TIBCO Foresight® Instream®

Dataswapper

Software Release 8.7
August 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 HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

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.

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

Introduction

Intended Audience

This document is intended for business analysts who are substituting values in EDI data, and for those who are configuring the TIBCO Foresight® Instream® data flow.

Before using this document, familiarize yourself with Instream® validation, which is described in **TIB_fsp-instream_ <*n.n*>_usersguide.pdf**.

You will also need access to someone familiar with creating business rules with TIBCO Foresight® EDISIM® Standards Editor.

System Requirements

Dataswapper has the same system requirements as Instream and it runs on the same platforms.

Capabilities

Dataswapper is an Instream program that:

- Replaces the value in a specified data element with another value in the EDI.
- Inserts a new segment into the EDI.
- Deletes a segment from the EDI.
- Creates a report of the data that was changed.

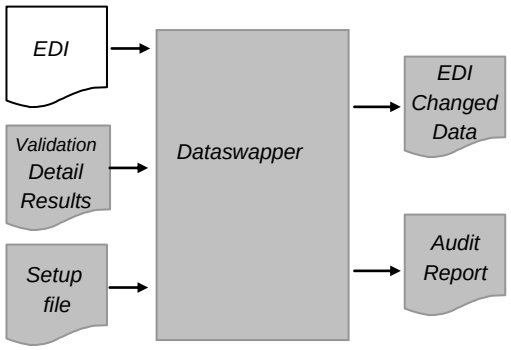
When substituting, in addition to assigning a known value that has already been captured, Dataswapper is smart enough to come back and assign a value that is captured later in the file to a data element that occurred earlier in the file.

Since an Instream variable can be assigned a value based on a database table lookup or a user exit to an external application, Dataswapper can be used in a variety of applications, such as:

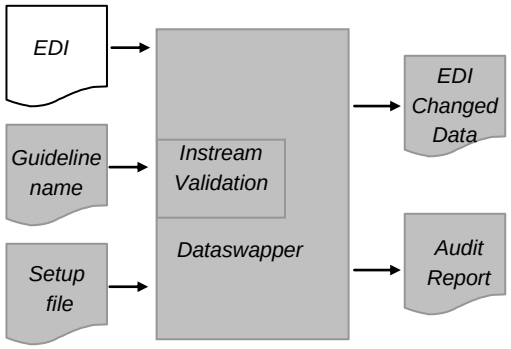
- Substituting a legacy Provider ID with a National Provider ID (NPI) based on a cross-reference table lookup.
- Performing the same substitution, but using an application or a commercial web service to obtain the National Provider ID.
- Assigning a new control number to a split or resubmitted transaction, based on a sequence maintained in a database.
- Automatically correcting frequently occurring coding errors.

Input and Output

This configuration supplies a validation detail results file as one input:



This configuration supplies a guideline name as one of the inputs:



In this scenario, Dataswapper calls Instream to validate the data with the guideline name supplied as input. It then uses the resulting detail file when swapping the data.

For input, Dataswapper uses:

- The EDI data itself, which can be based on any X12 data including a HIPAA addenda.
- The detail results file created by Instream validation. It uses these records to build a splitting pattern for the EDI data.

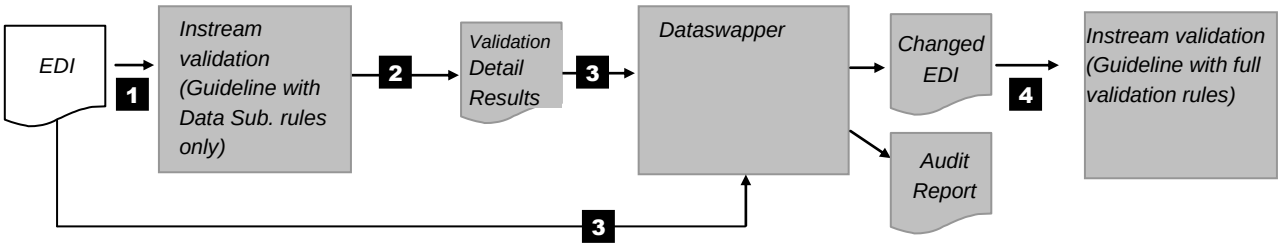
  or

  A guideline containing the data substitution business rules.
- A setup file (optional).

For output, Dataswapper creates:

- A changed EDI file containing the swapped data.
- An audit report showing what data was swapped.

Example Simple Implementation



1 Instream reads and processes EDI data using a guideline with the Dataswapper business rules but few or no other rules (for speed).

2 Validation produces detail results with SBST records needed by Dataswapper.

3 Dataswapper reads the validation detail results and the EDI data, and substitutes the EDI values. It creates a new EDI file with the changed data, along with an audit report showing the changes.

4 Instream validates the new EDI file with a guideline that does a full validation of all rules that you wish to enforce.

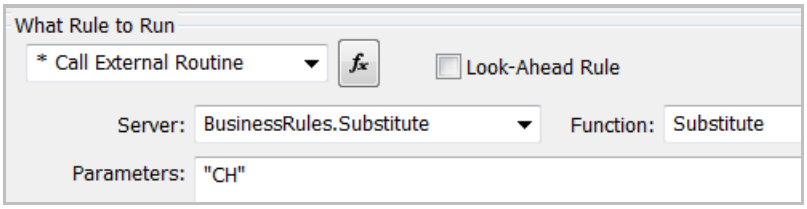
Splitting and Swapping

If you plan to split your data with Docsplitter, and then use Dataswapper to change the resulting split files, consider using the Docsplitter to perform both steps at once. See Appendix H: Split-and-Swap in **TIB_fsp-instream_*<n.n>*_docsplitter.pdf**.

Tutorial - HIPAA data

This simple example replaces the value in the BHT-06 with **CH**.

1. In EDISIM® Standards Editor, start a new guideline based on 837AQ320.
2. On the BHT-06, choose **Edit | Advanced | Business Rules | New | Invoke External Routine**.
3. Set up the rule like this:



4. Save the guideline as SUB1 and copy the STD file to Instream's Database directory. This guideline has already been installed for you in that directory, but feel free to overwrite.
5. Run Dataswapper on **2Interchanges837i.txt** or **837I_4010_H_2Interchanges.txt** in Instream's DemoData directory. It has two BHT segments, and the second one has a BHT-06 value of **RP**.

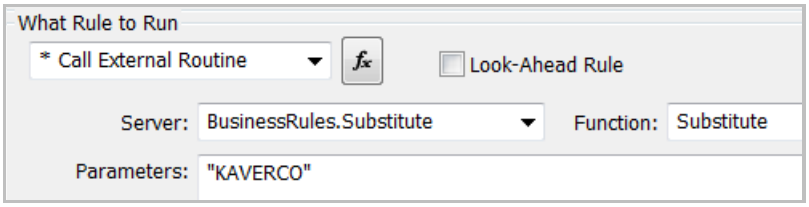
   You can use **V_Dswap_837I_4010_BHT06** in Instream's Scripts directory to perform the swap.
6. Go to Instream's **Output** directory and look at the output files.

   You now have a new EDI file with all BHT-06 values set to CH. With the data substitution finished, it is ready to validate.

Tutorial - X12 data

This simple example replaces the value in the BHT-06 with **CH**.

1. In EDISIM Standards Editor, start a new guideline based on X12-5050.
2. On the ST-03, choose **Edit | Advanced | Business Rules | New | Invoke External Routine**.
3. Set up the rule like this:



4. Save the guideline as **850_5050_SwapST03** and copy its STD file to Instream's Database directory.

   This guideline has already been installed for you in that directory, but feel free to overwrite.

5. Run Dataswapper on **850_5050_X_onePO1.txt** in Instream's DemoData directory.

   You can use **V_Swap_850_5050** in Instream's Scripts directory to perform the swap.

6. Go to Instream's **Output** directory and look at the output files.

   You now have a new EDI file 850_5050_X_onePO1NEW.txt with ST-03 set to KAVERCO.

   With the data substitution finished, it is ready to validate.

Creating a Dataswapper Guideline

Before you Begin

This activity requires someone who is familiar with EDISIM Standards Editor and understands the process of creating business rules.

BusinessRules.pdf contains details about creating business rules.

Creating a Rule

For processing speed, consider using a guideline with few or no business rules when validating for Dataswapper.

If processing speed is not an issue, you can add your Dataswapper rules to guidelines that you are already using.

Please see the tutorial on page 5 for an example of how to create a simple rule.

Swapping

Dataswapper reads through the validation detail results file to determine what values you want placed where. It then goes through the EDI file and makes the changes.

The information needed by Dataswapper is:

- A value to be swapped in.
- A location for the swap.

Your instructions to Dataswapper will vary according to the location of the find and replace values in the EDI. There are two possible scenarios.

Scenario	Description	Business Rules used
Scenario 1 (see page 9)	The value to be swapped in is already known when the location to be changed is reached.	Substitute See page 11
Scenario 2 (see page 10)	The value to be swapped in is not yet known when the location to be changed is reached.	SubstituteFind SubstituteReplace See pages 13 and 14

Swapping Scenarios

Scenario 1

Replacement value is already known when the value to be replaced is encountered

In this case, the value to be swapped in is either a literal or a variable that has already been set.

For example, the location for the swap is at ② in the diagram below, and the value to be swapped in is set in a variable at ①. When the change has to be made, the replacement value is already known.

Set a variable here ①

Substitute rule ②

To do this:

① Put a **SetVar** rule (or another rule that puts a value into a variable) here.

② Put a **Substitute** rule on location for the swap.

See MakeKey on page 11.

Scenario 2

Replacement value is not known when the value to be replaced is encountered

In this case, the replacement value has not been set when the value to be replaced is reached:

For example, the value to be replaced is at ① in the diagram below, but the value that replaces it is set at ②.

SubstituteFind rule (location of value to be replaced) ①

SubstituteReplace rule (value to use in replacement) ②

In this scenario, you will assign the find-replace pairs a "key" to link them together. This clears up ambiguities in case the file has repeating loops and therefore multiple find-replace pairs.

To do this:

- Put a SubstituteFind rule on the element containing the value to be replaced. A guideline can have multiple SubstituteFind rules with the same key.
- Set up a SubstituteReplace rule to define what value is used to replace the found value. This can be anywhere in the file. If the guideline has multiple SubstituteReplace rules, be sure that they have unique keys. Otherwise, only the last one will be used. Also, if the SubstituteReplace uses a variable, be sure that it is set before the SubstituteReplace rule executes.

  There is no point in using more than one SubstituteReplace with the same key. Since Dataswapper makes two passes through the file for Scenario 2, it the *last* SubstituteReplace that is used in all locations.
- The key acts as the binder between a find-replace pair.

Swapping Business Rules

MakeKey

(For an example, see Example 4 on page 37.)

Creates a unique key for SubstituteFind/SubstituteReplace pairs by incrementing a counter at the end of a string of characters. This assures that each key will be unique, a requirement if you have more than one find\replace pairs or if a find\replace pair is in a repeating loop.

Format of Parameters

Prefix KeyVar

Where:

Prefix	"Literal" or variable holding the base part of the key. MakeKey will automatically add an incrementing counter to this base.
KeyVar	Variable containing the key. If this guideline will be used with EDI files that have multiple interchanges, use GLOBAL_ before the prefix. Example: GLOBAL_ID. That way, it will not be cleared at the end of each interchange. When a MakeKey rule executes, if the prefix is not found in the KeyVar, it inserts the prefix and sets the increment to 1.

Substitute

(Used for Scenario 1 - see page 9)

Replaces the value in the current element or sub-element with a new value.

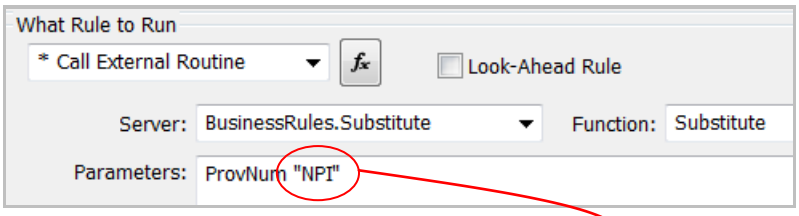
Format of Parameters

ReplaceValue MetaData

Where:

ReplaceValue	"Literal" or variable containing a value that is to replace the value in the current value. Examples: `"12345"` or `VAR1`
MetaData	Optional. "Literal" or variable containing the text of your choice. You can use this to identify the category of the swap: "ZIP" or "NPI" for example. It appears at the end of the SBST record in the detail results file.

Example: This example places the contents of the variable ProvNum into the current element and identifies it as a swap that we are categorizing as "NPI":



The detail output file will contain a record like this:

SBST 12|2010AA|NM1|9|||NPI|

SubstituteFind

(Used for Scenario 2 - see page 10)

SubstituteFind shows *where* a value is to be replaced. It specifies that the value in the current element or subelement is to be replaced.

The value that will replace it is identified with a SubstituteReplace that has the same *key*.

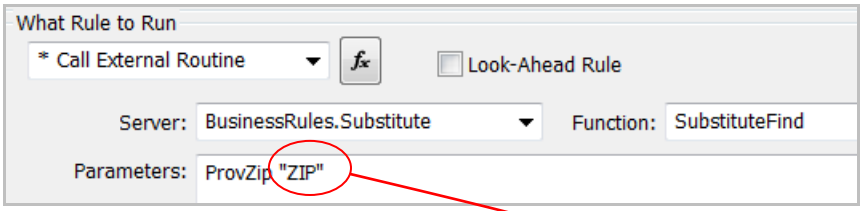
Format of Parameters

Key MetaData

Where:

Key	"Literal" or variable containing a key that will identify this element as the one where the value is to be replaced. A matching key is used in the SubstituteReplace rule. You can use the same key in multiple SubstituteFinds if you want the same value to be used at each location.
MetaData	Optional. "Literal" or variable containing the text of your choice. You can use this to identify the category of the swap: "ZIP" or "NPI" for example. It appears at the end of the SBSTF record in the detail results file.

Example: This example specifies that the key ProvZip marks the location where the data is to be changed and identifies it as a swap that we are categorizing as "ZIP":



If the quotes were omitted from around ZIP, it would mean that Dataswapper is to look up the value in ProvZip and use that value as the key.

The detail output file will contain a record like this:

SBSTF 14|ProvZip|2010AA|N4|3||ZIP|

SubstituteReplace

(Used for Scenario 2 - see page 10)

SubstituteReplace shows *what* is to be used to replace other values. It identifies the value that will replace a value identified with a SubstituteFind that has the same *key*.

Location of SubstituteReplace rule:

- If the SubstituteReplace uses a variable, be sure that the variable is set before the SubstituteReplace rule executes.
- If the guideline has multiple SubstituteReplace rules, be sure that they have unique keys. Otherwise, only the last one will be used.
- Other than the above considerations, this rule can be anywhere in the file.

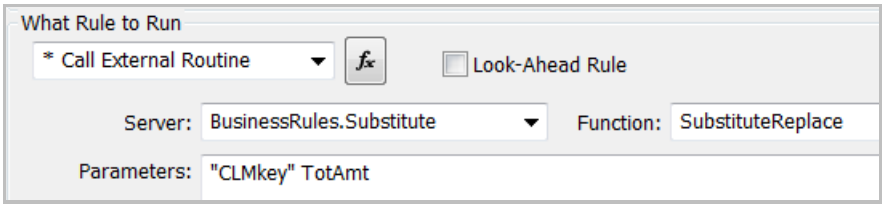
Format of Parameters

Key ReplaceValue

Where:

Key	"Literal" value or a variable containing the same key as the one that identifies the element to be replaced. This same key appears in the SubstituteFind rule.
ReplaceValue	"Literal" or variable containing a value to replace the one in the SubstituteFind element.

Example: This example specifies that the value in the variable TotAmt should replace the value in the element identified by CLMkey. CLMkey was set with a SubstituteFind business rule.

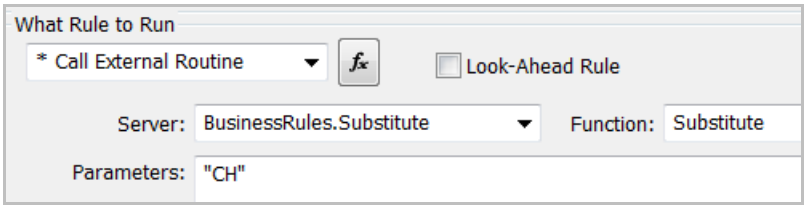


If the quotes were omitted around CLMkey, it would mean that Dataswapper is to look up the value in CLMkey and use that as the key.

Swapping Examples

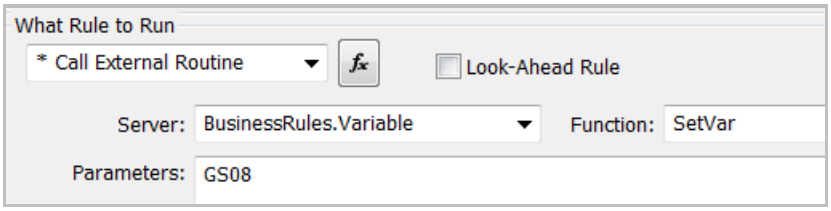
Example 1: BHT-06 must contain the literal value CH

Use Substitute on the BHT-06 to substitute in the literal value CH. Since the parameter is a literal, it has quotes:

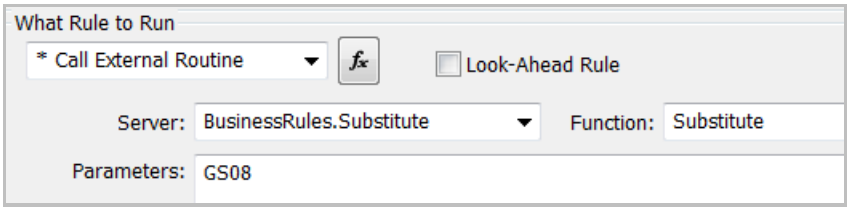


Example 2: Value in REF-02 must match the value in the GS-08

Put a SetVar on the GS-08 to capture its value into a variable:



Use Substitute on the REF-02 to substitute in the value from the variable from the GS-08. Since the parameter is a variable, it has no quotes:



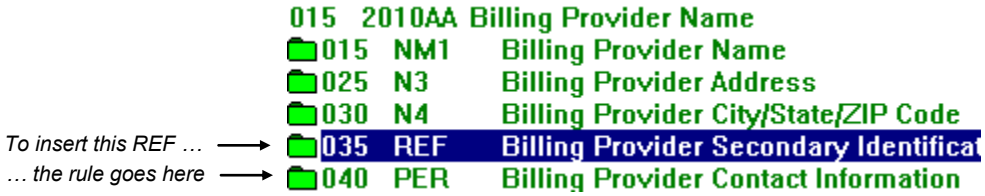
Inserting Segments

Dataswapper reads through the validation detail results file to find SBSTI records (see page 29), which specify segments that it should add to the EDI file.

It then adds the segments to the EDI and creates a new file with the segments.

Use EDISIM to create an InsertSegment business rule. The segment will be inserted before the location of the business rule by default. If you use Dataswapper's -a parameter, it will be inserted after the location of the business rule.

Example: To insert the highlighted REF, put the rule on the N4 or the PER.



InsertSegment Business Rule

Inserts a segment into the EDI above or below the current location.

Format of Parameters

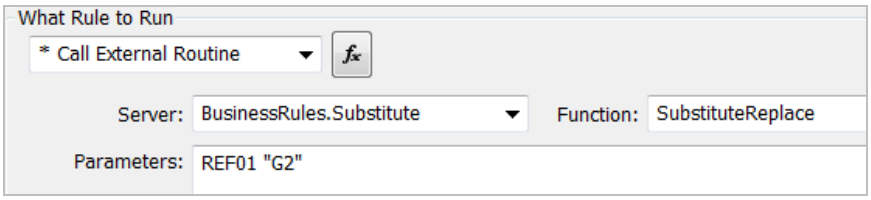
SegmentID (*Elements*) *MetaData*

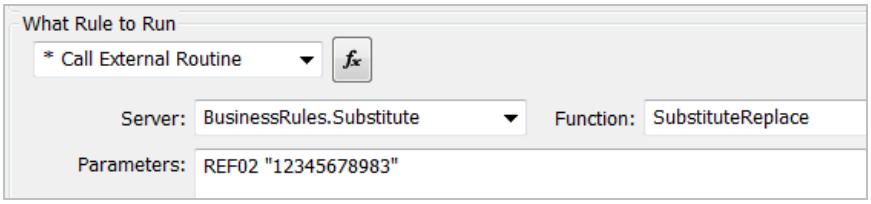
Where:

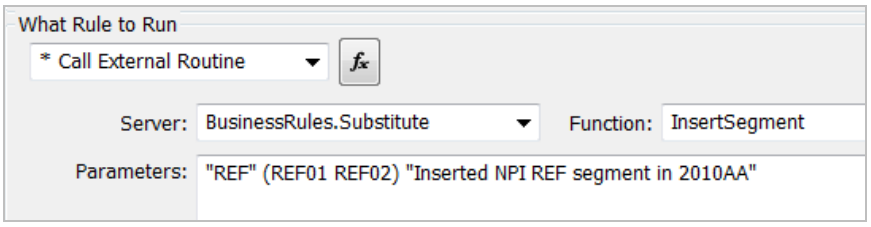
SegmentID	A "Literal" or variable containing the segment ID.
(*Elements*)	A series of FindKeys holding the values to replace. The series is surrounded by parentheses. To include sub-elements, surround them in a separate set of parentheses. See Example 2 below. The FindKeys are set with SubstituteReplace rules, which can come before or after the InsertSegment.
MetaData	Optional. "Literal" or variable containing text for your own use. It appears in the SBSTA record in the Dataswapper audit file.

InsertSegment Example

This example inserts a REF segment. It goes after the current segment if you use Dataswapper's -a parameter, and before it otherwise.







The detail output file will contain records like these (see page 29):

```
SBSTR        12|REF01|G2|
SBSTR        12|REF02|1234567893|
SBSTI        12|Inserted NPI REF segment in 2010AA|2010AA|REF|REF01|REF02
```

The audit file will contain a record like this (see page 31):

```
SBSTA        12|2010AA|REF||||REF*G2*1234567893~|Inserted NPI REF segment in
2010AA|
```

The REF segment inserted in the new EDI file will look like this:

```
REF*G2*1234567893~
```

Example 2. This example inserts a CTP segment. It goes after the current segment if you use Dataswapper's -a parameter, and before it otherwise.

On the segment below the CTP, create these business rules:

```
BusinessRules.Substitute SubstituteReplace "EMPTY" ""
BusinessRules.Substitute SubstituteReplace "CTP03" "0"
BusinessRules.Substitute SubstituteReplace "CTP04" "100"
BusinessRules.Substitute SubstituteReplace "CTP051" "UN"
BusinessRules.Substitute SubstituteReplace "CTP052" "1"
BusinessRules.Substitute SubstituteReplace "CTP053" "1"
```

```
BusinessRules.Substitute InsertSegment "CTP" ("EMPTY" "EMPTY"
"CTP03" "CTP04" ("CTPO51" "CTP052" "CTP052")) "Inserted CTP
segment"
```

The detail output file will contain records like these (see page 29):

```
SBSTR       180|EMPTY||
SBSTR       180|CTP03|0|
SBSTR       180|CTP04|100|
SBSTR       180|CTP051|UN|
SBSTR       180|CTP052|1|
SBSTR       180|CTP053|1|
SBSTI       180|Inserted CTP
segment|2410|CTP|EMPTY|EMPTY|CTP03|CTP04|CTP051:CTP052:CTP052
```

The audit file will contain a record like this (see page 31):

```
SBSTA       180|2410|CTP||||CTP***0*100:UN:1:1~|Inserted CTP segment|
```

The CTP segment inserted in the new EDI file will look like this:

```
CTP***0*100*UN:1:1~
```

Deleting Segments

Dataswapper reads through the validation detail results file to find SBSTD records (see page 27), which identify segments that it should remove from the EDI file.

It then removes the segments from the EDI and creates a new file without the segments.

Use EDISIM to create a DeleteSegment business rule for the segment that is to be deleted if it appears in the data. The rule may be on the segment itself or on one of its elements.

DeleteSegment Business Rule

Deletes the current segment from the EDI.

Format of Parameters

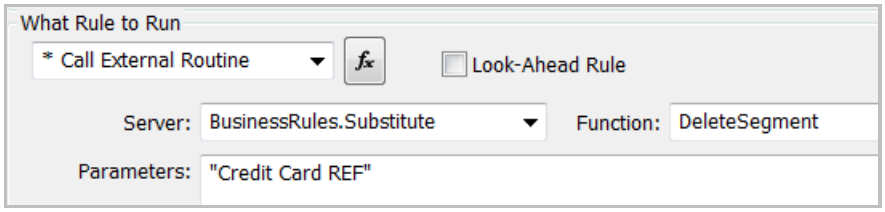
MetaData

Where:

MetaData	Optional. "Literal" or variable containing text for your own use. It appears in the SBSTA record in the Dataswapper audit file.

DeleteSegment Example

This example deletes the current segment if it appears in the EDI file.



The detail output file will contain a record like this (see page 27):

```
SBSTD        13|Credit Card REF|2010AA|REF
```

The audit file will contain a record like this (see page 31):

```
SBSTA        13|2010AA|REF|||REF*8U*123456789~||Credit Card REF|
```

Creating a Segment containing a Unique ID

Dataswapper can insert segments containing unique IDs:

```
NM1*QD*1*JOHNSON*BARBARA*T**PH.D~
N3*103 NORTH MAIN STREET*PO BOX 123~
N4*COLUMBUS*OH*43017*US~
CLM*1*100.00***11:A:1*N*A*N*A*********N**1~
ACT*fcf5cd0f-de20-11db-88de-915cc9ef8349~
DTP*096*TM*1230~
```

Please see GenerateFSUID in **BusinessRules.pdf** for details.

Executing Dataswapper

Command Line

Dataswapper is a standalone executable in Instream's **bin** directory. From the command line, it has the following case-sensitive parameters. Use quotation marks around paths or filenames that contain spaces.

-a	**a**fter - place inserted segments after the segment containing a GenerateFSUID or InsertSegment business rule. If -a is omitted, then the insertion will be before the segment that contains the business rule. This affects InsertSegment and GenerateFSUID business rules. See InsertSegment on page 16 or GenerateFSUID in **BusinessRules.pdf**.
-d	**d**ata file - path and filename of the corresponding EDI data file. Required.
-g	**g**uideline name. Required if you do not use -i. If both -i and -g are used, -g will override the guideline in the detail results file.
-i	**i**nput file - path and filename of an Instream validation detail results file.
-o	**o**utput file - path and filename of the EDI file with the changed data. Required.
-r	**r**eport file - desired path and filename for the audit report file. Required.
-s	**s**etup file - path and filename of the Dataswapper setup file. See page 23.
-v	**v**ersion - displays Dataswapper's version. Use without other parameters.
-?	**d**isplays command line help for Dataswapper.

Command Line Examples

```
C:\Foresight\Instream\Bin\DataSwapper.exe"
-i"C:\Output\ MyFile_Results.txt" -d"C:\Data\ MyFile.txt"
-r"C:\ Output\AuditReport.txt" -o"C:\ Output\ MyFileNEW.txt"
```

Sample Batch or Script Files

See **V_Dswap_837I_4010_BHT06** in Instream's Scripts folder for a demo.

Setup File Format

You can put many Dataswapper options into a setup file. Command line options that don't have a corresponding entry in a setup file include

`-v  ?  -s`

Options specified on a command line will overwrite options specified in an ini file.

Invoking a Dataswapper setup file

From Dataswapper, use **-s** as in this example:

```
C:\Foresight\Instream\Bin\DataSwapper.exe" -s"C:\FSConfigs\Dswap.ini" …
```

From Docsplitter when using the combination split-and-swap feature, use **-z** as in this example:

```
C:\Foresight\Instream\Bin\ DocSplitter.exe" -z"C:\FSConfigs\Dswap.ini" …
```

[Options] Section		
Text in INI file	**Explanation**	**Effect during Docsplitter split-and-swap**
[Options]	The Options section controls format of output	
OutputEDIWithCRLF=0 OutputEDIWithCRLF=1	Do you want a CR/LF to follow each segment terminator? 0 = no (wrap the output data) 1 = yes (default)	Always Ignored
EDIInputFileName	The original EDI file. Do not use wildcards in the name. Corresponds to Dataswapper command line option **-d** Example: `EDIInputFileName=C:\EDI\EDI_1.txt`	Always Ignored
EDIOutputFileName	The output EDI file with swapped data. Do not use wildcards in the name. Corresponds to Dataswapper command line option **-o** Example: `EDIOutputFileName=C:\Fixed\EDI_1_fixed.txt`	Always Ignored
DetailFileName	The validation detail file. Do not use wildcards in the name. Corresponds to Dataswapper command line option **-i** Example: DetailFileName`=C:\DTLfile\EDI_1_results.txt`	Always Ignored

GuidelineName	The guideline to use if you want Dataswapper to call Instream. Corresponds to Dataswapper command line option **-g** Example: GuidelineName=OUR837P	Always Ignored
InsertSegmentsAfter	If **1**, insert new segments *after* the triggering segment. If **0** (the default), insert new segments *before* the triggering segment. Corresponds to Dataswapper command line option **-a** Example: InsertSegmentsAfter=1	Always Used
[Debugging] Section		
Text in INI file	**Explanation**	**Effect during Docsplitter split-and-swap**
[Debugging]	Section that controls display of debugging information while running from the command line	
ProcessRecord=0 ProcessRecord=1	Do you want to display detail file record IDs as they are processed? 0 = no (default) 1 = yes	Always Ignored
[Suppress] Section		
Text in INI file	**Explanation**	**Effect during Docsplitter split-and-swap**
[Suppress]	Section that controls overrides to conditions that normally stop Dataswapper from processing	
MissingReplaceValue=0 MissingReplaceValue=1	Do you want to continue processing if no value is found to use for the replace? This can happen if Dataswapper was told to replace an element but was never given a value to use instead. The SubstituteReplace rule may be on an element that was not included in the EDI data. 0 = no (default). Stop processing the file and give a return code of 206 and an explanatory message. 1 = yes, continue processing the file.	Always Used

Appendix A: Dataswapper Records

Overview of Dataswapper Records

Record	Purpose	File containing Record	For details, see page …
Records for swapping rules			
SBST	Generated by a Substitute business rule	Validation detail results file	26
SBSTF	Generated by a SubstituteFind business rule	Validation detail results file	27
SBSTR	Generated by a SubstituteReplace business rule	Validation detail results file	29
SBSTA	Describes what was replaced	Dataswapper audit file	31
Records for inserting and deleting segments			
SBSTD	Generated by a DeleteSegment business rule	Validation detail results file	27
SBSTI	Generated by an InsertSegment business rule	Validation detail results file	29

Dataswapper Records in the Detail Results File

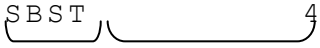
SBST Record

This record is generated during validation by a Substitute business rule, which is used for Scenario 1 (see page 9).

Fields in the SBST Record

Field	Length	Start	End	Contents
Record Tag	5	1	5	Literal value `SBST` for Substitute
Starting line #	10	6	15	Counter from beginning of EDI file to identify which segment contains the value to replace
Delimiter	1			| Vertical bar
Loop ID				ID of the loop containing the value to be replaced
Delimiter	1			| Vertical bar
SegId				Segment ID
Delimiter	1			| Vertical bar
ElmPos				Element position (within segment) that contains the value to be replaced
Delimiter	1			| Vertical bar
SubElmPos				Subelement position (within composite) that contains the value to be replaced
Delimiter	1			| Vertical bar
ReplaceValue				Replacement text
Delimiter	1			| Vertical bar
MetaData				Text of your choice inserted by the Substitute business rule
Delimiter				| Vertical bar

Example: In the BHT-06, which appears on line 4 of the EDI file, substitute in the value "CH". The S is meta data tacked on by the SubstituteFind business rule for your own use. Instream does not use it.



SBSTD Record

This record is generated during validation by a DeleteSegment business rule, which is described on page 19.

Fields in the SBSTD Record

Field	Length	Start	End	Contents
Record Tag	5	1	5	Literal value `SBSTD` for DeleteSegment
Line # of segment	10	6	15	Counter from beginning of EDI file to identify which segment is to be deleted
Delimiter	1			| Vertical bar
MetaData				Optional text to record in the corresponding SBSTA record in the audit file; for your own use
Delimiter	1			| Vertical bar
LoopID				Loop containing the segment to delete
Delimiter	1			| Vertical bar
SegmentID				Tag of the segment to delete

Example: Delete the REF for Credit/Debit Card Billing Information (Loop 2010AA). There should be a corresponding entry in the audit file containing the text `Credit Card REF`.

```
SBSTD          13|Credit Card REF|2010AA|REF
```

SBSTF Record

This record is generated during validation by a SubstituteFind business rule, which is used for Scenario 2 (see page 10).

Fields in the SBSTF Record

Field	Length	Start	End	Contents
Record Tag	5	1	5	Literal value `SBSTF` for Substitute Find
Line # of segment	10	6	15	Counter from beginning of EDI file to identify which segment contains the value to replace
Delimiter	1			| Vertical bar
Key				Unique value used to match with a SBSTR replacement value
Delimiter	1			| Vertical bar
Loop ID				ID of the loop containing the value to be replaced
Delimiter	1			| Vertical bar
SegId				Segment ID
Delimiter	1			| Vertical bar
ElmPos				Element position (within segment) that contains the value to be replaced
Delimiter				| Vertical bar
SubElmPos	1			Subelement position (within composite) that contains the value to be replaced
Delimiter				| Vertical bar
MetaData				Text of your choice inserted by the SubstituteFind business rule
Delimiter				| Vertical bar

Example: In the 2310A NM108, which appears on line 38 of the EDI file, substitute in the value 8. The S is meta data tacked on by the SubstituteFind business rule for your own use. Instream does not use it.

```
SBSTF          38|Q3|2310A|NM1|8||S
```

SBSTI Record

This record is generated during validation by an InsertSegment business rule (see page 16) or a GenerateFSUID business rule (see page 20).

Fields in the SBSTI Record

Field	Length	Start	End	Contents
Record Tag	5	1	5	Literal value `SBSTI` for InsertSegment
Line # of segment	10	6	15	Counter from beginning of EDI file to identify where the segment is to be inserted; inserted segment will go above this line
Delimiter	1	16	16	| Vertical bar
MetaData				Optional data to record in the corresponding SBSTA record in the audit file; for your own use InsertSegment and GenerateFSUID business rules write comments to this field
Delimiter	1			| Vertical bar
Loop ID				ID of loop that is to contain the inserted segment
Delimiter	1			| Vertical bar
SegmentID				ID of the new segment
Delimiter	1			| Vertical bar
Element Data				Data; subelement data is separated by a colon
Delimiter	1			| Vertical bar
Element Data				(may appear multiple times)

Example: Insert a REF segment in loop 2010AA at line 14 of the EDI file. `Inserted NPI REF` is meta data tacked on by the InsertSegment business rule for your own use. Instream does not use it.

```
SBSTI          14|Inserted NPI REF|2010AA|REF|REF01|REF02
```

There should be a corresponding entry in the audit file containing the text `Inserted NPI REF`.

SBSTR Record

This record is generated during validation by a SubstituteReplace business rule, which is used for Scenario 2 (see page 10).

Fields in the SBSTR Record

Field	Length	Start	End	Contents
Record Tag	5	1	5	Literal value `SBSTR` for Substitute Replace
Line # of segment	10	6	15	Counter from beginning of EDI file to identify which segment contains the value to replace
Delimiter	1			| Vertical bar
Key				Unique value used to match with a SBSTF Find Value
Delimiter	1			| Vertical bar
ReplaceValue				Replacement text
Delimiter	1			| Vertical bar

Example: Key CTP053 contains the value 1 at line 180 in the EDI file.

```
SBSTR          180|CTP053|1|
```

Dataswapper Record in the Audit File

SBSTA Record

This record in the audit report describes what was replaced.

Fields in the SBSTA Record

Field	Length	Start	End	Contents
Record Tag	5	1	5	Literal value SBSTA
Line # of segment	10	6	15	Counter from beginning of EDI file to identify which segment contains the value to replace
Delimiter	1			| Vertical bar
Loop ID				ID of loop that contains the changed value
Delimiter	1			| Vertical bar
SegId				Segment ID
Delimiter	1			| Vertical bar
ElmPos				Element position (within segment) that contains the value to be replaced
Delimiter	1			| Vertical bar
SubElmPos				Subelement position (within composite) that contains the value to be replaced
Delimiter	1			| Vertical bar
Old value				Value that was replaced
Delimiter	1			| Vertical bar
New value				Value that replaces the old value
Delimiter	1			| Vertical bar
MetaData				Text of your choice inserted by a Substitute, SubstituteFind, or DeleteSegment business rule.

Example: On line 63, in the BHT06, the value RP was replaced with the value CH:

```
SBSTA          63||BHT|6|0|RP|CH||
```

Appendix B: Dataswapper Return Codes

Return Codes

Return Code		Meaning
FSDXSUCCESS	100	Dataswapper ran successfully.
FSDXFAILED	110	Dataswapper did not run successfully.
FSDXLIBFAILED	150	Dataswapper did not run successfully because it could not load one of its internal libraries. This is probably an installation problem. Contact TIBCO Foresight Technical Support.
FSDXINITFAILED	180	Dataswapper did not run successfully because of a problem other than those listed below. Contact TIBCO Foresight Technical Support.
FSDXEDIINPUT	200	Dataswapper could not access the EDI input file.
FSDXEDIOUTPUT	201	Dataswapper could not open the EDI output file.
FSDXINSTREAMINPUT	202	Dataswapper could not open the Instream detail file.
FSDXREPORT	203	Dataswapper could not open the report file.
FSDXINIINPUT	205	Dataswapper could not open the INI configuration file.
FSDXNOFINDKEY	206	Dataswapper could not locate a key for Replacement.
FSDXBADMATCH	207	Dataswapper failed. The detail and EDI input files seem to mismatch. The segment for the replacement was not found in the expected location.

Troubleshooting information	Notes
Audit report	Use -r command-line parameter to specify the audit report file. Use ProcessRecord setup file parameter.
Validation detail results file	Look for Dataswapper records: SBST, SBSTD, SBSTF, SBSTI, SBSTR.

Displaying 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

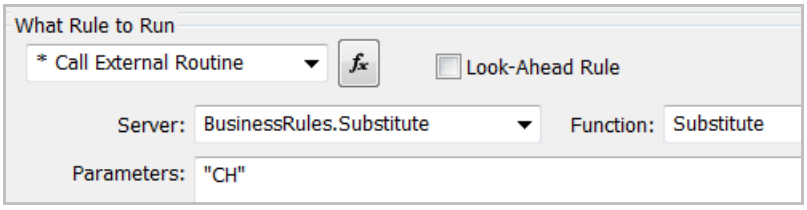
Exclude all TIBCO Foresight workflow subdirectories from virus checking.

Appendix C: Swapping Examples

Example 1: Replace value with Literal

This is an example of Scenario 1 on page 9.

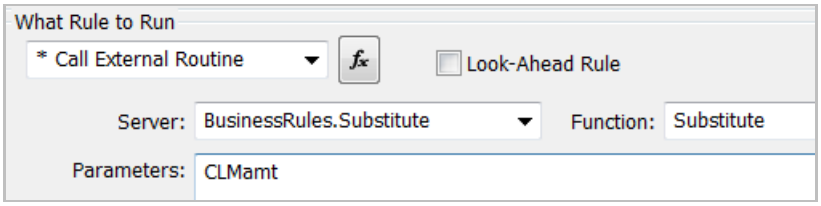
Replacing a value with a literal requires just the Substitute business rule. This example replaces the value in the current segment with CH, which is surrounded by quotes since it is to be interpreted literally.



Example 2: Replace Value with Previously set Variable

This is an example of Scenario 1 on page 9.

Replacing a value with a variable that has been set requires just the Substitute business rule. This example replaces the value in the current segment with the contents of the variable CLMamt, which is not surrounded with quotes:



Example 3: Find Value before Replace Value (not in repeating loop)

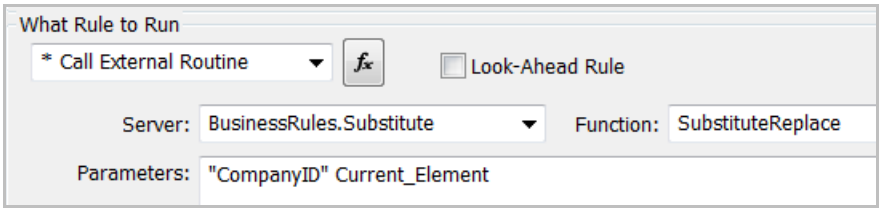
This is an example of Scenario 2 on page 10.

In a HIPAA 820, the BPR-10 (which is optional) and the TRN-03 (which is mandatory) must match.

Since the TRN is mandatory, we will use that as the replace value. If a BPR-10 is included in the data, a business rule on it will overwrite its data with the value in the TRN.

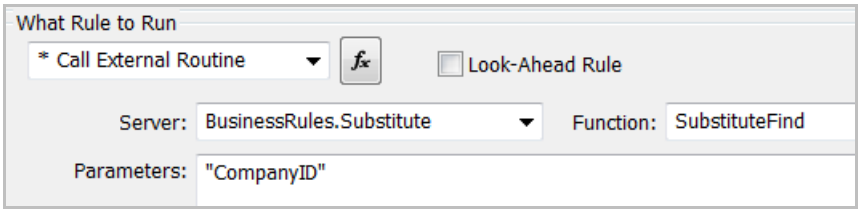
Mark the TRN-03 as the value to use as a replacement

Put a SubstituteReplace rule like this one on the TRN-03. The key is **CompanyID**. This will match the key in the BPR-10. The ReplaceValue is **Current_Element**, meaning the value in the TRN-03.



Mark the BPR-10 as the value to be replaced

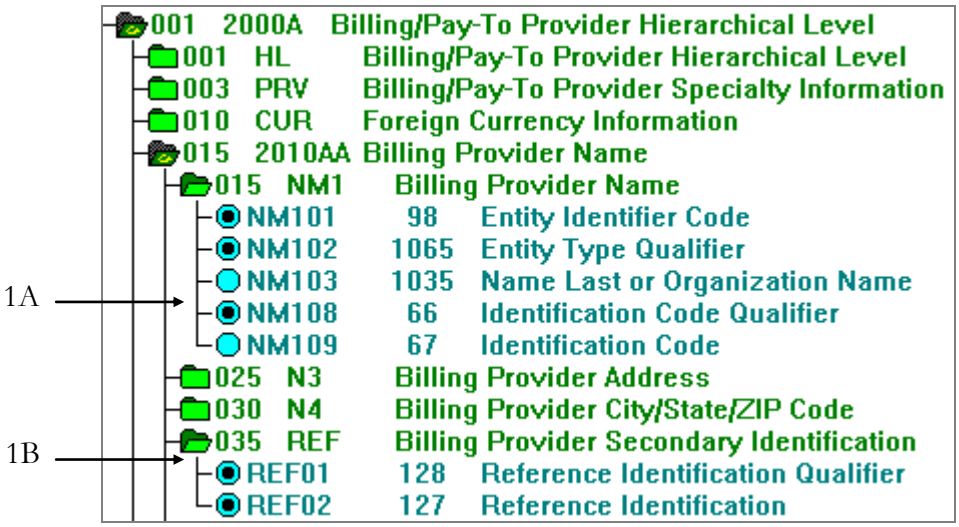
Put a SubstituteFind rule like this one on the BPR-10. The key is "CompanyID" to match the one used in SubstituteFind. Use quotation marks unless you are pulling the name of the key from a variable.



Example 4: Find Value before Replace Value (in repeating loop)

If your replace value is in a repeating loop or occurs more than once in the EDI, then use MakeKey to ensure that the key will be unique for each find-replace pair.

In this example, we replace the NM109 (see 1A below) with the REF02 (see 1B below).

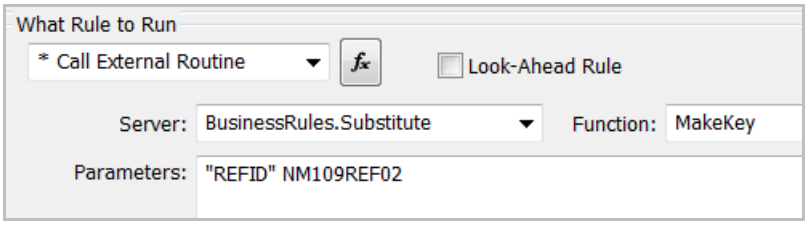


A key will be used to link the NM109 and REF02 together.

Since this is in a repeating loop (2000A), we use MakeKey to provide an incrementing key counter. This gives the NM109 and REF02 pair a unique key in each iteration of the loop.

Set up the MakeKey before using it

We put this MakeKey command on the NM1 segment since it is mandatory and occurs within the repeating loop but before the REF02 or NM109:

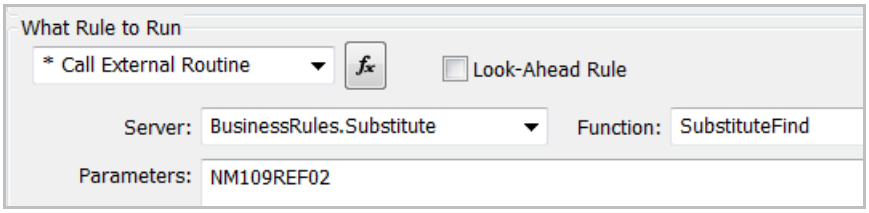


This sets up a variable NM109REF02 to hold the key, which will be:

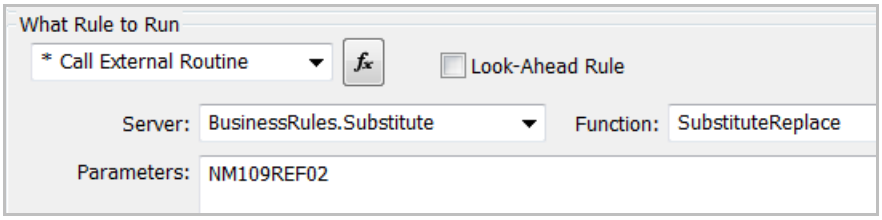
REFID*n*

where *n* is the incrementing counter.

Mark the NM109 as the value to find



Mark the REF02 as the value to use as a replacement



Other Rules

In a real life scenario, this example would require additional rules to:

- Ensure that the substitution only took place if 837I 2010AA NM108=XX.
- Handle the qualifiers as follows:

If REF01 contains…	Put this value in NM108:
EI	24
SY	34