TIBCO Foresight® ICD-10 Conversion Adapter

ICD-9 and ICD-10 at TIBCO Foresight

Software Release 1.3.0
October 2014

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.

USEOF TIBCO SOFTWARE AND THIS DOCUMENTIS SUBJECTTO THE TERMS AND CONDITIONSOFA LICENSE AGREEMENT FOUNDIN EITHERASEPARATELY 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 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, The Power of Now, TIBCO Foresight EDISIM, TIBCO Foresight HIPAA Validator Desktop, TIBCO Foresight ICD-10 Conversion Adapter, TIBCO Foresight Instream, TIBCO Foresight Studio, TIBCO Foresight Transaction Insight, and TIBCO Foresight Translator 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.

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 RELEASEDAT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITYOF THIS SOFTWARE VERSIONONASPECIFIC 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 FORAPARTICULAR 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©1999-2014 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

General Contact Information

TIBCO Software Inc., Foresight Group
655 Metro Place South
Suite 900
Dublin OH 43017
Phone: (614) 791-1600
Fax: (614) 791-1609

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

The TIBCO Foresight® ICD-10 Conversion Adapter works with TIBCO Foresight® Translator and/or TIBCO Foresight® Instream® to convert International Classification of Diseases, Ninth Revision (ICD-9) codes to International Classification of Diseases, Tenth Revision (ICD-10) codes, and vice versa.

To use this adapter, you must have installed one or both of these:

- Foresight® Translator 3.2.0 or later
- Instream® 8.2.0 or later

You will also need TIBCO Foresight® EDISIM® Standards Editor for business rule creation.

Additional TIBCO Foresight products may be required depending on your needs.

Overview

This document describes how to use TIBCO Foresight products to facilitate the conversion of ICD-9 codes to the ICD-10 revision, or vice versa.

Keep in mind:

- ICD-9 codes are 3-5 characters in length.
- ICD-10 codes are currently 3-7 alphanumeric characters. There are many more codes, and codes are much more specific than ICD-9 codes.

TIBCO Foresight products can be used to:

- Validate ICD codes using a Foresight-supplied types 1-7 guideline or a guideline merged with one.
- Translate an entire file including ICD conversions using the Translation Tool to create ICD business rules in map.
- Replace an ICD value with a converted value by creating ICD business rules in a guideline by using Instream to validate and then feeding the detail results file into Dataswapper.
- Translate or Swap ICD codes in an Automator workflow using TIBCO Foresight® Studio's GenericTranslator, Dataswapper, or DataswapperNoDTL components.
- View ICD codes in TIBCO Foresight® Transaction Insight® (TI) or TIBCO Foresight Test Asset Management Suite (TAMS) via the ICD pages, which are populated if ICD adapter rules are used during validation.

Validating ICD Codes

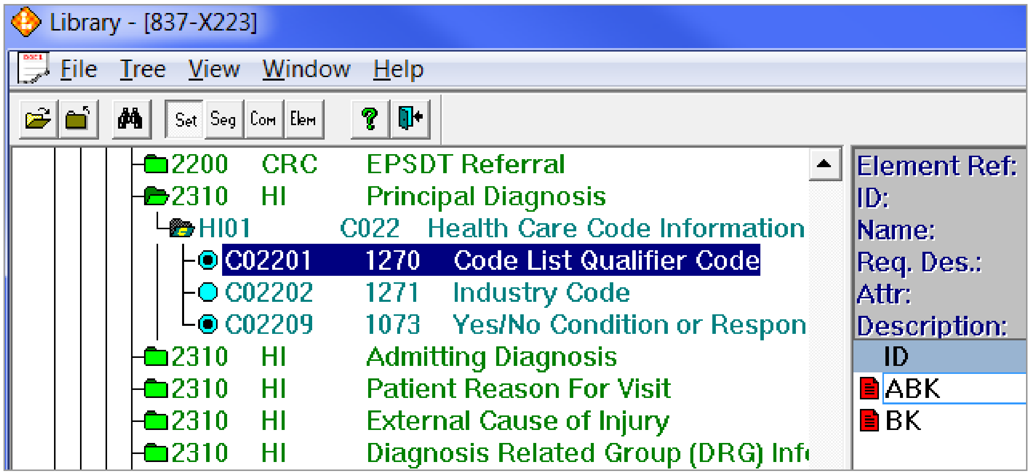
Task	Use these Programs:	Foresight® ICD-10 Conversion Adapter required?	Business rules required?	Notes
Validate	Instream or TIBCO Foresight® HIPAA Validator® Desktop	No	No	Validate with Foresight-supplied types 1-7 guidelines or guideline merged with one

HIPAA Validator® Desktop and Instream automatically validate ICD codes when using Foresight 4010 and 5010 types 1-7 guidelines. These guidelines use the qualifier and the date of service to determine which ICD version to use. You do not have to add these edits to your guidelines.

Please see **ForesightHIPAAguidelineList.pdf** for a list of types 1-7 guidelines.

Validation Example

In this Principal Diagnosis HI segment, the HI-01-01 can contain either contains BK (an ICD-9 code) or ABK (an ICD-10 code). If it contains ABK, the HI.01.02 will be validated against the ICD-10 code list. If it contains BK, it will be validated against the ICD-9 code list.



```
2200   CRC   EPSDT Referral
2310   HI    Principal Diagnosis
  HI01     C022  Health Care Code Information
    C02201   1270  Code List Qualifier Code
    C02202   1271  Industry Code
    C02209   1073  Yes/No Condition or Respon
2310   HI    Admitting Diagnosis
2310   HI    Patient Reason For Visit
2310   HI    External Cause of Injury
2310   HI    Diagnosis Related Group (DRG) Infı
```

Element Ref:
ID:
Name:
Req. Des.:
Attr:
Description:

ID
ABK
BK

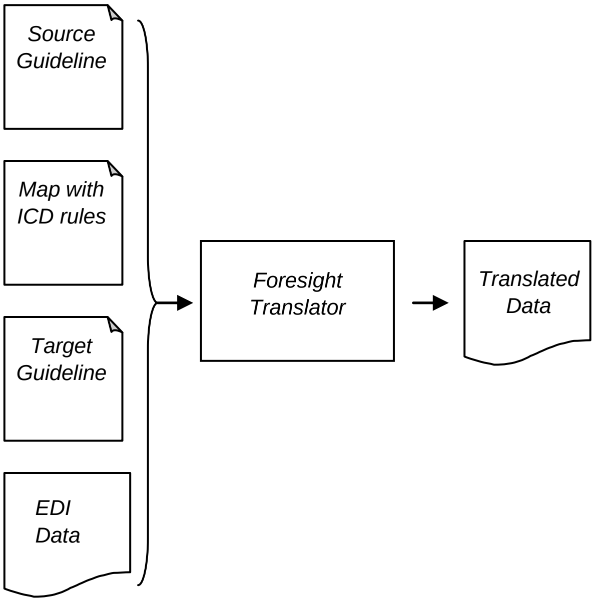
Using Foresight Translator to Convert ICD Codes

Task	Use these Programs:	ICD-10 Conversion Adapter required?	Business rules required?	Notes
Translate entire file including ICD conversion	Translator	Yes	Yes	Use Translation Tool to create ICD business rules in map

Use Foresight Translator if you want to translate the entire file, including translating between ICD-9 and ICD-10 codes.

To do this:

- Have Foresight Translator 3.2.0 or later.
- Have the ICD-10 Conversion Adapter.
- Add ICD business rules to the map with Translation Tool. The locations for these business rules vary by transaction (see page 43). Please see page 11 for details about the rules and **Translator.pdf** for general information about creating rules.



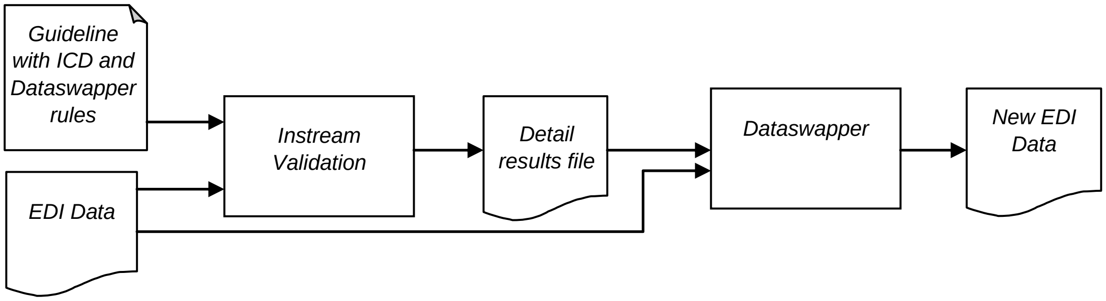
Using Dataswapper to Convert ICD Codes

Task	Use these Programs:	ICD-10 Conversion Adapter required?	Business rules required?	Notes
Replace ICD value with converted value	Instream, Desktop and Dataswapper	Yes	Yes	Use EDISIM® Standards Editor to create ICD business rules in guideline; use Instream to validate with it; feed detail results file into Dataswapper.

Use Dataswapper if you want to replace some individual data values, including ICD-9 and ICD-10 codes.

To do this:

- Have EDISIM 6.9.3 or later.
  (Note: For EDISIM 6.9.3 only, a supplemental FSBRD.dat file must be added in EDISIMs \bin directory. Contact TIBCO Foresight Support for additional information.)
- Have Instream 8.2.0 or later.
- Have the Foresight ICD-10 Conversion Adapter.
- Add business rules to the guideline with EDISIM Standards Editor. The locations for these business rules vary by transaction (see page 43). Please see page 11 for details about the rules, **BusinessRules.pdf** for general information about creating rules, and **Dataswapper.pdf** for information on swapping data.



Translating or Swapping in an Automator workflow

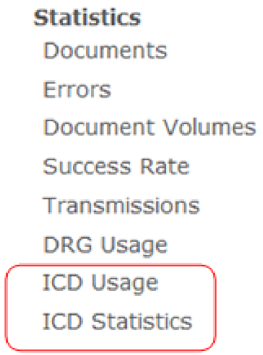
Task	Use these Programs:	ICD-10 Conversion Adapter required?	Business rules required?	Notes
Translate or Swap in Automator workflow	Foresight® Studio	Yes	Yes	Use GenericTranslator, Dataswapper, or DataswapperNoDTL components

Refer to **ForesightStudio.pdf** for additional information.

Viewing ICD Information in Transaction Insight®

Task	Use these Programs:	ICD-10 Conversion Adapter required?	Business rules required?	Notes
View in TI or TAMS	Transaction Insight or TAMS	Yes	Yes	(TI version 4.2 and later) ICD pages appear for TI and TAMS and are populated if you use ICD adapter rules when validating. Be sure you are in Healthcare mode on the **Settings | General Settings** page of **TIB_transactioninsight_*version*_commonadmin.pdf**.

You can import data containing ICD9 codes into TI or TAMS and then use it with these pages:



To populate these pages:

1. From the portal's Common Administration, choose **Settings | General settings** and turn these on:
   - Product Mode: Healthcare
   - Page Settings: Display ICD Usage
2. Validate with a guideline that has ICDConvert or ICDConvertOne rules that create GEN record 15016 in the DTL file as described on page 15.

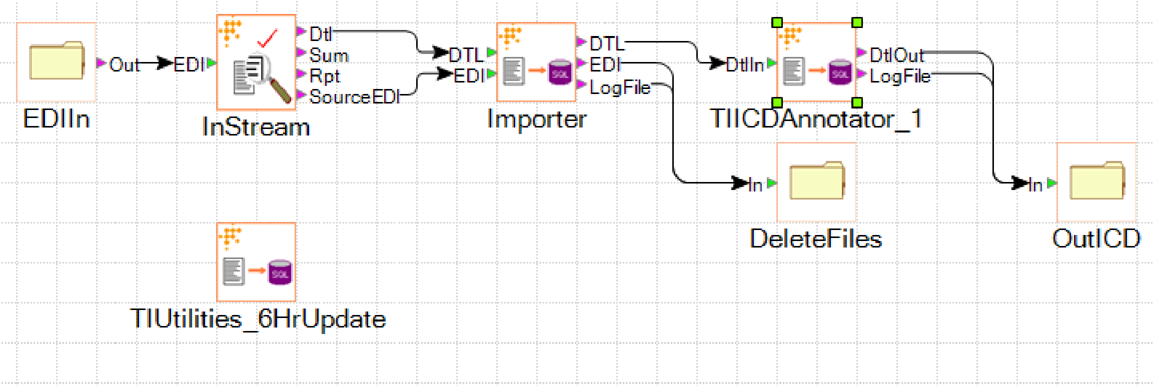
3. Import the data into TI or TAMS.

4. After importing completes, run the same DTL file into TIICDAnnotator.

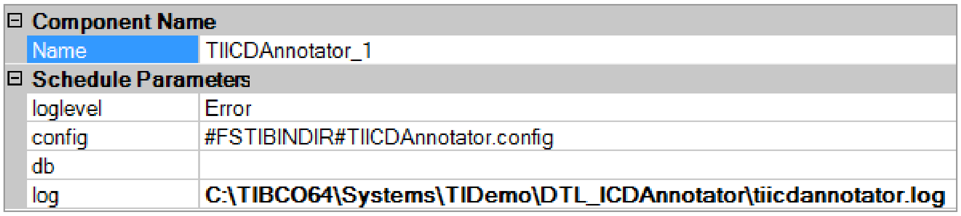
   TIICDAnnotator finds VIPS ICD 9 information in a validation detail results file and writes the code type, ICD9 code, ICD10 code, age, gender, effective date and claim charge amount into the TI database. It finds the VIPS information in the GEN record with error #15016. You can do this with a Foresight Studio workflow or from the command line.

Inserting ICD Information with a Workflow

In Foresight Studio, use the TIICDAnnotator component to update your database. Notice that it has to run after Importer.



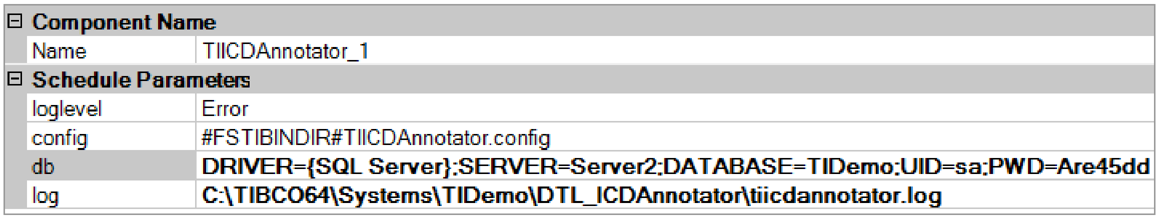
Its parameters would look something like this, where **#FSTIBINDIR#** is defined under **File | Edit Globals**.



⊟ **Component Name**	
Name	TIICDAnnotator_1
⊟ **Schedule Parameters**	
loglevel	Error
config	#FSTIBINDIR#TIICDAnnotator.config
db	
log	**C:\TIBCO64\Systems\TIDemo\DTL_ICDAnnotator\tiicdannotator.log**

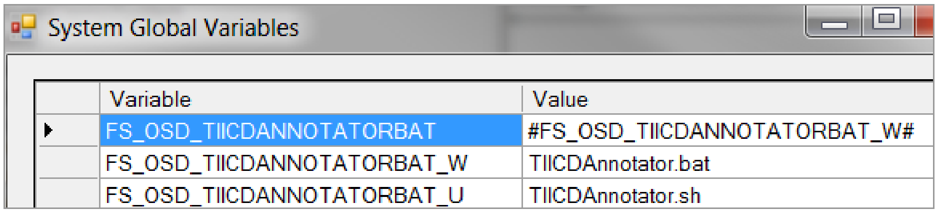
In this example, the database connection string is in TIICDAnnotator.config, since it is not given in the parameters.

To include it in the parameters, overriding the config file, it would look something like this:



⊟ **Component Name**	
Name	TIICDAnnotator_1
⊟ **Schedule Parameters**	
loglevel	Error
config	#FSTIBINDIR#TIICDAnnotator.config
db	**DRIVER={SQL Server};SERVER=Server2;DATABASE=TIDemo;UID=sa;PWD=Are45dd**
log	**C:\TIBCO64\Systems\TIDemo\DTL_ICDAnnotator\tiicdannotator.log**

Choose **File | Edit Globals** and set this variable to the name of your bat or sh file that contains the command line:



System Global Variables

Variable	Value
FS_OSD_TIICDANNOTATORBAT	#FS_OSD_TIICDANNOTATORBAT_W#
FS_OSD_TIICDANNOTATORBAT_W	TIICDAnnotator.bat
FS_OSD_TIICDANNOTATORBAT_U	TIICDAnnotator.sh

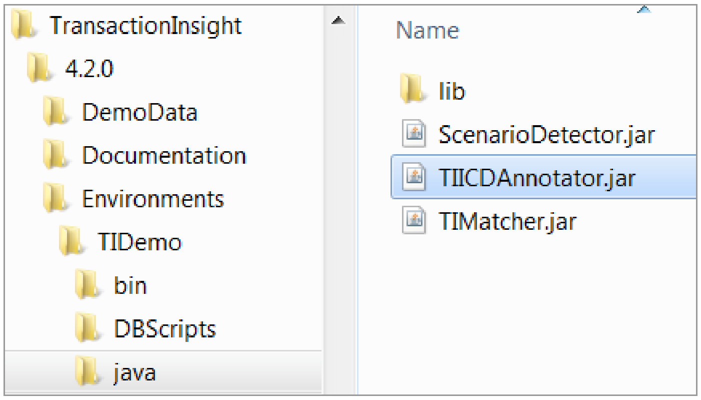
After TITulities runs with the -s parameter, the two ICD pages should be updated.

Inserting ICD Information from a Command Line

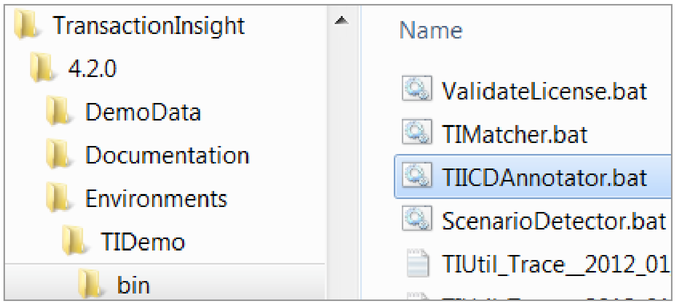
Important Files

TIICDAnnotator.jar goes into the TI environment's TIDemo\java directory.

Example:



It is run with TIICDAnnotator.bat, which goes in the TI environment's TIDemo\bin directory.



Command Format

Use double quotes around values containing spaces or special characters.

Command line settings take precedence over those in Importer.ini where there is a conflict.

```
TIICDAnnotator -db database_connection_string -r filename -log filename
-v log level -config filename
```

TIICDAnnotator Command Line Parameters

Command line option	Corresponding INI file option	Explanation	Examples
path and filename to TIICDAnnotator.jar	*none*		`"C:\TIBCO64\TransactionInsight\4.2.0\Environments\TIDemo\java\TIICDAnnotator.jar"`
-config	*none*	`-config` *filename* Either –config or –db is required. **-config** specifies the path and name of the configuration file.	`-config C:\TIBCO64\TransactionInsight\4.2.0\Environments\TIDemo\bin\TIICDAnnotator.config`
-db	-db	`-db` *database_connection_string* Either –config or –db is required. -db specifies the connection string to the TI database. **For Oracle:** *jdbc:oracle:thin:username/password@dbserver:port:instance* **For SqlServer:** *jdbc:sqlserver://dbserver[:port];databaseName=dbname;user=myuser;password=mypass;*	`jdbc:sqlserver://BigSrv2;databaseName=TIDemo;user=sa;password=asdf876Q;`
-r	*none*	`-r` *filename* Required. Instream results file. ICD data from this file is inserted into the database.	`-r "C:\DTLfiles\ File1.dtl"`
-log	*none*	`-log=`*filename* Optional. Log file path and name. Defaults to the value from the -r option + ".tiicdannotator.log".	`-log “C:\logs\ICD.log”`

TIICDAnnotator Command Line Parameters			
Command line option	**Corresponding INI file option**	**Explanation**	**Examples**
-v		`-v`*n*, where n is one of these: 1 = Errors only 2 = Warnings and Errors (default) 3 = Info, Warnings and Errors Optional. Amount of information to write to the log.	`-v1`

Example command line:

```
C:\TIBCO64\TransactionInsight\4.2.0\Environments\TIDemo\bin\ TIICDAnnotator.bat
"C:\TIBCO64\TransactionInsight\4.2.0\Environments\TIDemo\java\TIICDAnnotator.jar"
-r "C:\ValidationDTL\51_837I_20120201_5.dtl"
-config C:\TIBCO64\TransactionInsight\4.2.0\Environments\TIDemo\bin\
TIICDAnnotator.config
-log C:\logs\tiicdannotator.log
-v1
```

Understanding TI's ICD Pages

Please see ICD Usage and ICD Statistics Pages in **TIB_transactioninsight_*version*_userguide.pdf**.

Rules Reference Information

Foresight Translator ICD Rules

Foresight Translator rules for ICD codes are:

Please see Foresight Translator Demo on page 32.

HCPCS codes are not converted but can be used as selection criteria. Procedure codes of 5 characters are assumed to be CPT codes, which are not converted.

ConvertOne

This rule takes one ICD code as input and converts it to another ICD code, which it returns in a variable. It does not use any other selection criteria during the conversion. Use it when you are converting from ICD10 to ICD9 or when you want the most general code to be returned.

Format of Parameters

RuleTable Function Option InputVar TypeofinputData OutputVar

Where:

RuleTable	Name of the rule file to use for the ICD code conversion. Please see Rule File Names on page 23.
Function	How to convert. If you intend to get one code back to put into a single output element, you can use `ConvertFirstOnly`. Please see Appendix C – Convert Functions on page 30.
Option	Use `Normal`.
InputVar	Source of the code to be converted: `Current_Element`, literal in double quotes, or a variable.

TypeofinputData	Choose one of these to specify what kinds of ICD codes are to be converted:
	`Diagnosis`
	`PrincipalDiagnosis`
	`Procedure`
	`PrincipalProcedure`

OutputVar	Name of variable to hold the output ICD code.

Convert

Convert uses values from an array to select an output ICD code, which is returned in another array. Use this when converting from ICD9 to ICD10 and you want the returned code to be selected based on multiple values in the EDI, or you want multiple codes to be converted and/or multiple codes returned.

Format of Parameters

DiagnosisRuleTable ProcedureRuleTable Function Option Type InputArray OutputArray

Where:

DiagnosisRuleTable	Name of the table to be used for diagnostic code mapping. This field is ignored if you are not converting a diagnostic code. Please see Rule File Names on page 22.
ProcedureRuleTable	Name of the table to be used for procedure code mapping. This field is ignored if you are not converting a procedure code.
Function	How to convert. Use `ConvertFirstOnly` if you are only converting the code in cell 0,0 of the input array. If you want to convert all the ICD codes in the input array, see Appendix C – Convert Functions on page 30.
Option	Use `normal`.

Type	Type of code(s) to convert in the input array:
	`Diagnosis`
	`Procedure`
	`Both`
	Procedure codes of 5 characters are assumed to be CPT codes, which are not translated. They can be used in the InputArray as selection criteria. Procedure codes of other lengths can be used here.

InputArray	Name of the array containing the code to be converted plus other data to be used for conversion criteria (date of birth, sex, diagnosis codes, procedure codes, etc.). This array was created with InsertToArrayWithType (see page 13). Please see Input Array below.
OutputArray	Name of the array to hold the output ICD codes. The codes will be in the order specified under Function. Diagnosis codes will always be in row 0 and procedure codes will be in row 1. Use a 2DArray.Delete rule to clear the output array before issuing the Convert rule.

Input Array

Example input array:

	0	1	2	3	4	5
0	*Diagnosis or Procedure Code to convert* `D01484`	*Date of birth (converted to age)* `a8`	*Sex (M, F. or U)* `xF`	*Other Diagnosis or Procedure Code* `d01483`	*Other Diagnosis or Procedure Code* `d01484`	*Other Diagnosis or Procedure Code* `p0001`

Cell 0,0 of the array must include the ICD code to be converted, and other cells in row 0 can contain any of the values shown on page 6.

The array is created with InsertToArrayWithType (see page 13). The rule automatically adds the prefixes so that the Convert rule can interpret the value regardless of the order in which they appear in the array.

Please see Foresight Translator Example – Convert Translation with Selection Criteria on page 34.

InsertToArrayWithType

This rule inserts an ICD-related value into a cell in an array. The array can then be used as input for the Convert rule (page 12).

Format of Parameters

ArrayName Row Column Value Type

Where:

ArrayName	Name of the array you are populating. This can be a variable or a literal in double quotes.
Row	This will always be 0. Please read Array Reserved Variables in **BusinessRules.pdf** to see how to specify a row for array rules. The first row is row 0. The order in which data is to be input into the row is not important, except cell 0,0 is reserved for a code to be converted.
Column	The column where the value is to be inserted. This can be a variable, a literal in double quotes, or one of the array reserved variables like **N** for next. The first column is column 0.
Value	Value to put in the cell. This can be a variable, a literal in double quotes, or a reserved variable like `Current_Element`.
Type	A Dataswapper Identifier code representing the value type. Select from the drop-down menu. (See Appendix A – Prefixes and Identifiers for Data Conversions on page 21 for additional explanation of the codes.)

DOB
DateOfBirth (D8) - Identifies the input value as the 8-digit date of birth. The value is converted to an age, put in the array, and preceded with **a**. Example: If the date of birth means the person is 53 years old, the array cell will contain **a53**.

PDC
PrincipalDiagnosis - Identifies the input value as the principal diagnosis code. The value is put in the array and preceded with **D**. For example, if the input code (ICD-9) is 5770, the array cell will contain **D5770**.

DC
Diagnosis - Identifies the input value as a diagnosis code other than the principal diagnosis code. The value is put in the array and preceded with **d**. Example: If the input code (ICD-9) is 5770, the array cell will contain **d5770**.

DOS
DateOfService/Discharge - Identifies the input value as the date of service or discharge date. The value is put in the array as an 8-digit date and preceded with **e**. Example: If the input date is 20110422, the array cell will contain **e20110422**. When converting, this date is used to select a code that was valid on that date.

CPTP
CPTProcedureModifier - Identifies the input value as a procedure modifier, such as in the 4010 837I SV202.03-06. The value is put in the array and preceded with **m**. Example: If the input value is 0A, the array cell will contain **m0A**.

PPC
Principal Procedure - Identifies the input value as the principal procedure code. The value is put in the array and preceded with **P**. Example: If the input value is E8000, the array cell will contain **PE8000**.

PC
Procedure - Identifies the input value as a procedure code. The value is put in the array and preceded with the letter **p**. Example: If the input value is E8000, the array cell will contain **pE8000**.

SEX
Sex - Identifies the input value as a gender code. The value is put in the array and preceded with the letter **x**. If the patient is female, the value will contain **xF**.

Please see Foresight Translator Example – Convert Translation with Selection Criteria on page 34.

Instream ICD Rules

Instream business rules for ICD codes are:

ICDConvertOne

This rule takes one ICD code as input and converts it to another ICD code, which it returns in a variable. It does not use any other selection criteria during the conversion.

Format of Parameters

RuleTable Function Option InputVar TypeofinputData OutputVar

Where:

RuleTable	Name of the rule file to use for the ICD code conversion. Example: `RuleFile_CG2014_D9_Convert.txt` Please see Rule File Names on page 23.
Function	`ConvertFirstOnly` is the best choice for ICDConvertOne rules. Please see Appendix C – Convert Functions on page 30.
Option	Use `Normal`.
InputVar	Source of the code to be converted: `Current_Element`, a literal in double quotes, or a variable.

TypeofinputData	Choose the kind of ICD code to be converted:
	`Diagnosis`
	`PrincipalDiagnosis`
	`Procedure`
	`PrincipalProcedure`

OutputVar	Variable to hold the output ICD code.

ICDConvert

ICDConvert uses values from an array to select one or more corresponding output ICD codes, which are returned in an array.

Format of Parameters

DiagnosisRuleTable ProcedureRuleTable Function Option Type InputArray OutputArray

Where:

DiagnosisRuleTable	Name of the table to be used for diagnostic code mapping. This field is ignored if you are not converting a diagnostic code. Example: `RuleFile_BL2014_D9_Convert.txt` Please see Rule File Names on page 23.
ProcedureRuleTable	Name of the table to be used for procedure code mapping. This field is ignored if you are not converting a procedure code. Example: `RuleFile_CG2014_P9_Convert.txt`
Function	How to convert. If you are converting only the code in cell 0,0, `ConvertFirstOnly` is a good choice. If you want to convert all the ICD codes in the input array, see Appendix C – Convert Functions on page 30.
Option	Use `normal`.

Type	Type of codes to convert:
	`Diagnosis`
	`Procedure`
	`Both`
	They can be used in the input array as selection criteria. Procedure codes of other lengths can be converted.

InputArray	Name of the array containing a code to be converted (in cell 0,0) plus other data to be used for conversion criteria (date of birth, sex, diagnosis codes, etc.). This array was created with ICDInsertToArrayWithType (see page 18) and can contain procedure and/or diagnostic codes plus other information to help Foresight Translator select the best ICD codes. Please see Input Array below.
OutputArray	Name of the array to hold the output ICD codes. The codes will be in the order specified under Function. Diagnosis codes will always be in row 0 and procedure codes will always be in row 1.

Input Array

Example input array:

	0	1	2	3	4	5
0	*Diagnosis or Procedure Code to convert* `D01484`	*Date of birth* `a8`	*Sex (M, F. or U)* `xF`	*Other Diagnosis or Procedure Code* `d01483`	*Other Diagnosis or Procedure Code* `d01484`	*Other Diagnosis or Procedure Code* `p0001`

The array must include one or more ICD codes and can contain any of the values shown on page 21.

The array is created with ICDInsertToArrayWithType (see page 18). This rule automatically adds the prefixes (the first letter in each cell) to the values. This lets the ICDConvert rule interpret the value regardless of the order in which they appear in the array.

Please see Dataswapper Example – ICDConvert Translation with Selection Criteria on page 38.

ICDInsertToArrayWithType

This rule inserts an ICD-related value into a cell in an array. The array can then be used as input for the ICDConvert rule (page 16).

Format of Parameters

ArrayName Row Column Value Type

Where:

ArrayName	Name of the array you are populating. This can be a variable or a literal in double quotes.
Row	This will always be 0.
Column	The column where the value is to be inserted. This can be a variable, a literal in double quotes, or one of the array reserved variables like **N** for next. The first column is column 0.
Value	Value to put in the cell. This can be a variable, a literal in quotes, or a reserved variable like `Current_Element`.
Type	A Dataswapper Identifier code representing the value type. (See Appendix A – Prefixes and Identifiers for Data Conversions on page 21 for additional explanation of the codes.) **DOB** DateOfBirth(D8) - Identifies the input value as the 8-digit date of birth. The value will be converted to an age, put in the array, and preceded with the letter **a**. For example, if the date of birth means the person is 53 years old, the array cell will contain **a53**. **PDC** PrincipalDiagnosis - Identifies the input value as the principal diagnosis code. The value will be put in the array and preceded with the capital letter **D**. For example, if the input code (ICD-9) is 5770, the array cell will contain **D5770**. **DC** Diagnosis - Identifies the input value as a diagnosis code other than the principal diagnosis code. The value will be put in the array and preceded with the letter **d**. For example, if the input code (ICD-9) is 5770, the array cell will contain **d5770**. **DOS** DateOfService/Discharge - Identifies the input value as the date of service or discharge date. The value will be put in the array as an 8-digit date and preceded with the letter **e**. For example, if the input date is 20110422, the array cell will contain **e20110422**. When converting, this date is used to select a code that was valid on that date.

CPTP
CPTProcedureModifier - Identifies the input value as a procedure modifier, such as in the 4010 837I SV202.03-06. The value will be put in the array and preceded with the letter **m**. For example, if the input value is 0A, the array cell will contain **m0A**.

PPC
Principal Procedure - Identifies the input value as the principal procedure code. The value will be put in the array and preceded with the capital letter **P**. For example, if the input value is E8000, the array cell will contain **PE8000**.

PC
Procedure - Identifies the input value as a procedure code. The value will be put in the array and preceded with the letter **p**. For example, if the input value is E8000, the array cell will contain **pE8000**.

SEX
Sex - Identifies the input value as a gender code. The value will be put in the array and preceded with the letter **x**. If the patient is female, the value will contain **xF**.

Please see Dataswapper Example – ICDConvert Translation with Selection Criteria on page 38.

GEN Records Created by ICD Rules

Each time a Convert or ConvertOne rule executes, the Instream validation detail file contains a GEN record with number 15016:

```
SBST         46|2300|HI|1|1|ABK||
GEN          4615016 1 0,D01484,A1832,a8,xF,     ←
SBST         46|2300|HI|1|2|A1832||
```

Where:

`46` = Line number

`15016` = Message number

`1` = Severity

`0` = Type

`D01484` = ICD value to be converted. It has a letter in front identifying what it represents:

D	Principal diagnostic code
d	Diagnostic code
P	Principal procedure code
p	Procedure code

(in this example, it is a D, so 01484 is a principal diagnosis code)

`A1832` = Value to replace 01484

`A8` = Selection criteria (in this case age 8)

`xF` = Selection criteria (in this case, Female)

Appendix A - Prefixes and Identifiers for Data Conversions

Example input array:

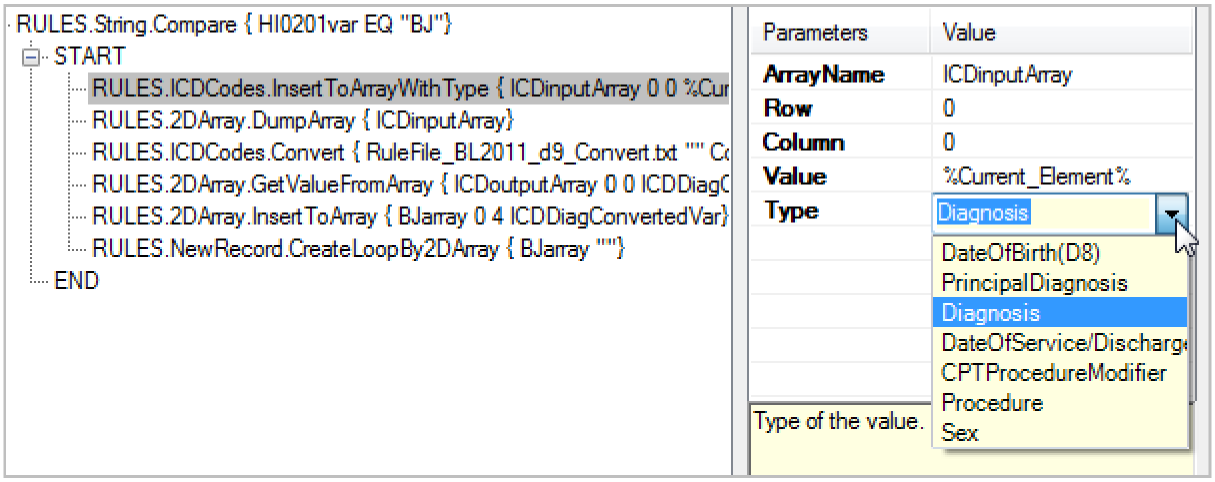
	0	1	2	3	4	5
0	*Diagnosis or Procedure Code to convert* `D01484`	*Date of birth (converted to age)* `a8`	*Sex (M, F. or U)* `xF`	*Other Diagnosis or Procedure Code* `d01483`	*Other Diagnosis or Procedure Code* `d01484`	*Other Diagnosis or Procedure Code* `p0001`

Each entry in an ICD input array has a one-letter prefix that identifies the data. The example array above contains **D01484** in cell 0,0. The prefix is D, meaning the value 01484 is a principal diagnosis code. Prefixes are listed in the chart below.

Data Prefixes for ICD Input Arrays

Data	Prefix	DataSwapper Identifier	Notes
CPT procedure modifier	m	CPTP	You may include any number of procedure modifier codes.
Diagnosis	d	DC	You may include any number of diagnosis codes.
Age in years	a	DOB	Automatically calculated from the patient's birth date.
Date of service or discharge in yyyymmdd format	e	DOS	Used to see if the code was valid on the date of service.
Procedure (CPT or ICD)	p	PC	You may include any number of procedure codes. Procedure codes of 5 characters are assumed to be CPT codes and are not converted, but they are used as selection criteria. Others are ICD-9 or ICD-10 procedure codes.
Principal diagnosis	D	PDC	You may include a principal diagnosis code.
Principal procedure	P	PPC	You may include a principal procedure code.
Sex M, F, U	s	SEX	Patient's gender.

To specify a value's prefix in Foresight Translator, use the **Type** field in the **InsertToArrayWithType** rule to identify the type of data. In this example, we are selecting Diagnosis so the prefix in the array will be **d**.



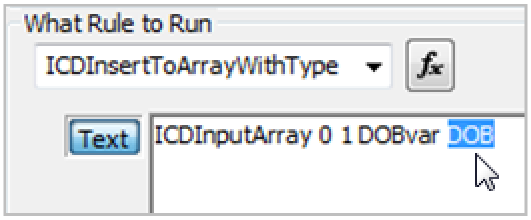
RULES.String.Compare { HI0201var EQ "BJ"}
START
- RULES.ICDCodes.InsertToArrayWithType { ICDinputArray 0 0 %Cur
- RULES.2DArray.DumpArray { ICDinputArray}
- RULES.ICDCodes.Convert { RuleFile_BL2011_d9_Convert.txt "" C
- RULES.2DArray.GetValueFromArray { ICDoutputArray 0 0 ICDDiag
- RULES.2DArray.InsertToArray { BJarray 0 4 ICDDiagConvertedVar}
- RULES.NewRecord.CreateLoopBy2DArray { BJarray ""}

END

Parameters	Value
ArrayName	ICDinputArray
Row	0
Column	0
Value	%Current_Element%
Type	Diagnosis

Type of the value.

To specify a value's prefix for a DataSwapper conversion, type its identifier as the last parameter in the ICDInsertToArrayWithType rule. This rule inserts the contents of DOBvar into an array and specifies that it is a DOB (date of birth), so the value will be prefixed with **a** when placed in the array:



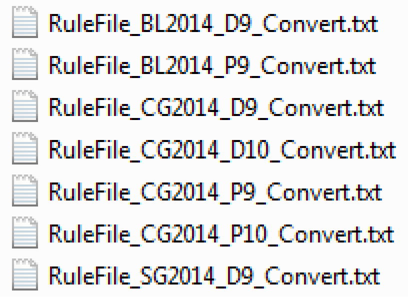
What Rule to Run: ICDInsertToArrayWithType

Text: ICDInputArray 0 1 DOBvar DOB

Appendix B - Rules Files and Functions

Rule File Names

Rules files use 2014 General Equivalence Mapping (GEM) and are located in the VIP directory under Foresight Translator or Instream:



RuleFile_BL2014_D9_Convert.txt
RuleFile_BL2014_P9_Convert.txt
RuleFile_CG2014_D9_Convert.txt
RuleFile_CG2014_D10_Convert.txt
RuleFile_CG2014_P9_Convert.txt
RuleFile_CG2014_P10_Convert.txt
RuleFile_SG2014_D9_Convert.txt

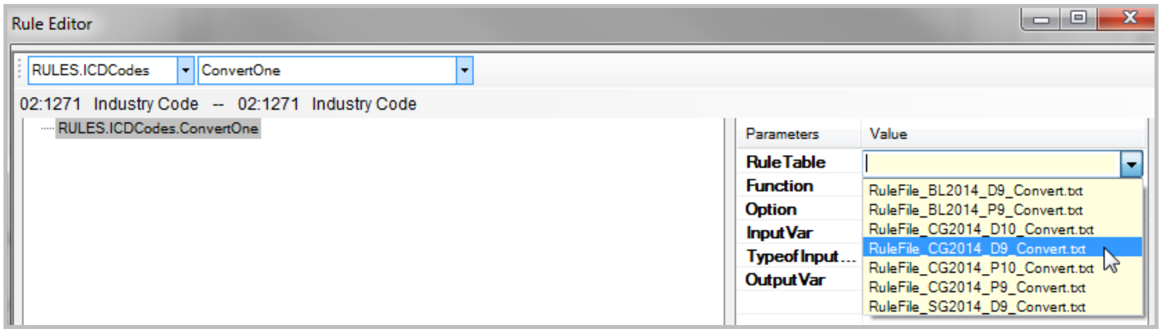
Rule file names have this format:

RuleFile_*RulesType*_*ICDinput*_**Convert.txt**

Where:

RuleFile_	Literal text.	
RulesType	One of these, followed by the year when the rules were established:	
	BL	Baseline rules, proprietary content that uses additional intelligence to determine the best match for conversions.
	CG	Combined GEM mapping that matches based on knowledge gained analyzing both forward and reverse GEM mappings.
	SG	Straight GEM mapping.
ICDinput	One of these:	
	d9	Converts an ICD-9 diagnostic code to an ICD-10 diagnostic code.
	d10	Converts an ICD-10 diagnostic code to an ICD-9 diagnostic code.
	p9	Converts an ICD-9 procedure code to an ICD-10 procedure code.
	p10	Converts an ICD-10 procedure code to an ICD-9 procedure code.
_Convert.txt	Literal text.	

In Foresight Translator, the center part of the filename is used in the business rule. This rule will use RuleFile_CG2014_D9_Convert.txt:


Rule Editor

RULES.ICDCodes ConvertOne

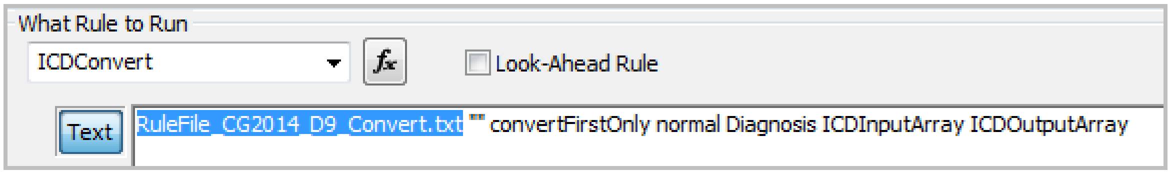
02:1271 Industry Code -- 02:1271 Industry Code

RULES.ICDCodes.ConvertOne

Parameters	Value
RuleTable	
Function	
Option	
InputVar	
TypeofInput...	
OutputVar	

RuleFile_BL2014_D9_Convert.txt
RuleFile_BL2014_P9_Convert.txt
RuleFile_CG2014_D10_Convert.txt
RuleFile_CG2014_D9_Convert.txt
RuleFile_CG2014_P10_Convert.txt
RuleFile_CG2014_P9_Convert.txt
RuleFile_SG2014_D9_Convert.txt

For DataSwapper, type the whole rule file name, like this:


What Rule to Run

ICDConvert

☐ Look-Ahead Rule

Text: RuleFile_CG2014_D9_Convert.txt "" convertFirstOnly normal Diagnosis ICDInputArray ICDOutputArray

Examples:

- **RuleFile_CG2014_D9_Convert.txt** means:

CG2014	The conversion will be done according to the rules of the 2014 General Equivalence Mapping.
D9	ICD-9 procedure codes are being converted to ICD-10.

- **RuleFile_BL2014_D9_Convert.txt** in the name means:

BL2014	The conversion will be done according to the rules of the 2014 Baseline.
D9	ICD-9 diagnosis codes are being converted to ICD-10.

Rule Files and Selection Criteria

If selection criteria is not included (as in the ConvertOne rule, for example), a default match is returned. In this case:

- Diagnosis 9 to 10 should provide an accurate returned code. RuleFile_BL2014_D9 provides the most accurate default for Diagnosis 9 to 10.
- Procedure 9 to 10 conversions may produce a misleadingly precise ICD-10 code because procedure ICD10 codes generally have no “unspecified” codes.
- Diagnosis or procedure 10 to 9 conversions will return the CMS published “reimbursement” code(s).

Conversion	Rule File to Use	Recommended Selection Criteria
Diagnosis 9 to 10	RuleFile_BL2014_D9_Convert - gives the most accurate converted default value RuleFile_CG2014_D9_Convert - contains only the Combined GEM rules	All ICD-9 diagnosis and ICD-9 or CPT procedure codes and their modifiers Age, sex, and date of service Other ICD-9 diagnostic codes
Diagnosis 10 to 9	RuleFile_CG2014_D10_Convert	Other ICD-10 diagnostic codes Date of service
Procedure 9 to 10	RuleFile_CG2014_P9_Convert	Other ICD-9 procedure codes Date of service
Procedure 10 to 9	RuleFile_CG2014_P10_Convert	Other ICD-10 procedure codes Date of service

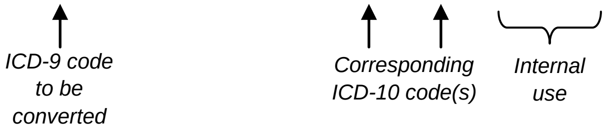
Rule File Technical Details

Note: The examples given in the following sections are illustrative only. Actual content of the rules files will change as updated versions are released.

Basic Rule File Format

Each line in a rule file contains a possible choice for conversion. This section of the CG2014_D9 rule file is typical. It shows how the ICD-9 diagnosis codes 24911 and 24930 will be converted:

```
24911|1||||||||||||||||E0810|E0865|||Y|N|1|N
24911|2||||||||||||||||E0910|E0865|||Y|N|1|N
24911|3||||||||||||||||E0810||||Y|N|1|E
.
.
.
24930|1||||||||||||||||E0811||||Y|N|1|B
24930|2||||||||||||||||E08641||||Y|N|1|B
24930|3||||||||||||||||E0911||||Y|N|1|B
24930|4||||||||||||||||E09641||||Y|N|1|B
24930|5||||||||||||||||E1311||||Y|N|1|B
24930|6||||||||||||||||E13641||||Y|N|1|B
```



- The first column is the code to be converted.
- The second column is the rule number for that code.
- The next values shown in the example above are possible ICD-10 codes.

  If multiple ICD-10 codes appear in one line, the first one is generally the best match.
- The last four values are for internal use.

Selection Criteria in Rule Files

Rule file columns may include selection criteria such as age, sex, and other values. The prefixes do not appear in the rule files, but the meaning is determined by column.

All selection criteria must be included in the Convert rule, and must match the rule file line, in order for the line to be selected.

Here are a few lines out of the many for code 8788.

```
8788|1|E9060||||||||76|||||M||S31551D||||Y|N|1|A
8788|2|E9063||||||||76|||||M||S31551D||||Y|N|1|A
8788|3|E9064||||||||76|||||M||S31551D||||Y|N|1|A
8788|4|E9283||||||||76|||||M||S31551D||||Y|N|1|A
8788|5|E9060||||||||76|||||F||S31552D||||Y|N|1|A
8788|6|E9063||||||||76|||||F||S31552D||||Y|N|1|A
8788|7|E9064||||||||76|||||F||S31552D||||Y|N|1|A
8788|8|E9283||||||||76|||||F||S31552D||||Y|N|1|A
8788|9|||||||||76|||||M||S31501D||||Y|N|3|A
8788|10|||||||||76|||||F||S31502D||||Y|N|1|A
.
.
.
07021|2|||||||||||||||<20101001|B1911||||I|N| |Y
```

Selection criteria is shown in bold above, and includes:

- An additional ICD-9 code in column 3.
- A CPT modifier of `76`.
- Sex – `M` or `F`.
- Birth date – `<20101001` means birth date must be less than Oct. 1, 2010.

How this example's selection criteria is used in converting codes:

- The additional ICD-9 codes in column 3 help to match certain rule numbers, and therefore help determine which ICD-10 codes to return.

  Only rule 7 will be matched if the input array contains:

  D8788,a61,x**F**,d**E9064**

  8788|7|**E9064**||||||||76|||||**F**||S31552D||||Y|N|1|A

  The age of 61 (`a61`) in the input array is not a consideration in matching to particular rules, since the rule file does not provide date of birth information for ICD-9 code 8788.

- The 76 is a CPT modifier. If sent in the input array, modifiers help to match to particular rules.

  However, all the lines shown above have a modifier of 76, so that will not actually help in matching unless the input array contains a different modifier. For example, if the input array contains `D8788,m77,xF,dE9064`, then no rule shown above is matched.

- The M or F is the patient's sex. If the input array contains `D8788,xF`, then rules 5, 6, 7, 8, 10, and 11 are matched.
- The date of birth, supplied in the input with prefix **a**, can be a selection criteria in a rule file line, as in this example:

  `3596|6|||||||||||||`**`<18`**`|||M3302||||Y|N|1|A`

  ↑

How Convert uses Matched Rules

Since codes may have multiple lines in the rule file, the input array data may match to one rule, multiple rules, or no rules.

How the Convert rule uses a match:

If the input matches …	Convert uses these RuleFile lines …
One line	It returns the corresponding codes in that one line according to the way the chosen function works.
Multiple lines	**ConvertFirstOnly**: Converts: Only the first ICD-9 code in the input array Returns: All ICD-10 codes From: The first matched rule for the input code **ConvertNoDups**: Converts: All ICD-9 codes in the input array Returns: All ICD-10 codes, with duplicates omitted From: The first matched rule for each input code **ConvertPreserveOrder**: Converts: All ICD-9 codes in the input array Returns: All ICD-10 codes From: The first matched rule for each input code **Note:** Use of this function is not recommended. It may return multiple codes and result in running out of space. **ConvertAllTargets**: Converts: All ICD-9 codes in the input array Returns: All ICD-10 codes From: All matching rules for each input code
No lines	No converted codes will be returned. “NORULES” will appear in the output.

When Multiple Codes are Returned

In some cases, two or more ICD_10 codes may be returned for one ICD-9 code. This is controlled by the rule files and not by the Convert or ConvertOne rule. The modified rule file will return the first ICD-10 code in the line.

Example:

```
80616|2||||||||||||||||S12401B|S14115A|||Y|N|1|N
```

Two codes match 80616 when there is no additional selection criteria provided. Both will be returned.

To prevent this, you have two options:

A. Remove the unwanted code from the line, like this:

   ```
   80616|2||||||||||||||||S12401B||||Y|N|1|N
   ```

   You can request modified rule files that will only return one code.

B. Write business rule that check for a space, substring the value before the space, and then use that value in the output.

When a Code has no Match

No equivalent

Some ICD-9 procedure codes do not have an equivalent ICD-10 code. In that case, the value NOPCS is returned instead of a code.

Bad ICD-9 code

Some ICD codes in the input may be invalid. In that case, the value NORULES is returned instead of a code.

Appendix C - Convert Functions

The function used in the Convert rule determines:

- What codes are actually converted from the input string
- What codes that match the input code are actually returned.

Convert returns converted codes as follows.

The output array will not contain prefixes. Row 0 will contain converted diagnostic codes and row 1 will contain converted procedure codes. Cell 0,0 or 0,1 always contains the conversion of the code in the input array cell 0,0. If multiple codes are returned, they are placed in separate cells of the output array.

Function	Input Code Converted	Rows used from Rule File	Codes Returned
ConvertFirstOnly	First code in input array; others are used as selection criteria or ignored	First matching row	All output codes from the first matched row.
ConvertNoDups	All unique codes in input array	First matched row for each input code	All unique codes that match the input codes. Duplicates output codes are omitted.
ConvertPreserveOrder	All codes in input array	First matched row for each input code	All codes that match the input codes, in the same order in which they were input. Duplicate output codes are included if they correspond to different input codes. **Note:** Use of this function is not recommended. It may return multiple codes and result in running out of space.
ConvertAllTargets	All codes in input array	All matched rows for each input code	Convert all codes to all defined mappings, and preserve input order. Duplicate output codes are included if they correspond to different input codes. This call is useful for generating test data.

Example Convert Results

Pertinent lines in the rule file:

```
24911|1||||||||||||||||E0810|E0865|||Y|N|1|N
24911|2||||||||||||||||E0865|E0910|||Y|N|1|N
24911|3||||||||||||||||E0865||||Y|N|1|E
.
.
.
24930|1|24930|||||||||||||||E08641||||Y|N|1|T
24930|2||||||||||||||||E0811||||Y|N|1|B
24930|3||||||||||||||||E08641||||Y|N|1|B
24930|4||||||||||||||||E0911||||Y|N|1|B
24930|5||||||||||||||||E09641||||Y|N|1|B
```

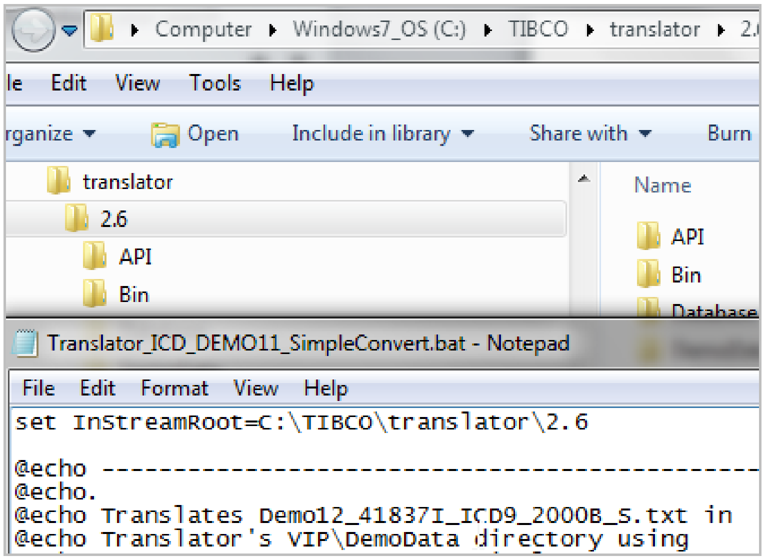
Function	**Result when using Convert function**	**Codes returned from example above**
Assume input array contains `D24911, a61, xF, d24930, d24911` *Principal diagnosis ICD-9 code* *Age 62* *Female* *ICD-9 code* *ICD-9 code (Same as Principal diag. code)*		
ConvertFirstOnly	Output array row 0: `DE0810,DE0865` This example only converted the first input code (24911). The other input ICD-9 codes (24930 and the other 24911) were ignored. It used the first row in the rule file for 24911 and returned both values from that line, separated by a space. Each is preceded with D for principal diagnosis code.	
ConvertNoDups	Output array row 0: `DE0810,DE0865` This example converted all unique codes in the input (24911 and 24930). It used rule 1 for code 24911 and rule 1 for 24930, and returned all values from those two rules. Because it had two input values of 24911, there were duplicate values and those were omitted from the returned codes.	
ConvertPreserveOrder	Output array row 0: `DE0810,DE0865` This example converted each input code. It used rule 1 for code 24911, rule 1 for 24930, and rule 1 for the second code 24911, and returned all values from these rules. There were duplicates in the returned codes.	
ConvertAllTargets	Output array row 0: `DE0810,DE0865,DE0910` This example used rules 1 and 2 for 24911, and all 5 rules for 24930. Duplicates were removed.	

Appendix D - Demos and Extended Examples

Foresight Translator Demo

Map	Demo_ICD_9to10_837I_Convert.map
Guidelines	Source 837AQ320.std Target 837-X223.std
Script	Translator_ICD_demo.bat (In VIP directory) Update the top line as shown on page 32.
Rules in map	Convert InsertToArrayWithType
EDI File to Translate	Demo12_41837I_ICD9_2000B_S.txt (In VIP/DemoData directory)

Update the path in the top line of the scripts before executing them. The InstreamRoot should point to the directory that contains Foresight Translator 's Bin directory. Example:



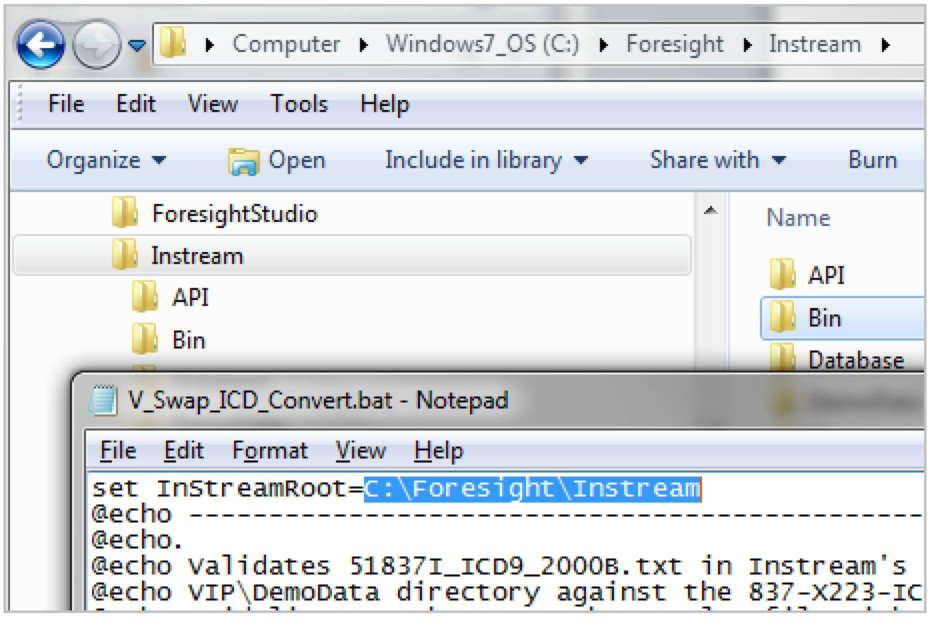
```
set InStreamRoot=C:\TIBCO\translator\2.6

@echo ----------------------------------------------
@echo.
@echo Translates Demo12_41837I_ICD9_2000B_S.txt in
@echo Translator's VIP\DemoData directory using
```

Dataswapper Demo

Guidelines	837-X223-ICD.std (5010)
Script	InStream_ICD_demo.bat (In VIP directory)
Rules in Guideline	Convert InsertToArrayWithType
EDI File to Process	51837I_ICD10_HL22_Clean.txt (In VIP/DemoData directory)

Update the path in the top line of the scripts before executing them. The InstreamRoot should point to the directory that contains Instream's Bin directory. Example:


Computer ▸ Windows7_OS (C:) ▸ Foresight ▸ Instream ▸
File Edit View Tools Help
Organize ▾ Open Include in library ▾ Share with ▾ Burn
ForesightStudio
Instream
API
Bin
Name
API
Bin
Database
V_Swap_ICD_Convert.bat - Notepad
File Edit Format View Help

```
set InStreamRoot=C:\Foresight\Instream
@echo ---------------------------------------------
@echo.
@echo Validates 51837I_ICD9_2000B.txt in Instream's
@echo VIP\DemoData directory against the 837-X223-IC
```

Foresight Translator Example – Convert Translation with Selection Criteria

Demo This example is in map Demo_ICD_9to10_837I_Convert.map, found in Foresight Translator's Database directory. For simplicity, the map only contains the rules shown below and not the other rules necessary for a full conversion of 4010 to 5010 data. For a full step-up map, contact TIBCO Foresight Support.

You can test by translating 4010 EDI file Demo12_41837I_ICD9_2000B_S.txt in Foresight Translator's VIP\DemoData directory.

This example uses Convert to map the first two composites in the HI segment for Principal, Admitting, E-Code and Patient Reason For Visit Diagnosis Information. It determines the new principal diagnosis code based on the subscriber's date of birth, sex, and all diagnosis codes in the Principal Diagnosis Code HI segment.

Collecting Date of Birth and Sex

1. Collect the subscriber's date of birth and gender.

   DMG02

   ```
   RULES.Variable.Set { %Current_Element% DOBvar}
   ```

   DMG03

   ```
   RULES.Variable.Set { %Current_Element% SEXvar}
   ```

2. Set up an array for selection criteria.

   We now set up an array to contain all selection criteria that might be used to convert the ICD-9 codes. These selection criteria are further down in the claim, so we find the values with Search rules.

   We leave the 0,0 cell empty. Just before converting each ICD-9 code, we will put that code in cell 0,0 and then execute a Convert rule.

   CLM01

   Delete the selection criteria array if it exists, and then start a new one, adding the date of birth and the sex to the array. Leave 0,0 empty:

   ```
   RULES.2DArray.Delete { ICDinputArray}
   RULES.ICDCodes.InsertToArrayWithType { ICDinputArray 0 1 DOBvar DateOfBirth(D8)}
   RULES.ICDCodes.InsertToArrayWithType { ICDinputArray 0 2 SEXvar Sex}
   ```

   ICDinputArray now contains this:

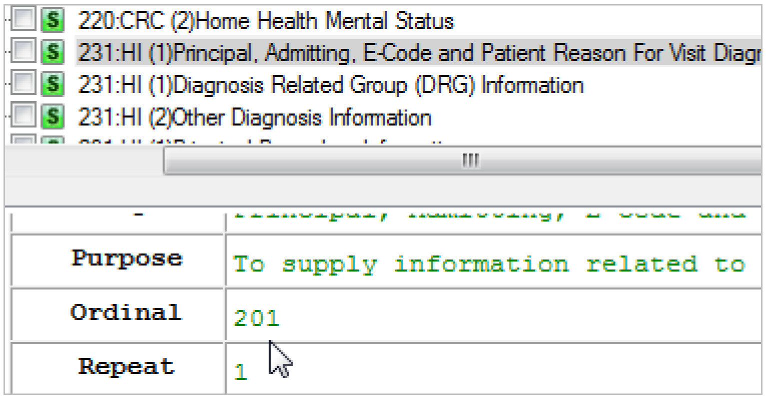
	Column 0	Column 1	Column 2
Row 0		**Sex**	**Date of birth**

Search forward to find the codes in the HI and SV2 segments.

CLM01

These rules all go on the CLM01.

1. Determine the ordinal number for the relevant HI segment:



☐ S 220:CRC (2)Home Health Mental Status
☐ S 231:HI (1)Principal, Admitting, E-Code and Patient Reason For Visit Diagr
☐ S 231:HI (1)Diagnosis Related Group (DRG) Information
☐ S 231:HI (2)Other Diagnosis Information

Purpose	To supply information related to
Ordinal	201
Repeat	1

Do a forward search for the value in the HI-01-02 and put the value into variable HI0102var. If there is actually a value in the variable, put it in the next cell of the array and label it as a principal diagnosis code.

```
RULES.Search.Element { "HI" 201 1-2 HI0102var}]
RULES.String.HasValue { HI0102var}
   START
      RULES.ICDCodes.InsertToArrayWithType { ICDinputArray 0 N HI0102var
      PrincipalDiagnosis}
   END
```

2. Do the same for the value in the HI-02-02

```
RULES.Search.Element { "HI" 201 2-2 HI0202var}]
RULES.String.HasValue { HI0202var}
   START
      RULES.ICDCodes.InsertToArrayWithType { ICDinputArray 0 N HI0202var
      Diagnosis}
   END
```

3. Do another forward search to see if there is a SV2 segment at ordinal 273. If so, we put it in SV2array and put a 1 in SV2foundVar. Check SV2foundVar to see if the SV2 existed; if so, put the number of rows (SV2 segments) it contains into variable SV2rowCount.

4. Then execute a loop that checks each array row to see if the subelement at SV2-02-01 contains HC. If so, we get the next value and put it in the next column of the array.

```
RULES.Search.SegmentsInLoop { "SV2" 273 SV2foundVar SV2array}]
RULES.String.Compare { SV2foundVar EQ "1"}
   START
      RULES.2DArray.GetRowCounter { SV2array SV2rowCount}
      RULES.Counter.Reset { CurrentCounter 0}
      RULES.String.Loop { "CurrentCounter LT SV2rowCount"}
         START
            RULES.2DArray.GetSubElementValue { SV2array CurrentCounter 4 0
            SV2qualVar}
            RULES.String.Compare { SV2qualVar EQ "HC"}
               START
                  RULES.2DArray.GetSubElementValue { SV2array
                  CurrentCounter 4 1 SV2procCodeVar}
                  RULES.ICDCodes.InsertToArrayWithType { ICDinputArray 0 N
                  SV2procCodeVar Procedure}
               END
            RULES.Counter.Increment { CurrentCounter 1}
         END
   END
```

ICDinputArray might now contain something like this:

	Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
Row 0		*Sex*	*Date of birth*	*Principal Diagnosis*	*Diagnosis*	*Procedure Code*	*Procedure Code*	*Procedure Code*

Perform Conversions

HI-01-01 (Principal, Admitting, E-Code and Patient Reason For Visit Diagnosis Information)

Convert the qualifier from BK to ABK.

```
RULES.Value.ReplaceWith { "ABK"}
```

HI-01-02

Insert the principal diagnosis code into first cell of the array. This cell is reserved for the code that is to be converted.

```
RULES.ICDCodes.InsertToArrayWithType { ICDinputArray 0 0
%Current_Element% PrincipalDiagnosis}
```

ICDinputArray might contain:

	Column 0	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7
Row 0	***Principal Diagnosis***	*Sex*	*Date of birth*	*Principal Diagnosis*	*Diagnosis*	*Procedure Code*	*Procedure Code*	*Procedure Code*

With the array fully stocked, we use it to convert the value in cell 0,0, putting it into ICDoutputArray. Since this is a diagnosis code, it will come back in row 0 of the output array.

```
RULES.ICDCodes.Convert { RuleFile_BL2011_d9_Convert.txt ""
ConvertFirstOnly normal Diagnosis ICDinputArray [ICDoutputArray}
```

We get the value from this cell and put it in a variable. We then use it in the target data.

```
RULES.2DArray.GetValueFromArray { ICDoutputArray 0 0
ICDprincipalDiagConvertedVar}

RULES.Value.ReplaceWith { ICDprincipalDiagConvertedVar}
```

HI02-01

If the HI-02-01 contains BJ, we start making an HI segment in an array called BJarray. We will use this later to create a separate HI segment to conform to 5010. We also capture the value in a variable for use on the next element.

```
RULES.String.Compare { %Current_Element% EQ "BJ"}
START
   RULES.2DArray.InsertToArray { BJarray 0 0 "HI"}
   RULES.2DArray.InsertToArray { BJarray 0 1 %FS_ELM_SEP%}
   RULES.2DArray.InsertToArray { BJarray 0 2 "ABJ"}
   RULES.2DArray.InsertToArray { BJarray 0 3 %FS_COMP_SEP%}
END
RULES.Variable.Set { %Current_Element% HI0201var}
```

BJarray might contain:

	Column 0	Column 1	Column 2	Column 3
Row 0	HI	*	ABJ	:

HI-02-02

Insert another diagnosis codes from the same segment into the input array to convert, convert the diagnosis code to ICD10, add it to the segment we are assembling in BJarray, and then write the new HI record to the target:

```
RULES.String.Compare { HI0201var EQ "BJ"}
START
   RULES.ICDCodes.InsertToArrayWithType { ICDinputArray 0 0
   %Current_Element% Diagnosis}

   RULES.ICDCodes.Convert { RuleFile_BL2011_d9_Convert.txt ""
   ConvertFirstOnly normal Diagnosis ICDinputArray
   ICDoutputArray}

   RULES.2DArray.GetValueFromArray { ICDoutputArray 0 0
   ICDDiagConvertedVar}

   RULES.2DArray.InsertToArray { BJarray 0 4 ICDDiagConvertedVar}

   RULES.NewRecord.CreateLoopBy2DArray { BJarray ""}
END
```

Dataswapper Example - ICDConvert Translation with Selection Criteria

These rules are in guideline 837-X223-ICDconvert. To perform the swap, execute InStream_ICD_demo.bat in the VIP directory.

This example uses lookahead rules as described in **BusinessRules.pdf**.

It uses an array to capture information needed during conversion of diagnosis and procedure codes.

Rules in Order of Execution

DMG02

Capture the subscriber's birth date from the DMG02.

```
BusinessRules.Variable.SetVar DOBvar
```

DMG03

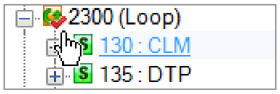
Do the same for the subscriber's gender at DMG03:

```
BusinessRules.Variable.SetVar SEXvar
```

Lookahead Rules

Before doing any conversions, we are going to assemble all data that we want to consider when converting diagnosis and procedure codes. This data varies according to the contents of a particular CLM loop, so we mark a lookahead range of the entire 2300 loop.

To mark the lookahead range, we right-click on the 2300 loop header and select **DSR mark**. The loop header will then have a red check mark:



2300 (Loop)
130 : CLM
135 : DTP

CLM

On the claim segment, we begin setting up an array to hold selection criteria for the conversion. We clear the array and then insert the date of birth and sex into cells 0,1 and 0,2. The DOB and SEX at the end of the rule are indicators.

```
BusinessRules.Lookahead.ClearArray ICDInputArray
BusinessRules.Lookahead.ICDInsertToArrayWithType ICDInputArray 0 1 DOBvar DOB
BusinessRules.Lookahead.ICDInsertToArrayWithType ICDInputArray 0 2 SEXvar SEX
```

The array now contains:

0	1	2	3	4	5	6	7	8	9	10
	DOB	**SEX**								

HI Principal Diagnosis

01.02

We add the principal diagnosis code to the next cell in the array so that it can be used as selection criteria.

```
Always call External Routine BusinessRules.Lookahead.ICDInsertToArrayWithType
ICDInputArray 0 N Current_Element PDC
```

The array now contains:

0	1	2	3	4	5	6	7	8	9	10
	DOB	SEX	**PDC**							

HI Admitting Diagnosis

01.01

Put this value in a variable to be used in the next rule.

```
BusinessRules.Lookahead.SetVar 2300BHI01_0101
```

01.02

```
BusinessRules.Lookahead.CompareString 2300BHI01_0101 EQ "BJ"
(BusinessRules.ICDCodes ICDInsertToArrayWithType ICDInputArray 0 N
Current_Element DC)
```

0	1	2	3	4	5	6	7	8	9	10
	DOB	SEX	PDC	**DC**						

HI Other Diagnosis Information

01.01

```
BusinessRules.Lookahead.SetVar "2300BHI01_0101"
```

01.02

```
BusinessRules.Lookahead.ICDInsertToArrayWithType ICDInputArray 0 N
Current_Element DC
```

0	1	2	3	4	5	6	7	8	9	10
	DOB	SEX	PDC	DC	**DC**					

HI Principal Procedure Information

01.01

```
BusinessRules.Lookahead.SetVar HI02_0102_P
```

01.02

```
BusinessRules.Lookahead.CompareString HI02_0102_P EQ "BP"
(BusinessRules.ICDCodes ICDInsertToArrayWithType ICDInputArray 0 N
Current_Element PC)
```

0	1	2	3	4	5	6	7	8	9	10
	DOB	SEX	PDC	DC	DC	**PC**				

HI Other Procedure Information

01.01

```
BusinessRules.Lookahead.SetVar HI03_0102_P
```

01.02

```
BusinessRules.Lookahead.CompareString HI03_0102_P EQ "BO"
(BusinessRules.ICDCodes ICDInsertToArrayWithType ICDInputArray 0 N
Current_Element PC)
```

0	1	2	3	4	5	6	7	8	9	10
	DOB	SEX	PDC	DC	DC	PC	**PC**			

SV2 Institutional Service Line

02.01

```
Lookahead.SetVar 2400_HC
```

02.02

```
BusinessRules.Lookahead.CompareString 2400_HC EQ "HC" (BusinessRules.ICDCodes
ICDInsertToArrayWithType ICDInputArray 0 N Current_Element PC)
```

02.03

```
Routine BusinessRules.Lookahead.CompareString 2400_HC EQ "HC"
(BusinessRules.ICDCodes ICDInsertToArrayWithType ICDInputArray 0 N
Current_Element CPTP)
```

0	1	2	3	4	5	6	7	8	9	10
	DOB	SEX	PDC	DC	DC	PC	PC	**PC**	**CPTP**	

DPT Service Date

03

```
BusinessRules.Lookahead.ICDInsertToArrayWithType ICDInputArray 0 N
Current_Element DOS
```

	0	1	2	3	4	5	6	7	8	9	10
0		DOB	SEX	PDC	DC	DC	PC	PC	PC	CPTP	**DOS**

Non-Lookahead Rules

We now have all selection criteria for this claim in ICDInputArray. We can go back to the top of the lookahead range and start executing the non-lookahead rules.

HI Principal Diagnosis

01.01

```
BusinessRules.Substitute.Substitute "ABK"
```

01.02

Put the primary diagnosis code in cell 0,0 of the input array so that it can be converted:

```
BusinessRules.ICD.ICDInsertToArrayWithType ICDInputArray 0 0 Current_Element
PDC
```

	0	1	2	3	4	5	6	7	8	9	10
0	**PDC**	DOB	SEX	PDC	DC	DC	PC	PC	PC	CPTP	DOS

Convert the primary diagnosis code and put the converted code in cell 0,0 of ICDOutputArray:

```
BusinessRules.ICDCodes.ICDConvert RuleFile_BL2014_D9_Convert.txt""
convertFirstOnly normal Diagnosis ICDInputArray ICDOutputArray
```

ICDOutputArray contains:

	0
0	**PDC**

Get the converted code from ICDOutputArray and put it in a variable HI0102_PDC10Var:

```
BusinessRules.ICDCodes.GetVarFromArray ICDOutputArray 0 0 HI0102_PDC10Var
```

Substitute the converted code for the Principal Diagnosis Code:

```
BusinessRules.Substitute.Substitute HI0102_PDC10Var
```

HI Admitting Diagnosis

01.01

```
Always call External Routine BusinessRules.Substitute.Substitute "ABJ"
```

01.02

This is similar to the strategy for the principal diagnosis code. We put the admitting diagnosis in cell 0,0 so that it can be converted and substituted in the output EDI.

```
BusinessRules.ICDCodes.ICDInsertToArrayWithType ICDInputArray 0 0
Current_Element DC
```

```
BusinessRules.ICDCodes.ICDConvert RuleFile_BL2014_D9_Convert.txt""
convertFirstOnly normal Diagnosis ICDInputArray ICDOutputArray
```

```
BusinessRules.ICDCodes.GetVarFromArray ICDOutputArray 0 0 HI0202_PDC10Var
```

```
BusinessRules.Substitute.Substitute HI0202_PDC10Var
```

HI Other Diagnosis Information

01.01

```
Always call External Routine BusinessRules.Substitute.Substitute "ABF"
```

01.02

Now convert this diagnosis code and substitute it in the output EDI.

```
BusinessRules.ICDCodes.ICDInsertToArrayWithType ICDInputArray 0 0
Current_Element DC
```

```
BusinessRules.ICDCodes.ICDConvert RuleFile_BL2014_D9_Convert.txt""
convertFirstOnly normal Diagnosis ICDInputArray ICDOutputArray
```

```
BusinessRules.ICDCodes.GetVarFromArray ICDOutputArray 0 0 HI0302_PDC10Var
```

```
BusinessRules.Substitute.Substitute HI0302_PDC10Var
```

HI Principal Procedure Information

01.01

```
Always call External Routine BusinessRules.Substitute.Substitute "BBP"
```

01.02

Convert the procedure code and put it in the output EDI. Notice that we are using a different rule file that converts procedure codes.

```
BusinessRules.ICDCodes.ICDInsertToArrayWithType ICDInputArray 0 0
Current_Element PC
```

```
BusinessRules.ICDCodes.ICDConvert BL2014_D9 RuleFile_BL2014_P9_Convert.txt
convertFirstOnly normal Procedure ICDInputArray ICDOutputArray
```

Since this is a procedure code, it comes back in row 1 of ICDOutputArray:

	0
0	
	PC

```
BusinessRules.ICDCodes.GetVarFromArray ICDOutputArray 1 0 HI0402_PDC10Var
```

```
BusinessRules.Substitute.Substitute HI0402_PDC10Var
```

HI Other Procedure Information

01.01

```
Always call External Routine BusinessRules.Substitute.Substitute "BBO"
```

01.02

```
BusinessRules.ICDCodes.ICDInsertToArrayWithType ICDInputArray 0 0
Current_Element PC
```

```
BusinessRules.ICDCodes.ICDConvert BL2014_D9 RuleFile_BL2014_P9_Convert.txt
convertFirstOnly normal Procedure ICDInputArray ICDOutputArray
```

```
BusinessRules.ICDCodes.GetVarFromArray ICDOutputArray 1 0 HI0502_PDC10Var
```

```
BusinessRules.Substitute.Substitute HI0502_PDC10Var
```

Appendix E - Location of ICD Codes in 5010 HIPAA Guidelines

Transaction	**Location**
270	2100C and 2100D HI segment
271	2100C and 2100D HI segment
278 RQ	2000E HI segment
278 RS	2000E HI segment
837I	2300 HI segments: Principal Diagnosis Admitting Diagnosis Patient Reason for Visit External Cause of Injury Other Diagnosis Information Principal Procedure Information Other Procedure Information
837P	2300 HI segment – Health Care Diagnosis Code
837D	2300 HI segment – Health Care Diagnosis Code