANSACTION FILTER VALUES	
Transaction Filter Name	Transaction Filter Value
Item_ID	TCIN12345

4. Updating the Transaction Filter

To update a transaction filter:

1. Choose **Filters | Transaction Filters**.
2. Select the **Standard** and **Transaction Set**.
3. Click the Edit pencil, make the changes, and save.

Transaction Filters and EDI Versions

When viewing a transaction filter, you do not select an EDI version.

For example, this transaction filter is for 214:

Transaction Filters (214)

Carrier Code

All ▾

Assume that the filter has index 2:

Partner Filters	Transaction Filters	Gro
Standard: X12 ▾ Transaction Set: 214 ▾		
Index	Filter Name	Desc
 2	Carrier Code	B10-

This means any 214 data conveyed to Transaction Insight by a validation detail file record ZT02 will be selected by this filter. It does not matter whether the data is 4010 or 5010 or any other version.

What matters is whether the 214 guideline used for validation generated the ZT01 record.

2 TIBCO Foresight® Archive and Retrieval System Filters

Skills Needed

This is an advanced task requiring these skills:

- Creating guidelines
- Validating with Instream
- Using databases
- Using and Administering Foresight® Archive and Retrieval System

Consider asking TIBCO Foresight Technical Support about having TIBCO Foresight create your filters.

When a file is imported, the **ArchiveTypes** table is automatically updated with a new row, if it does not have an existing row that matches the data. Data type and version are identified by the X12 GS-08 and ST-01 and the EDIFACT UNH-02.

This table should automatically take care of all archived files without any intervention. Please check with TIBCO Foresight Technical Support if you have questions about how this table is operating for you.

Actual envelope values that may be searched are automatically stored in TRANSFILTER* tables.

Universal Type Filters

Universal Type Filters let Foresight Archive and Retrieval System portal users search enveloping for sender and receiver values. These are available for X12 and EDIFACT documents.

Standard Type:	Archive Type - Version:	Start Date:
All <input type="button" value="v"/>	All <input type="button" value="v"/>	10/24/2012
Universal Type Filters		
Sender ID	X12	PRN:ISA06
Receiver ID	X12	PRN:ISA08
Sender Code	X12	PRN:GS02
Receivers Code	X12	PRN:GS03
Interchange Sender	EDIFACT	PRN:UNB2-1
Interchange Recipient	EDIFACT	PRN:UNB3-1
Application Sender	EDIFACT	PRN:UNG2-1
Application Recipient	EDIFACT	PRN:UNG3-1

During installation, the **ArchiverLoadFilterMaps_version.sql** script is run on the Foresight Archive and Retrieval System database to create this set of default filters for all EDI-based detail files.

This script sets up these rows in the **ArchiveFilterMap** table, which defines all Foresight Archive and Retrieval System filters.

ArchiveFilterID	ExternalReference	ArchiveReference	IsDocLevel	Description
1	PRN:ISA06	TRANSFILTER001	0	Sender ID
2	PRN:ISA08	TRANSFILTER002	0	Receiver ID
3	PRN:GS02	TRANSFILTER003	0	Sender Code
4	PRN:GS03	TRANSFILTER004	0	Receivers Code
5	PRN:UNB2-1	TRANSFILTER005	0	Interchange Sender
6	PRN:UNB3-1	TRANSFILTER006	0	Interchange Recipient
7	PRN:UNG2-1	TRANSFILTER007	0	Application Sender
8	PRN:UNG3-1	TRANSFILTER008	0	Application Recipient

When imports a file, it automatically extracts the ISA-06, ISA-08, GS-02, and GS-03 from X12 documents and the UNB-2-1, UNB-3-1, UNG-2-1, and UNG-3-1 from EDIFACT data and stores the values in TRANSFILTER* tables:

FilterID	FileSetID	FilterData	CreateDate	DocInfold	FileInfold
25	37	HILLSDALEMARKET	2013-02-07 08:38:21.960	NULL	107
26	38	Org Agriculture	2013-02-07 09:36:26.877	NULL	110
27	39	HILLSDALEMARKET	2013-02-12 11:00:52.430	NULL	113
28	40	TOPCATSYSTEMS	2013-02-12 14:09:48.807	NULL	115
29	40	1234567890123	2013-02-12 14:09:48.873	NULL	115
30	41	TOPCATSYSTEMS	2013-02-12 15:16:26.287	NULL	117
31	41	1234567890123	2013-02-12 15:16:26.313	NULL	117

The user can type the value into the appropriate universal type filter to find a file:

Universal Type Filters				
Sender ID	X12	PRN:ISA06		HILLSDALEMARKET
Receiver ID	X12	PRN:ISA08		
		↑		↑
		ArchiveFilterMap table		TRANSFILTER### table

Type-Specific Filters

These let Foresight Archive and Retrieval System users search specific document types for files containing a specific value in a specific element in an EDI file.

For example, notice the Type Specific Filter that appears at the bottom of this Filter Search page once the user selects X12 and ANSI 850-005010. This is a custom filter created for 5010 850s.

Standard Type:	Archive Type - Version:	Start Date:
X12	ANSI 850 - 005010	10/24/2012
Universal Type Filters		
Sender ID	X12	PRN:ISA06
Receiver ID	X12	PRN:ISA08
Sender Code	X12	PRN:GS02
Receivers Code	X12	PRN:GS03
Type Specific Filters		
850 Purchase Order Number BEG-03	X12	ZZT50

The following information explains how to set up Type Specific Filters like this.

The example filter that we will create below allows the user to search for 5010 850 files by purchase order number, a value that is stored in the BEG-03.

You have two options for accomplishing this. They have the same result.

Option 1: using a Z-record to identify the data (see page 22):

Type Specific Filters		
850 Purchase Order Number BEG-03	X12	ZZT50
		99AKDF9DAL001

Option 2: using an SVALU record to identify the data (see page 26):

850 Purchase Order Number BEG-03	X12	SVALU:BEGBeginningSegment:3	99AKDF9DAL001
----------------------------------	-----	-----------------------------	---------------

Option 1: Using a Z-Record to create a Filter

1. Create a guideline

- a. Use EDISIM Standards Editor to add Z-record rules to a PDSA guideline or one that is equivalent (see **Data_Types_and_TI.pdf**).

The record can have any ID of up to 4 digits and letters. When Instream writes out its validation detail file, it will add a Z to the front of this ID, and that Z in the first column will allow Foresight Archive and Retrieval System to find it.

Example

We will use ZT50 for our example record ID, since this ID will also work with a Transaction Insight transaction filter.

These rules will put the BEG-03 value into a record called ZZT50 in the validation DTL file. An additional Z is automatically added to the front of the ID during validation so we use ZT50 in our business rule.

ST segment: define the record

What Rule to Run		
DefineCustomRec	<input type="checkbox"/> <i>fx</i>	<input type="checkbox"/> Look-Ahead Rule
<input type="text"/>	Parameter Name	Parameter Value
	ID	ZT50
	Flag	M
	VarInfo	BEG03var/22

BEG-03 element: populate the variable and write out the record

What Rule to Run		
SetVar	<input type="checkbox"/> <i>fx</i>	<input type="checkbox"/> Look-Ahead Rule
<input type="text"/>	Parameter Name	Parameter Value
	VarToAssign	BEG03var
	Value	Current_Element

What Rule to Run		
OutputCustomRec	<input type="checkbox"/> <i>fx</i>	<input type="checkbox"/> Look-Ahead Rule
<input type="text"/>	Parameter Name	Parameter Value
	ID	ZT50

- b. Copy the guideline to Instream's Database directory.

Be sure the validation profile (APF file) used has these settings in the [Detail Record Output] section:

```
IDENT=1
ZREC=1
```

- c. Validate some data with it, and check the validation detail results (DTL file) to be sure the ZZT50 record appears with the PO number:

```

5010_850_19.dtl - Notepad
File Edit Format View Help
GEN      311001 1 0Loaded Transaction S
GEN      311213 1 0start of Transaction
SVALU    3|STST|1|ST*850*600000001*HILL
IDENT    3|I|d7f5f9b9-7ad3-11e2-beab-d9
SVALU    4|BEGBeginningSegment|2|BEG*00
ZZT50    499AKDF9DAL001
SVALU    9|FOBRelatedInstructions|7|FOB
SVALU    10|CTPPricingInformation|8|

```

2. Make the filter tables in the Foresight Archive and Retrieval System database

This is a job for a database administrator.

- a. Edit **ArchiverLoadFilterMaps_n.n.n.sql**, which comes with the installation files.

We are going to make a separate table in which to store the ID of each document that matches your filter, plus the actual value that matched.

Read the comments in the script carefully.

- b. **If using SQL Server**, comment out all **EXECUTE** lines that have been previously used:

```

]BEGIN TRANSACTION
--
-- for all EDI. These are the defaults you can use.
--
--EXECUTE [dbo].[MakeFilter]      'PRN:ISA06',          'TRANS
--EXECUTE [dbo].[MakeFilter]      'PRN:ISA08',          'TRANS
--EXECUTE [dbo].[MakeFilter]      'PRN:GS02',          'TRANS
--EXECUTE [dbo].[MakeFilter]      'PRN:GS03',          'TRANS

--EXECUTE [dbo].[MakeFilter]      'PRN:UNB2-1',        'TRANSFILTER00
--EXECUTE [dbo].[MakeFilter]      'PRN:UNB3-1',        'TRANSFILTER00
--EXECUTE [dbo].[MakeFilter]      'PRN:UNG2-1',        'TRANSFILTER00
--EXECUTE [dbo].[MakeFilter]      'PRN:UNG3-1',        'TRANSFILTER00
--GO

```

If using Oracle, comment out all **call MakeFilter** lines that have been previously used:

```

ALTER SESSION SET CURRENT_SCHEMA=&schema;
--call MakeFilter('PRN:ISA06',          'TRANSFILTER001',  0,
--call MakeFilter('PRN:ISA08',          'TRANSFILTER002',  0,
--call MakeFilter('PRN:GS02',          'TRANSFILTER003',
--call MakeFilter('PRN:GS03',          'TRANSFILTER004',  0,
--call MakeFilter('PRN:UNB2-1',        'TRANSFILTER005',
--call MakeFilter('PRN:UNB3-1',        'TRANSFILTER006',  0,
--call MakeFilter('PRN:UNG2-1',        'TRANSFILTER007',
--call MakeFilter('PRN:UNG3-1',        'TRANSFILTER008',  0,

```

- c. Create a new MakeFilter line like the ZZT50 one below, using the same ZZT ID that will appear in the DTL file. The columns are explained in the script's comments.

If using SQL Server:

```
EXECUTE MakeFilter 'ZZT50','DOCFILTER050',1,'850 Purchase Order Number
BEG-03','005010', 'EDI';
```

If using Oracle:

```
call MakeFilter('ZZT50','DOCFILTER050',1,'850 Purchase Order Number BEG-
03','005010', 'EDI');
```

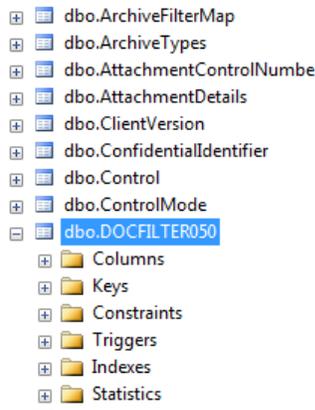
Our example Z record in the DTL file is ZZT50.

We used 1, indicating we are recording all instances of the filter information in a file rather than just the first one.

We decide to start our new table's name with DOCFILTER plus a unique number. However, this table can have the name of your choice.

The last value, EDI, is for X12 documents.

- d. Execute this script against the Foresight Archive and Retrieval System database and check to be sure that it made an empty table with the name that you chose:



You now have a table to hold this filter's information as data comes in.

- e. Check for a new record in the ArchiveFilterMap Table, which defines all Foresight Archive and Retrieval System filters.

In our example:

- Rows 1-8 are the EDI Universal Type Filters
- Row 9 was added with our script

ArchiveFilterID	ExternalReference	ArchiveReference	IsDocLevel	Description	ArchiveTypeID
1	PRN:ISA06	TRANSFILTER001	0	Sender ID	28
2	PRN:ISA08	TRANSFILTER002	0	Receiver ID	28
3	PRN:GS02	TRANSFILTER003	0	Sender Code	28
4	PRN:GS03	TRANSFILTER004	0	Receivers Code	28
5	PRN:UNB2-1	TRANSFILTER005	0	Interchange Sender	64
6	PRN:UNB3-1	TRANSFILTER006	0	Interchange Recipient	64
7	PRN:UNG2-1	TRANSFILTER007	0	Application Sender	64
8	PRN:UNG3-1	TRANSFILTER008	0	Application Recipient	64
9	ZZT50	DOCFILTER050	1	850 Purchase Order Number BEG-03	65

3. Check the Portal

Go to **Archive | Search | Filter Search** and pick the **Standard Type** and **Archive Type – Version**. Confirm that the Type Specific Filter is there.

Archive Search

FILTER SEARCH
FILE SEARCH
SEARCH PREFERENCES

Standard Type:

X12 ▼

Archive Type - Version:

ANSI 850 - 005010 ▼

Start Date:

10/22/2012

Universal Type Filters

Sender ID	X12	PRN:ISA06
Receiver ID	X12	PRN:ISA08
Sender Code	X12	PRN:GS02
Receivers Code	X12	PRN:GS03

Type Specific Filters

850 Purchase Order Number BEG-03	X12	ZZT50
----------------------------------	-----	-------

4. Archive some data that was validated with your Z-record guideline

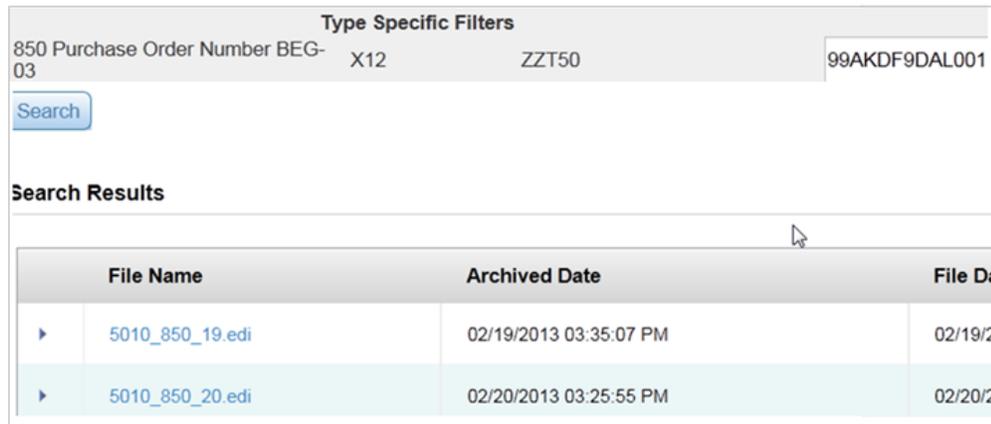
- a. In the file that you test with your filter, note the value that should be in the filter. In our example, this is the BEG03:

```

ISA*00*                *00*
GS*PO*HILLSDALEMARKET*TOPCATM
ST*850*600000001*HILLSDALE000
BEG*00*BK*99AKDF9DAL001*39483
CUR*AC*USA*.2939*SE*USA*IMF*0
REF*AB*3920394930203*GENERAL F
PER*AC*ARTHUR JONES*TE*(614)5
TAX*229438383*IP*02535*IP*362
```

- b. Archive the file.

- c. Try out the filter and see if it finds your file.



Option 2: Using an SVALU Record to create a Filter

1. Create a guideline

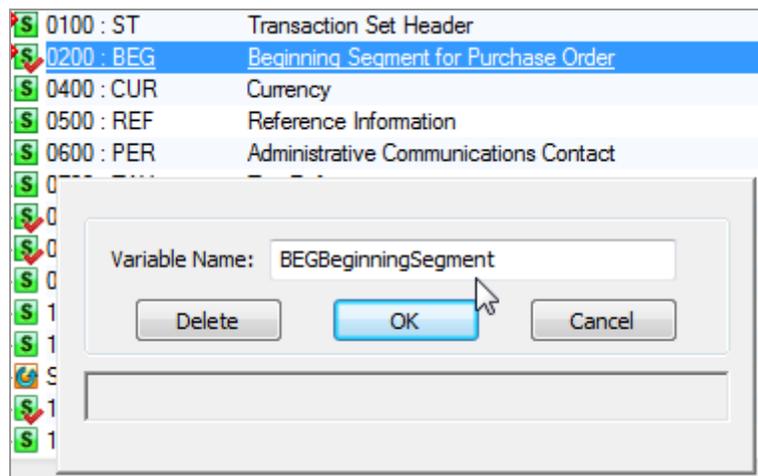
- a. Use EDISIM Standards Editor to add a DSR mark to the segment containing the value that you want to filter. Use a PDSA guideline or one that is equivalent (see **Data_Types_and_TI.pdf**).

This mark will create an SVALU-record in the validation detail results (DTL) file. This record will include the value for each element in the segment.

Example

Assume that we are filtering the BEG-03 in a 5010 850. Right-click on the BEG segment and choose **DSR Mark/Unmark**.

If it has a Variable Name, note it. If not, add one:



- b. Copy the guideline to Instream's **Database** directory.

Be sure the validation profile (APF file) used has these settings in the [Detail Record Output] section:

```
IDENT=1
SVALU=1
```

- c. Validate some data with it, and be sure a SVALU record appears for the segment containing the filter value. It will contain the DSR variable plus the whole segment:

```
SVALU      3 |STST|1|ST*850*600000001*HILLSDALE0001
IDENT      3 |I|a12f90a1-7ac6-11e2-a38a-ef9720ba6cc7|1|
SVALU      4 |BEGBeginningSegment|2|BEG*00*BK*99AKDF9DAL001*3948392019
SVALU      9 |FOBRelatedInstructions|7|FOB*CA*IP*GENERAL PURPOSE*01*CF
SVALU     10 |CTPPricingInformation|8|CTP*AG*ALT*100.00*100*DP:1387084
```

2. Make the Filter

This is a job for a database administrator.

- a. Edit **ArchiverLoadFilterMaps_***version*.**sql**, which comes with the installation files.

We are going to make a separate table in which to store the ID of each document that match your filter, plus the actual value that matched.

Read the comments in the script carefully.

- b. If using SQL Server, comment out all **EXECUTE** lines that have been previously used:

```
BEGIN TRANSACTION
--
-- for all EDI. These are the defaults you can use.
--
--EXECUTE [dbo].[MakeFilter]      'PRN:ISA06',          'TRANS
--EXECUTE [dbo].[MakeFilter]      'PRN:ISA08',          'TRANS
--EXECUTE [dbo].[MakeFilter]      'PRN:GS02',          'TRANS
--EXECUTE [dbo].[MakeFilter]      'PRN:GS03',          'TRANS

--EXECUTE [dbo].[MakeFilter]      'PRN:UNB2-1',        'TRANSFILTER00
--EXECUTE [dbo].[MakeFilter]      'PRN:UNB3-1',        'TRANSFILTER00
--EXECUTE [dbo].[MakeFilter]      'PRN:UNG2-1',        'TRANSFILTER00
--EXECUTE [dbo].[MakeFilter]      'PRN:UNG3-1',        'TRANSFILTER00
--GO
```

If using Oracle, comment out all **call MakeFilter** lines that have been previously used:

```
ALTER SESSION SET CURRENT_SCHEMA="&schema";
--call MakeFilter('PRN:ISA06',      'TRANSFILTER001',  0,
--call MakeFilter('PRN:ISA08',      'TRANSFILTER002',  0,
--call MakeFilter('PRN:GS02',        'TRANSFILTER003',
--call MakeFilter('PRN:GS03',        'TRANSFILTER004',  0,
--call MakeFilter('PRN:UNB2-1',      'TRANSFILTER005',
--call MakeFilter('PRN:UNB3-1',      'TRANSFILTER006',  0,
--call MakeFilter('PRN:UNG2-1',      'TRANSFILTER007',
--call MakeFilter('PRN:UNG3-1',      'TRANSFILTER008',  0,
```

- c. Create a new MakeFilter line containing the DSR mark variable, like this example. The columns are explained in the script's comments.

If using SQL Server:

```
EXECUTE MakeFilter 'SVALU:BEGBeginningSegment:3','DOCFILTER051',1,'850
Purchase Order Number BEG-03','005010','850','EDI';
```

If using Oracle:

```
call MakeFilter('SVALU:BEGBeginningSegment:3','DOCFILTER051',1,'850
Purchase Order Number BEG-03','005010','EDI');
```

Since we used 1, indicating we are recording all instances of the filter information in a file rather than just the first one, we decide to start our new table's name with DOCFILTER plus a unique number. However, this table can have the name of your choice.

Note the **SVALU:BEGBeginningSegment:3**

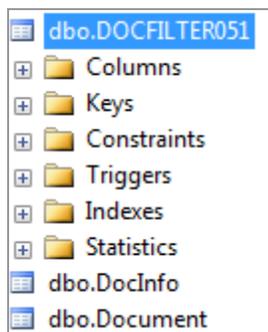
In this example:

- SVALU A literal.
- BEGBeginningSegment The DSR Mark variable for the segment.
- 3 The third element in the segment. If the value had been a composite sub-element, there would be another colon and position number after the 3.

The last value, EDI, is for X12 documents.

The script contains comments that explain how to use it.

- d. Execute this against the Foresight Archive and Retrieval System database and check to be sure that it made an empty table with the name that you chose:



You now have a table to hold this filter's information as data comes in.

- e. Check for a new record in the ArchiveFilterMap table, which defines all Foresight Archive and Retrieval System filters.

In our example:

- Rows 1-8 are the EDI Universal Type Filters
- Row 9 is a previously-added custom filter using a Z-record
- Row 10 is our new custom filter based on an SVALU record

ArchiveFilterID	ExternalReference	ArchiveReference	IsDocLevel	Description	ArchiveTypeID
1	PRN:ISA06	TRANSFILTER001	0	Sender ID	28
2	PRN:ISA08	TRANSFILTER002	0	Receiver ID	28
3	PRN:GS02	TRANSFILTER003	0	Sender Code	28
4	PRN:GS03	TRANSFILTER004	0	Receivers Code	28
5	PRN:UNB2-1	TRANSFILTER005	0	Interchange Sender	64
6	PRN:UNB3-1	TRANSFILTER006	0	Interchange Recipient	64
7	PRN:UNG2-1	TRANSFILTER007	0	Application Sender	64
8	PRN:UNG3-1	TRANSFILTER008	0	Application Recipient	64
9	ZZT50	DOCFILTER050	1	850 Purchase Order Number BEG-03	65
10	SVALU:BEGBeginningSegment:3	DOCFILTER051	1	850 Purchase Order Number BEG-03	65

3. Check the Portal

Go to **Archive | Search | Filter Search** and pick the **Standard Type** and **Archive Type – Version**. Confirm that the Type Specific Filter is there:

Type Specific Filters		
850 Purchase Order Number BEG-03	X12	SVALU:BEGBeginningSegment:3

4. Archive data that was validated with your guideline

- In the file that you test with your filter, note the value that should be in the filter. In our example, this is the BEG03:

```
ISA*00*                *00*
GS*PO*HILLSDALEMARKET*TOPCATM
ST*850*600000001*HILLSDALE000
BEG*00*BK*99AKDF9DAL001*39483
CUR*AC*USA*.2939*SE*USA*IMF*0
REF*AB*3920394930203*GENERAL P
PER*AC*ARTHUR JONES*TE*(614)5
TAX*229438383*IP*02535*IP*362
```

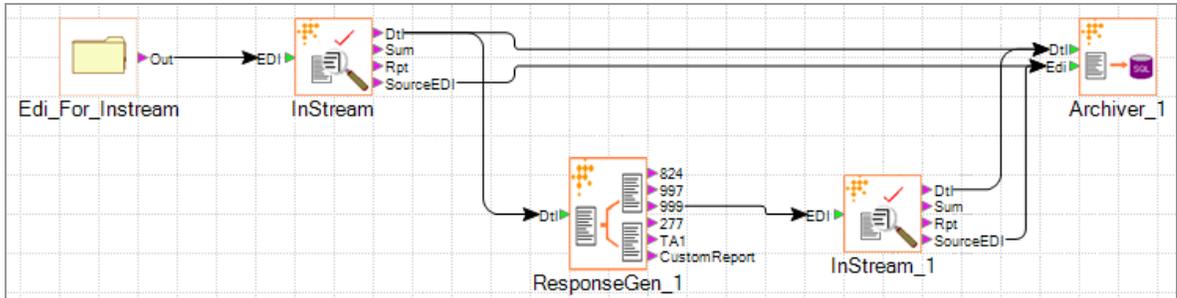
- Archive the file.
- See if the filter finds your file:

Type Specific Filters		
850 Purchase Order Number BEG-03	X12	SVALU:BEGBeginningSegment:3 99AKDF9DAL001
<input type="button" value="Search"/>		
Search Results		
File Name	Archived Date	File De
▶ 5010_850_19.edi	02/19/2013 02:00:31 PM	02/19/2

Foresight Archive and Retrieval System Workflow Considerations

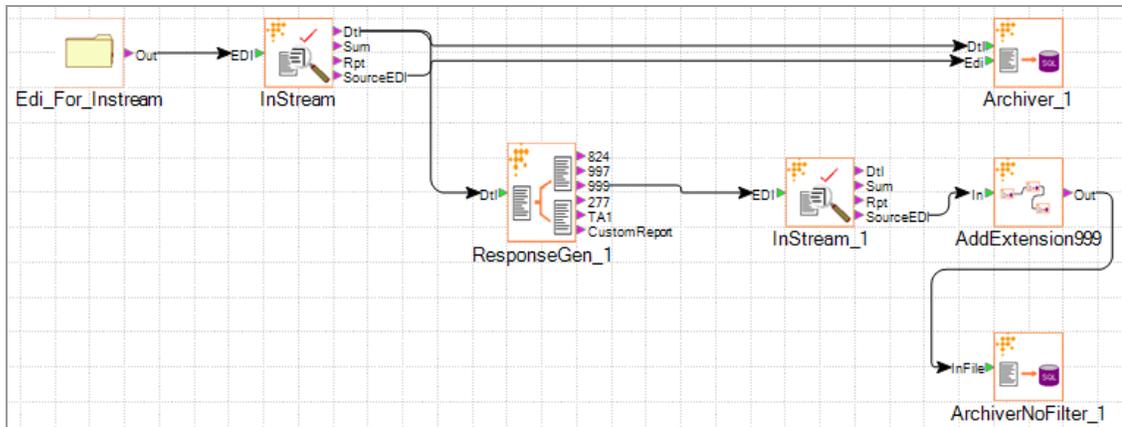
This TIBCO Foresight® Studio workflow archives two EDI files, along with their validation detail results (DTL) files:

1. The originating data – let's say a 5010 850.
2. A 999 created by Response Generator.



Since two different transaction-version combinations are being Archived with the same jobID, you will not be able to find these documents by searching with type-specific filters.

Instead, consider using this similar workflow that allows you to use type-specific filters on the originating data:



How Foresight Archive and Retrieval System Assigns Filters to Documents

When you archive a validation DTL file, Foresight Archive and Retrieval System first finds a row from the ArchiveTypes table and then uses it to get the Z-record and SVALU records from the file. If there is a match, it writes the relevant data from the line to the table defined for this filter.

This includes several steps:

1. It identifies the type of data by looking at records or data in the validation DTL file:

XML File Foresight Archive and Retrieval System uses the standard from the validation DTL file's GEN record 11001 to determine the filter set. This includes:

```
from Standard standardName (XML)
```

Example:

```
GEN          011001 1 0Loaded Message VRI from Standard  
XML_PO_f (XML)
```

When it queries the database for this file, it will look for a record where the transaction set is **XML**, the Data Standard is **XML**, and the DataVersion is **XML_PO_f**.

Flat File For flat files, Foresight Archive and Retrieval System looks in the validation DTL file for a GEN record with the number 11001 that includes the text:

```
from Standard standardName (Date=yyyy/mm/mm hh:mm)
```

Example:

```
GEN          211001 1 0Loaded Transaction Set NAME from  
Standard VETDELIM (Date=2013/01/23 18:48)
```

When it queries the database for this file, it will look for a record where the transaction set is **FLAT**, the Data Standard is **FLAT**, and the DataVersion is **VETDELIM**.

X12 File For X12-based detail files, Foresight Archive and Retrieval System uses the GS-08 and ST-01 fields to clarify the type. It records the ISA-06, ISA-08, GS-02, and GS-03 values. It also records the entire payload of SVALU and Z records.

EDIFACT File For EDIFACT-based detail files, Foresight Archive and Retrieval System uses the UNH-02-1 through UNH-02-3 to clarify the type. It records the UNB-2-1, UNB-3-1, UNG-2-1 and UNG-3-1. It also records the entire payload of SVALU and Z records.

- It then looks in the ArchiveTypes table to find the row with that DataVersion and TransactionSet.

ArchiveTypeID	ArchiveTypeName	DataStandard	DataVersion	TransactionSet	DataStandard
63	ANSI 837D	EDI	005010X22...	837	X12
64	EDIFACT ALL	EDIFACT	ALL	ALL	EDIFACT
65	ANSI 850	EDI	005010	850	X12
66	ANSI INVOIC	EDIFACT	D.96A	INVOIC	EDIFACT

It will have a unique ArchiveTypeID in the first column – in this example, 65.

- It then it retrieves all rows from the ArchiveFilterMap table that match that ArchiveTypeID:

ArchiveFilterID	ExternalReference	ArchiveReference	IsDocLevel	Description	ArchiveTypeID
1	PRN:ISA06	TRANSFILTER001	0	Sender ID	28
2	PRN:ISA08	TRANSFILTER002	0	Receiver ID	28
3	PRN:GS02	TRANSFILTER003	0	Sender Code	28
4	PRN:GS03	TRANSFILTER004	0	Receivers Code	28
5	PRN:UNB2-1	TRANSFILTER005	0	Interchange Sender	64
6	PRN:UNB3-1	TRANSFILTER006	0	Interchange Recipient	64
7	PRN:UNG2-1	TRANSFILTER007	0	Application Sender	64
8	PRN:UNG3-1	TRANSFILTER008	0	Application Recipient	64
9	ZZT50	DOCFILTER050	1	850 Purchase Order Number BEG-03	65
10	SVALU:BEGBeginningSegment:3	DOCFILTER051	1	850 Purchase Order Number BEG-03	65

- It checks the detail file for values from the **ExternalReference** column. If it finds a matching Z-record or SVALU record, it places that line in the table named in the **ArchiveReference** column.
- For each EDI file, Foresight Archive and Retrieval System captures the values in the ISA-06, ISA-08, GS-02, and GS-03, or the UNB-2-1, UNB-3-1, UNG-2-1 and UNG-3-1 so users will have them as search filters. These go in the tables shown in the ArchiveReference column above.

ArchiveTypes Table

Do not change this table manually.

Filtering starts with the table **ArchiveTypes**, which classifies files handled by Archive. It is preloaded with rows that identify certain data types, and additional rows are automatically added when new types of data are archived.

ArchiveTypeID	ArchiveTypeName	DataStandard	DataVersion	TransactionSet	DataStandard
63	ANSI 837D	EDI	005010X22...	837	X12
64	EDIFACT ALL	EDIFACT	ALL	ALL	EDIFACT
65	ANSI 850	EDI	005010	850	X12
66	ANSI INVOIC	EDIFACT	D.96A	INVOIC	EDIFACT

It consists of these columns:

ArchiveTypeID System generated identifier.

ArchiveTypeName Description for this filter, such as ANSI 837P or EDIFACT ORDERS. This is the first part of the Archive Type – Version filter value. Be consistent in your naming pattern so the users can find the filter value easily.

Archive Type - Version:

DataStandard One of these types of data: EDI, XML, FLAT, EDIFACT, or OTHER. EDI means X12 EDI.

DataVersion For X12, this is the GS-08 value.

For EDIFACT, this is UNH-02-02 and UNH-02-03.

For XML and FLAT, this is the standard that appears in the validation detail file (described on page 30). This can also be ALL for all versions.

TransactionSet One of these:

For X12 data, this is the ST-01 value or ALL for all transaction sets.

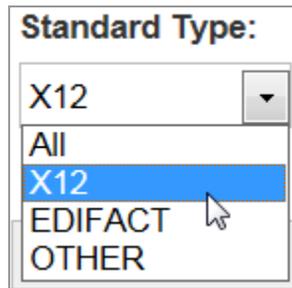
For EDIFACT data, this is the UNH-02-01 or ALL for all messages.

For Flat files, this is FLAT.

For XML files this is XML.

DataStandardDisplayName

The name that displays on the Standard type filter for this standard:



Information from this table populates these drop-downs on the FILTER SEARCH page:

Standard Type: X12 **Archive Type - Version:** ANSI 850 - 005010

ArchiveTypeID	Archive TypeName	DataStandard	DataVersion	TransactionSet	DataStandardDisplayName
63	ANSI 837D	EDI	005010X224A2	837	X12
64	EDIFACT ALL	EDIFACT	ALL	ALL	EDIFACT
65	ANSI 850	EDI	005010	850	X12
66	ANSI INVOIC	EDIFACT	D.96A	INVOIC	EDIFACT

Red arrows indicate the mapping from the table to the filters: 'X12' from the 'DataStandardDisplayName' column of row 65, and 'ANSI 850 - 005010' from the 'Archive TypeName', 'DataStandard', and 'DataVersion' columns of row 65.

The ArchiveTypes table is initialized when the Foresight Archive and Retrieval System database is created. Look at **ArchiverCreateSchema.sql** for the INSERT statements that populate this table.

Each row must be unique. Do not re-run **ArchiverCreateSchema.sql** on an existing database without first commenting these lines out of the script. This would cause duplicates and Foresight Archive and Retrieval System will fail. If this happens, call TIBCO Foresight Technical Support for advice.

ArchiveFilterMap Table

This table is populated with the MakeFilter stored procedure, which is used when you add filters.

This table defines all filters for Archive.

Rows 1 – 8 in this example are the EDI Universal Type Filters.

Row 9 and 10 are custom filters.

ArchiveFilterID	ExternalReference	ArchiveReference	IsDocLevel	Description	ArchiveTypeID
1	PRN:ISA06	TRANSFILTER001	0	Sender ID	28
2	PRN:ISA08	TRANSFILTER002	0	Receiver ID	28
3	PRN:GS02	TRANSFILTER003	0	Sender Code	28
4	PRN:GS03	TRANSFILTER004	0	Receivers Code	28
5	PRN:UNB2-1	TRANSFILTER005	0	Interchange Sender	64
6	PRN:UNB3-1	TRANSFILTER006	0	Interchange Recipient	64
7	PRN:UNG2-1	TRANSFILTER007	0	Application Sender	64
8	PRN:UNG3-1	TRANSFILTER008	0	Application Recipient	64
9	ZZT50	DOCFILTER050	1	850 Purchase Order Nu...	65
10	SVALU:BEGBeginningS...	DOCFILTER051	1	850 Purchase Order Nu...	65

When data is archived, this is how this example ArchiveFilterMap table is used:

1. The Foresight Archive and Retrieval System client looks in the DTL file for any records that start with:
 - ZZT50
 - SVALU # | BEGBeginningSegment
2. When it finds one, it inserts the relevant value into the table identified by the ArchiveReference column.

Example

Using the table above, if the detail file contained this line:

ZZT50 499AKDF9DAL001

... the table DOCFILTER050 would contain the value 99AKDF9DAL001.

If the detail file contained this line:

SVALU 4 | BEGBeginningSegment | 2 | BEG*00*BK*99AKDF9DAL001*394839201938 ...

... the table DOCFILTER051 would contain the value 99AKDF9DAL001, assuming the filter was set up to snag the third element.

A Foresight Archive and Retrieval System portal user could then search for the document that contains 99AKDF9DAL001 for the BEG-03.

ArchiveFilterMap Columns

ArchiveFilterID	System generated identifier.
ExternalReference	References to items in the detail file: Currently these values can be in this column: <ul style="list-style-type: none"> - The envelope values that are automatically saved for all EDI documents: PRN:ISA-06, PRN:ISA-08, PRN:GS-02, PRN:GS-03, PRN:UNB-2-1, PRN:UNB-3-1, PRN:UNG-2-1, and PRN:UNG-3-1. These are the familiar Sender and Receiver fields from X12 and EDIFACT data. - SVALU: <i>svalueID:location</i>. This is an SVALU record from a detail file. Example: SVALU:S009:3:2 - Zxxxxx. A Z record. Example: ZZT28.
ArchiveReference	The table where the data that passes this filter will reside.
IsDocLevel	Determines which instances of the data being filtered will be captured. <ul style="list-style-type: none"> 0 Only the file's first instance of the filtered item is to be stored 1 All instances of the filtered item will be stored (lets the portal user search for any of the instances rather than just the first)
Description	A user-friendly description that will appear on the portal to identify this filter. Keep it short if you don't want it to wrap to a second line when displayed in the portal.
ArchiveTypeId	A reference back to the ArchiveTypes table.

Universal Type Filters			
Sender ID	X12	PRN:ISA06	
Receiver ID	X12	PRN:ISA08	
Sender Code	X12	PRN:GS02	
Receivers Code	X12	PRN:GS03	
Type Specific Filters			
850 Purchase Order Number BEG-03	X12	ZZT50	
850 Purchase Order Number BEG-03	X12	SVALU:BEGBeginningSegment:2	

↑
↑
↑
Description
From ArchiveTypes Table
ExternalReference

Populating the ArchiveFilterMap Table

The **MakeFilter** and **DropFilter** stored procedures in the Foresight Archive and Retrieval System database add and remove rows from the ArchiveFilterMap table.

MakeFilter

As the name implies, this stored procedure is used to add an item to the ArchiveFilterMap table.

Permissions needed to execute MakeFilter:

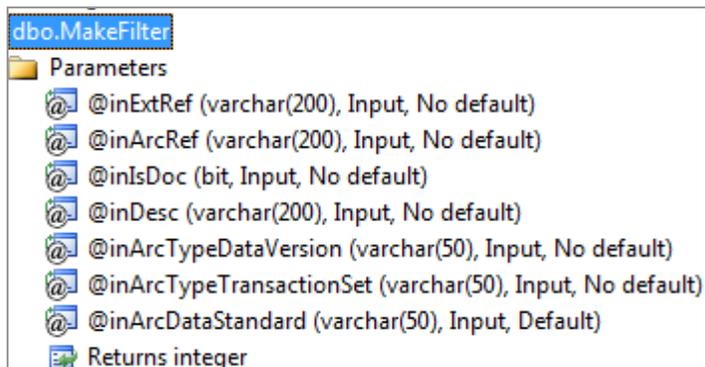
SQL Server

- Run stored procedure
- Insert or Perform DML operations on Foresight Archive and Retrieval System database tables
- Create indexes

Oracle

- Run stored procedure
- Perform DML operations on Foresight Archive and Retrieval System database tables
- Create table
- Create a sequence
- Create an index
- Create trigger

It takes the following parameters:



The parameters are the values for columns in the ArchiveFilterMap table, as described on page 36, and the ArchiveTypes table, as described on page 33:

- | | |
|--------------------------|---|
| 1. ExtRef | Value to go in the ExternalReference column in ArchiveFilterMap |
| 2. ArcRef | Value to go in the ArchiveReference column in ArchiveFilterMap |
| 3. IsDoc | Value to go in the IsDocLevel column in the ArchiveFilterMap table |
| 4. Desc | Value to go in the Description column in the ArchiveFilterMap table |
| 5. ArcTypeDataVersion | Value to go in the DataVersion column in the ArchiveTypes table |
| 6. ArcTypeTransactionSet | Value to go in the TransactionSet column in the ArchiveTypes table |
| 7. ArcDataStandard | Value to go in the DataStandard column in the ArchiveTypes table |

The parameters for this file and examples of how to use it are documented extensively in the file **ArchiverLoadFilterMaps.sql**, which is in Foresight Archive and Retrieval System's DBScripts directory.

SQL Server Examples

```
EXECUTE MakeFilter
'ZZT28',                -- the filter
'DOCFILTER005',        -- the table for this filter
1,                      -- indicates that all instances will be used
'837I Billing Provider Tax ID', -- the description
'004010X096A1',        -- the data version
'837';                 -- the data set

EXECUTE MakeFilter 'ZZT28','DOCFILTER005',1,'837I Billing Provider Tax
ID','004010X096A1','837';

EXECUTE MakeFilter 'ZZT52','DOCFILTER006',1,'837I COBA ID','004010X096A1','837';

EXECUTE MakeFilter 'SVALU:S009:2:1','DOCFILTER007',1,'Subscriber
Claim','004010X096A1','837';
```

Oracle Example

```
CALL <schemaName>.MakeFilter(
'ZZT28',                -- the filter
'DOCFILTER005',        -- the table for this filter
1,                      -- indicates that all instances will be used
'837I Billing Provider Tax ID', -- the description
'004010X096A1',        -- the data version
'837';                 -- the data set
);

call ARCHIVERE2E.MakeFilter('ZZT28','DOCFILTER005',1,'837I Billing Provider
Tax ID','004010X096A1','837');

call ARCHIVERE2E.MakeFilter('ZZT52','DOCFILTER006',1,'837I COBA
ID','004010X096A1','837');

call ARCHIVERE2E.MakeFilter('SVALU:S009:2','DOCFILTER007',1,'Subscriber
Claim','004010X096A1','837');
```

The data version and data set indicate that this filter would be applied to any file that is an EDI 837I 4010. The filter is ZZT28, a custom filter for these files. DOCFILTER005 is the name of the table that will contain all of the filtered data for this filter.

Filters have to correspond to what guidelines put into the validation detail files. In the examples above, these filters will only be useful if some detail files actually contained ZZT28 or ZZT52 records, or SVALU S009 records.

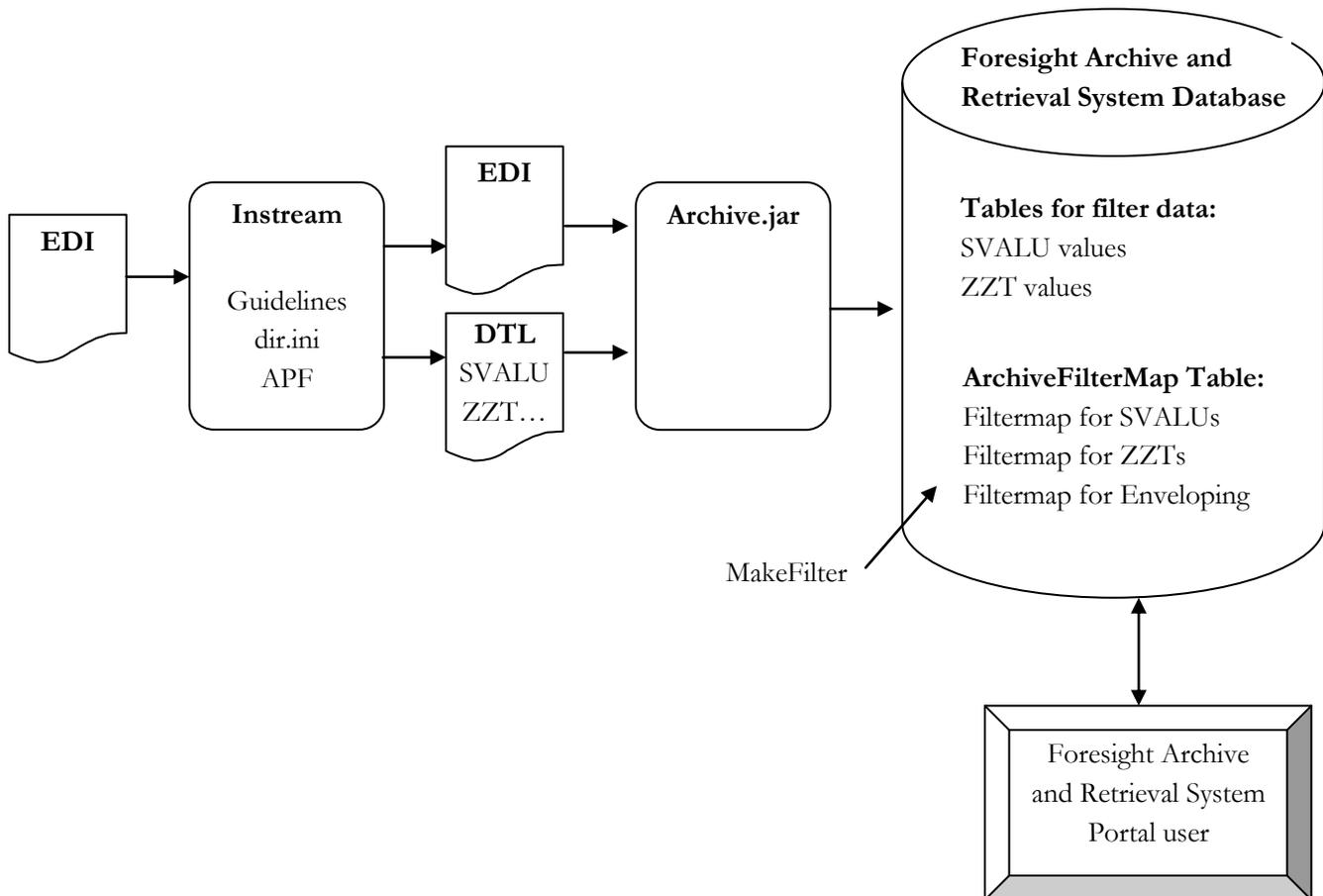
Please feel free to contact TIBCO Foresight Technical Support for guidance.

DropFilter

This stored procedure is used to remove a filter (and its associated table) from the database. It takes one parameter, the **ArchiveFilterId** from the ArchiveFilterMap table.

ArchiveFilterID	ExternalReference
7	PRN:UNG2-1
8	PRN:UNG3-1
9	ZZT50
10	SVALU:BEGBeginningSegment:3

Conceptual Diagram of Foresight Archive and Retrieval System Filters

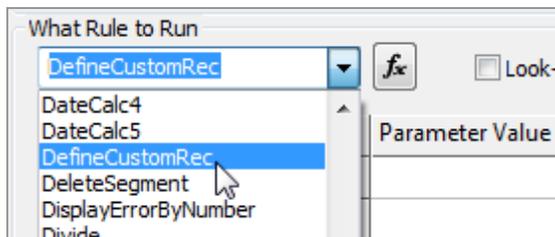


3 How to Create ZT Records

This requires someone familiar with EDISIM Standards Editor.

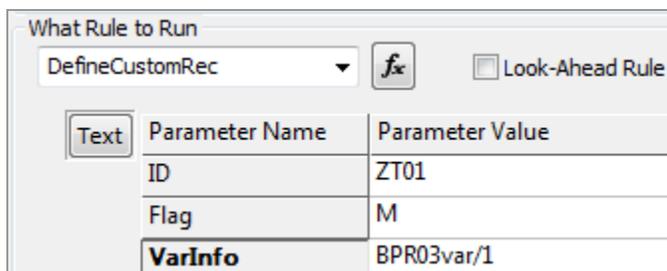
Rules used below are documented in **BusinessRules.pdf**. Also see the example on page 3.

1. From EDISIM Standards Editor, open the guideline that will be used for validation, or a guideline that you will merge into the guideline that will be used for validation.
2. Right-click the ST or UNH record and choose **Business Rules | New ...**
3. Under What Rule to Run, choose **DefineCustomRec**:



4. Click the **Text** button until a grid appears and fill it out.

Example:



Where:

- ZT01 Record name.
- M Written out manually by an OutputCustomRec rule for ZT01.
- BPR03var Variable that holds the value that is the payload of this record.
- /1 Maximum length of the value is 1 character.

5. Capture the value in a variable. This is usually with a SetVar rule but it can be any other rule that puts a value into a variable.

Example:

Text	Parameter Name	Parameter Value
	VarToAssign	BPR03var
	Value	Current_Element

Where:

BPR03var Variable to hold the value

Value Current location of the value

6. Output the record with an OutputCustomRec rule.

4 Filters and New Databases

When you move to a new database, you will need to run the scripts again to put the filters into the new database.

Therefore, you will need to keep a complete set of all filter-related statements you executed against your databases.

Suggestion: Save the TIBCO Foresight-provided `ArchiverLoadFilterMaps_n.n.n.sql` script to a new name and comment out sections after they have executed. Add new statements as needed for new filters. That way, you will have one script containing all of your filters and can uncomment and run all necessary filter statements when you start a new database.