TIBCO Foresight® Instream®

Instream Validation Technical Manual

Software Release 8.7
August 2017

Two-Second Advantage®

Important Information

General Contact Information

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

Technical Support

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

Contents

1 Introduction

Intended Audience

This document is intended for technical staff implementing TIBCO Foresight® Instream® validation. A general familiarity with EDI is assumed.

System Requirements

Instream is available for Windows or UNIX platforms and requires the following:

- Java JDK (if using the Java API or ReportMailer)
- SMTP server (if using ReportMailer functionality)

For a complete list of supported operating systems and hardware/memory requirements see the Instream® Readme file.

Recommended Software

Recommended software for Instream sites:

- **TIBCO Foresight® EDISIM® 6.6 or later.**

  This will let you develop your own business rules for use with Instream.

- **TIBCO Foresight® HIPAA Validator® Desktop**

  This will let you view errors, the corresponding EDI, and guideline information in a graphical user interface. Look for it under **Start | Programs | Foresight | Desktop**.

Checking Instream's Version

Execute the **Version** file in Instream's Scripts directory.

License File

Beginning with Instream version 8.0.0, a license file is no longer required.

Capabilities

The validation module of Instream is a command line validator for many different file formats, including (but not limited to) HIPAA (health care) data, other X12 formats, EDIFACT, TRADACOMS, HL7, flat files, and XML.

Please see **FileFormatsAtForesight.pdf** for a complete list.

For UNICODE handling information, see **UNICODE_at_Foresight.pdf**.

You can add your own edits to these formats with the help of EDISIM®, TIBCO Foresight's guideline authoring tool.

This document describes validation with Instream. The other Instream modules are documented separately.

Delimiters and Separators

Instream can validate data with separators and delimiters that are keyboard characters or ASCII up to hex equivalent 252.

Command Line Overview

If you start Instream from a command line, you will take data files as input and write result files. See page 24 for details about starting from the command line.

Your calling software can scan the resulting summary report to determine if the data passed validation.

If your software determines that the data failed validation, the calling software can:

- Use the detail results file to generate e-mails or response documents. The detail results file is consistently formatted and contains information needed to fill out all optional and mandatory elements.
- Call Instream's Docsplitter module to split good data from bad, and/or call Instream's Response Generator module to generate a response for the sender.

API Overview

You can integrate Instream validation with another software package by using Instream's C++, C#, VB.NET, or Java API.

Please see **TIB_fsp-instream_<*n.n*>_api.pdf**.

Customizing for Specific Partners

Instream validation lets you customize for specific partners by:

- Using different TIBCO Foresight-supplied guidelines to strengthen or weaken compliance checking.
- Using guidelines with partner-specific edits that you define with EDISIM, the TIBCO Foresight guideline-authoring tool.
- Customizing diagnostic messages.
- Adjusting error severity levels for a partner or group of partners. This controls the conditions that cause a data to be rejected and the response generated after rejection.
- Creating and enforcing new external code lists.
- Amending the contents of TIBCO Foresight-supplied external code lists.

Trading Partner Automation: To automatically select a guideline or profile based on some information in the EDI data, see **TIB_fsp-instream_<*n.n*>_tpa.pdf** and **APF.pdf**.

Customizing for Specific Guidelines

You can save the profile you want to use for a specific standard or guideline and have Instream automatically load it whenever you use that standard for validation. See **Specifying which APF File to Use** in **APF.pdf**.

CCI Edits

Instream will enforce:

- Correct Coding Initiatives (CCI) for Part B Medicare Carriers from the Centers for Medicare and Medicaid Services (CMS).
- National Correct Coding Initiative (NCCI) edits for Hospital Outpatient Prospective Payment System (OPPS).

TIBCO Foresight guidelines that enforce CCI are marked with *CCI in **ForesightHIPAAguidelinelist.pdf**. You can enforce CCI in your own guidelines by merging them with these or by using BusinessRules.CCI as described in **BusinessRules.pdf**.

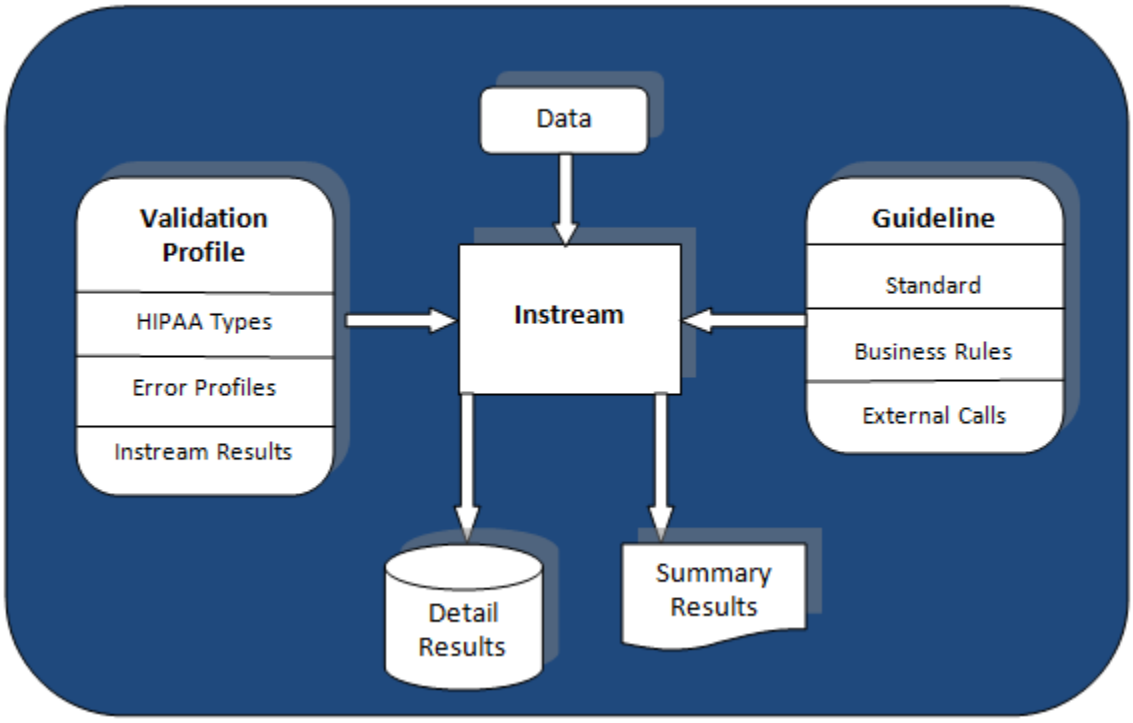
TIBCO Foresight CCI checking includes:

- Check CCI pairs to see if they can go together.
- For pairs that need a modifier to go together:
  - Is a modifier provided in the data?
  - Are the modifiers legal CCI modifiers?
  - Both members of a pair cannot use the same anatomical modifier.

CCI and NCCI checking significantly reduce speed of validation.

Please refer to the National Correct Coding Policy Manual for information about modifiers.

Input and Output



The validation program is HVInStream (for UNIX) or HVInStream.exe (for Windows).

Input

Instream validation reads data in various formats (see **FileFormatsAtForesight.pdf**).

When running from the command line, EDI data includes one or more complete interchanges per file. Each interchange can contain one or more functional groups, each with one or more transactions or messages.

For X12-based data, when using the C++ API, the data need not have ISA, GS, GE, or IEA if you use **DocumentLevelOnly** or **FSDOCUMENTONLY**.

Output

Instream validation creates two files or output streams:

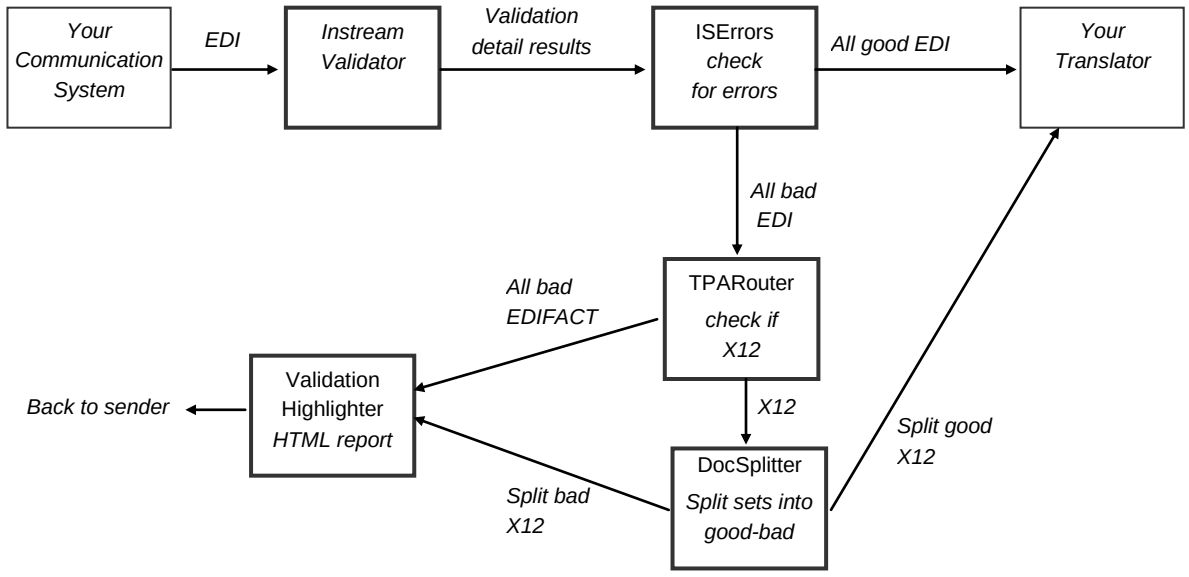
- **Detail results** describing each error and warning found during validation, plus general messages and statistics. You can include custom records that contain actual values from the data. For example, you can output claim number s or patient numbers so that post-processors can identify the specific claim or patient to which an error applies. Please see Detail Results on page 32.

- **Summary results** containing a summary of the validation, including validation start and end times and number of errors, warnings, and other messages. Please see Summary Results on page 31.

For validation, Instream uses:

- (for HIPAA data) More than 50 TIBCO Foresight-supplied data code tables such as Taxonomy and ICD-9. For directions on changing these tables, see **ExtendingCodeTables.pdf**.
- (for EDI data) TIBCO Foresight-supplied guidelines containing the rules from the underlying standard (such as X12, EDIFACT, or TRADACOMS); for HIPAA data, this includes the HIPAA rules. For a list, see **ForesightHIPAAguidelinelist.pdf**.
- (Optional) Your own rules. You can use the TIBCO Foresight guideline-authoring tool EDISIM to add your own rules to a guideline. Contact TIBCO Foresight Support for details.

Typical Inbound Implementation - X12 and EDIFACT Example

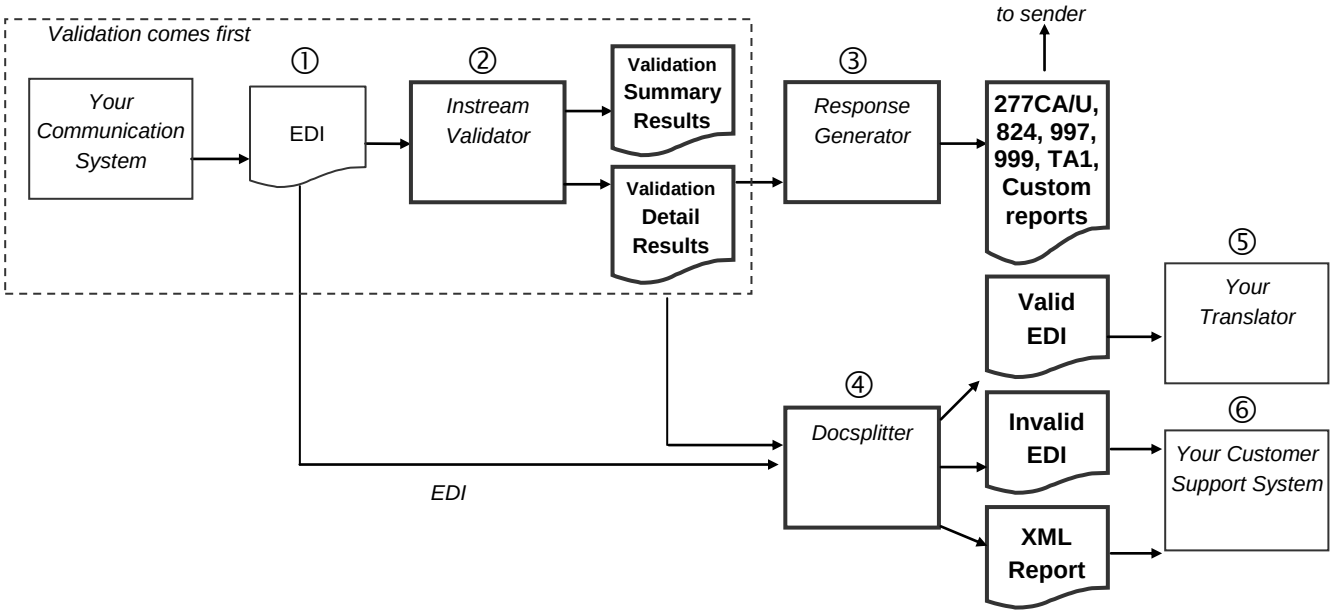


The **bold** objects above are part of Instream or are generated by Instream.

Typical Inbound Implementation - HIPAA Example

Instream can validate *inbound* data as it comes into your organization to protect your translator and application systems from bad data.

Instream is designed to be highly configurable, with several components that work together. This example shows Instream's Validator, Response Generator, and Docsplitter working together to process inbound EDI.



The **bold** objects above are part of Instream or are generated by Instream.

Typical Steps:

① EDI comes in through your company communication system.

② Instream Validator reads the EDI and creates detail and summary results.

The dotted outline in the graphic above indicates these steps, which precede use of Docsplitter or Response Generator.

After Validator creates the detail results file, either Docsplitter or Response Generator can start in any order or simultaneously. Assume that Response Generator starts first.

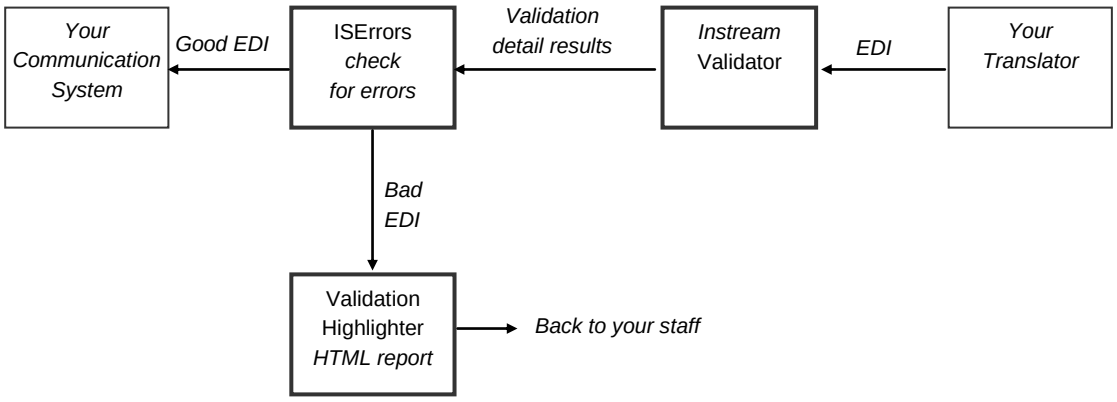
③ Response Generator reads the detail results created by Validator and generates the appropriate response document. Your application dispatches it to the sender.

④ Docsplitter reads the detail results created by Validator and the original EDI file and generates a file containing the valid EDI, a file containing the invalid EDI, and an XML report.

⑤ The valid EDI goes to your translator.

⑥ The invalid EDI and the XML report go to your customer support system.

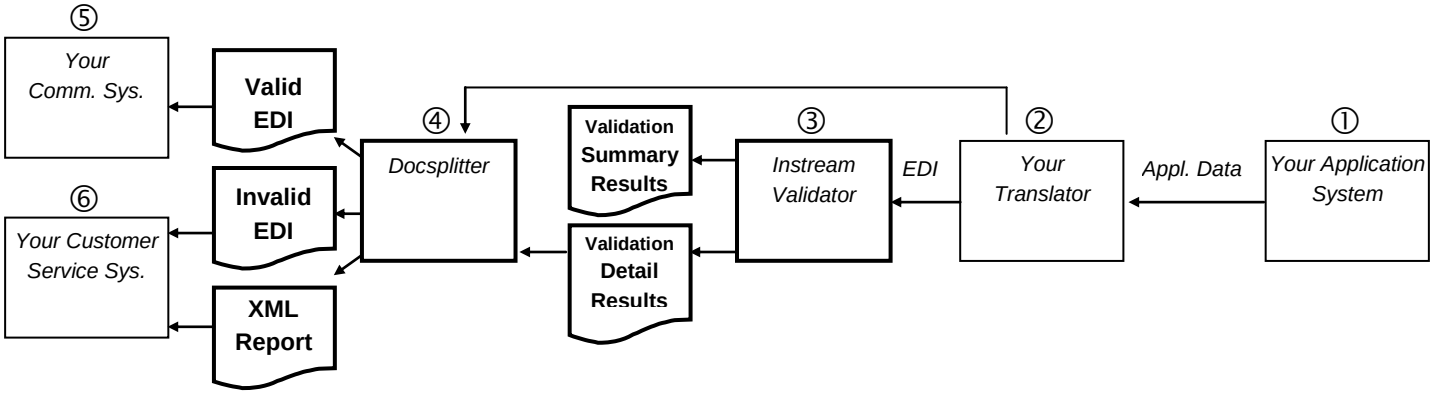
Typical Outbound Implementation - X12 and EDIFACT Example



The **bold** objects above are part of Instream or are generated by Instream.

Typical Outbound Validation - Docsplitter Implementation

This example shows Instream and Docsplitter working together to validate outbound EDI data as a final check before it goes into the world.



The **bold** objects above are part of Instream or are generated by Instream.

Typical Steps:

① Your application system creates application data.

② Your translator reads the application data and creates EDI.

③ Instream Validator reads the EDI and creates summary and detail results files.

④ Docsplitter reads the detail results file and the EDI file and splits the data into valid and invalid EDI. It also creates an XML report.

⑤ The valid EDI is sent to your communications system so that it can be sent out to the recipient.

The invalid EDI and the XML report go to your customer service system.

Acknowledgements

You have two options for generating acknowledgements:

- Use your own acknowledgement handling procedures, which can make use of the summary and detail results files created by validation.
- Use Instream's Response Generator (X12- and EDIFACT-based data).

Response Generator can create:

- HIPAA data:
  - 824, 997, and 999 (version 5010 only)
  - 277CA, 277U, and 277X228 (for 837 transactions only)
  - TA1
  - custom text reports

- Non-HIPPA data:
  - 824, 997, and 999 (version 5010 only)
  - TA1
- EDIFACT data:
  - CONTRL message

You have control of many aspects of the response, including the severity of errors that cause a response.

Please see page 8 for an example of where Response Generator might fit. Also see **TIB_instream_<*n.n*>_respgen.pdf**.

Conditions that will Stop Validation

Instream validation will send a return code of 140 and stop validating if it cannot fulfill your requests for APF, guideline, partner automation, error file, database lookup, or other setup information or resources. **Errors in data will not stop validation.**

The following list gives examples of circumstances that **will stop** validation:

- You ask for guideline ABC, which does not exist.
- You ask for partner automation, but the setup file is not in a usable format.
- Your APF file is a Microsoft Word document.
- Communication issues between validation business rules DBExecute and DBQuery and the SQL database, such as:
  - A business rule points to a database that cannot be accessed.
  - Your business rule queries a database table that does not exist.

  If you prefer that validation continue when a communication issue is encountered, enable the option ContinueAfterSQLErrors in your validation profile.

  ```
  ContinueAfterSQLErrors=1
  ```

  See **APF.pdf** for more information.

Performance

If you do NOT use HIPAA guidelines, you can improve the efficiency of Instream by putting this line in the [Options] section of the $dir.ini, or by removing any colon in front of the line if it is already there:

```
HipaaCodeTable=0
```

Windows Tutorials

HIPAA Windows Tutorial

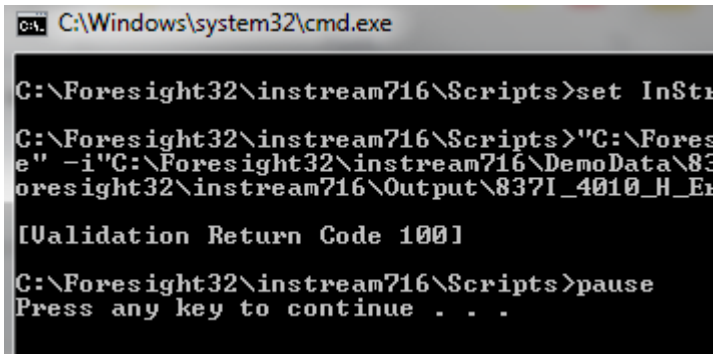
1. Go to Instream's **DemoData** directory and look at the EDI data in 837I_4010_H_ErrorEvenClms.txt.

   This contains one 837I transaction set with one provider, one subscriber, and 10 claims. The claims are numbered from 1 to 10 and the even-numbered claims have errors.

2. Validate the data:

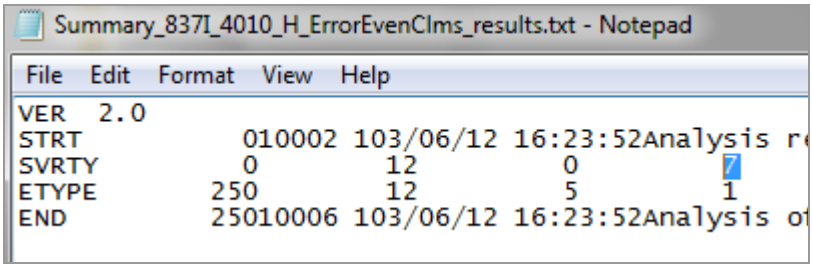
   Go to Instream's **Scripts** directory and double-click on **V_837I_4010_H_1.bat**.

   This batch runs Instream validation with GuidelinePlus PDSA837I, the HIPAA X12-4010 addenda. When it finishes successfully, you should see the message **Validation Return code = 100**.



3. Go to Instream's Output directory and use a text editor to look at the validation summary file **Summary_837I_4010_H_ErrorEvenClms_results_out.txt**.

   It has 7 errors:



   For details about the format of the summary file, see **Summary Results** on page 31.

4. Look for the specific errors in the detailed validation results file:

   Open **837I_4010_H_ErrorEvenClms_results.txt**.

   Search for **DTL**. This line and the next three explain an error. This error is on line 69 of the EDI file, a DTP segment. The segment is missing.

```
DTL          69  2300 DTP                66             2
10811 3  1731 3 1007     488

EMSG         69Missing Segment DTP (Statement Dates) at 2-135, though
marked "Must Be Used"
```

   Search for the other DTL segments in order to see additional errors.

   For details about the format of the detail file, see **Detail Results** on page 32.

Create your own Batch File

Create a batch file to validate **834_4010_H_2members.txt**, which is in Instream's **DemoData** directory.

Since its GS08 contains **004010X095A1**, you know it is based on an 834 Addenda (see **ForesightHIPAAguidelinelist.pdf**).

Your job is to create and execute a batch file to validate this 834.

1. Go to Instream's **Scripts** directory and copy **V_837I_4010_H_1.bat** to **V_834_4010_H_1.bat**.
2. Open the new file in a text editor like Notepad and make these changes:

```
"%InStreamRoot%\Bin\HVInStream.exe"
 -i"%InStreamRoot%\DemoData\ 834_4010_H_2members.txt"
-o"%InStreamRoot%\Output\ 834_4010_H_2members_out.txt"
-gPDSA834
```

3. Save, close, and run **V_837I_4010_H_1.bat**.
4. View the new output files added to Instream's **Output** folder:

The Instream validation command format is described on page 24.

EDIFACT Windows Tutorial

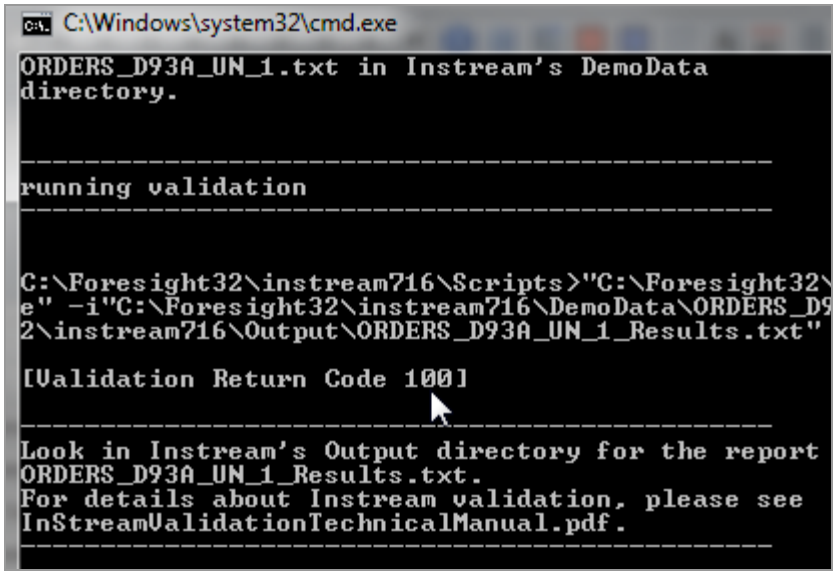
1. Go to Instream's **DemoData** directory and look at the EDI data in **ORDERS_D93A_UN_1.txt**.

   This contains one simple ORDERS message that we are going to check for errors.

2. Validate the ORDERS data:

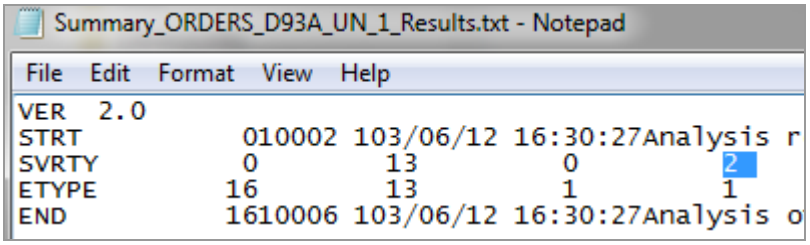
   Go to Instream's **Scripts** directory and double-click on **V_ORDERS_D93A_UN_1.bat**.

   This batch file uses Instream to validate this data against the EDIFACT standard D93A. When it finishes successfully, you should see the message **Return code = 100**.



3. Go to Instream's Output directory and use a text editor to look at the validation summary file **Summary_ORDERS_D93A_UN_1_Results.txt**.

   It has two errors:



   For details about the format of the summary file, see Summary Results on page 31.

4. Look for the specific errors in the detailed validation results file:

   Open **ORDERS_D93A_UN_1_Results.txt**.

   Search for **DTL**. This line and the next two explain an error. This error is on line 4 of the EDI file, the BGM segment. There is an excess sub-element separator:

```
DTL           4       BGM1225             1 3           0
10623 3    21 8 6848

EMSG          4Sub-element separator seen in elementary data element
at BGM03 (D.E. 1225) at col. 17.  Excess ignored.

ESEG          4BGM+220+KC11111+9:'
```

   The next DTL line is about line 7 in the EDI file, a NAD segment with a bad code value in the NAD01.

   For details about the format of the detail file, see Detail Results on page 32.

Create your own Command Line Batch File

We are going to create a batch file to validate **INVOIC_D96A_UN_1.txt** in Instream's DemoData directory.

Copy **V_ORDERS_D93A_UN_1.bat** to **V_INVOIC_D96A_UN_1.bat**. Adjust the bold parts below to point to the input file (-i), the output file (-o), and the guideline (-g).

```
"%InStreamRoot%\Bin\HVInStream.exe" -
i"%InStreamRoot%\DemoData\INVOIC_D96A_UN_1.txt "
-o"%InStreamRoot%\Output\INVOIC_D96A_UN_1_Results.txt"
 -gD96A
```

(The guideline D96A must be in Instream's Static directory, since it is a basic, unmodified EDIFACT standard. Modified guidelines go in the Database directory.)

The Instream validation command format is described on page 24.

X12 Windows Tutorial

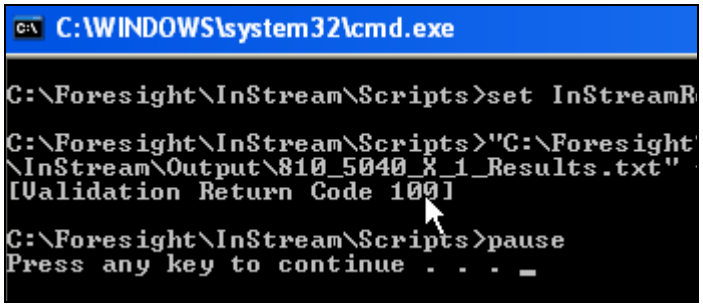
1. Go to Instream's **DemoData** directory and look at the EDI data in **810_5040_X_1.txt**.

   This contains one simple X12-5040 810 invoice that we are going to check for errors.

2. Validate the 810 data:

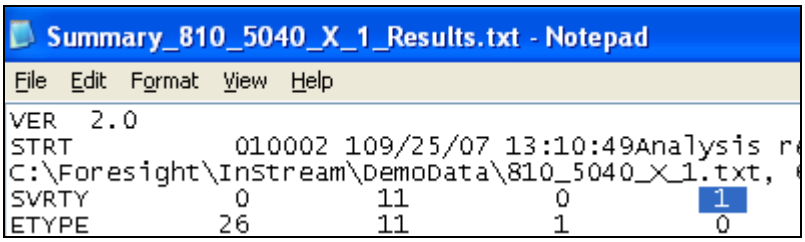
   Go to Instream's **Scripts** directory and double-click on **V_810_5040_X_1.bat.**

   This batch file uses Instream to validate this data against standard X12-5040. When it finishes successfully, you should see the message **Return code = 100**.



3. Go to Instream's Output directory and use a text editor to look at the validation summary file **Summary_810_5040_X_1_Results.txt**.

   It has one error:



   For details about the format of the summary file, see Summary Results on page 31.

4. Look for the specific error in the detailed validation results file:

   Open **810_5040_X_1_Results.txt**.

   Search for **DTL**. This line and the next three explain an error. This error is on line 26 of the EDI file, the IEA segment. The control number doesn't match the one in the ISA.

```
DTL          26       IEA I12            23 2           0
10912 3    01-100011

EMSG         26Interchange Ctl No. 000000003 in IEA doesn't match
ISA's 000000001

EDAT         26000000003

ESEG         26IEA*1*000000003!
```

   This is the only error. If there had been others, they would be documented similarly, starting with a DTL line.

   For details about the format of the detail file, see Detail Results on page 32.

Create your own Command Line Batch File

We will make a batch file to validate an 850 purchase order: **850_4010_X_1.txt**.

Copy **V_810_5040_X_1.bat** to a new name **V_850_4010_X_1.bat**. Adjust the bold parts below to point to **850_4010_X_1.txt** for the input file (-i) and modify the output filename accordingly (-o). The guideline is X12-4010.

```
"%InStreamRoot%\Bin\HVInStream.exe"
-i"%InStreamRoot%\DemoData\850_4010_X_1.txt"
-o"%InStreamRoot%\Output\850_4010_X_1_Results.txt"
-gX12-4010
```

The Instream validation command format is described on page 24.

UNIX Tutorials

Each TIBCO Foresight-supplied script identifies the location of Instream with LIBPATH.

Setting LIBPATH on UNIX	
AIX	export LIBPATH=/HVInStream/bin:$LIBPATH
SUN	LD_LIBRARY_PATH=/HVInStream/bin:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH
Red Hat Enterprise Linux	export FSINSTREAMINI=/HVInStream/bin export LD_LIBRARY_PATH=/HVInStream/bin:$LD_LIBRARY_PATH

HIPAA UNIX Tutorial

1. Go to Instream's **DemoData** directory and look at the EDI data in **837I_4010_H_ErrorEvenClms.txt**.

   This contains one 837I transaction set with one provider, one subscriber, and 10 claims. The claims are numbered from 1 to 10 and the even-numbered claims have errors.

2. Validate the data:

   Go to Instream's **Scripts** directory and look at the contents of **V_837I_4010_H_1.sh**. Execute this script by typing its name.

   This script runs Instream validation with GuidelinePlus PDSA837I. When the script finishes successfully, you should see the message **validation return code=100**.

3. Go to Instream's Output directory and use a text editor to look at the validation summary file **Summary_ 837I_4010_H_ErrorEvenClms_results.txt**. The SVRTY line shows that it has 7 errors (the 4th number is the error count).

4. Look for the specific errors in the detailed validation results file:

   In **837I_4010_H_ErrorEvenClms_results.txt**, find the first line that starts with DTL. This line and the next three explain an error. This error is on line 69 of the EDI file, a DTP segment. The segment is missing.

   Search for the other DTL segments and see what other errors were found.

   For details about the format of the detail file, see **Detail Results** on page 32.

Create your own Script File

Create a script file to validate **834_4010_H_2members.txt**, which is in Instream's **DemoData** directory.

Since its GS08 contains **004010X095A1**, you know it is based on an 834 Addenda (see **ForesightHIPAAguidelinelist.pdf**).

Your job is to create and execute a script file to validate this 834.

1. Go to Instream's **Scripts** directory and copy **V_837I_4010_H_1.sh** to **V_834_4010_H_1.sh**.
2. Make these changes to the new file

```
"%InStreamRoot%\Bin\HVInStream.exe"
 -i"%InStreamRoot%\DemoData\ 834_4010_H_2members.txt"
-o"%InStreamRoot%\Output\ 834_4010_H_2members_out.txt"
-gPDSA834
```

3. Save, close, and run **V_837I_4010_H_1.sh**.
4. View the new output files added to Instream's **Output** folder:

The Instream validation command format is described on page 24.

EDIFACT UNIX Tutorial

1. Go to Instream's **DemoData** directory and look at the EDI data in **ORDERS_D93A_UN_1.txt**.

   This contains one simple ORDERS message that we are going to check for errors.

2. Validate the ORDERS data:

   Go to Instream's **Scripts** directory and double-click on **V_ORDERS_D93A_UN_1.sh**.

   This script file uses Instream to validate this data against the EDIFACT standard D93A. When it finishes successfully, you should see the message **Return code = 100**.

3. Go to Instream's Output directory and look at the validation summary file **Summary_ ORDERS_D93A_UN_1_Results.txt**. The SVRTY line shows two errors (the fourth number in the line).

   For details about the format of the summary file, see Summary Results on page 31.

4. Look for the specific errors in the detailed validation results file:

   Open **ORDERS_D93A_UN_1_Results.txt**.

   Search for **DTL**. This line and the next two explain an error. This error is on line 4 of the EDI file, the BGM segment. There is an excess sub-element separator:

```
DTL           4       BGM1225             1 3           0
10623 3    21 8 6848

EMSG          4Sub-element separator seen in elementary data element
at BGM03 (D.E. 1225) at col. 17.  Excess ignored.

ESEG          4BGM+220+KC11111+9:'
```

The next DTL line is about line 7 in the EDI file, a NAD segment with a bad code value in the NAD01.

For details about the format of the detail file, see Detail Results on page 32.

Create your own Script File

Copy **V_ORDERS_D93A_UN_1.SH** to **V_INVOIC_D96A_UN_1.SH**. Adjust the bold parts below to point to the input file (-i), the output file (-o), and the guideline (-g).

```
"%InStreamRoot%\Bin\HVInStream.exe" -
i"%InStreamRoot%\DemoData\INVOIC_D96A_UN_1.txt " -
o"%InStreamRoot%\Output\INVOIC_D96A_UN_1_Results.txt"
 -gD96A
```

(The guideline D96A must be in Instream's Static directory, since it is a basic, unmodified EDIFACT standard. Modified guidelines go in the Database directory.)

The Instream validation command format is described on page 24.

X12 UNIX Tutorial

1. Go to Instream's **DemoData** directory and look at the EDI data in **810_5040_X_1.txt**.

   This contains one simple X12-5040 810 invoice that we are going to check for errors.

2. Validate the 810 data:

   Go to Instream's **Scripts** directory and double-click on **V_810_5040_X_1.sh**

   This batch file uses Instream to validate this data against standard X12-5040. When it finishes successfully, you should see the message **Return code = 100**.

3. Go to Instream's Output directory and look at the validation summary file **Summary_810_5040_X_1_Results.txt**. The SVRTY line shows one error (the fourth number in the line).

   For details about the format of the summary file, see Summary Results on page 31.

4. Look for the specific error in the detailed validation results file:

   Open **810_5040_X_1_Results.txt**.

   Search for **DTL**. This line and the next three explain an error. This error is on line 26 of the EDI file, the IEA segment. The control number doesn't match the one in the ISA.

```
DTL          26       IEA I12            23 2           0
10912 3    01-100011

EMSG         26Interchange Ctl No. 000000003 in IEA doesn't match
ISA's 000000001

EDAT         26000000003

ESEG         26IEA*1*000000003!
```

   This is the only error. If there had been others, they would be documented similarly, starting with a DTL line.

   For details about the format of the detail file, see Detail Results on page 32.

Create your own Script File

We will make a batch file to validate an 850 purchase order: **850_4010_X_1.txt**.

Copy **V_810_5040_X_1.sh** to a new name **V_850_4010_X_1.sh** . Adjust the bold parts below to point to **850_4010_X_1.txt** for the input file (-i) and modify the output filename accordingly (-o). The guideline is X12-4010.

```
"%InStreamRoot%\Bin\HVInStream.exe"
-i"%InStreamRoot%\DemoData\850_4010_X_1.txt"
-o"%InStreamRoot%\Output\850_4010_X_1_Results.txt"
-gX12-4010
```

The Instream validation command format is described on page 24.

TIBCO Foresight-Supplied Guidelines

Instream ships with a database of EDI guidelines that contains information about the valid X12 and EDIFACT syntax and rules.

- Some X12 guidelines contain HIPAA validation rules. Please see **ForesightHIPAAguidelinelist.pdf** for a list of HIPAA guideline names.
- Please see **ForesightGeneralguidelinelist.pdf** for a list of general guideline names supported by Instream on TIBCO Foresight® Transaction Insight®.

Customizing Guidelines

You can use EDISIM Standards Editor to add your own rules to TIBCO Foresight-supplied guidelines.

HIPAA customers can merge a guideline that they customize with the corresponding TIBCO Foresight-supplied HIPAA guideline. This yields a guideline that contains all X12 and HIPAA rules plus your own rules.

See **BusinessRules.pdf** and **TIB_fsp_edisim_ *<n.n>*_fseditor.pdf** for more information.

2 Validating with Instream

Methods of Execution

Instream validation can be:

- Run as a command line, usually from within a batch file or script
- Executed from another program via a system call.

Validating from a Command Line

The command line to start validation is usually in a script or batch file.

Instream issues one or more return codes (see page 77) when the validation is complete.

The executable for the command line is:

Windows **HVInStream.exe**

UNIX **HVInStream**

It is a standalone executable in Instream's **Bin** directory. Control returns when the validation completes, and your own procedures then examine the output files to determine the success or failure of the validation.

The command line has the following case-sensitive parameters. Use quotation marks around paths or filenames that contain spaces.

InstreamExecutable **-i***InFile* **-o***Outpath* **-g(or -xg)***Guideline* **-s***StartupProfile* **-a -c -D -d -f -m -r -u**

Important	All HVInStream options are **case sensitive**. **Spaces** are significant. Use **double quotation marks** around paths that have spaces.
InstreamExecutable	Instream executable, including path: **HVInStream.exe** (Windows) **HVInStream** (UNIX)
i	**i**nput file - path and filename of an EDI data file. **Required**.
o	**o**utpath - desired path and filename for the results file. **Required**. The detail file will use the filename that you specify and the summary file will use the same filename preceded with "Summary_". The directory must already exist. If the output files already exist, they will be overwritten. Filename is optional. Default output filenames are FS_Output and Summary_Output.
xo	**X**ML **o**utput – desired path and filename for the results file. Optional but recommended. See **XMLatForesight.pdf** for details.

g	**g**uideline – guideline to use for validation. Required for flat file data. Optional but recommended for EDI data. For XML data, use the **xg** parameter instead. If using a standard, it must be located in Instream's Static directory. If using a guideline, it must be in Instream's Database directory. If **-g** is omitted, Instream checks to see if trading partner automation is enabled and if a match can be found that way. If not, it checks the $dir.ini or fsdir.ini for a GuidelineBestFit=0 setting. If it finds it, validation is cancelled and a return code of 140 is issued. Otherwise, Instream picks the guideline that best fits: ▪ For X12-based data, it matches the data's GS08 (X12) or UNG07 (EDIFACT) with the VRI of a guideline. ▪ For EDIFACT-based data, it matches the UNG UNG07 (01 and 02, plus 03 if it exists). If there is no UNG, it uses the UNH02 (02, 03, and 05 if available). The detail results file shows a message confirming the guideline for each GS or UNH segment for which a best fit guideline is used. If you are using envelope-based partner automation (see **TIB_fsp-instream_ <*n.n*>_tpa.pdf**), using –g on the command line causes Instream to ignore the partner automation entirely. If you are using content-based partner automation, you can use –g to load the guideline that has the content-based partner automation rules. **Recommendation**: Since TIBCO Foresight distributes HIPAA guidelines that include types 1 and 2 and also HIPAA guidelines for types 1 through 7, it is advisable to use **–g**.
xg	**XML g**uideline – guideline to use for validating XML data. Optional but recommended. See **XMLatForesight.pdf** for details.

a	**(Primarily for TIBCO Foresight use)** **A**utomator mode – uses these file extensions for output, regardless of what is specified with the –o parameter: DTL for the detailed results file RPT for the report card SUM for the summary file
c	**c**onfiguration file location – the high-level directory containing $dir.ini (Windows) or fsdir.ini (UNIX). See Specifying Alternate INI Files on page 29.
d	Not available for EDIFACT-based data. **D**ocument-only validation – ignore the ISA and GS and only validate the segments between ST and SE. If you use **-d**, you must also use **-D**). Please see Document-Only Processing on page 29 for details.
D	Not available for EDIFACT-based data. **D**elimiters for document-only validation (segments ST-SE, no ISA and GS). Parameters: *"segTerminator,elemSeparator,componentSeparator"* Possible formats: integer Example: `-D"29,30,31"` hexadecimal Example: `-D"0x1E,0x1F,0x1D"` character Example: `-D" ~*:"` **Important**: **-D** must come before **–d** in the command-line. Please see Document-Only Processing on page 29 for details.
f	table **f**ile server – use the Table File Server. This gives faster processing if you are validating many very small files. See page 77.

m	Original file information - lets you insert the literal filename, date and hour of file creation, the EDI file's size, and its path name into a GEN record with number 15077. This is for your own use; Transaction Insight® displays it and lets you use it in searches on the Transmissions Summary page. Parameters: *"mm/dd/yyyy hh:mm:ss fileSize pathName"* All parameters are required. Example: `-m"02/14/2005 14:55:19 2032 C:/HVInStream/DemoData/835-DEMO1.TXT"` Instead of **-m** on the validation command line, you can use `useinputfileasoriginal=1` in Importer.ini. This collects the information from the STRT record in the validation detail file. If you import data validated with **–m** into Transaction Insight, it takes up additional database space.
r	**r**eport card – creates a formatted report summarizing the results. The report card will be in a file called **Report**_*resultsfilename*.**txt**.
s	**s**tartup profile – profile (APF) file. Optional; default is $fsdeflt.apf (Windows and UNIX) Parameter: *"path"* Example: `-s"S:\shared profiles\MyProfile.apf"` Please see **APF.pdf** for details.
tr	Transaction Insight revalidation (error correction).

u	**u**ser message – free-form text. Allows you to insert whatever text you'd like in a GEN record with number 15078. This is for your own use. Transaction Insight can display it.
	Parameter: *"some text"*
	Example:
	`-u"From Sock 2"`
	Resulting GEN record:
	`GEN 015078 1 0From Sock 2`

-version	Displays the release/version of the Instream software.
	Example:
	`-version`
	Sample output:
	`HIPAA Validator InStream version 8.3.0 [Build 900r(64 bit): 08/14/2013]`

If you execute Instream with no parameters, you will see the version and a list of parameters.

Command Line Examples

The commands below validate this file: `EDI.txt`

And creates these results files: `Results.txt`
`Summary_Results.txt`

Windows Example

All input and output files are in the **C:\Files** directory.

```
"C:\Foresight\Instream\Bin\HVInStream.exe" -i"C:\Files\EDI.txt"
-o"C:\Files\Results.txt" -gPDSA837I
```

UNIX Example

All input and output files are in the **/Files** directory.

```
export FSINSTREAMINI=/HVInStream/bin

export LIBPATH=/HVInStream/bin:$LIBPATH

/HVInStream/bin/HVInStream -i"/Files/EDI.txt" -o"/Files/Results.txt"
-gPDSA837I
```

Sample Batch or Script Files

The files that TIBCO Foresight installs in Instream's **Scripts** folder contain examples of validation. Some also execute Docsplitter, DataSwapper, or Response Generator after validating. Please see **Demo_Index.pdf** for details.

Document-Only Processing

Available for X12 messages only.

The command-line parameters -D and -d work together to validate EDI with no delimiters as follows. If the command-line has -d, it must also have -D.

ISA/GS present in data		✓		✓		✓
-D on command-line	✓	✓	✓	✓		
-d on command-line	✓	✓			✓	✓
Delimiters used for validation	-D	-D	-D	ISA	*Error*	*Error*

Specifying Alternate INI Files

Many configuration features of Instream are defined in the $Dir.ini file (Windows) and the fsdir.ini file (UNIX). This file allows you to specify the name and location of error message files, set up partner automation, and set certain other options.

Instream automatically reads this file from the Instream Bin directory when you request a validation.

When you do not want to use the settings in your usual $Dir.ini or fsdir.ini file in the Instream Bin directory, you can point to a different one with the -c command-line parameter.

The format is:

-c"path"

Where *path* is the high-level directory containing a Bin directory which must contain:

- $Dir.ini or fsdir.ini
- fscint.ini (only needed here if you are using the table file server)

Example

```
set InStreamRoot=C:\Foresight\Instream

"%InStreamRoot%\Bin\HVInStream.exe"
-i"%InStreamRoot%\DemoData\Tutorial837IA.txt"
-o"%InStreamRoot%\Output\Tutorial837IA_Results.txt"
-gPDSA837I -c"C:\OtherINI"
```

C:\OtherINI\Bin contains:

- $dir.ini or fsdir.ini

EDIFACT Escape Characters

If the segment terminator, data element separator, or subelement separator appears in the data itself, it should be preceded with an escape character (usually a question mark) to indicate that it is data rather than a separator. For instance, your segment terminator is a single quote. Your data is **`Bob's Restaurant`** so your data should read **`Bob?'s Restaurant`**.

To include a `?` in the data, use two consecutive `??`. To include `?` in the data, use two consecutive `??`. To include two `??`, use `????` since each `?` escapes only one character.

To change the escape character from a question mark, the data file being validated should start with a UNA Service String Advice that contains a different escape character.

Application Program Interface

You can integrate Instream validation into another of your applications either statically (for C/C++) or dynamically (for C/C++, C#, and Java).

An advantage of an API interface: you can read and act on results as they come back, rather than waiting for a file to be created. You can also stop the validation anytime, based on the type or number of errors.

The API can pass both the inbound data and resulting output via common memory or file. Common memory is normally the most efficient way of connecting, but *large* documents should be called as files.

For details, see **TIB_fsp-instream_<*n.n*>_api.pdf**.

3 Validation Results

Results Files

Instream Validator writes a summary and a detail results file for each EDI file validated.

Summary File Check it to see how many errors, warnings, etc., were found. See Summary Results below.

Detail File Check it to see specifics about the validation: error and warning messages, general messages, etc. See Detail Results on page 32.

Both files:

- Consist of records that start with a one- to five-character record identifier or "tag," left justified in a field five characters wide.
- Contains the output from the validation of a single EDI data file.

Summary Results

This file is named **Summary-***xxx* where *xxx* is the name of the detail output file. This file contains these five records that summarize the validation results:

VER	Version number of the output file format	see page 71
STRT	Date and time when validation started, name and location of file validated	see page 60
SVRTY	Message count by severity	see page 70
ETYPE	Message count by type	see page 54
END	Information about the file just validated	see page 51

To see an example, run one of the files in Instream's **Scripts** directory and then look in Instream's **Output** directory for a filename that contains the word **Summary**.

Detail Results

The detail results file contains the details of the validation. Its name and location are specified with the **-o** command line parameter.

The file has the following structure.

VER
STRT
body (GEN, DTL, EDAT, EMSG, ESEG, SVRTS, ETYPS, Custom Z records)
SVRTY
ETYPE
END

Each DTL record will be followed by an EMSG and ESEG and possibly an EDAT. Don't assume the order of EMSG, ESEG, and EDAT.

TA1

Instream produces a TA1 file under these conditions:

- GuidelineBestFit=0 in the $dir.ini/fsdir.ini
- No guideline is provided on the command line
- No guideline is identified by trading partner automation (GS01 or GS08 is missing)

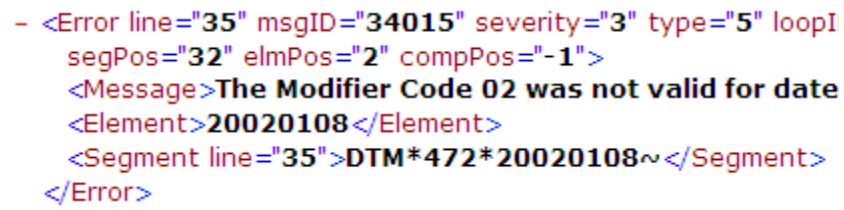
Line Numbers

Line numbers appear throughout Instream's validation detail file, XML report, and delimited report.

Example from detail file

```
DTL          21  2100 NM11036            18 4        ...
EMSG         21Element NM104 (D.E. 1036) at col. 18  ...
```

Example from validation XML report



This is the line number where the error was discovered, but not always the line number where the problem actually occurred. In some cases, it cannot be determined that a segment has an error until later in the transaction.

For example, balancing rules have to be run at the end of a loop. Instream becomes aware of the loop ending when it encounters the first segment after the loop ends. It then makes the balancing comparison and uses its current line number – the first segment after the loop. This is the line number that will appear in the DTL record.

Record Definitions

CSEG: Current Segment Data Record

The CSEG record is variable length and contains the actual contents of the EDI segment currently being processed.

By default, CSEG records are not output since their presence inflates the size of the detail file. To output CSEG records, set CSEG=1 in the APF file. For details, please see the **Detail Record Output** section of the **APF.pdf**.

CSEG Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (CSEG)	5	1	5
Line #	10	6	15
Segment Data	*n*	16	*EOL*

Record Tag

Contains **CSEG** to identify the type of record.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Segment Data

Contains the complete EDI segment.

CTX: Context Record for Response Generator 999

In previous releases, Response Generator created a CTX segment in the 999 if the IK403 started with "I" for Implementation syntax errors. Now, you have three options for 5010 999 Errata CTX records:

- No CTX records (the default)
- TIBCO Foresight-supplied CTX records
- Your own custom CTX records

These are explained in **TIB_instream_ <*n.n*>_respgen.pdf**, Appendix H: CTX Segments in Response Generator 999s.

Example

The underlined information is static text.

CTX record in a DTL file:

```
CTX          19|CTX02|14,32001
```

Corresponding segment in a 999:

```
CTX*SITUATIONAL TRIGGER*SBR*14**2*1069~
```

CTX Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (CTX)	5	1	5
Line #	10	6	15
Vertical Bar	1	16	17
CTX02	5	17	21
Vertical Bar	1	22	23
Line # for 1st seg.	varies	23	
, (*comma*)	1		
Error #	5		

Record Tag

Identifies the type of record: CTX.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

CTX02

Currently, this is always CTX02.

Line # for 1st seg

Line number in the EDI file for the first segment in the situational relationship. The ST is segment 1.

Error #

Error number generated when the situational relationship is violated. This must be in the [RespGen Overrides] section of the validation APF used with Response Generator.

Example: `CTX50001=CTX,32001,SBR,,2,,,1069`

DTL: Detailed Message Record

The DTL record is fixed length and contains detailed information about a warning, error, or information message generated by Instream validation.

The DTL record will be followed by an EMSG and ESEG record. It may also be followed by an EDAT record and one or more EMSG, EDAT, and ESEG records, depending on the error encountered.

DTL Record Layout				
Field	**Length**	**Start**	**End**	**Notes**
Record Tag (DTL)	5	1	5	
Line #	10	6	15	
Loop/Group ID	6	16	21	
Seg ID	4	22	25	
Elem ID	4	26	29	
Comp ID	4	30	33	
Seg Pos	10	34	43	
Elem Pos	2	44	45	
SubElem Pos	2	46	47	
Loop/Group Repeat Count	10	48	57	
Element Repeat	10	58	67	4020 or later
999 IK3-04	3	68	70	not in EDIFACT or TRADACOMS
999 IK4-03	3	71	73	not in EDIFACT or TRADACOMS
Filler	4	74	77	
Error #	5	78	82	
Severity	2	83	84	
Seg Ordinal Number	5	85	89	
HIPAA Type	1	90	90	not in EDIFACT or TRADACOMS
997 AK304	2	91	92	not in EDIFACT or TRADACOMS
997 AK403	2	93	94	not in EDIFACT or TRADACOMS
824 TED01	3	95	97	not in EDIFACT or TRADACOMS
824 TED02	3	98	100	not in EDIFACT or TRADACOMS
277 STC01-02	5	101	105	not in EDIFACT or TRADACOMS
Filler	5	106	110	
Application Data	20	111	130	right justified

Record Tag

Contains **DTL** to identify the type of record.

Line #

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Loop/Group ID

Contains the identifier of the lowest level loop or group when the DTL record was generated. This field will be blank when the message is associated with an object that is outside any loop or group.

Seg ID

Contains the segment identifier in effect when the DTL record was generated. This segment ID may not be the one that caused the error. Instead, it is the segment being processing when the error was detected.

Example: An N2 segment is required in the NM1 loop. Instream does not know that it is missing until it encounters the N3 segment and notices that the N2 segment was not seen. The Segment ID in the 'Missing Segment' message will be N3, not N2. Also see the Segment Ordinal Number field, below.

Elem ID

Contains the element identifier corresponding to the message in the DTL record. If the message is not related to an element, this field is blank.

Comp ID

Contains the composite identifier corresponding to the message in the DTL record. If the message is not related to a composite, this field is blank.

Seg Pos

Contains a sequential number indicating where the segment is located in the input file. This is the segment being processed when the error was detected, not necessarily the segment that *caused* the message to be generated. The numbering starts at **0** at each ST or UNH and increments through the set or message. It starts over at the next ST or UNH.

Elem Pos

Contains the position number of the element or composite in the segment that caused the message, starting with **1**. If the message is not related to an element, this field is blank.

SubElem Pos

Contains the position number of the subelement within the composite that caused the message, starting with **1**. If the message is not related to a composite, this field is blank.

Loop/Group Repeat Count

Shows the iteration of the loop or group. For example, if the error is within the second iteration of the claim loop for a particular dependent, then this value is 2.

Element Repeat

X12-4020 and later. Shows which position a repeating element occupies. Example: If the element repeats three times and the error is in the second one, this field will contain 2.

999 IK3-04

Used in X12 999 responses only. The Implementation Segment Syntax Error Code for this error number. This field is mapped from the **Warning Levels** section of the **APF** file used. See **APF.pdf**.

999 IK4-03

Used in X12 999 responses only. The Implementation Data Element Syntax Error Code for this error number. This field is mapped from the **Warning Levels** section of the **APF** file used. See **APF.pdf**.

Filler

Blank field reserved for future development.

Error #

Contains the ID number of the error.

Severity

Contains the severity of the error, one of these:

- **0** Ignore
- **1** Informational
- **2** Warning
- **3** Error
- **4** Fatal Error
- **5** User Level 1
- **6** User Level 2

Seg Ordinal Number

Contains the ordinal (internal key) number of the segment that *caused* the message. The ordinal number is mainly for internal use by TIBCO Foresight. Unlike the Seg ID field, this segment will actually be the one that is referenced by the message, not the current segment being processed.

Let's use the same example from the Seg ID field above. An N2 segment is required in the NM1 loop. Validator does not know that it is missing until it encounters the N3 segment and notices that the N2 segment was not seen. The Segment Ordinal Number in the 'Missing Segment' message will be for the N2, not the N3.

HIPAA (WEDI SNIP) Type

The WEDI SNIP Type field contains the HIPAA type of the error. The valid types and their meanings are:

- **0** General messages. TIBCO Foresight assigns messages to Type 0 if they do not deal with WEDI SNIP type errors or warnings.
- **1** EDI Syntax
- **2** Syntactical Requirement (within HIPAA Validator® Desktop, this is combined with 1)
- **3** Balancing
- **4** Situation
- **5** Code Set
- **6** Product Types or Lines of Service
- **7** Payer Specific
- **8** Partner Specific (within HIPAA Validator Desktop, this is shown as a P)

997 AK304

Used in X12 997 responses only. The Segment Syntax Error Code for this error number. This field is mapped from the **Warning Levels** section of the **APF** file used. See **APF.pdf**.

997 AK403

Used in X12 997 responses only. The Element Syntax Error Code for this error number. This field is mapped from the **Warning Levels** section of the **APF** file used. See **APF.pdf**.

824 TED01

Used in X12 824 responses only. Application Error Condition Code for this error number. This field is mapped from the **Warning Levels** section of the **APF** file used. See **APF.pdf**.

824 TED02

Used in X12 824 responses only. Technical Error Description Free Form Message for this error number. This field is mapped from the **Warning Levels** section of the **APF** file used. See **APF.pdf**.

277 STC01-02 code

Used in X12 277 responses only. Health care claim status code. This field is mapped from the **Warning Levels** section of the **APF** file used. See **APF.pdf**.

Application Data

For your use. This field is mapped from the APF file's **Warning Levels** section. It is the last item in the error's definition. The data is right-justified in the field.

Example usage: Store your own error number that corresponds to the TIBCO Foresight error number. See **APF.pdf**.

EDAT: EDI Error Data Record

The EDAT record is variable length and contains the actual data that caused the warning, error, or information message referred to by the preceding DTL record.

The EDAT record follows, and further describes, a DTL record. Only one EDAT record occurs for a DTL record.

If no specific data is involved in the error (example: missing segment), then no EDAT record is written; since there would be no data to show.

EDAT Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (EDAT)	5	1	5
Line #	10	6	15
Error Data	*n*	16	*EOL*

Record Tag

Contains **EDAT** to identify the type of record.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Error Data

Contains the data that caused the error.

EDTL: Detailed Message Record for XML Data

For details about XML, see ***XMLatForesight.pdf***

The EDTL record is variable length and contains detailed information about a warning, error, or information message generated by Instream when validating XML or flat file data.

The flat file format is documented separately on page 45.

The EDTL record will be followed by an EMSG record ESEG record and may be followed by an EDAT record, depending on the error encountered.

Example:

```
EDTL          16|/PO/BT_COUNTRY/||BT_COUNTRY|13|||||18244|3|
```

EDTL Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (EDTL)	5	1	5
Line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar			
Path from root		17	
Group Repeat Count			
“Segment” ID			
Position			
Not used for XML			
Not used for XML			
Not used for XML			
Not used for XML			
Error #			
Severity			

Record Tag

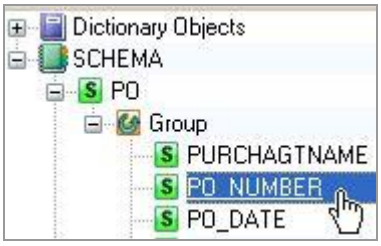
Contains **EDTL** to identify the type of record.

Line

Contains the number of physical lines in the XML data file up to the point where this record is generated.

Path from root

Contains the path from the root to the element containing the error. Slashes separate each part of the path. If this element had an error, the EDTL would contain a path of `/PO/PO_NUMBER/`:

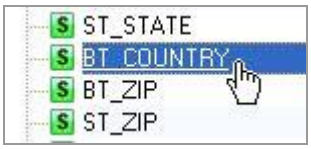


Group Repeat Count

The iteration of the group, if applicable.

"Segment" ID

The ID of the "segment" to which the message pertains:



Position

The first character position where the error was detected. In this example, USA is not an acceptable value so the position is 13 because the "U" is the 13th position from the start of the tag:

```
<BT_COUNTRY>USA</BT_COUNTRY>
             ↑
```

Error#

Contains the ID number of the error.

Severity

Contains the severity of the error, one of these:

0	Ignore	**4**	Fatal Error
1	Informational	**5**	User Level 1
2	Warning	**6**	User Level 2
3	Error		

Application Data

For your use. This field is mapped from the APF file's **Warning Levels** section. It is the last item in the error's definition. Example usage: Store your own error number that corresponds to the TIBCO Foresight error number. See **APF.pdf**.

EDTL: Detailed Message Record for Flat File Data

For details about flat files, see ***FlatFilesAtForesight.pdf***

The EDTL record is variable length and contains detailed information about a warning, error, or information message generated by Instream when validating XML or flat file data.

The XML version of this record is documented separately on page 43.

The EDTL record will be followed by an EMSG record ESEG record and may be followed by an EDAT record, depending on the error encountered.

Example:

```
EDTL              5||0|PETS|SPEC|2||0|2|10605|3|
EMSG             5Code Value "PARAKEET" not found in the dictionary ...
EDAT              5PARAKEET
ESEG             5PETS*JENNY*PARAKEET*BLUE!
```

EDTL Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (EDTL)	5	1	5
Line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar			
Loop ID		17	
Loop Repeat Count			
Segment ID			
Element ID			
Element Position			
Subelement ID			
Subelement Position			
Ordinal Number			
Error #			
Severity			
Application Data			

Record Tag

Contains **EDTL** to identify the type of record.

Line #

Contains the number of physical lines in the data file up to the point where this record is generated.

Loop ID

Contains the identifier of the lowest level loop where the EDTL record was generated. This field will be blank when the message is associated with an object that is outside any loop.

Loop Repeat Count

Shows the iteration of the loop containing the error. For example, if the error is within the second iteration of a repeating loop, then this value is 2.

Segment ID

Contains the record identifier in effect when the EDTL record was generated. This segment ID may not be the one that caused the error. Instead, it is the segment being processing when the error was detected.

Element ID

Contains the field tag that generated the message. This may not be what caused the error, but it was being processed when the error was detected.

Element Position

The position number of the field in the record that caused the message, starting with **1**. If the message is not related to an element, this field is blank.

Subelement ID

Contains the identifier of the field within a complex field.

Subelement Position

Contains the position number of the field within the complex field that caused the message, starting with **1**.

Ordinal Number

Contains the ordinal (internal key) number of the record that caused the message. The ordinal number is mainly for internal use by TIBCO Foresight. This will actually be the one that is referenced by the message, not the current record being processed.

Error #

Contains the ID number of the error.

Severity

Contains the severity of the error, one of these:

- **0** Ignore
- **1** Informational
- **2** Warning
- **3** Error
- **4** Fatal Error
- **5** User Level 1
- **6** User Level 2

Application Data

For your use. This field is mapped from the APF file's **Warning Levels** section. It is the last item in the error's definition. Example usage: Store your own error number that corresponds to the TIBCO Foresight error number. See **APF.pdf**.

ELOC: Error Location Record

The ELOC record is variable length and contains the names of elements, composites, segments, and loops to identify the location of errors in business language rather than for an EDI specialist. It is generated for errors in the 10000-29999 range only.

If ELOC=1 in the validation APF:

- An ELOC record appears in the detail file after a EMSG record that refers to an error.
- An ELOC record appears in a Response Generator custom report if the custom report template contains the %ErrMsg_NonTech% variable.
- Transaction Insight Portal users will see this information if they selects Non-technical messages.

Demo V_RG_837P_4010_textrpt_ELOC in Instream's Scripts directory.

Output Examples using ELOC=1 in APF

1. **Detail file entry. Data is EDI.**

   Example:

```
EMSG          6Element NM103 (D.E. 1035) at col. 10 is missing, though
marked "Must Be Used"
ELOC          6\Name Last or Organization Name\\Submitter Name
```

2. **Response Generator custom report entry with %ErrMsg_NonTech% variable in report template. Data is EDI.**

   Example:

   %ErrMsg% `Clm: The field Name Last or Organization Name field of the Other Payer Name`

   %ErrMsg_NonTech% **Clm: The field Name Last or Organization Name field of the Other Payer Name information in the Claim Information/Other Subscriber Information area at col. 10 is missing though it was marked in the guideline as "Must Be Used".**

3. **Detail file entry. Data is XML with error in tag.**

   Example:

```
EDTL          13|/PO/ST_CITY/||ST_CITY|25|||||18202|3|
EMSG          13Expected end of tag 'ST_CITY'
EDAT          13
ESEG          13<BT_CITY>DAMASCUS</BT_CITY>
ELOC          13ST_CITY/PO/ST_CITY/
```

Record Layout

ELOC Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (ELOC)	5	1	5
Line #	10	6	15
Location Text	*n*	16	*EOL*

Record Tag

Contains **ELOC** to identify the type of record.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Location Text

A variable length field that contains the name of the loop, segment, composite, and element referred to by the preceding DTL record.

EMSG: Error Message Record

The EMSG record is variable length and contains the actual text of the warning, error, or information message referred to by the preceding DTL record.

The EMSG record only occurs after a DTL record. It further describes the DTL record. One EMSG record occurs for each DTL record.

EMSG Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (EMSG)	5	1	5
Line #	10	6	15
Error Message Text	*n*	16	*EOL*

Record Tag

Contains **EMSG** to identify the type of record.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Error Message Text

A variable length field that contains the text of the error, warning, or informational message referred to by the preceding DTL record.

END: End Validation Record

The END record is variable length and occurs once at the end of the detail and summary files. It contains various pieces of information about the file just validated.

END Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (END)	5	1	5
Line #	10	6	15
Error #	5	16	20
Severity	2	21	22
Date/Time	17	23	39
FileName Msg	*n*	40	*EOL*

Record Tag

Contains **END** to identify the type of record.

Line

The number of physical lines in the EDI data file. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Error

Contains the error number of the 'Analysis Completed' message (by default, this is 10006).

Severity

Contains the severity code of the 'Analysis Completed' message (by default, this is 1).

Date/Time

Contains the data and time that the validation completed in the format *MM/DD/YY HH:MM:SS* (a single space separates the date and time).

FileName Msg

Contains the name and size of the EDI data file just validated. The format of this message is 'Analysis of file *filename* complete' where *filename* is the full path and filename of the EDI data file.

ENDS: End Record for Transaction Set or Message

The ENDS record is fixed length and occurs once at the end of each transaction set or message.

ENDS Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (ENDS)	5	1	5
Line #	10	6	15
Segment Count	10	16	25
ST/SE Control #	9	26	34

Record Tag

Contains **ENDS** to identify the type of record.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Segment Count

The actual number of segments in this transaction set or message, including:

X12: ST and SE
EDIFACT: UNH and UNT

This is NOT simply a repeat of the value in the SE01 or UNT01 and it does not include:

X12: ISA, GS, GE, or IEA
EDIFACT: UNB, UNG, UNE, and UNZ

ST/SE Control

X12: The SE02 value for the transaction set for this ENDS record.

EDIFACT: The UNT02 value for the message for this ENDS record.

ESEG: Error Segment Data Record

The ESEG record is variable length and contains the actual contents of the EDI segment that caused the warning, error, or information message referred to by the preceding DTL record.

If the message refers to specific data, then an ESEG record will follow, and further describe, a DTL record. Only one ESEG record will occur for a DTL record.

Some errors don't refer to particular data, and these will have no ESEG record. For example, a missing segment won't generate a EMSG record since it has no data to show.

ESEG Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (ESEG)	5	1	5
Line #	10	6	15
Segment Data	*n*	16	*EOL*

Record Tag

Contains **ESEG** to identify the type of record.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Segment Data

Contains the complete EDI segment where the message was generated.

ETYPE: Error Type Summary Record for File

The ETYPE record is fixed length and occurs once at the end of the detail and summary files. It contains a count of messages by type, including messages referring to enveloping segments.

You can see an error's type in the Warning Levels section of the APF file being used for the validation. See **APF.pdf**.

ETYPE Record Layout				
Field		**Length**	**Start**	**End**
Record Tag (ETYPE)		5	1	5
Line #		10	6	15
Type 0 Count	type 0	10	16	25
EDI Syntax Count	type 1	10	26	35
Syntactical Requirement	type 2	10	36	45
Balancing Count	type 3	10	46	55
Situation Count	type 4	10	56	65
Code Set Count	type 5	10	66	75
Product Count	type 6	10	76	85
Payer Count	type 7	10	86	95
Partner Count	type 8	10	96	105

Record Tag

Contains **ETYPE** to identify the type of record.

Line

Contains the number of physical lines in the entire EDI data file, including all enveloping. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Type 0 Count

Contains the number of Type 0 messages in the entire EDI data file. TIBCO Foresight uses Type 0 for general messages that do not deal with EDI errors or warnings.

Error Type Count Fields

Contain the number of errors for each error type in the entire EDI data file.

ETYPS: Error Type Summary Record for Transaction Set or Message

The ETYPS record is fixed length and occurs on every SE or UNT segment in the EDI data file. It contains the total number of errors, by type, in the transaction set or message. It does not include errors in interchange or functional group enveloping.

ETYPS Record Layout				
Field		**Length**	**Start**	**End**
Record Tag (ETYPS)		5	1	5
Line #		10	6	15
Type 0 count	type 0	10	16	25
EDI Syntax Count	type 1	10	26	35
Syntactical Requirement	type 2	10	36	45
Balancing Count	type 3	10	46	55
Situation Count	type 4	10	56	65
Code Set Count	type 5	10	66	75
Product Count	type 6	10	76	85
Payer Count	type 7	10	86	95
Partner Count	type 8	10	96	105

Record Tag

Contains **ETYPS** to identify the type of record.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Type 0 Count

Contains the number of Type 0 messages in the transaction set or message. TIBCO Foresight uses Type 0 for general messages that do not deal with EDI errors or warnings.

Error Type Count Fields

Contain the number of errors, by type, in the transaction set or message.

EVALU: Element Value Record

EVALU records display the data in an EDI element. Currently, no TIBCO Foresight guidelines create EVALU records and TIBCO Foresight programs do not use them after they are created in the validation detail results file.

If you are working with XML data, refer to SVALU: Segment Value Record on page 67.

They will be generated if:

- The APF file used for validation contains EVALU=1.
- You have added EVALU records to the guideline with EDISIM Standards Editor (see below).

To create EVALU records for your own use:

1. Right-click on an element in Standards Editor and select **DSR Mark**.

   A pop-up box allows you to change the default structure ID. You will need the Standards Editor that ships with EDISIM 5.14 or later.

2. (If HIPAA)
   Merge the guideline.
   Use Instream to validate with the merged guideline.

3. (If not HIPAA)
   Copy the guideline to Instream's Database directory and validate with it.

Example: This displays the data for the data in the ST-02 element. It shows that the element is in the segment at line 3 in the data; the second element in the segment; and has a value of 0386.

```
EVALU          3|ST_02|1|2|0|0386
```

You can prevent EVALU segments from being created by setting EVALU=0 in the APF file.

EVALU Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (EVALU)	5	1	5
Line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar			
Structure ID		17	
Segment Position			
Element position			
Subelement position			
Element value			end of record

Record Tag

Contains **EVALU** to identify the type of record.

Line#

The number of physical lines in the EDI data file up to the point where the EVALU is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Structure ID

The variable name assigned to the element. See page 84.

Segment Position

The ordinal position of the segment that contains the element. This is roughly equal to the position of the segment in the guideline when used and unused segments are all taken into account.

Element position

The position of the element within its segment.

Subelement position

The position of the subelement within its composite, if the data is in a subelement. Otherwise, this will be 0.

Element data

The data in the element.

GEN: General Message Record

The GEN record is fixed length and contains informational messages generated by the validation. They mark such things as start of interchange.

Please see Displaying Version Information in the Results File on page 73 for some additional fields that Instream can add to the end of GEN records.

Please see **ICD_at_Foresight.pdf** for the format of the GEN record created by ICD business rules.

GEN Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (GEN)	5	1	5
Line #	10	6	15
Error #	5	16	20
Severity	2	21	22
Type	2	23	24
Message	*n*	25	*EOL*

Record Tag

Contains **GEN** to identify the type of record.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Error

Contains the ID number of the error message that is included within the GEN record.

Severity

Contains the severity of the error message that is included within the GEN record, one of the following numbers:

- **0** Ignore
- **1** Informational
- **2** Warning
- **3** Error
- **4** Fatal Error
- **5** User Level 1
- **6** User Level 2

Type

Contains the type of the informational message encountered. Generally, this will be 0 in a GEN record, but you can map the error number to a HIPAA type in the APF file. The valid type codes and their meanings are:

0 General message; not an EDI error or warning

1 EDI Syntax

2 Syntactical Requirement (within HIPAA Validator Desktop, this is combined with 1)

3 Balancing

4 Situation

5 Code Set

6 Product Types or Lines of Service

7 Payer Specific

8 Partner Specific (within HIPAA Validator Desktop this is shown as a **P**)

Message

A variable length field containing the actual message text.

If you have this line in the [Options] section of your $Dir.ini in Instream's Bin directory, this will be a FSUID. Please see **FSUID_and_AppDocs.pdf**.

GEN record 17021

Document types and message references:

- EDIFACT
  Document type(1), Message reference(UN)
- FlatFile
  Document type(2), Message reference(flatfile)
- XML
  Document type(3), Message reference(XML)
- TRADACOM
  Document type(4), Message reference(DT)
- X12
  The message is not written out

IDENT: Unique Identifier Record

The IDENT record contains a unique identifier (TIBCO Foresight Unique ID or FSUID) to identify a part of the input data passed to Instream.

In this example, an IDENT record is generated for each subscriber CLM segment:

```
STRUS          31|2300|0|1|1159
SVALU          31|S009|464|CLM*2235057*460.00***25:B:1*N*A*N*I*
IDENT          31|I|32f31986-efdc-11de-a384-f131e23d4046|1|
DTL            31  2300 CLM1332C023         28 5 2          1
```

IDENT Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (IDENT)	5	1	5
Line #	10	6	15
The remaining fields are separated by vertical bars			
RuleID	1	17	
FSUID	varies	19	at vertical bar
SystemID	varies		at vertical bar
Reserved	varies		end of record

Record Tag

Contains **IDENT** to identify the type of record.

Line

The number of physical lines in the EDI data file up to the point where the IDENT is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

RuleID

Shows which business rule created it. Contains:

- **M** if the record was created by a Match business rule
- **I** if the record was created by an Identify business rule

FSUID

TIBCO Foresight User ID, a unique 37-character ID. Should never be repeated in an IDENT record.

SystemID

ID to identify Instream system used for validation. It always contains 1 for the initial Instream run (this is the only system defined by default in TI). If the record contains a SystemID other than 1, define the system in TI under Settings | External System Setting before attempting to import the detail file.

Reserved

Reserved for future development.

STRT: Start Validation Record

The STRT record is variable length and occurs once in the detail and summary results files. It is the second record in the file. It contains the date and time that validation started and the name, location, and size of the EDI data file.

STRT Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (STRT)	5	1	5
Line #	10	6	15
Error #	5	16	20
Severity	2	21	22
Date/Time	17	23	39
FileName Msg	*n*	40	*EOL*

Record Tag

Contains **STRT** to identify the type of record.

Line

Always contains 0, since this record is generated before validation of any segments.

Error

Contains the error number of the 'Analysis Requested' message (10002).

Severity

Contains the severity code of the 'Analysis Requested' message (1 by default).

Date/Time

Contains the data and time that the validation was started in the format *MM/DD/YY HH:MM:SS* (a single space separates the date and time).

FileName Msg

Contains the name and size of the EDI data file being validated. The format of this message is 'Analysis requested on file *filename, size* bytes long' where *filename* is the full path and file name of the EDI data file, and *size* is the number of bytes in the file.

STRUE: Structure End Record

STRUE records mark the end of every interchange, functional group, transaction set or message, and loop or group.

They will be generated if the APF file used for validation contains STRUE=1. This is true regardless of which guideline is used for validation.

The start of the structure is marked with a STRUS record.

Example

This marks the end of loop 1000A, which occurs at line 8 in the EDI data:

```
STRUE          8|1000A|0|1|337|0:0:0:0:0:0:0:0|0:0:0:0:0:0:0:0:0:0|PER
```

STRUE Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (STRUE)	5	1	5
Line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar.			
Structure ID		17	
Document flag			
Instance			
Ending position			
Errors by severity			
Errors by type			
ID of ending segment			end of record

Record Tag

Contains **STRUE** to identify the type of record.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Structure ID

Contains the ID of the segment or loop/group that starts the structure. This field starts at character position 17 and ends with a vertical bar. See page 84.

Document flag

This marks structures that you designate as application documents. The flag will match the one in the corresponding STRUS record (see page 65).

Document flags are for your own use and for future enhancements within TIBCO Foresight products.

Instance

The first instance of the structure that is ending will be 1, the second will be 2, etc.

Ending position

The number of bytes from the beginning of the EDI data to the end of this structure.

Errors by severity

The number of errors of each severity in the structure that is ending. Colons separate each severity.

Errors by type

The number of errors of each type in the structure that is ending. Colons separate each type.

Ending segment ID

The segment tag for the last segment in the structure.

STRUS: Structure Start Record

STRUS records mark the start of every interchange, functional group, transaction set or message, and loop or group. A corresponding STRUE record marks the end of each structure.

They will be generated if the APF file used for validation contains STRUS=1. This is true regardless of which guideline is used for validation.

You can change the document flag in a STRUS record as described under **Document flag** below.

Example: This marks the start of loop 1000A, which occurs at line 6 in the EDI data:

```
STRUS          6|1000A|0|1|255
```

STRUS Record Layout			
Field	**Length**	**Start**	**End**
Record (STRUS)	5	1	5
Starting line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar.			
Structure ID		17	
Document flag			
Instance			
Starting position			end of record

Record Tag

Contains **STRUS** to identify the type of record.

Starting line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Structure ID

Contains the ID of the segment or loop/group. This field starts at character position 17 and ends with a vertical bar. See Appendix C: SVALU Record Structure IDs on page 83.

Document flag

This marks loops or groups that you consider application documents. The flag is set to 0 by default, and is set to 1 if you have designated this loop as an application document.

To designate this loop or group as an application document:

1. Right-click on the loop or group in Standards Editor and select **DSR Mark**. You will need the Standards Editor that ships with EDISIM 5.14 or later.
2. Merge the guideline with a TIBCO Foresight PD (e.g., PDSA5010834) guideline.
3. Validate with the merged guideline. The document flag for this STRUS will be 1 in the detail results file.

Document flags are for your own use and for future enhancements within TIBCO Foresight products.

Instance

If the structure occurs more than once in the transaction set or message, the instance identifies which one is being shown. For example, the first instance of a claim loop will be 1, the second instance will be 2, etc.

Starting position

This is the number of bytes from the beginning of the EDI data to the beginning of this structure.

SVALU: Segment Value Record

SVALU records display the data in an EDI segment or XML element.

They will be generated if:

- The APF file used for validation contains SVALU=1.
- The guideline used for validation is a PD guideline, a user guideline that generates SVALU records (see below), or a guideline merged from either of them.

To create SVALU records for your own use:

1. Right-click on a segment in Standards Editor and select **DSR Mark**. A pop-up box allows you to change the default structure ID. You will need the Standards Editor that ships with EDISIM 5.14 or later.
2. (If HIPAA)
   Merge the guideline.
   Use Instream to validate with the merged guideline.
3. (If not HIPAA)
   Copy the guideline to Instream's Database directory and validate with it.

Example (EDI): This displays the data for a claim segment, which has been given Structure ID S009 and is 37 lines and 464 bytes from the beginning of the EDI data.

```
SVALU         37|S009|464|CLM*1*100.00***11:A:1*N*A*N*A*********N**1
```

You can prevent SVALU segments from being created by setting SVALU=0 in the APF file. Do not do this if you are using Docsplitter.

SVALU Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (SVALU)	5	1	5
Line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar.			
Structure ID	varies	17	varies
Structure position	varies	varies	varies
Segment data	varies	varies	end of record

Record Tag

Contains **SVALU** to identify the type of record.

Line#

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Structure ID

Contains the ID of the structure. This matches the ID of the corresponding TIBCO Foresight-supplied Z-record that was discontinued in Instream version 4.5.

Appendix C: SVALU Record Structure IDs on page 83 shows where records with each Structure ID are generated.

Structure position

The ordinal position of the segment within the transaction set or message, according to the guideline.

Segment data

The entire EDI segment.

SVRTS: Error Severity Summary Record for Transaction Set or Message

The SVRTS record is fixed length and occurs on each SE or UNT segment. It contains the number of messages generated in that transaction set or message for each severity.

Severities are set in the validation's APF file. See **APF.pdf** for details.

SVRTS Record Layout				
Field		**Length**	**Start**	**End**
Record Tag (SVRTS)		5	1	5
Line #		10	6	15
Ignore Count	severity 0	10	16	25
Info Count	severity 1	10	26	35
Warning Count	severity 2	10	36	45
Error Count	severity 3	10	46	55
Fatal Count	severity 4	10	56	65
User1 Count	severity 5	10	66	75
User2 Count	severity 6	10	76	85

Record Tag

Contains **SVRTS** to identify the type of record.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Severity Error Count Fields

Contain the number of errors for each severity level.

SVRTY: Error Severity Summary Record for File

The SVRTY record is fixed length and occurs once toward the end of the summary and detail results file. It contains a count of messages by severity level.

It is usually the next-to-last record in the file, just before the End record.

SVRTY Record Layout				
Field		**Length**	**Start**	**End**
Record Tag (SVRTY)		5	1	5
Ignore Count	severity 0	10	6	15
Info Count	severity 1	10	16	25
Warning Count	severity 2	10	26	35
Error Count	severity 3	10	36	45
Fatal Count	severity 4	10	46	55
User1 Count	severity 5	10	56	65
User2 Count	severity 6	10	66	75

Record Tag

Contains **SVRTY** to identify the type of record.

Count Fields

Contain the number of errors for each severity.

VER: Version Record

The VER record is variable length and occurs once in the detail and summary results files. It is the first record in the file and contains the version number of the output file format.

VER Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (VER)	5	1	5
Version	*n*	6	*EOL*

Record Tag

Contains **VER** to identify the type of record.

Version

Contains the version number of the output file in the format *n1.n2*, where *n1* is the major version number, which can be one or more digits, and *n2* is the minor version number.

Z: Custom Data Record

You can include the contents of actual data fields in the output file. See **CustRec** in **BusinessRules.pdf**.

Z Record Layout			
Field	**Length**	**Start**	**End**
Record Tag (Z*aaaa*)	5	1	5
Line #	10	6	15
Field 1 Data			
Field *n* Data			

Record Tag

Identifies the type of record. For this record, this field always starts with **Z** followed by one-to-four alphanumeric characters.

Line

Contains the number of physical lines in the EDI data file up to the point where this record is generated. If the EDI data is wrapped to fixed length blocks, or has segment terminators that do not include a new-line character, the physical line number may not be a segment count.

Field Data

Contains the contents of the specified variable name as defined in the guideline.

Displaying Version Information in the Results File

You can display version, date, and time information at the end of the GEN records at the top of the validation detail results file. To do this, change this line in the $dir.ini file:

```
:ShowVersion=0
```

to:

```
ShowVersion=1
```

Be sure to remove the leading colon.

This causes:

- Version information to appear at the end of these GEN records near the top of the file:

```
GEN           015074 1 0Message file loaded :
C:\Foresight\InStream\bin\FSANERRS.TXT (Version=6.2.0
Date=2007/09/26 13:44)

GEN           015075 1 0Message file loaded :
C:\Foresight\InStream\bin\FSBRERRS.TXT (Version=6.2.0
Date=2007/09/26 13:44)

GEN           015010 1 0Message file loaded :
C:\Foresight\InStream\bin\CustomerFSBRERRS.TXT (Date=2004/08/26
14:27)

GEN           311001 1 0Loaded Message ORDERS from Standard
myorders (Date=2007/10/01 15:03)
```

- This record to appear:

```
GEN           015073 1 0HIPAA table loaded
C:\Foresight\InStream\Bin\fs_hipaa.dat (Version=6.1.0.1)
```

This information refers to the version and date of the file mentioned in the message.

4 HIPAA External Code Tables

Table File Server

The Table File Server is an application that loads the external code tables used during HIPAA validation. It provides faster processing of many very small EDI files, since it preloads the code tables used during validation. It may slow processing for large files. See Appendix B: Table File Server on page 79.

Extending or Modifying Code Tables

You can add to or override codes in the HIPAA external code tables provided by TIBCO Foresight.

Example situations where you might want to do this:

- You receive an error message indicating that a code is invalid, but you are accepting the code.
- Instream does not flag a code as an error, and you want to have it flagged.

For details, see **ExtendingCodeTables.pdf**.

Creating your own Code Tables

You can create your own code tables and use EDISIM Standards Editor to create rules that enforce them.

This is explained in detail in **BusinessRules.pdf**.

5 Appendix A: Return Codes

Instream Validation

Return Code	Meaning
100	Validation ran successfully.
110	Validation did not run successfully. The command line syntax is incorrect.
112	A JNI EXECPTION was encountered when using the Java API.
115	MISSING TERMINATOR in the input data. (Note: This error causes validation to stop.)
120	There was a problem loading HVInStream.DLL library.
129	$dir.ini or fsdir.ini was found but cannot be opened.
130	There was a problem with initialization of the validation engine caused by a setup problem (i.e. registry setup error) or $dir.ini or fsdir.ini is missing or contains invalid paths.
131	$dir.ini or fsdir.ini cannot be found.
132	Cannot access the "BASEROOT" specified in the $dir.ini or fsdir.ini.
133	Cannot access the Database directory.
134	Generic failure to read ini settings.
140	A critical error prevented Validation from running successfully.
150	Cannot find or open FS_HIPAA.dat in the Bin directory.
180	The activity was cancelled by the user (when using API).
185	When running validation, Docsplitter, and Response Generator together from an API, Docsplitter failed.
186	When running validation, Docsplitter, and Response Generator together from an API, Response Generator failed.

Return Code	Meaning
187	When running validation, Docsplitter, and Response Generator together from an API, Response Generator and Docsplitter both failed.
188	DataSwapper failed when run from an API.
191	A critical trading partner automation error prevented Validation from running successfully. For more information, see *.TA1 in the same directory as the validation detail results file.
195	Cannot find the error message file. Check the [ErrMsgFile] of your $dir.ini or fsdir.ini file.
200	Configuration path error when using the -c command-line parameter.
201	The input file could not be accessed. Check filenames and paths. Put quotes around paths that contain spaces.
202	Output file could not be opened. Check filenames and paths. Put quotes around paths that contain spaces.

Troubleshooting information	Notes
Validation detail results file	Check GEN records. Check for EMSG records near the top.
TA1 file	Look for a TA1 file if no guideline is specified.
Report file	Use -r command-line parameter.

Seeing Return Codes

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

UNIX echo "return code = " $?

Windows @echo [Return Code = %ERRORLEVEL%]

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

Virus Checking and TIBCO Foresight Products

Exclude all TIBCO Foresight workflow subdirectories from virus checking.

6 Appendix B: Table File Server

HIPAA only

The Table File Server is an application that loads the external code tables that can be used during HIPAA validation. It provides faster processing of many very small EDI files by preloading code tables used during validation.

For best performance when processing medium or large files, do not use the Table File Server. Instream processes files larger than 1048578 bytes without using the file server.

When using an API to run Instream validation, the Table File Server is used if it has been started before the application program that uses the API.

These files are associated with the Table File Server:

Windows	UNIX	Purpose
fsFileServer.exe	fsFileServer	Table File Server executable. Installed in Instream's **Bin** folder but can be moved to another machine.
fscint.ini	fscint.ini	INI file that Instream uses to locate **fsFileServer** or **fsFileServer.exe** on local machine or on one or more remote server locations. Must stay in Instream's **Bin** folder.
fsFileServer.ini	fsFileServer.ini	INI file that goes in same folder as **fsFileServer** or **fsFileServer.exe**. Specifies the number of connections and the port to which it listens.
fsFileServerDebug Mode.bat	n/a	Batch file to start the Table File Server in debug mode.
n/a	RunServDemo	Instream's **API\FsServer** directory. Script file to test your Table File Server setup.
n/a	startfServer shutdownfServer	In Instream's **API\FsServer** directory. Script files to start / stop the Table File Server.

fscint.ini Setup

As installed, the Table File Server runs from Instream's **Bin** directory. You can configure it to run on a different machine on the network as long as it is accessible by the main Instream installation.

To adjust its location, edit **fscint.ini**, which must be in Instream's **Bin** directory. The file lists servers and ports where Instream can find the Table File Server running. For example, you might replace `LocalHost 5850` with one or more IP addresses and ports. Multiple machines could be set to the same port.

If you list multiple machines and ports, the Table File Server will try to access the top one. If it is accessible and the maximum number of users (see MAXCONNECTION on page 81) hasn't been exceeded, then it uses that one. Otherwise, it continues down the list until it finds one that it can use.

It will look only if **fscint.ini** has **USINGFSFILESERVER=ON**.

Sample Windows fscint.ini
(in Instream's Bin directory)

```
#list of host and port to connect to
# using fsFileServer USINGFSFILESERVER should be set
to ON/OFF
USINGFSFILESERVER=ON

LocalHost  5850
255.255.255.0   5855
```

Sample UNIX fscint.ini

```
# using fsFileServer USINGFSFILESERVER should be set to
ON/OFF
USINGFSFILESERVER=ON

#list of host and port to connect to
YOURCOMPUTER  5850
255.255.255.0   5855
255.255.255.1   5855
```

fsFileServer.ini Setup

The server configuration specifies the port to which the server is "listening" and the number of connections you are allowing. The server port number must match in **fscint.ini** and **fsFileServer.ini**. It can listen on only one port.

Example fsFileServer.ini.

(put in same directory as File Server)

```
#server port
PORT      5850
#max connection
MAXCONNECTION     8
```

Starting the Table File Server

Windows

1. Start the Table File Server by running **fsFileServer.exe**.

   Debug mode: Run fsFileServer.exe with a **–d** parameter and start it from a batch file so the Table File Server maintains control of the console window.

   To do this, you can use **fsFileServerDebugMode.bat** in Instream's API\FsServer directory.

2. Run Instream with the Table File Server by including the **-f** command line parameter when starting Instream.

3. Stop the Table File Server by running **ShutdownfsFileServer.bat** in the API/FsServer directory.

UNIX

1. Start the Table File Server by running **./startfServer** from Instream's **API/FsServer** directory.

2. Run Instream with the Table File Server by including the **-f** command line parameter when starting Instream.

3. Stop the Table File Server by running **shutdownfServer** in the API\FsServer directory.

Table File Server Start-up Scripts

These scripts are installed under Instream's API/FsServer directory.

- **Starting the Table File Server with Debugging Turned On**

  Windows: fsFileServerDebugMode.bat
  UNIX: startfServer

- **Running Instream with the Table File Server**

  Windows: RunFileServerDemo.bat
  UNIX: RunServDemo

  This should yield detail and summary files in Instream's output directory. If your **Dir.ini** file has **ShowVersion=1**, your detail files will have a GEN record like this:

```
GEN           015080 1 0Connect to fsFileServer on Host
```

- **Stopping the Table File Server**

  Windows: ShutdownfsFileServer.bat
  UNIX: shutdownfServer

How to tell if the Table File Server is running

Use Task Manager to see whether **fsFileServer.exe** is running.

7 Appendix C: SVALU Record Structure IDs

Structure IDs

HIPAA only

The SVALU record in the detail results file contains a Structure ID (see page 65) that identifies where the record was generated.

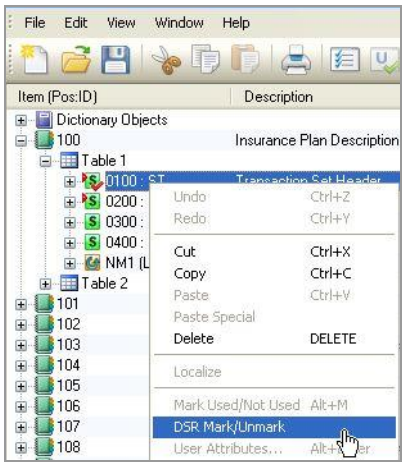
Example:

```
STRUS        6|1000A|0|1|255
```

This ID matches the ID of the old custom Z records that were generated by Instream 4.4 and earlier.

The sources of SVALU records are:

- Your own SVALU records that you created in EDISIM Standards Editor by right-clicking on a record and choosing DSR Mark:



In this case, you assign a variable to the location. This variable appears as the structure ID in the SVALU record.

- Certain HIPAA guidelines that place SVALU records in the detail results file. The chart below lists each structure ID supplied by these guidelines and shows where they are generated.

HIPAA Structure ID Chart

All Documents	
ID	**Called on segment …**
ISA1	Interchange Information (ISA)
GSSG	Functional Group Header (GS)
STST	Transaction Set Header (ST)
TRSE	Transaction Set Trailer (SE segment)

270 Eligibility, Coverage or Benefit Inquiry (PDSA270 and PDSA5010270X279)	
ID	**Called on segment …**
0021	Transaction Type Code (BHT)
ISST	Information Source Level (Loop 2000A, HL segment)
HLIS	Information Source Level (Loop 2000A, HL segment)
ISNM	Information Source Name (Loop 2100A, NM1)
HLIR	Information Receiver Level (Loop 2000B, HL segment)
IRNM	Information Receiver Name (Loop 2100B, NM1 segment)
SBST	Subscriber Level (Loop 2000C, HL segment)
SUBTRN	X12_5010: Subscriber Trace Number (Loop 2000C, TRN segment)
SBNM	Subscriber Name (Loop 2100C, NM1 segment)
SUBPRV	Provider Information (Loop 2100C, PRV segment)
SBSV	Subscriber Eligibility (Loop 2110C, EQ segment)
DPST	Dependent Level (Loop 2000D, HL segment)
DEPTRN	X12_5010: Dependent Trace Number (Loop 2000D, TRN segment)
DPNM	Dependent Name (Loop 2100D, NM1 segment)
DEPPRV	Provider Information (Loop 2100D, PRV segment)
DPSV	Dependent Eligibility (Loop 2110D, EQ segment)

271 Eligibility, Coverage or Benefit Response to Inquiry (PDSA271 and PDSA5010271X279)	
ID	**Called on segment …**
0021	Transaction Set Purpose Code (BHT segment)
IHLIS	Information Source Level (Loop 2000A, HL segment)
ISNM	Information Source Name (Loop 2100A, NM1 segment)
HLIR	Information Receiver Level (Loop 2000B, HL segment)
IRNM	Information Receiver Name (Loop 2100B, NM1 segment)
SBST	Subscriber Level (Loop 2000C, HL segment)
SUBTRN	X12_5010: Subscriber Trace Number (Loop 2000C, TRN segment)
SBNM	Subscriber Name (Loop 2100C, NM1 segment)
SBSV	Subscriber Eligibility (Loop 2110C, EB segment)
DPST	Dependent Level (Loop 2000D, HL segment)
DEPTRN	X12_5010: Dependent Trace Number (Loop 2000D, TRN segment)
DPNM	Dependent Name (Loop 2100D, NM1 segment)
DPSV	Dependent Eligibility (Loop 2110D, EB segment)

275 Patient Information (PDSA5010275X210.STD and PDSX5010275X210.STD)	
ID	**Called on segment …**
BGNSEGMENT	Beginning Segment (BGN segment)
1000APAYER	Payer Name (Loop 1000A, NM1 segment)
1000BSUBMITTER	Submitter Information (Loop 1000B, NM1 segment)
1000CPROVIDER	Provider Name Information (Loop 1000C, NM1 segment)
1100CPROVIDERID	Provider Identification (Loop 1100C, NX1 segment)
1000DPATIENT	Patient Name (Loop 1000D, NM1 segment)
1000DCONTROLNUMBER	Patient Control Number (Loop 1000D, REF segment)
1000DDTPClaimServiceDate	Claim Service Date (Loop 1000D, DTP segment)
2000ATRN	Payer Claim Control Number/Provider Attachment Control Number (Loop 2000A, TRN segment)
2000ANUMBER	Assigned Number (Loop 2000A, LX segment)
2000ASTC	Status Information (Loop 2000A, STC segment)
2100ADTP	Service Line Date of Service (Loop 2100A, DTP segment)
2100BDTP	Additional Information Submission Date (Loop 2100B, DTP segment)
2110BEFI	Electronic Format Identification (Loop 2110B, EFI segment)

275-X314 Health Care Claim or Encounter (PDSA6020-275X314.STD and PDSX6020-275X314.STD)	
ID	**Called on segment …**
BGNSEGMENT	BGN - Beginning Segment
1000APAYER	Payer Name (Loop 1000A, NM1 segment)
1000BSUBMITTER	Submitter Information (Loop 1000B, NM1 segment)
1000CPROVIDER	Provider Name Information (Loop 1000C, NM segment 1)
1000DPATIENT	Patient Name (Loop 1000D, NM1 segment)
1000DCONTROLNUMBER	Provider's Assigned Claim Identifier (Loop 1000D, REF segment)
1000DDTPClaimServiceDate	Claim Service Date (Loop 1000D, DTP segment)
2000ANUMBER	Assigned Number (Loop 2000A, LX segment)
2000ATRN	Payer Claim Control Trace Number/Provider Attachment Control Trace Number (Loop 2000A, TRN segment)
2000ASTC	Status Information (Loop 2000A, STC segment)
2100ADTP	Service Line Service Date (Loop 2100A, DTP segment)
2100BDTP	Additional Information Submitted Date (Loop 2100B, DTP segment)
2110BOOI	Associated Object Type Identification (Loop 2110B, OOI segment)

275-X316 Health Care Services Review (PDSA6020-275X316.STD and PDSX6020-275X316.STD)	
ID	**Called on segment …**
BGNSEGMENT	BGN - Beginning Segment
1000AINFOSOURCENAME	Information Source Name (Loop 1000A, NM1 segment)
1000BRECEIVERNAME	Information Receiver Name (Loop 1000B, NM1 segment)
1000CPATIENTNAME	Patient Name (Loop 1000C, NM1 segment)
1000CCONTROLNUMBER	Patient Account Number (Loop 1000C, REF segment)
2000ANUMBER	Assigned Number (Loop 2000A, LX segment)
2000ATRN	Payer Claim Control Trace Number/Provider Attachment Control Trace Number (Loop 2000A, TRN segment)
2000ASTC	Status Information (Loop 2000A, STC segment)
2100ADTP	Additional Information Submitted Date (Loop 2100A, DTP segment) -
2110AOOI	Associated Object Type Identification (Loop 2110B, OOI segment)

276 Health Care Claim Status Request (PDSA276 and PDSA5010-276X212)	
ID	**Called on segment …**
0021	Beginning of Hierarchical Transaction (BHT segment)
HLIS	Information Source Level Starts (Loop 2000A, HL segment)
ISPN	Payer Name (Loop 2100A, NM1 segment)
HLIR	Information Receiver Level Starts (Loop 2000B, HL segment)
IRNM	Information Receiver Name (Loop 2100B, NM1 segment)
HLSP	Service Provider Level Starts (Loop 2000C, HL segment)
SPNM	Provider Name (Loop 2100C, NM1 segment)
HLSB	Subscriber Level Starts (Loop 2000D, HL segment)
SBNM	Subscriber Name (Loop 2100D, NM1 segment)
SBTN	Claim Submitter Trace Number (Loop 2200D, TRN segment)
SPCREF	Payer Claim Identification Number (Loop 2200D, REF segment)
SDOS	Claim Service Date (Loop 2200D, DTP segment)
SBSV	Service Line Information Starts (Loop 2210D, SVC segment)
HLDP	Dependent Level Starts (Loop 2000E, HL segment)
DPNM	Dependent Name (Loop 2100E, NM1 segment)
DPTN	Claim Submitter Trace Number (Loop 2200E, TRN segment)
DPCREF	Payer Claim Identification Number (Loop 2200E, REF segment)
DDOS	Claim Service Date (Loop 2200E, DTP segment)
DPSV	Service Line Information Starts (Loop 2210E, SVC segment)

277 Health Care Claim Status Response (PDSA277 and PDSA5010277X212)	
ID	**Called on segment …**
0021	Beginning of Hierarchical Transaction (BHT segment)
HLIS	Information Source Level Starts (Loop 2000A, HL segment)
ISPN	Payer Name (Loop 2100A, NM1 segment)
HLIR	Information Receiver Level Starts (Loop 2000B, HL segment)
IRNM	Information Receiver Name (Loop 2100B, NM1 segment)
HLSP	Service Provider Level Starts (Loop 2000C, HL segment)
SPNM	Provider Name (Loop 2100C, NM1 segment)
HLSB	Subscriber Level Starts (Loop 2000D, HL segment)

277 Health Care Claim Status Response (PDSA277 and PDSA5010277X212)	
ID	**Called on segment …**
SDMG	X12-4010: Subscriber demographic Information (Loop 2000D, DMG segment)
SBNM	Subscriber Name (Loop 2100D, NM1 segment)
IRTRN	X12-5010: Information Receiver Trace Identifier (Loop 2200B, TRN segment)
IRSTC	X12-5010: Information Receiver Status Information (Loop 2200B, STC segment)
SPTRN	X12-5010: Provider of Service Trace Identifier (Loop 2200C, TRN segment)
SPSTC	X12-5010: Provider Status Information (Loop 2200C, STC segment)
SBTN	Claim Submitter Trace Number (Loop 2200D, TRN segment)
SSTC	Claim Level Status Information (Loop 2200D, STC segment)
SPCREF	X12-4010: Payer Claim ID (Loop 2200D, REF segment) X12-5010: Payer Claim Control Number (Loop 2200D, REF segment)
SLREFCLAIMID	X12_5010: Claim Identification Number For Clearinghouses and Other Transmission Intermediaries (Loop 2200D, REF segment)
SBTREF	Institutional Bill Type Identification (Loop 2200D, REF segment)
SDOS	Claim Service Date (Loop 2200D, DTP segment)
SBSV	Service Line Information Starts (Loop 2220D, SVC segment)
HLDP	X12-4010: Dependent Level Starts (Loop 2000E, HL segment)
DDMG	X12-4010: Dependent demographic Information (Loop 2000E, DMG segment)
DPNM	Dependent Name (Loop 2100E, NM1 segment)
DPTN	Claim Submitter Trace Number (Loop 2200E, TRN segment)
DSTC	Claim Level Status Information (Loop 2200E, STC segment)
DPCREF	X12-4010: Payer Claim ID (Loop 2200E, REF segment) X12-5010: Payer Claim Control Number (Loop 2200E, REF segment)
DLREFCLAIMID	X12_5010: Claim Identification Number For Clearinghouses and Other Transmission Intermediaries (Loop 2200E, REF segment)

277 Health Care Claim Status Response (PDSA277 and PDSA5010277X212)	
ID	**Called on segment …**
DBTREF	Institutional Bill Type Identification (Loop 2200E, REF segment)
DDOS	Claim Service Date (Loop 2200E, DTP segment)
DPSV	Service Line Information Starts (Loop 2220E, SVC segment)

277-X313 Health Care Claim Request for Additional Information (PDSA6020-277-X313.STD and PDSX6020-277-X313.STD)	
ID	**Called on segment …**
HLIS	Information Source Level Starts (Loop 2000A, HL segment)
HLIR	Information Receiver Level Starts (Loop 2000B, HL segment)
HLSP	Service Provider Level Starts (Loop 2000C, HL segment)
HLSB	Patient Level Starts (Loop 2000D, HL segment)
DPTN	Payer Claim Control Number (Loop 2200D, TRN segment)
DPSV	Service Line Information Starts (Loop 2220D, SVC segment)

277CA (5010) Health Care Claim Acknowledgement (PDSA5010277CAX214.STD and PDSX5010277CAX214.STD)	
ID	**Called on segment …**
0021	Beginning of Hierarchical Transaction (BHT segment)
HLIS	Information Source Level (Loop 2000A, HL segment)
ISPN	Information Source Name (Loop 2100A, NM1 segment)
InfoSourceTRN	Transmission Receipt Control Identifier (Loop 2200A, TRN segment)
HLIR	Information Receiver Level (Loop 2000B, HL segment)
IRNM	Information Receiver Name (Loop 2100B, NM1 segment)
IRTRN	Information Receiver Application Trace Identifier (Loop 2200B, TRN segment
IRSTC	Information Receiver Status Information (Loop 2200B, STC segment)
HLBP	Billing Provider of Service Level (Loop 2000C, HL segment)
BPNM	Billing Provider Name (Loop 2100C, NM1 segment)
BPTRN	Provider of Service Information Trace Identifier (Loop 2200C, TRN segment)
BPSTC	Billing Provider Status Information (Loop 2200C, STC segment)
HLPT	Patient Level (Loop 2000D, HL segment)
PTNM	Patient Name (Loop 2100D, NM1 segment)
PTTN	Claim Status Tracking Number (Loop 2200D, TRN segment)
PTSTC	Claim Level Status Information (Loop 2200D, STC segment
PTPCREF	Payer Claim Control Number (Loop 2200D, REF segment)
PTREFCLAIMID	Claim Identifier Number For Clearinghouse and Other (Loop 2200D, REF segment)
PTBTREF	Institutional Bill Type Identification (Loop 2200D, REF segment)
PTDOS	Claim Level Service Date (Loop 2200D, DTP segment)
PTSV	Service Line Information (Loop 2220D, SVC segment)

277U (5010) Health Care Claim Status Response (PDSX5010-277UX212 only)	
ID	**Called on segment …**
0021	Beginning of Hierarchical Transaction (BHT segment)
HLIS	Information Source Level (Loop 2000A, HL segment)
ISPN	Information Source Name (Loop 2100A, NM1 segment)
HLIR	Information Receiver Level (Loop 2000B, HL segment)
IRNM	Information Receiver Name (Loop 2100B, NM1 segment)
IRTRN	Information Receiver Application Trace Identifier (Loop 2200B, TRN segment
IRSTC	Information Receiver Status Information (Loop 2200B, STC segment)
HLSP	Service Provider Level (Loop 2000C, HL segment)
SPNM	Provider Name (Loop 2100C, NM1 segment)
SPTRN	Provider of Service Trace Identifier (Loop 2200C, TRN segment)
SPSTC	Provider Status Information (Loop 2200C, STC segment)
HLSB	Subscriber Level (Loop 2000D, HL segment)
SBNM	Subscriber Name (Loop 2100D, NM1 segment)
SBTN	Claim Status Tracking Number (Loop 2200D, TRN segment)
SSTC	Claim Level Status Information (Loop 2200D, STC segment
SPCREF	Payer Claim Control Number (Loop 2200D, REF segment)
SBTREF	Institutional Bill Type Identification (Loop 2200D, REF segment)
SLREFCLAIMID	Claim Identifier Number For Clearinghouse and Other (Loop 2200D, REF segment)
SDOS	Claim Level Service Date (Loop 2200D, DTP segment)
SBSV	Service Line Information (Loop 2220D, SVC segment)
HLDP	Dependent Level (Loop 2000E, HL segment)
DPNM	Dependent Name (Loop 2100E, NM1 segment)
DPTN	Claim Status Tracking Number (Loop 2200E, TRN segment)
DSTC	Claim Level Status Information (Loop 2200E, STC segment)
DPCREF	Payer Claim Control Number (Loop 2200E, REF segment)
DBTREF	Institutional Bill Type Identification (Loop 2200E, REF segment)
DLREFCLAIMID	Claim Identification Number For Clearinghouses and Other Transmission Intermediaries (Loop 2200E, REF segment)
DDOS	Claim Service Date (Loop 2200E, DTP segment)
DPSV	Service Line Information (Loop 2220E, SVC segment)

278X215I: Health Care Services Review Information - Inquiry (PDSA5010-278X215I.STD and PDSX5010-278X215I.STD)	
ID	**Called on segment …**
0021	Beginning of Hierarchical Transaction (BHT segment)
HLUM	Utilization Management Organization (UMO) Level (Loop 2000A, HL segment)
UMNM	Utilization Management Organization (UMO) Name (Loop 2010A, NM1 segment)
HLRQ	Requester Level (Loop 2000, HL segment)
RQNM	Requester Name (Loop 2010, NM1 segment)
HLSub	Subscriber Hierarchical Level (Loop 2000C, HL segment)
SUBSCRIBERTRN	Subscriber Trace Number (Loop 2000C, TRN segment)
SBNM	Subscriber Name (Loop 2010C, NM1 segment)
HLDP	Dependent Hierarchical Level (Loop 2000D, HL segment)
DEPENDENTTRN	Dependent Trace Number (Loop 2000D, TRN segment)
DPNM:	Dependent Name (Loop 2010D, NM1 segment)
HLPELevel	Patient Event Level (Loop 2000E, HL segment)
PETRN	Patient Event Tracking Number (Loop 2000E, TRN segment)
PEDATE	Event Date (Loop 2000E, DTP segment)
PEProvName	Patient Event Provider Name (Loop 2010EA, NM1 segment)
HLSS	Service Level (Loop 2000F, HL segment)
SSTN	Service Trace Number (Loop 2000F, TRN segment)
ServProvName	Service Provider Name (Loop 2010F, NM1 segment)

278X215R: Health Care Services Review Information - Response (PDSA5010-278X215R.STD and PDSX5010-278X215R.STD)	
ID	**Called on segment …**
0021	Beginning of Hierarchical Transaction (BHT segment)
HLUM	Utilization Management Organization (UMO) Level (Loop 2000A, HL segment)
UMA1	Request Validation (Loop 2000A, AAA segment)
UMNM	Utilization Management Organization (UMO) Name (Loop 2010A, NM1 segment)
UMA2	Utilization Management Organization (UMO) Request Validation (Loop 2010A, AAA segment)
HLRQ	Requester Level (Loop 2000B, HL segment)
RQNM	Requester Name (Loop 2010B, NM1 segment)
RQAA	Requester Request Validation (Loop 2010B, AAA segment)
HLSB	Subscriber Hierarchical Level (Loop 2000C, HL segment)
SUBSCRIBERTRN	Subscriber Trace Number (Loop 2000C, TRN segment)
SBNM	Subscriber Name (Loop 2010C, NM1 segment)
SBA2	Subscriber Request Validation (Loop 2010C, AAA segment)
HLDP	Dependent Hierarchical Level (Loop 2000D, HL segment)
DEPENDENTTRN	Dependent Trace Number (Loop 2000D, TRN segment)
DPNM	Dependent Name(Loop 2010D, NM1 segment)
DPA2	Dependent Request Validation (Loop 2010D, AAA segment)
HLPELevel	Patient Event Level (Loop 2000E, HL segment)
PETRN	Patient Event Tracking Number (Loop 2000E, TRN segment)
PEAAA1	Patient Event Request Validation (Loop 2000E, AAA segment)
PEDATE	Event Date (Loop 2000E, DTP segment)
PEProvName	Patient Event Provider Name (Loop 2010EA, NM1 segment)
PEProvAAA1	Patient Event Provider Request Validation (Loop 2010EA, AAA segment)
HLSS	Service Level (Loop 2000F, HL segment)
SSTN	Service Trace Number (Loop 2000F, TRN segment)
SSAA	Service Request Validation (Loop 2000F, AAA segment)
SPNM	Service Provider Name (Loop 2010F, NM1 segment)
SPAA	Service Provider Request Validation (Loop 2010F, AAA segment)

278X216A: Health Care Services Review Information - Acknowledgement (PDSA5010-278X216A.STD and PDSX5010-278X216A.STD)	
ID	**Called on segment …**
0021	Beginning of Hierarchical Transaction (BHT segment)
HLUM	Information Source Level (Loop 2000A, HL segment)
UMA1	Request Validation (Loop 2000A, AAA segment)
UMNM	Information Source Name (Loop 2010A, NM1 segment)
UMA2	Information Source Validation (Loop 2010A, AAA segment)
HLRQ	Information Receiver Level (Loop 2000B, HL segment)
RQNM	Information Receiver Name (Loop 2010B, NM1 segment)
RQAA	Information Receiver Notification Validation (Loop 2010B, AAA segment)
HLSB	Subscriber Hierarchical Level (Loop 2000C, HL segment)
SUBSCRIBERTRN	Subscriber Trace Number (Loop 2000C, TRN segment)
SBA1	Subscriber Notification Validation (Loop 2000C, AAA segment)
SBNM	Subscriber Name (Loop 2010C, NM1 segment)
SBA2	Subscriber Request Validation (Loop 2010C, AAA segment)
HLDP	Dependent Hierarchical Level (Loop 2000D, HL segment)
DEPENDENTTRN	Dependent Trace Number (Loop 2000D, TRN segment)
DPA1	Dependent Notification Validation (Loop 2000D, AAA segment)
DPNM	Dependent Name (Loop 2010D, NM1 segment)
DPA2	Dependent Notification Validation (Loop 2010D, AAA segment)
HLPELevel	Patient Event Level (Loop 2000E, HL segment)
PETRN	Patient Event Tracking Number (Loop 2000E, TRN segment)
PEAAA1	Patient Event Request Validation (Loop 2000E, AA segment)
PEDATE	Event Date (Loop 2000E, DTP segment)
PEProvName	Patient Event Provider Name (Loop 2010E, NM1 segment)
PEProvAAA1	Patient Event Provider Request Validation (Loop 2010E, AAA segment)
HLSS	Service Level (Loop 2000F, HL segment)
SSTN	Service Trace Number (Loop 2000F, TRN segment)
SSAA	Service Request Validation (Loop 2000F, AAA segment)
SPNM	Service Provider Name (Loop 2010F, NM1 segment)
SPAA	Service Provider Request Validation (Loop 2010F, AAA segment)

278X216N: Health Care Services Review Information - Notification (PDSA5010-278X216N.STD and PDSX5010-278X216N.STD)	
ID	**Called on segment …**
0021	Beginning of Hierarchical Transaction (BHT segment)
HLUM	Information Source Level (Loop 2000A, HL segment)
UMNM	Information Source Name (Loop 2010A, NM1 segment)
HLRQ	Information Receiver Level (Loop 2000B, HL segment)
RQNM	Information Receiver Name (Loop 2010B, NM1 segment)
HLSB	Subscriber Hierarchical Level (Loop 2000C, HL segment)
SBNM	Subscriber Name (Loop 2010C, NM1 segment)
HLDP	Dependent Hierarchical Level (Loop 2000D, HL segment)
DPNM	Dependent Name (Loop 2010D, NM1 segment)
HLPELevel	Patient Event Level (Loop 2000E, HL segment)
PETRN	Patient Event Tracking Number (Loop 2000E, TRN segment)
PEDATE	Event Date (Loop 2000E, DTP segment)
PEProvName	Patient Event Provider Name (Loop 2010E, NM1 segment)
HLSS	Service Level (Loop 2000F, HL segment)
SSTN	Service Trace Number (Loop 2000F, TRN segment)
SPNM	Service Provider Name (Loop 2010F, NM1 segment)

278 Health Care Services Review Information Request (PDA278RQ and PDSA5010-278X217Q)	
ID	**Called on segment …**
0021	Beginning of Hierarchical Transaction (BHT segment)
UMST	Utilization Management Organization (UMO) Level Starts (Loop 2000A, HL segment)
HLUM	Utilization Management Organization (UMO) Level (Loop 2000A, HL segment)
UMNM	Utilization Management Organization (UMO) Name (Loop 2010A, NM1 segment)
HLRQ	Requester Level (Loop 2000B, HL segment)
RQNM	Requester Name (Loop 2010B, NM1 segment)
SBST	Subscriber Level Starts (Loop 2000C, HL segment)
HLSB	Subscriber Level (Loop 2000C, HL segment)
SBTN	XX12:4010: Patient Event Tracking Number (Loop 2000C, TRN segment) X12-5010: Not used
SBNM	X12-4010: Subscriber Name (Loop 2010CA, NM1 segment) X12-5010: Subscriber Name (Loop 2010C, NM1 segment)
HLDP	Dependent Level (Loop 2000D, HL segment)
DPTN	X12-4010: Patient Event Tracking Number (Loop 2000D, TRN segment) X12-5010: Not used
DPNM	X12-4010: Dependent Name (Loop 2010DA, NM1 segment) X12-5010: Dependent Name (Loop 2010D, NM1 segment)
HLPELevel	X12-4010: Not used X12-5010: Patient Event Level (Loop 2000E, HL segment)
PETRN	X12-4010: Not used X12-5010: Patient Event Tracking Number (Loop 2000E, TRN segment)
PEDATE	X12-5010: Event Date (Loop 2000E, DTP segment)

278 Health Care Services Review Information Request (PDA278RQ and PDSA5010-278X217Q)	
ID	**Called on segment …**
PEProvName	X12-4010: Not used X12-5010: Patient Event Provider Name (Loop 2010EA, NM1 segment)
HLSP	Service Provider Level (Loop 2000E, HL segment)
SPNM	Service Provider Name (Loop 2010E, NM1 segment)
HLSS	Service Level (Loop 2000F, HL segment)
ServProvName	X12-4010: Not used X12-5010: Service Provider Name (Loop 2010F, NM1 segment)
SSTN	Service Trace Number (Loop 2000F, TRN segment)

278 Health Care Services Review Information Response (PDA278RP and PDSA5010-278X217R)	
ID	**Called on segment …**
0021	Beginning of Hierarchical Transaction (BHT segment)
HLUM	Utilization Management Organization (UMO) Level (Loop 2000A, HL segment)
UMA1	Request Validation (Loop 2000A, AAA segment)
UMNM	Utilization Management Organization (UMO) Name (Loop 2010A, NM1 segment)
UMA2	Utilization Management Organization (UMO) Request Validation (Loop 2010A, AAA segment)
HLRQ	Requester Level (Loop 2000B, HL segment)
RQNM	Requester Name (Loop 2010B, NM1)
RQAA	Requester Request Validation (Loop 2010B, AAA segment)
HLSB	Subscriber Level (Loop 2000C, HL segment)
SBTN	Patient Event Tracking Number (Loop 2000C, TRN segment)
SBA1	Subscriber Request Validation (Loop 2000C, AAA segment)
SBNM	X12-4010: Subscriber Name (Loop 2010CA, NM1 segment) X12-5010: Subscriber Name (Loop 2010C, NM1 segment)
SBA2	X12-4010: Subscriber Request Validation (Loop 2010CA, AAA segment) X12-5010: Subscriber Request Validation (Loop 2010C, AAA segment)
HLDP	Dependent Level (Loop 2000D, HL segment)

278 Health Care Services Review Information Response (PDA278RP and PDSA5010-278X217R)	
ID	**Called on segment …**
DPTN	Patient Event Tracking Number (Loop 2000D, TRN segment)
DPA1	Dependent Request Validation (Loop 2000D, AAA segment)
DPNM	X12-4010: Dependent Name (Loop 2010DA, NM1 segment) X12-5010: Dependent Name (Loop 2010D, NM1 segment)
DPA2	X12-4010: Dependent Request Validation (Loop 2010DA, AAA segment) X12-5010: Dependent Request Validation (Loop 2010D, AAA segment)
HLSP	X12-4010: Service Provider Level (Loop 2000E, HL segment) X12-5010: See HLPELevel
HLPELevel	X12-4010: Not used; see HLSP X12-5010: Patient Event Level (Loop 2000E, HL segment)
PEAAA1	X12-4010: Not used X12-5010: Patient Event Request Validation (Loop 2000E, AAA segment)
SPNM	Service Provider Name (Loop 2010E, NM1 segment)
PEProvName	X12-4010: Not used X12-5010: Patient Event Provider Name (Loop 2010EA, NM1 segment) (See ProvAAA1 also)
ProvAAA1	X12-4010: Not used X12-5010: Patient Event Provider Request Validation (Loop 2010EA, AAA segment)
SPAA	Service Provider Request Validation (Loop 2010E, AAA segment)
HLSS	Service Level (Loop 2000F, HL segment)
SSTN	Service Trace Number (Loop 2000F, TRN segment)
SSAA	Service Request Validation (Loop 2000F, AAA segment)
SPNM	X12-4010: Not used X12-5010: Service Provider Name (Loop 2010FA, NM1 segment)

278 Health Care Services Review Information Response (PDA278RP and PDSA5010-278X217R)	
ID	**Called on segment …**
SPAA	X12-4010: Not used X12-5010: Service Provider Request Validation (Loop 2010FA, AAA segment)

820 Premium Payments (PDSA820) (4010)	
ID	**Called on segment …**
0010	Financial Information (BPR segment)
TRN	Trace (Header Table 1, TRN segment)
0002	Premium Receiver (Loop 1000A, N1 segment)
0004	Premium Payer (Loop 1000B, N1 segment)
0006	Organizational Summary Remittance Starts (Loop 2000A, ENT segment)
0011	Organization Summary Remittance Detail (Loop 2300A, RMR segment)
0013	Organization Summary Remittance Level Adjustment (Loop 2320A, ADX01 segment)
0008	Individual Remittance (Loop 2000B), ENT segment)
0014	Individual Premium Remittance Detail (Loop 2300B, RMR segment)
0016	Individual Premium Adjustment (Loop 2320B, ADX01 segment)

820X218 Premium Payments (PDSA5010820X218)	
ID	**Called on segment …**
0010	Financial Information (BPR segment)
TRN	Trace (Header Table 1, TRN segment)
0002	Premium Receiver (Loop 1000A, N1 segment)
0004	Premium Payer (Loop 1000B, N1 segment)
0006	Organizational Summary Remittance Starts (Loop 2000A, ENT segment)
0011	Organization Summary Remittance Detail (Loop 2300A, RMR segment)
0013	Organization Summary Remittance Level Adjustment (Loop 2320A, ADX01 segment)
0008	Individual Remittance (Loop 2000B), ENT segment)
2100BINDName	2100B, NM1 Individual Name
0014	Individual Premium Remittance Detail (Loop 2300B, RMR segment)
0016	Individual Premium Adjustment (Loop 2320B, ADX01 segment)

820X306 Premium Payments (PDSA5010820X306)	
ID	**Called on segment …**
0010	Financial Information (BPR segment)
TRN	Trace (Header Table 1, TRN segment)
0002	Premium Receiver (Loop 1000A, N1 segment)
0004	Premium Payer (Loop 1000B, N1 segment)
0006	Loop 2000, ENT= Remittance Information
2100IndName	Loop 2100, NM1 Individual Name
0011	Loop 2300, RMR Remittance Detail

834 Benefit Enrollment (PDSA834 and PDSA5010-834)	
ID	**Called on segment …**
BGNS	Beginning segment (BGN segment)
ZTPN	Transaction Set Policy Number (REF segment)
ZFED	File Effective Date (DTP segment)
0002	Sponsor Name (Loop 1000A, N1 segment)
0004	Payer Name (Loop 1000B, N1 segment)
0006	TPA/Broker Name Starts (Loop 1000C, N1 segment)
0007	Member Level Detail Starts (Loop 2000, INS segment)
0008	Subscriber Number (Loop 2000, REF segment)
ZMLD	Member Level Dates (Loop 2000, DTP segment)
0010	Member Name (Loop 2100A, NM1 segment)
ZPER	Member Communications Numbers (Loop 2100A, PER segment)
ZN3	Member Residence Street Address (Loop 2100A, N3 segment)
N4	Member Residence City, State, ZIP Code (Loop 2100A, N4 segment)
ZDMG	Member Demographics (Loop 2100A, DMG segment)
ZIMN	Incorrect Member Name (Loop 2100B, NM1 segment)
ZIDG	Incorrect Member Demographic (Loop 2100B, DMG segment)
ZDSB	Disability Information (Loop 2200, DSB segment)
ZDDT	Disability Date (Loop 2200, DTP segment)
PTHD	Health Coverage Starts (Loop 2300, HD segment)
HDDT	Health Coverage Dates (Loop 2300, DTP segment)
HCPN	Health Coverage Policy Number (Loop 2300, REF segment)
0013	Provider Information (Loop 2310, LX segment)
0014	Provider Name (Loop 2310, NM1segment)
ZCOB	Coordination of Benefits (Loop 2320, COB segment)
ZCN1	X12-4010: Other Insurance Company Name (Loop 2320, N1 segment) X12-5010: Coordination of Benefits Related Entity (Loop 2330, NM1 segment)
ZCDT	Coordination of Benefits Eligibility Dates (Loop 2320, DTP segment)
ZPMN	Member Policy Number (Loop 2000, REF segment where REF01='1L')

835 Health Care Claim Payment/Advice (PDSA835, PDSX835, PDSA5010835, PDSX5010835)	
ID	**Called on segment …**
0002	Financial Information (BPR segment)
STRN	Re-association Trace Number (Header TRN segment)
SREF	Receiver Identification (Header REF segment)
0004	Payer Name and ID (Loop 1000A, N1 segment)
0006	Payee Name and ID (Loop 1000B, N1 segment)
0020	Payee Additional Identification (Loop 1000B, REF segment)
0007	LX (Loop 2000, LX segment)
TS3	Provider Summary Information (Loop 2000, TS3 segment)
TS2	Provider Supplemental Summary Information (Loop 2000, CLP segment)
0009	Claim Payment Information (Loop 2100, CLP segment)
0011	Claim Adjustment (Loop 2100, CAS segment)
Patient	Patient Name (Loop 2100, NM1 Patient Name segment)
Insured	Insured Name (Loop 2100, NM1 Insured Name segment)
2100DTM	Claim Date (Loop 2100, DTM segment)
REND	Rendering Provider (Loop 2100, NM1 Service Provider name)
2110DTM	Service Date (Loop 2100, DTM segment)
MIA	Inpatient Adjudication Information (Loop 2100 Inpatient Adjudication Information)
0017	Service Payment Information (Loop 2110, SVC segment)
0014	Service Adjustment (Loop 2110, CAS segment)
0018	Provider Adjustment Starts (Table 3, PLB segment)

837 – All 837s (PDSA837D and PDSA5010837D, PDSA837I and PDSA5010837I, PDSA837P and PDSA5010837P)	
These records are used by Docsplitter, by Transaction Insight, and by Response Generator, which uses them to create 277s from data in 837s. In the chart below, “Subscriber loop” means 2000B and “Dependent loop” means 2000C.	
ID	**Called on segment …**
0021	Transaction Type Code (BHT segment)
ZRTR	Transmission Type Identifier (Header REF segment)
ZRT	Trading Partner Information (Loop 1000A, NM1 segment)
ZRP	Payer information (Loop 1000B, NM1 segment)
PRST	Billing/Pay-to Provider HL (Loop 2000A, HL segment)
2000APRV	Billing Provider Specialty Information (Loop 2000A, PRV segment)
0001	Billing Provider information (Loop 2010AA, NM1 segment)
2010AAN3	Billing Provider Address (Loop 2010AA, N3 segment)
2010AAN4	Billing Provider City, State, Zip Code (Loop 2010AA, N4 segment)
2010BBREFBILLING	Proprietary Provider ID (Loop 2010BB, REF segment for Billing Provider Secondary Identification) (5010 only.)
ZREF	Billing Provider Secondary Identification (Loop 2010AA, REF segment)
ZRFB	X12-4010: Pay-to-Provider Secondary Identification (Loop 2010AB, REF segment)
0002	Pay-To provider information (Loop 2010AB, NM1 segment)
2010ABN3	Pay-To Address – ADDRESS (Loop 2010AB, N3 segment)
2010ABN4	Pay-To Address City, State, ZIP Code (Loop 2010AB, N4 segment)
SBST	Subscriber HL (Loop 2000B, HL segment)
SBRInfo	Subscriber Information (Loop 2000B, SBR segment)
0003	Subscriber Name (Subscriber loop 2010BA, NM1 segment)
ZRSG	Subscriber DMG information (Subscriber loop 2010BA, DMG segment)
ZSRF	Subscriber Secondary Identification (Subscriber loop 2010BA, REF segment)
SPWK	Subscriber Claim Supplemental Information (Subscriber loop 2300, PWK segment)
S020	Subscriber Clearinghouse Claim Number (Subscriber loop 2300, Subscriber Claim Identification Number for Clearinghouses REF)
RFPV	Attending/Referring Physician Name (Subscriber loop 2310A, NM1 segment)
RNPV	Operating/Rendering Provider Name (Subscriber loop 2310B, NM1 segment)
PSPV	Service Facility Location (Dependent loop 2310C, NM1 segment)

837 – All 837s (PDSA837D and PDSA5010837D, PDSA837I and PDSA5010837I, PDSA837P and PDSA5010837P)	
S2330ANM1	Other Subscriber Name (Loop 2330A, NM1 segment)
S2330BNM1	Other Payer Name (Loop 2330B, NM1 segment)
S2330BREF	Other Payer Contact Information (Loop 2330B, REF segment)
ZRSM	Subscriber Claim Medical Rec # (Subscriber loop 2300, Medical Record Number REF)
ZRSS	Service information (Subscriber loop 2400; SV1, SV2, or SV3 segment)
ZRSV	Service Date information (Subscriber loop 2400, Service Date DTP)
PATInfo	Patient Information (Dependent loop 2000C, PAT segment)
ZRSI	Original Reference Number ICN/DCN (Subscriber loop 2300, REF segment)
DPST	Dependent HL loop (Dependent loop 2000C, HL segment)
ZRDI	Original Reference Number ICN/DCN (Dependent loop 2300, REF segment)
0006	Dependent Patient Name (Dependent loop 2010CA, NM1 segment)
ZRDG	Dependent DMG information (Dependent loop 2010CA, DMG segment)
ZPRF	X12-4010: Subscriber Secondary Identification (Dependent loop 2010CA, REF segment)
P009	Dependent Claim (Dependent loop 2300, CLM segment)
PPWK	Dependent Claim Supplemental Information (Dependent loop 2300, PWK segment)
P020	Dependent Clearinghouse Claim Number (Dependent loop 2300, Dependent Claim Identification Number for Clearinghouses REF)
ZRDM	Dependent Claim Medical Rec # (Dependent loop 2300, Medical Record Number REF)
RFPC	Attending/Referring Provider Name (Dependent loop 2310A, NM1 segment)
RNPC	Operating/Rendering Provider Name (Dependent loop 2310B, NM1 segment)
PSPC	Purchased Service Provider Name (Dependent loop 2310C, NM1 segment)
P2330ANM1	Other Subscriber Name (Dependent loop 2330B, NM1 segment)
P2330BNM1	Other Payer Name (Dependent loop 2330B, NM1 segment)
P2330BREF	Other Payer Secondary Identifier (Dependent loop 2330B, REF segment)
0005	Dependent Service Line (Dependent loop 2400, LX segment)
ZRDV	Service Date information (Dependent loop 2400, Service Date DTP)

837 Dental (PDSA837D and PDSA5010837D) The “All 837s” information above is the same for the 837 Dental except for the following.	
ID	**Called on segment …**
SubmitterPER	Submitter EDI Contact Information (Loop 1000A, PER segment)
0004	Payer Name information (Subscriber loop 2010BB, NM1 segment)
SREFREF	Referring Provider Secondary Identification (Subscriber loop 2310A, REF segment)
S2310BPRV	Rendering Provider Specialty Information (Subscriber loop 2310B, PRV segment)
SREND	Rendering Physician Secondary Identification (Subscriber loop 2310B, REF segment)
S010	Service Date (Subscriber loop 2300, DTP segment where DTP01=472)
S011	Subscriber Service Line information (Subscriber loop 2400, SV3 segment)
S012	Subscriber Claim date information (Subscriber loop 2400, DTP segment)
SREFLINEITEM	Line Item Control Number (Subscriber loop 2400, REF segment)
P2310BPRV	Rendering Provider Specialty Information (Subscriber loop 2310B, PRV segment)
PREFREF	Referring Provider Secondary Identification (Dependent loop 2310A, REF segment at dependent level)
PREND	Rendering Provider Secondary Identification (Dependent loop 2310B, REF segment)
P010	Service Date (Dependent loop 2300, DTP segment where DTP01=472)
P011	Dependent Patient Service Line information (Dependent loop 2400, SV3 segment)
P012	Dependent Patient Claim date information ((Dependent loop 2400, DTP segment)
PREFLINEITEM	Line Item Control Number (Loop 2400 in 2000C, REF segment)

837 Institutional (PDSA837I and PDSA5010837I)

The “All 837s” information above is the same for the 837 Institutional except for the following.

ID	Called on segment …
SubmitterPER	Submitter EDI Contact Information (Loop 1000A, PER segment)
2010AAPER	Billing Provider Contact Information (Loop 2010AA, PER segment)
0004	Payer Name (Subscriber loop 2010BC, NM1 segment)
S009	Subscriber Claim (Subscriber loop 2300, CLM segment)
S010	Subscriber Claim Date (Subscriber loop 2300, DTP segment)
S020	Subscriber Claim Identification Number For Clearinghouses (Subscriber loop 2300, REF segment)
SATTEND	Attending Physician Secondary Identification (Subscriber loop 2310A REF segment)
SOPER	Operating Physician Secondary Identification (Subscriber loop 2310B REF segment)
SPPV	Service Facility Name (Subscriber loop 2310E, NM1 segment)
SREFLINEITEM	Line Item Control Number (Subscriber loop 2400, REF segment)
SSERVFAC	Service Facility Secondary Identification (Subscriber loop 2310E, REF segment)
S2330BREFG1	X12_5010: Other Payer Prior Authorization Number (Subscriber loop 2330B, REF segment)
S2330BREF9F	X12_5010: Other Payer Referral Number (Subscriber loop 2330B, REF segment)
S2330BREFT4	X12_5010: Other Payer Claim Adjustment Indicator (Subscriber loop 2330B, REF segment)
S2330BREFF8	X12_5010: Other Payer Claim Control Number (Subscriber loop 2330B, REF segment)
0007	Subscriber Service Line (Subscriber loop 2400, LX segment)
S011	Subscriber Service Line information (Subscriber loop 2400), SV2 segment)
P010	Dependent Claim Date (Dependent loop 2300, DTP segment)
P020	Dependent Claim Identification Number For Clearinghouses (end of Dependent loop 2300)
PATTEND	Attending Physician Secondary Identification (Dependent loop 2310A REF segment)
POPER	Operating Physician Secondary Identification (Dependent loop 2310B REF segment)

837 Institutional (PDSA837I and PDSA5010837I) The “All 837s” information above is the same for the 837 Institutional except for the following.	
ID	**Called on segment …**
P2330BREFG1	X12_5010: Other Payer Prior Authorization Number (Dependent loop 2330B, REF segment)
P2330BREF9F	X12_5010: Other Payer Referral Number (Dependent loop 2330B, REF segment)
P2330BREFT4	X12_5010: Other Payer Claim Adjustment Indicator (Dependent loop 2330B, REF segment)
P2330BREFF8	X12_5010: Other Payer Claim Control Number (Dependent loop 2330B, REF segment)
PREFLINEITEM	Line Item Control Number (Loop 2400 in 2000C, REF segment)
SPPC	Service Facility Name (Dependent loop 2310E, NM1 segment)
PSERVFAC	Service Facility Secondary Identification (Dependent loop 2310E, REF segment)
P011	Dependent Service Line (Dependent loop 2400, SV2 segment)

837 Professional (PDSA837P and PDSA5010837P)

The “All 837s” information above is the same for the 837 Professional except for the following

ID	Called on segment …
SubmitterPER	Submitter EDI Contact Information (Loop 1000A, PER segment)
2010AAPER	Billing Provider Contact Information (Loop 2010AA, PER segment)
0004	Payer Name information (Loop 2010BB, NM1 segment)
SREFREF	Referring Provider Secondary Identification (Subscriber loop 2310A, REF segment)
S2310BPRV	Rendering Provider Specialty Information (Subscriber loop 2310B, PRV segment)
SREND	Rendering Physician Secondary Identification (Subscriber loop 2310B, REF segment)
SPPV	Supervising Provider Name (Subscriber loop 2310E, NM1 segment)
SSUPPRV	Supervising Provider Secondary Identification (Subscriber loop 2310E, REF segment)
S2330BREFG1	X12_5010: Other Payer Prior Authorization Number (Subscriber loop 2330B, REF segment)
S2330BREF9F	X12_5010: Other Payer Referral Number (Subscriber loop 2330B, REF segment)
S2330BREFT4	X12_5010: Other Payer Claim Adjustment Indicator (Subscriber loop 2330B, REF segment)
S2330BREFF8	X12_5010: Other Payer Claim Control Number (Subscriber loop 2330B, REF segment)
S011	Subscriber Service Line (Subscriber loop 2400, SV1 segment)
S010	Subscriber Claim Date (Subscriber loop 2400, DTP segment)
SREFLINEITEM	Line Item Control Number (Subscriber loop 2400, REF segment)
PREFREF	Referring Provider Secondary Identification (Dependent loop 2310A, REF segment at dependent level)
SPPC	Supervising Provider Name (Dependent loop 2310E, NM1 segment)
P2310BPRV	Rendering Provider Specialty Information (Dependent loop 2310B, PRV segment)
PREND	Rendering Provider Secondary Identification (Dependent loop 2310B, REF segment)
PSUPPRV	Supervising Provider Secondary Identification (Dependent loop 2310E, REF segment)

837 Professional (PDSA837P and PDSA5010837P) The “All 837s” information above is the same for the 837 Professional except for the following	
ID	**Called on segment …**
P2330BREFG1	X12_5010: Other Payer Prior Authorization Number (Dependent loop 2330B, REF segment)
P2330BREF9F	X12_5010: Other Payer Referral Number (Dependent loop 2330B, REF segment)
P2330BREFT4	X12_5010: Other Payer Claim Adjustment Indicator (Dependent loop 2330B, REF segment)
P2330BREFF8	X12_5010: Other Payer Claim Control Number (Dependent loop 2330B, REF segment)
P011	Dependent Patient Service Line (Dependent loop 2400, SV1 segment)
P010	Dependent Patient Claim Date (Dependent loop 2400, DTP segment)
PREFLINEITEM	Line Item Control Number (Loop 2400 in 2000C, REF segment)

837 Dental, Institutional, and Professional Post-adjudication Claims Data Reporting (PDSA5010-837X300, PDSA5010-837X299, PDSA5010-837X298)	
ID	**Called on segment …**
Header	
STST	Transaction Set Header
0021	Beginning of Hierarchical Transaction
Loop 1000A	
ZRT	Submitter Name
SubmitterPER	Submitter EDI Contact Information
Loop 1000B	
ZRP	Receiver Name
Loop 2000A	
PRST	Billing Provider Hierarchical Level
2000APRV	Billing Provider Specialty Information
Loop 2010AA	
0001	Billing Provider Name
2010AAN3	Billing Provider Address
2010AAN4	Billing Provider City, State, Zip Code

837 Dental, Institutional, and Professional Post-adjudication Claims Data Reporting (PDSA5010-837X300, PDSA5010-837X299, PDSA5010-837X298)	
ZREF	Billing Provider Tax Identification
Loop 2000B	
SBST	Subscriber Hierarchical Level
SBRInfo	Subscriber Information
Loop 2010BA	
0003	Subscriber Name
ZRSG	Subscriber Demographic Information
ZSRF	Subscriber Social Security Number
Loop 2010BB	
2010BBData Receiver	Data Receiver
Loop 2300 Subscriber	
S009	Claim information
S010	Date - Service Date (X300 and X299 only)
SPWK	Claim Supplemental Information (X298 and X299 only)
ZRSI	Payer Claim Control Number (X299 and X298 only)
SREPRICED CLAIMNUMBER	Repriced Claim Number (X299 and X298 only)
S020	Claim Identifier For Transmission Intermediaries
ZRSM	Medical Record Number (X299 and X298 only)
HI201	Principal Diagnosis (X299 only)
Loop 2310A (X300 and 298 only)	
RFPV	Referring Provider Name
SREFREF	Referring Provider Secondary Identification
Loop 2310A (X299 only)	
RFPV	Attending Provider Name
S2310APRV	Attending Provider Speciality Information
SATTEND	Attending Provider Secondary Identification
Loop 2310B (X300 and X298 only)	
RNPV	Rendering Provider Name
S2310BPRV	Rendering Provider Specialty Information

837 Dental, Institutional, and Professional Post-adjudication Claims Data Reporting (PDSA5010-837X300, PDSA5010-837X299, PDSA5010-837X298)	
SREND	Rendering Provider Secondary Identification
Loop 2310B (X299 only)	
RNPV	Operating Physician Name
SOPER	Operating Physician Secondary Identification
Loop 2310C (X300 only)	
SSPV	Service Facility Location Name
Loop 2310C (X299only)	
SSPV	Other Operating Physician Name
Loop 2330A	
S2330ANM1	Other Subscriber Name
Loop 2330B (X300 only)	
S2330BNM1	Other Payer Name
S2330BREF	Other Payer Secondary Identifier
S2330BREFF8	Other Payer Control Number
Loop 2330B (X299 only)	
S2330BNM1	Other Payer Name
S2330BREF	Other Payer Secondary Identifier
S2330BREFT4	Other Payer Claim Adjustment Indicator
S2330BREFF8	Other Payer Claim Control Number
S2330BREFBP	Other Payer Adjusted Claim Control Number
S2330BREF1N	Other Payer Adjudicated DRG
Loop 2330B (X298 only)	
S2330BNM1	Other Payer Name
S2330BREF	Other Payer Secondary Identifier
S2330BREFT4	Other Payer Claim Adjustment Indicator
S2330BREFF8	Other Payer Control Number
Loop 2400	
0007	Service Line Number
S011	Dental Service (X300 only)
S011	Institutional Service Line (X299 only)
S011	Professional Service (X298 only)

837 Dental, Institutional, and Professional Post-adjudication Claims Data Reporting (PDSA5010-837X300, PDSA5010-837X299, PDSA5010-837X298)	
ZRSV	Date - Service Date
SREFLINEITEM	Line Item Control Number (X298 and X299 only)
Loop 2000C – Dependent Level	
DPST	Patient Hierarchical Level
PATInfo	Patient Information
Loop 2010CA	
0006	Patient Name
ZRDG	Patient Demographic Information
Loop 2300 Dependent	
P009	Claim information
P010	Date - Service Date (X300 and X299 only)
PPWK	Claim Supplemental Information (X299 and X298 only)
ZRDI	Payer Claim Control Number (X299 and X298 only)
DREPRICED CLAIMNUMBER	Repriced Claim Number (X299 and X298 only)
P020	Claim Identifier For Transmission Intermediaries
ZRDM	Medical Record Number (X299 and X298 only)
HI76	Principal Diagnosis (X299 only)
Loop 2310A (X300 and X298 only)	
RFPC	Referring Provider Name
PREFREF	Referring Provider Secondary Identification
Loop 2310A (X299 only)	
RFPC	Attending Provider Name
P2310APRV	Attending Provider Speciality Information
PREFREF	Attending Provider Secondary Identification
Loop 2310B (X300 and X298 only)	
RNPC	Rendering Provider Name
P2310BPR	Rendering Provider Specialty Information
PREND	Rendering Provider Secondary Identification
Loop 2310B (X299 only)	
RNPC	Operating Physician Name

837 Dental, Institutional, and Professional Post-adjudication Claims Data Reporting (PDSA5010-837X300, PDSA5010-837X299, PDSA5010-837X298)	
POPER	Operating Physician Secondary Identification
Loop 2310C (X300 only)	
PSPC	Service Facility Location Name
Loop 2310C (X299 only)	
PSPC	Other Operating Physician Name
Loop 2330A	
P2330ANM1	Other Subscriber Name
Loop 2330B (X300 only)	
P2330BNM1	Other Payer Name
P2330BREF	Other Payer Secondary Identifier
P2330BREFF8	Other Payer Control Number
Loop 2330B (X299 only)	
P2330BNM1	Other Payer Name
P2330BREF	Other Payer Secondary Identifier
P2330BREFT4	Other Payer Claim Adjustment Indicator
P2330BREFF8	Other Payer Control Number
P2330BREFBP	Other Payer Adjusted Claim Control Number
P2330BREF1N	Other Payer Adjudicated DRG
Loop 2330B (X298 only)	
P2330BNM1	Other Payer Name
P2330BREF	Other Payer Secondary Identifier
P2330BREFT4	Other Payer Claim Adjustment Indicator
P2330BREFF8	Other Payer Control Number
Loop 2400	
0005	Service Line Number
P011	Dental Service
P011	Institutional Service Line (X299 only)
P011	Professional Service
ZRDV	Date - Service Date
PREFLINEITEM	Line Item Control Number (X299 and X298 only)
TRSE	Transaction Set Trailer

997 Functional Acknowledgement (PDSA997 and PDSX5010997X231)	
ID	**Called on segment …**
AK1	Functional Group Response Header (AK1 segment)
AK2	Transaction Set Response Header (AK2 segment)
AK3	X12-4010: Data Segment Note (AK3 segment)
AK4	X12-4010: Data Element Note (AK4 segment)
AK5	X12-4010: Transaction Set Response Trailer (AK5 segment)
AK9	Functional Group Response Trailer (AK9 segment)

999 Application Acknowledgement PDSA5010997X231 and PDSX5010999X231)	
ID	**Called on segment …**
AK1	Functional Group Response Header (AK1 segment)
AK2	Transaction Set Response Header (AK2 segment)
IK3	Error Identification (IK3 segment)
IK4	Implementation Data Element Note (IK4 segment)
IK5	Implementation Transaction Set Response (IK5 segment)
AK9	Functional Group Response Trailer (AK9 segment)

8 Appendix D: Record Definition Summary

Record Layout Summary, Version 2.0

For details about each record, see page 33.

CSEG

Field	Length	Start	End
Record (CSEG)	5	1	5
Line #	10	6	15
Segment Data	*n*	16	*EOL*

DTL

Field	Length	Start	End	Notes
Record (DTL)	5	1	5	
Line #	10	6	15	
Loop/Group ID	6	16	21	
Seg ID	4	22	25	
Elem ID	4	26	29	
Comp ID	4	30	33	
Seg Pos	10	34	43	
Elem Pos	2	44	45	
SubElem Pos	2	46	47	
Loop/Group Repeat Count	10	48	57	
Element Repeat	10	58	67	
999 IK3-04	3	68	70	Not in EDIFACT
999 IK4-03	3	71	73	Not in EDIFACT
Filler	4	74	77	
Error #	5	78	82	
Severity	2	83	84	
Seg Ordinal	5	85	89	
WEDI SNIP Type	1	90	90	Not in EDIFACT
997 AK304	2	91	92	Not in EDIFACT
997 AK403	2	93	94	Not in EDIFACT
824 TED01	3	95	97	Not in EDIFACT
824 TED02	3	98	100	Not in EDIFACT
277 STC code	5	101	105	Not in EDIFACT

Filler	5	106	110	
Application Data	20	111	130	Right justified

EDAT

Field	Length	Start	End
Record (EDAT)	5	1	5
Line #	10	6	15
Element Data	*n*	16	*EOL*

EDTL

EDTL for XML			
Field	**Length**	**Start**	**End**
Record Tag (EDTL)	5	1	5
Line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar			
Path from root		17	
Group Repeat Count			
"Segment" ID			
Position			
Not used for XML			
Not used for XML			
Not used for XML			
Not used for XML			
Error #			
Severity			
EDTL for Flat File			
Field	**Length**	**Start**	**End**
Record Tag (EDTL)	5	1	5
Line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar			
Loop ID		17	
Loop Repeat Count			
Segment ID			
Element ID			
Element Position			
Subelement ID			
Subelement Position			
Ordinal Number			
Error #			
Severity			

ELOC

Field	Length	Start	End
Record (ELOC)	5	1	5
Line #	10	6	15
Location Text	*n*	16	EOL

EMSG

Field	Length	Start	End
Record (EMSG)	5	1	5
Line #	10	6	15
Error Message	*n*	16	*EOL*

END

Field	Length	Start	End
Record (END)	5	1	5
Line #	10	6	15
Error #	5	16	20
Severity	2	21	22
Date/Time	17	23	39
FileName Msg	*n*	40	*EOL*

ENDS

Field	Length	Start	End
Record (ENDS)	5	1	5
Line #	10	6	15
Seg Count	10	16	25
ST Control #	9	26	34

ESEG

Field	Length	Start	End
Record (ESEG)	5	1	5
Line #	10	6	15
Segment Data	*n*	16	*EOL*

ETYPE/ETYPS

Field	Length	Start	End
Record Tag (ETYPS)	5	1	5
Line #	10	6	15
Type 0 count	10	16	25
EDI Syntax Count	10	26	35
Syntactical	10	36	45
Balancing Count	10	46	55
Situation Count	10	56	65
Code Set Count	10	66	75
Product Count	10	76	85
Payer Count	10	86	95
Partner Count	10	96	105

EVALU

Field	Length	Start	End
Record (EVALU)	5	1	5
Line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar.			
Structure ID		17	
Segment Position			
Element position			
Subelement position			
Element value			*EOL*

GEN

Field	Length	Start	End
Record (GEN)	5	1	5
Line #	10	6	15
Error #	5	16	20
Severity	2	21	22
Type	2	23	24
Message	*n*	25	*EOL*

IDENT

IDENT Record Layout			
Field	Length	Start	End
Record Tag (IDENT)	5	1	5
Line #	10	6	15
The remaining fields are separated by vertical bars			
RuleID	1	17	
FSUID		19	
SystemID			
Reserved			

SBST

Field	Length	Start	End
Record (SBST)	5	1	5
Starting line #	10	6	15
Delimiter |	1		
Loop/Group ID			
Delimiter |	1		
SegId			
Delimiter |	1		
ElmPos			
Delimiter |	1		
SubElmPos			
Delimiter |	1		
ReplaceValue			
Delimiter |	1		
MetaData			

SBSTA

Field	Length	Start	End
Record (SBSTA)	5	1	5
Line # of segment	10	6	15
Delimiter	1		
Loop/Group ID			
Delimiter	1		
SegId			
Delimiter	1		
ElmPos			
Delimiter	1		
SubElmPos			
Delimiter	1		
Old value			
Delimiter	1		
New value			
Delimiter	1		
MetaData			

SBSTD

Field	Length	Start	End
Record (SBSTD)	5	1	5
Line # of segment	10	6	15
Delimiter |	1		
MetaData			
Delimiter |	1		
Loop/GroupID			
Delimiter |	1		
SegmentID			

SBSTF

Field	Length	Start	End
Record (SBSTF)	5	1	5
Line # of segment	10	6	15
Delimiter	1		
Key			
Delimiter	1		
Loop/Group ID			
Delimiter	1		
SegId			
Delimiter	1		
ElmPos			
Delimiter			
SubElmPos	1		
Delimiter			
MetaData			

SBSTI

Field	Length	Start	End
Record (SBSTI)	5	1	5
Line # of segment	10	6	15
Delimiter	1	16	16
MetaData			
Delimiter	1		
Loop/Group ID			
Delimiter	1		
SegmentID			
Delimiter	1		
Element Data			
Delimiter	1		
Element Data			

SBSTR

Field	Length	Start	End
Record (SBSTR)	5	1	5
Line # of segment	10	6	15
Delimiter	1		
Key			
Delimiter	1		
ReplaceValue			
Delimiter	1		

STRT

Field	Length	Start	End
Record (STRT)	5	1	5
Line #	10	6	15
Error #	5	16	20
Severity	2	21	22
Date/Time	17	23	39
FileName Msg	*n*	40	*EOL*

STRUE

Field	Length	Start	End
Record (STRUE)	5	1	5
Line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar.			
Structure ID		17	
Document flag			
Instance			
Ending position			
Errors by severity			
Errors by type			
ID of ending segment			*EOL*

STRUS

Field	Length	Start	End
Record (STRUS)	5	1	5
Starting line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar.			
Structure ID		17	
Document flag			
Instance			
Starting position			*EOL*

SVALU

Field	Length	Start	End
Record (SVALU)	5	1	5
Line #	10	6	15
The remaining fields are variable length and each is preceded with a vertical bar.			
Structure ID		17	
Structure position			
Segment data			*EOL*

SVRTS

Field	Length	Start	End
Record (SVRTS)	5	1	5
Line #	10	6	15
Ignore Count	10	16	25
Info Count	10	26	35
Warning Count	10	36	45
Error Count	10	46	55
Fatal Count	10	56	65
User1 Count	10	66	75
User2 Count	10	76	85

SVRTY

Field	Length	Start	End
Record (SVRTY)	5	1	5
Ignore Count	10	6	15
Info Count	10	16	25
Warning Count	10	26	35
Error Count	10	36	45
Fatal Count	10	46	55
User1 Count	10	56	65
User1 Count	10	66	75

VER

Field	Length	Start	End
Record (VER)	5	1	5
Version	*n*	6	*EOL*

Z

Field	Length	Start	End
Record (*Zaaaa*)	5	1	5
Line #	*10*	6	15
Field 1 Data	*n*1	16	*n*
Fiend *n* Data	*n*2	*n*	*n*

(Note: ***aaaa*** represents the Custom Record Tag assigned to the record when it is defined in EDISIM. Examples: ZCLM, ZPATN, etc.)