TIBCO Foresight® Instream®

Response Generator

Software Release 8.7
August 2017

Two-second advantage®

Important Information

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

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

TIBCO, Two-Second Advantage, TIBCO Foresight EDISIM, TIBCO Foresight HIPAA Validator Desktop, TIBCO Foresight Instream, TIBCO Foresight Instream MAC Adaptor, and TIBCO Foresight Translator are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

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

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

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

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

Contents

1 Introduction

Intended Audience

This document is intended for developers or business analysts who are setting up or running Response Generator.

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

Capabilities

The EDI

The original EDI can include one or more interchanges, functional groups, or transaction sets or messages from any version X12 or EDIFACT that can be validated by Instream.

The Responses

Based on functional group

Response Generator will generate any of these in response to EDI validated by Instream, based on functional group.

Input EDI	Output files created by Response Generator							
	277CA	277U	824	997	999 (5010)	TA1	CONTRL Syn 3,4	Custom report
HIPAA EDI except 837			✓	✓	✓	✓		✓ 270, 276, 277CA, 277U (4010 only), 277X228, 278, 820x218, 820x306, 834, 835, 837
HIPAA 837	✓	✓	✓	✓	✓	✓		✓
Non-HIPAA X12 EDI			✓	✓	✓	✓		
EDIFACT except inbound CONTRL							✓	

Based on a whole document

Response Generator generates properly-formatted 277, 824, 997, 999 (5010 only), and TA1 response documents for the following X12 versions, based on a whole document:

- 3040
- 3070
- 4010
- 4040
- 4050
- 4060
- 5010
- 6020

Based on mixed X12 source

Response documents can be generated from mixed X12 source within a single interchange, provided the source contains all of a single transaction type.

Note that:

- 997s are not generated for 99x sources
- 824s are not generated for 824s
- 277s are generated for 837s only

If no valid response documents can be created, such as requesting a 277 on a document that contains no 837s, an error is presented. This error can be overridden by using the **-floose_src_check** flag.

Use of the -fver_pass command

Response documents can be generated for any X12 version (2000 – 5010) by using the Response Generator command **-fver_pass**. This command causes Response Generator to pass the ISA and GS version numbers for the source document through unchanged. See **-fver_pass** in Command Line Format on page 13.

Output Version

Source document ISA12	Output document version
00401	X12-4010 (Exception: 999 is always X12-5010 errata)
anything else	X12-5010

You can override the output version on the command line.

X12 HIPAA Responses

277CA Health Care Claim Acknowledgement (277CA)

Note: A 277CA is generated only when responding to HIPAA 837 transactions.

The 277 CA is sent by a payer in response to an 837 to report on whether pre-adjudication validation found them acceptable for adjudication.

Claims failing pre-adjudication validation are not sent for adjudication and therefore are never reported in an 835. Claims that pass this step are forwarded for adjudication and generate an 835.

To generate a 277CA, use these command line parameters: **-o277** or **-fca277**

If requested, the 277CA is created regardless of whether the 837 contained errors. It lists all claims, good and bad, because they are needed for balancing.

The 277CA will be version 5010 (only) and can be accompanied by a 999. It cannot be generated along with a 277U.

It is based on the Foresight guideline 5010-277CAX214, which you can view in TIBCO Foresight® HIPAA Validator® Desktop's Library.

277U Unsolicited Health Care Claim Status Notification (277U)

Note: A 277U is generated only when responding to HIPAA 837 transactions.

Instream will validate a 277U. Response Generator will generate a response in X12 versions 3040 to 5010.

The 277U transaction is used in addition to the 997 and 824 to report which claims are not being paid because they (or their 'parent') have errors. Any claim that contains an error or whose owner (subscriber, provider, etc.) contains an error will be included in this response. Which error level determines a rejection is identified with the use of the **-er** command line parameter (it is Severity 3 by default).

The 277 is not good at describing *why* a claim is being rejected – the 824 is best for that.

If no errors are present in the 837, an empty 277 will be generated unless you use Response Generator's -**fsa** command line parameter. You can delete any zero byte files with Response Generator's -**nz** command line parameter.

There is no HIPAA 277U implementation guide, so there are no HIPAA code sets or HIPAA rules to describe it. See Appendix E: Response Document Contents on page 143 for the structure and contents of TIBCO Foresight's 277U.

824 Application Advice

Note: An 824 is generated when responding to HIPAA EDI (including 837) and non-HIPAA X12 EDI transactions.

The 824 transaction reports on error types 3 and above. These include application errors. One or more RED segments follow each TED segment. By default, each RED01 element will contain a claim number if the error occurred in a claim. Otherwise, it will contain "Not Claim Specific."

You can use a setup file to request all RED01 elements contain claim numbers, regardless of where the error occurs. See RED Segments on page 49.

If no errors (types 3 and above) are present in the 837, an empty 824 will be generated unless you use Response Generator's **-fsa** command line parameter. You can delete any zero byte files with Response Generator's -**nz** command line parameter.

An 824 can be requested for any X12 transaction except a 997 or another 824.

997 Functional Acknowledgement

Note: A 997 is generated when responding to HIPAA EDI (including 837) and non-HIPAA X12 EDI transactions.

The 997 transaction acknowledges the receipt, and acceptance and/or rejection, of functional groups and transaction sets. It also reports some errors - typically X12 syntax errors (types 1 and 2). It can be requested for any X12 transaction except another 997 or a 999.

Response Generator cannot generate a 997 and 999 for the same input file.

See 997 Structure and Data Sources on page 160 for information about the structure and contents of TIBCO Foresight's 997.

999 Implementation Acknowledgement

Note: A 999 is generated when responding to HIPAA EDI (including 837) and non-HIPAA X12 EDI transactions.

Per HIPAA rules, Response Generator will not generate a 997 and 999 for the same input file.

The 999 created by Response Generator is based on the Implementation Acknowledgement for Health Care Insurance, 005010X231A1 errata. The only difference between this 999 and the previous non-errata 999 is the contents of the GS08 and the ST03 (now 005010X231A1). To produce a non-errata 999, use the **-dav** command line qualifier to specify 005010X231.

It can be used as a response to any HIPAA X12-5010 data, and it answers these questions:

- Does this functional group have any errors (syntactical and implementation) errors that fall below the error threshold? If so, what are they?
- Do the transaction sets that it contains have any errors that fall below the error threshold? If so, what are they?

By default, it reports all errors that the guideline detected during validation. Response Generator's **-s** command line option causes the output 999 to report only types 1-2 errors.

For an example, run **V_RG_837I_5010_999** in Instream's Scripts directory.

See 999 Structure and Data Sources on page 160 for information about the structure and contents of TIBCO Foresight's 999.

TA1 Interchange Acknowledgement

Note: A TA1 is generated when responding to HIPAA EDI (including 837) and non-HIPAA X12 EDI transactions.

The TA1 response acknowledges the receipt of an interchange, and reports on whether the interchange was accepted, accepted with errors, or rejected. This is a single segment that you can choose to have placed within the 997 or 999, or surrounded by ISA and IEA in a separate file.

Guidelines for the Response Generator 824 and 997 guidelines are described in Instream's Doc directory in a file called **RSGN4010.SEF**. If you have TIBCO Foresight® EDISIM®, you can import it into Standards Editor for review.

Custom Report

Note: A custom report can be generated when responding to HIPAA EDI transactions for 270, 276, 277CA, 277U (4010 only), 277X228, 278, 820x218, 820x306, 834, 835, and 837 validations only.

A text report is generated with the **-otext** command. Its purpose is to capture output as text and then, using a user-built template (as specified with the **-tpl** parameter), format a custom report. See Report Templates on page 65.

EDIFACT Responses (CONTRL Document)

Note: A CONTRL document is generated when responding to EDIFACT validation (except Inbound CONTRL).

Response Generator will generate a CONTRL document in response to an EDIFACT validation performed by Instream. The CONTRL document will be syntax level 3 or 4, based on the originating interchange's UNB01.

Based on the value in the original EDI's UNB01, the CONTRL will be:

- Syntax version 4 if the original EDI was version 4.
- Syntax version 3 if the original EDI was versions 1, 2, or 3.

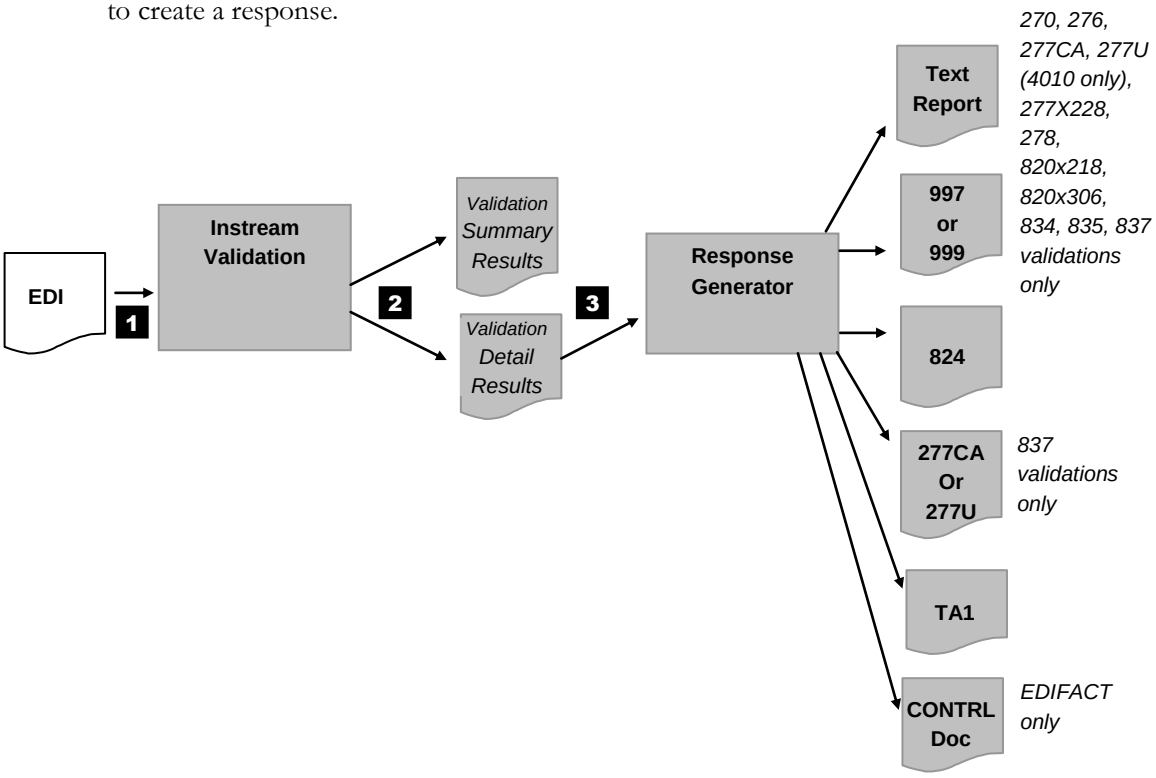
X12 Non-HIPAA Responses

Response Generator will generate the following in response to non-HIPAA EDI validated by Instream. It can create any or all of these (described in the previous section):

- 824 Application Advice
- 997 Functional Acknowledgement
- 999 Functional Acknowledgement
- TA1 Interchange Acknowledgement

Example input and output

Response Generator runs after Instream validation and uses the validation detail results to create a response.



1 Instream reads and validates EDI data using a GuidelinePlus.

2 Validation produces summary and detail results about the data's compliance.

3 Response Generator reads the validation detail results and creates the response documents that you requested on the command line.

What you need before using Response Generator

Guideline that generates SVALU Records

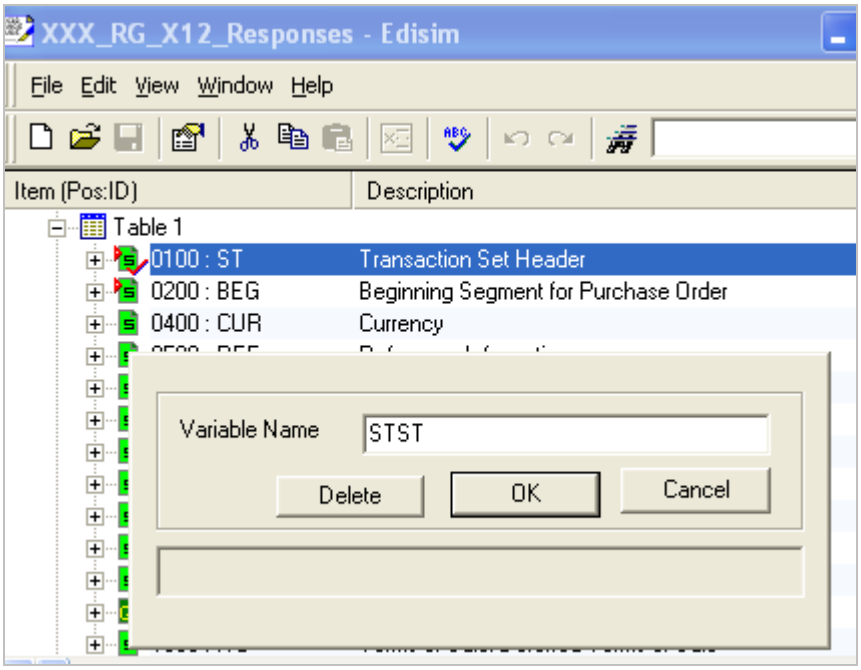
When using Response Generator, have Instream validate the EDI data using a guideline that writes SVALU records to the validation detail results file. This report then serves as input to Response Generator and Docsplitter, as well as your own applications.

For HIPAA data:

- These guidelines ship with Instream and have names that start with **PD**. For example, the guideline for a 4010 835 is called **PDSA835**.
- The guideline names are listed in the **GuidelinePlus** column in **ForesightHIPAAguidelinelist.pdf** in Instream's Doc directory.
- You can use these guidelines for validation, or a guideline created by merging a company guideline with a PD guideline. See **GuideMerge.pdf** for details about merging.

For X12 data:

1. Edit the guideline with Standards Editor, right-click on the **ST** segment, and choose **DSR Mark/Unmark**. Use Variable Name **STST**:



2. Open **Dictionary Object**s and then **Segments** and add these DSR marks also:

   GS Segment DSR Variable Name **GSSG**
   ISA Segment DSR Variable Name **ISA1**

3. Save the guideline and copy it to Instream's **Database** directory.
4. Use this one for validation when you want to generate a 997, 999, or TA1 response.

For EDIFACT data:

1. Edit the guideline with Standards Editor, right-click on the UNH segment, and choose **DSR Mark/Unmark** and use variable name **UNH1**.
2. Open **Dictionary Object**s and then **Segments** and put a DSR mark on the UNB with variable **UNB**.
3. Go to the dictionary's UNG segment and add a DSR mark with variable **UNG**.
4. Save the guideline and copy it to Instream's **Database** directory.
5. Use this one for validation when you want to generate a CONTRL response.

Validation Profile Settings to use for Response Generator

You can specify a profile on the validation command line or validation API call. If you don't specify one, Response Generator uses the default profile **$fsdeflt.apf** in Instream's **Bin** directory.

Caution: Do not change **$fsdeflt.apf**, which is overwritten during updates. Instead, save it to a new name when making profile changes.

Be sure that your profile is generating the proper detail results for Response Generator. To do this, open the profile file and be sure that WT_Error and WT_Fatal are set to 1. This tells Instream validation to write DTL records for errors and fatal errors:

```
[Warning Allow]
WT_Message=1
WT_NonCritical=1
WT_Warning=1
WT_Error=1
WT_Fatal=1
WT_User1=1
WT_User2=1
```

These should be set to 1 so they will be included in the validation detail file.

In the [Detail Record Output] section, be sure the options for Structure Start Records, Structure End Records, Segment Value Records, and Detail Records are set to 1 (true).

```
STRUS=1
STRUE=1
SVALU=1
DTL=1
```

In the [Types Allows] section, be sure Type0 is set to 1. This tells Instream validation to include informational messages in the detail records.

```
[Types Allow]
Type0=1
Type1=1
Type2=1
Type3=1
Type4=1
Type5=1
Type6=1
Type7=1
Type8=1
```

In the [Warning Levels] section, be sure the severity values for 31990 and 31991 are set to 1 (for informational):

```
31081=3,4,8,7,848,,148,8, ,
31082=3,0,,,,,,,
31990=1,0,,,,,,,
31991=1,0,,,,,,,
31992=3,0,,,,,,,
31993=3,0,8,7,024,X1,,,,
```

The APF file's [Warning Levels] section also lets you specify values for:

- The X12 997's AK304 and AK403
- The X12 999's IK304 and IK403
- The 824's TED01 and TED02.

In the example shown above, error 31081 has AK304 set to 8, AK403 set to 7, TED01 set to 848, STC-01-02 set to 148, and IK304 set to 8.

You need not go to the APF file and specify values for all possible errors. Response Generator can still generate 997s, 999s, and 824s by using these default values:

Transaction	Element	Default Value
997	AK304	8 if error refers to an element 1 if error refers to the segment
997	AK403	7
999	IK304	8 if error refers to an element 1 if error refers to the segment
999	IK403	7
824	TED01	024
824	TED02	*blank*

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 837 data and create responses.

   Go to Instream's **Scripts** directory and double-click **V_RG_837I_4010_CommandLineOnly.bat**.

   Go to Instream's **Output** directory and look at the two files created by Response Generator: **EDI_824.txt** and **EDI_997.txt**.

3. For an example of EDIFACT response generation, execute **V_RG_CUSCAR_D93A** in the Scripts directory and look for the results in the Output directory.

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 837 data and create responses:

   Go to Instream's **Scripts** directory and execute **V_RG_837I_4010_CommandLineOnly.sh**.

   Go to Instream's **output** directory and look at the two files created by Response Generator: **EDI_824.txt** and **EDI_997.txt**.

3. For an example of EDIFACT response generation, execute **V_RG_CUSCAR_D93A.sh** in Instream's Scripts directory and look for the results in the Output directory.

Other Demos

See Instream's Scripts directory for many other demos. The ones with "RG" in the filenames include Response Generator.

2 Executing Response Generator

API

Response Generator shares a Java API, a C++ API, and a C# API with Instream validation and Docsplitter. See **TIB_fsp-instream_ <*n.n*>_api.pdf**.

Command Line Format

RespGen.exe is a standalone executable in Instream's **bin** directory. From the command line, it has the following case-sensitive parameters. Use quotation marks around paths or filenames that contain spaces. Curly brackets in the command format below indicate optional parameters.

Parameters can appear in any order. The minimum required parameters are **-i** and one of the **-o** parameters. The one exception to this is the **-version** parameter, which is used alone (no **-i** or **-o**).

RespGen **-i** *fn* **-o**<*option*> *fn* {**-a**} {**-apf** *fn*} {-c *n*} {**-cd** *directory*} {**-dac** *n* } {**-dar** *dar*} {**-das** *das*} {**-dav** *string*} {**-dcc** *contact_info*} {**-dcd** *descrip*} {**-dcg** *n*} {**-dcm** *n*} {**-dco** *orig_fn*} {**-dcp** *code*} {**-de**<*option*> *value*} {**-dic** ***start*** *min/max*} {**-diq** *on/off* } {**-dir** *qq*nn..*} {**-dis** *qq*nn..*} {**-diu** *indicator*} {**-e**<*option*>} {**-f**<*option*>} {**-ge**} {**-gTA1** *n*} {**-k**} {**-l** "*option*"} {**-lsource**} {**-n**} {**-np**} {**-nz**} {**- otext -tpl tplfilename** {-te *n* -tel *nn* -tev *nn*} *filename*}{**-pt** <*option*>} {**-pt824**} {**-s**} {**-stc**} {**-TPA** "*filepath*"} {**-u** "*dirpath*"} {**-v** *n*} {**-version**} {**-w**} {**-y**} {**-z** "*filepath*"}

Parameters

-a (Append)

Optional

Requests that the generated EDI data be appended to the output file (specified with the **-o** parameter) if it already exists. This overrides the default behavior of replacing the file if it exists.

Format of Parameters

-a

Example

Append the generated EDI data to the existing EDI_997.txt output file instead of replacing it:

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -a
```

-apf (APF File)

Optional, unless you are generating CTX segments for a 999 or overriding values in Response Generator output files (see Appendix H: Response Generator Overrides File on page 177).

Allows you to signal the use of a Validation Profile (.apf) file. A similar entry in the TPA setup file, called 'APFPath', will also be added.

Important: If Response Generator is run with both the **-stc** and **-apf** options, it will use the file defined with the **-stc** option for the STC overrides, and ignore any STC entries in the .apf file.

Format of Parameters

-apf *filename*

filename The APF file to be used, including path and filename.

Example

Use the specified .apf file when processing this request:

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -apf
"C:\profiles\custom.apf"
```

-c (Control Number)

Optional

Specifies the starting transaction set control number for the outbound EDI file.

For X12, this is the starting ST02 number.
For EDIFACT, this is the starting UNH01 number.

Format of Parameters

-c *n*

Where:

n A number greater than zero. Default is 1.
OR
Is the starting number control number. It can be either of these:

- A control number.
- The name of a file that contains the control number. If this file does not exit, Response Generator creates it and places the first control number in it. If it does exist, Response Generator uses the number it contains for the first control number. It then updates the file each time it uses a control number. This lets you run multiple instances of Response Generator, all using unique control numbers.

Example

Make 500 the starting ST02 number for the outbound EDI file:

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -c 500
```

-cd (Configuration Directory)

Optional

Specifies the directory in which RespGen should look to find the configuration files ErrMsgTrans.txt and ErrRespXref.txt.

Important: -cd and **-u** cannot be used at the same time. If neither is used, RespGen.exe will look in its own directory for these configuration files.

Format of Parameters

-cd *directory*

Where:

directory Path to the configuration files ErrMsgTrans.txt and ErrRespXref.txt.

Example

Direct RespGen to look in c:\configs for the necessary error configurations files:

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -cd "c:\configs"
```

-csovr (Control Segment Override)

Optional, X12 and EDIFACT responses only.

Directs Response Generator to use the value of the specified field in the interchange and/or group header segment(s).

Format of Parameters

-csovr *key=value;key=value*

Where:

key Mnemonic of X12 or EDIFACT interchange or group header segment field. See X12 Key Information on page 16 and EDIFACT Key Information on page 17.

value Value to be used in the specified field.

Example 1

Direct Response Generator to look in c:\configs for the necessary error configurations files:

```
RespGen -csovr"ISENDQ='01';ISEND='12345';IRECVQ='01';
IRECV='00WERQ'"
```

Example 2

Use the backslash character (\) as an escape character to include a single quote or a backslash in a value:

```
-csovr"ISEND='JIM\'SCO'"
```

Example 3

Use two consecutive single quotes to send nothing in a field (i.e., force the field to be blank):

```
-csovr"-IDATE='';ITIME=''"
```

X12 Key Information

KEY= (X12 Field Mnemonic)	X12 Element ID	X12 Element Name	X12 Version Information
Segment ISA Interchange Control Header			
AUTHQ	ISA01:I01	Authorization Information Qualifier	
AUTH	ISA02:I02	Authorization Information	
SECQ	ISA03:I03	Security Information Qualifier	
SEC	ISA04:I04	Security Information	
ISENDQ	ISA05:I05	Interchange ID Qualifier	
ISEND	ISA06:I06	Interchange Sender ID	
IRECVQ	ISA07:I05	Interchange ID Qualifier	
IRECV	ISA08:I07	Interchange Receiver ID	
IDATE	ISA09:I08	Interchange Date	

KEY= (X12 Field Mnemonic)	X12 Element ID	X12 Element Name	X12 Version Information
ITIME	ISA10:I09	Interchange Time	
ISTDRSEP	ISA11:I10	Interchange Control Standards Identifier	pre-4020
IVER	ISA12:I11	Interchange Control Version Number	
ICTL	ISA13:I12	Interchange Control Number	
ACK	ISA14:I13	Acknowledgement Requested	
USAGE	ISA15:I14	Interchange Usage Indicator	
SESEP	ISA16:I15	Component Element Separator	
Segment GS Functional Group Header			
FUNCID	GS01:479	Functional Identifier Code	
GSEND	GS02:142	Application Sender's Code	
GRECV	GS03:124	Application Receiver's Code	
GDATE	GS04:373	Date	
GTIME	GS05:337	Time	
GCTL	GS06:28	Group Control Number	
RAC	GS07:455	Responsible Agency Code	
GVER	GS08:480	Version/Release/Industry Identifier Code	

EDIFACT Key Information

KEY= (EDIFACT Field Mnemonic)	EDIFACT Element ID	EDIFACT Element Name	EDIFACT Version Info
Segment UNB Interchange Header			
	UNB01:S001		
SYNID	UNB01.01:0001	Syntax Identifier	
SYNVER	UNB01.02:0002	Syntax Version Number	
SCVER	UNB01.03:0080	Service Code List Directory Version Number	V4
ENC	UNB01.04:0133	Character Encoding, Coded	V4
SYNREL	UNB01.05:0076	Syntax Release Number	V4
	UNB02:S002		
ISEND	UNB02.01:0004	Interchange Sender Identification	
ISENDQ	UNB02.02:0007	Identification Code Qualifier	
ISENDID	UNB02.03:0008	Interchange Sender Internal Identification	
ISENDSUBID	UNB02.04:0042	Interchange Sender Internal Sub-Identification	V4
	UNB03:S003		
IRECV	UNB03.01:0010	Interchange Recipient Identification	

KEY= (EDIFACT Field Mnemonic)	EDIFACT Element ID	EDIFACT Element Name	EDIFACT Version Info
IRECVQ	UNB03.02:0007	Identification Code Qualifier	
IRECVID	UNB03.03:0014	Interchange Recipient Internal Identification	
IRECVSUBID	UNB03.04:0046	Interchange Recipient Internal Sub-Identification	V4
	UNB04:S004		
IDATE	UNB04.01:0017	Date	
ITIME	UNB04.02:0019	Time	
ICTL	UNB05:0020	Interchange Control Reference	
	UNB06:S005		
ACK	UNB09:0030	Acknowledgement Requested	
USAGE	UNB11:0035	Interchange Usage Indicator	
RECPW	UNB06.01:0022	Recipient Reference/Password	
RECPWQ	UNB06.02:0025	Recipient Reference/Password Qualifier	
APPLREF	UNB07:0026	Application Reference	
PROCPTY	UNB08:0029	Processing Priority Code	
AGREEID	UNB10:0032	Interchange Agreement Identifier	
Segment UNG Functional Group Header			
MSGGRPID	UNG01:0038	Message Group Identification	
	UNG02:S006		
GSEND	UNG02.01:0040	Application Sender Identification	
GSENDQ	UNG02.02:0007	Application Code Qualifier	
	UNG03:S007		
GRECV	UNG03.01:0044	Application Recipient Identification	
GRECVQ	UNG03.02:0007	Application Code Qualifier	
	UNG04:S004		
GDATE	UNG04.01:0017	Date	
GTIME	UNG04.02:0019	Time	
GCTL	UNG05:0048	Group Reference Number	
RAC	UNG06:0051	Controlling Agency, Coded	
	UNG07:S008		
MSGVNO	UNG07.01:0052	Message Version Number	
GVER	UNG07.02:0054	Message Release Number	
AAC	UNG07.03:0057	Association Assigned Code	
APPLPW	UNG08:0058	Application Password	

-dac (Specify Starting Group Control Number)

Optional

Specifies the Starting Group Control Number for the first outbound group envelope in each interchange.

For X12, this is the starting GS06 number.
For EDIFACT, this is the starting UNG05 number.

Format of Parameters

-dac *n*

Where:

n Is a number greater than 0 and less than or equal to 999999999 (that's nine 9s). The default is 1.

OR

Is the starting number control number. It can be either of these:

- A control number.
- The name of a file that contains the control number. If this file does not exit, Response Generator creates it and places the first control number in it. If it does exist, Response Generator uses the number it contains for the first control number. It then updates the file each time it uses a control number. This lets you run multiple instances of Response Generator, all using unique control numbers.

Example group control numbers:

	no -dac on command line	-dac 5 on command line
ISA		
GS	1	5
GS	2	6
ISA		
ISA		
GS	1	5
GS	2	6
ISA		

Example

When generating this output, have Functional Group Control Numbers start at 1001.

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -dac 1001
```

-dar (Application Receiver ID)

Optional

Specifies the Application Receiver ID (GS03) for the outbound GS.

Format of Parameters

-dar ***dar***

Where:

dar Desired Application Receiver ID. 2 to 15 alphanumeric characters. Default is the first Application Sender ID (GS02) in the inbound EDI file.

Example

Make `PURCH1256` the Application Receiver ID for the outbound GS:

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -dar
PURCH1256
```

-das (Application Sender ID)

Optional

Specifies the Application Sender ID (GS02) for the outbound GS.

Format of Parameters

-das ***das***

Where:

das Desired Application Sender ID. 2 to 15 alphanumeric characters. Default is the first Application Receiver ID (GS03) in the inbound EDI file.

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -das
PURCH1258
```

-dav (Application Version)

Optional

Specifies the Application Version (GS08) for the outbound GS. Regardless of the contents of the GS08 being created, Response Generator will use the rules described on page 3. If the transaction has an ST03, this will be populated by the same value as the GS08.

Note: **-dav** is overridden by the **-f** parameter when it specifies a document type, such as -fdo824x186.

Format of Parameters

-dav *string*

Where:

string Desired Application Version. A string from 1 to 12 characters long. Default is 005010.

Example

Make 004010 the application version for the outbound GS:

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -dav 004010
```

-dcc (Contact Information)

Optional

Specifies the contact information needed for the custom report trailers.

Format of Parameters

-dcc *contact_info*

Where:

contact_info User defined contact information.

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -dcc
AlphaHealthcare
```

-dcd (864 Description)

Optional

Allows you to specify the 864 description field.

Format of Parameters

-dcd *description*

Where:

description The 864 description field. (Up to 80 alphanumeric characters.)

Example

```
-i input_file -o997 output_file -dcd Claim123
```

-dcg (GS Control Number)

Optional

Allows you to specify a GS Control Number. This user-specified GS Control Number should be thought of as a GS-level trace number; it is not the same as the GS06 number.

Format of Parameters

-dcg *n*

Where*:*

n The desired GS Control Number. (1-9 numbers.)

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -dcg 0009
```

-dcm (864 MIT Description)

Optional

Allows you to specify the 864 MIT description field.

Format of Parameters

-dcm *n*

Where*:*

n The 864 MIT description field. (Up to 80 alphanumeric characters.)

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -dcm Claim46
```

-dco (Original filename)

Optional

Specifies the original filename value as needed for the Custom Report headers.

Format of Parameters

-dco *original_filename*

Where*:*

Original_filename The original filename.

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -
dco "AllClaims030510"
```

-dcp (864 Purpose Code)

Optional

Allows you to specify the 864 purpose code.

Format of Parameters

-dcp *code*

Where*:*

code The desired 864 purpose code.

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -dcp DCPC1
```

-de<*option*> [-dea, -dec, -der, and dev] (New UNH Values)

Optional

Allows you to specify a new UNH*x* value.

Format of Parameters

-de *<option> value*

Where:

<option> **-de*a*** = Specifies a new UNH02.05 association assigned code value.

-de*c* = Specifies a new UNH02.04 controlling agency code value.

-de*r* = Specifies a new UNH02.03 release number value.

-de*v* = Specifies a new UNH02.02 version number value.

value The appropriate number or code.

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -der 101
```

-dic (Interchange Control Number)

Optional

Allows you to specify the 864 MIT description field.

Format of Parameters

-dic *start* *(min max)*

Where:

start Mandatory if -dic is used. It is the starting number control number. It can be either of these:

- A 9-digit control number. This number must be within the min and max if they are used.
- The name of a file that contains a 9-digit control number. Response Generator uses the number it contains for the first control number. It then updates the file each time it uses a control number, incrementing by one, no matter how many files are processed.

  Example:

  File 1 is run through Response Generator – The first ISA within the file is assigned interchange control number 000000001 and the second is assigned 000000002.

  File 2 is run through Response Generator – Response Generator continues the numbering that began with File 1. The first ISA within the file is assigned interchange control number 000000003, the third is assigned 000000004, and so on.

min Minimum number for a range of control numbers. Required if *max* is used.

max Maximum number for a range of control numbers. Required if *min* is used.

The control number automatically increments until it reaches the value you specified in *max* or until it reaches 999999999. At that point, it starts over at *min* (if used) or 1.

Example

When generating this output, have Interchange Control Numbers start with 9001:

-i "C:\Files\HV_Results.txt" -o824 "C:\Files\EDI_824.txt" -o997 "C:\Files\EDI_997.txt" -o277 "C:\Files\EDI_277.txt" -ge -dic 9001 -dac 1001 -s

Here are additional examples of responses with and without the **-dic** command:

Inbound data	Response	ISA13 without -dic	ISA13 with -dic 148, 100, 150
ISA			
837	997	1	148
	824	2	149
837	997	3	150
	824	4	100
IEA			
ISA			
837	997	5	101
	824	6	102
IEA			

-diq (generate a TA1 for the document)

Optional

Sets a flag in the ISA14 element to notify the receiver that the sender is expecting a TA1 to be returned for the document.

-diq <on|off>

Format of Parameters

-diq <on|off>

Where:

on If **on**, ISA14 is set to '1', indicating the sender is requesting a TA1 be returned for the document.

off If **off**, ISA14 is set to '0' (default), indicating the sender is not requesting a TA1 for the document.

Example

```
RespGen -i "HV_Results.txt" -o824 "C:\Files\EDI_824.txt" -ge
-y -diq on -fUseShortGS05"
```

-dir and -dis (Interchange Receiver ID and Qualifier or Sender ID and Qualifier)

Optional

Change the:

Interchange Receiver ID and qualifier (**-dir**) to be used in the outbound ISA07 and ISA08

Or

Interchange Sender ID and qualifier (**-dis**) to use in the outbound ISA05 and ISA06.

Defaults are the values from the ISA07 and ISA08 or the ISA05 and ISA06 in the originating interchange.

Format of Parameters

-dir *qq*nnnnnnnn*

or

-dis *qq*nnnnnnnn*

Where:

*qq** The qualifier followed by an asterisk. The qualifier is 2 characters long. If you omit *qq* to accept the qualifier from the originating document, also omit the asterisk.

Note: The asterisk character is required after *qq*; it is not a variable and does not represent an element separator. Response Generator will automatically use the appropriate element separator. Do not substitute a different character or the string will be interpreted incorrectly. See **Incorrect Example**, below.

nnnnnnnn The receiver or sender ID, 2-15 alphanumeric. If shorter than 15, it will be padded on the right with spaces to make it 15 characters.

Examples

ISA07 will be XX and ISA08 will be 123456789012345:

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -dir
"XX*123456789012345"
```

ISA07 will be whatever was used in the originating interchange. ISA08 will be 1234567890 followed by 5 spaces of padding:

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -dir
"1234567890"
```

Incorrect example

This command line incorrectly attempts to use a character other than an asterisk between the *qq* and *nnn…* values.

When Response Generator finds no * in the **-dir** command, the string is treated as one value. Therefore the ISA07 will be whatever was used in the originating interchange and the ISA08 will be XX\1234567890 followed by 2 spaces of padding.

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -dir
"XX\1234567890"
```

-diu (Interchange Usage Indicator)

Optional

Specifies the Interchange Usage Indicator (ISA15) for the outbound ISA.

Format of Parameters

-diu *indicator*

Where:

Indicator One of the following usage indicators:

I = Information (leaves the ISA15 unchanged, regardless of validity)

P = Production (default)

T = Testing

X = Use the value from the input file's ISA15

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -diu T
```

-e<option> (Error Severity and Encoding)

Optional

Error Severity

Sets the error severity that will result in

- a rejected transaction set (**-er**)
- an 'Accepted but Errors were Noted' acknowledgement for the transaction set (**-ew**).

See Acceptance or Rejection Criteria on page 57 for details about how **-er** and **-ew** work together.

Format of Parameters

-e<*option***>** *n*

Where:

<option>	**-e*r*** = reject transaction set
	-e*w* = accept transaction set with errors noted
n	One of the following error severity codes:
	0 = Ignore
	1 = Information
	2 = Warning
	3 = Error (Default)
	4 = Serious Error
	5 = User Level #1
	6 = User Level #2

Example

Accept sets with severities of 2 and 3 with errors. Reject sets with severities of 4 and above:

```
RespGen -i "HV_Results.txt" -o999 "EDI_999.txt" -er 4 -ew 2
```

Encoding

Sets the encoding for the output data.

See **UNICODE_at_Foresight.pdf** for details about encoding.

Format of Parameters

-e<*option***>**

Where:

<option> **-eUTF8** = Output should be UTF8 with BOM.

-eUTF16 = Output should be UTF16 with BOM

-eTEXT = Output should be ASCII with 999 IK4-04 populated even if its input was encoded.

If **-e** is omitted, the output will match the originating file but without any BOM.

Example

Creates a 999 encoded as UTF16:

```
RespGen -i "HV_Results.txt" -o999 "EDI_999.txt" -eUTF16 -ge
```

-f (Format)

Optional

The **-f** parameter can be used to specify numerous **f**ormatting options for the Response Generator output.

Format of Parameters

-f<*option***>**

Where -f*option* can be any of the following:

-f997_group_only	Causes 997s and 999s to have AK1 and AK9 segments only (no AK2/AK3/AK4/AK5).
-fak3	For warnings (severity of 2), Inserts AK3/AK4 in 997s and IK3/IK4 in 999s.
-fak901E	Causes Response Generator to use E in the 997/999 AK901 to indicate Accepted with Errors when all AK501s generated for the functional group are set to A or E. By default, the AK901 contains A under these circumstances.
-fca277	Generates a 277CA instead of a 277U (must be accompanied by -o277).
-fdo_gs08_passthrough	Causes GS08 for created 997 to mirror the GS08 of the source document. This option is incompatible with the -dav option, and Response Generator will fail if both are set.
-fdo277x070	Creates a 277 that conforms to the 3070x070A standard.
-fdo277x167	Creates a 277 that conforms to the 4040x167 standard.
-fdo824x166	Creates an 824 that conforms to the 824x166 standard.
-fdo824x186	Creates an 824 that conforms to the 824x186 standard.
-fdo864wrapper	Creates an 864 wrapper around a custom report. All enveloping flags are valid if this parameter is used.
-f997_no_ST03	Suppresses the ST03 in cases where the response document is greater than 4060.
-fdo999ne	Creates a non-errata version (based upon 005010x231) of the HIPAA 999.
-fdo_ctlnum_passthrough	If set, all control numbers (ISA, GS, ST) are passed through to all generated response documents.
-fdogs05_hhmm	If present, the format of the GS05 is set to four characters (HHMM).
-fdogs05_hhmmss	If present, the format of the GS05 is set to six characters (HHMMSS).

-fdogs05_hhmmssd	If present, the format of the GS05 is set to seven characters (HHMMSSD).
-fdogs05_hhmmssdd	If present, the format of the GS05 is set to eight characters (HHMMSSDD).
-fdo_gs06_passthrough	If set, outgoing response documents (864, 997, 824, and 277) will use the GS06 (Group Control Number) of the incoming document. MIT01 is currently set to GS06, so using the -fdo_gs06_passthrough flag will cause the MIT01 to be set to the incoming GS06.
-fdo_HIPAA_824	Generates an 824 that conforms to either the 824x166 or 824x186 standard, based upon the version of the source document.
-fdo_loop_rollup	Causes 277CA, 277H, and 277U responses to have one loop per provider, even if the incoming document had multiple 2000A loops for a single provider. This parameter should be used with caution as it can cause differences in the interpretation of bad claims between the response document and DocSplitter/Importer.
-fdostrict999	(Instream Healthcare Edition/HIPAA Only) If possible, generates a syntactically correct (types 1-2) 999 response that is valid according to the base X12 5010 999 syntactical requirements. **Note:** The generation of a 999 response with this specification is entirely dependent on your inbound data**.** The strict 999 implementation attempts to create a syntactically correct 999 response based on your inbound data. If any inbound data required for use in the 999 does not meet syntax (i.e., min/max lengths, missing mandatory elements, invalid characters, etc.), an error is generated and no 999 is created. **IMPORTANT:** It is recommended that you test use of this parameter thoroughly before promoting it to your production environment.
-fdostrictTA1	(Instream Healthcare Edition/HIPAA Only) If possible, generates a syntactically correct TA1 response that is valid according to the base X12 5010 TA1 syntactical requirements. **You must use the -oTA1 parameter with this parameter.** Example: `-oTA1 TA1.txt -fdostrictTA1` **Note:** The generation of a TA1 response with this specification is entirely dependent on your inbound data. The strict TA1 implementation attempts to create a syntactically

	correct TA1 response based on your inbound data. If any inbound data required for use in the TA1 does not meet syntax (i.e., min/max lengths, missing mandatory elements, invalid characters, etc.), an error is generated and no TA1 is created. **IMPORTANT:** It is recommended that you test use of this parameter thoroughly before promoting it to your production environment.
-fedifact_gen_una	Generates a UNA segment in the EDIFACT CONTRL response. (If this is not set, Response Generator generates a UNA if any delimiters are different from the defaults.)
-fedifact_ic_only	Causes the EDIFACT CONTRL document to generate a response for the status of the Interchange only.
-fedifact_no_ucm	Causes the EDIFACT CONTRL document to generate a response for the Interchange/Functional Group.
-fhm277	Causes the requested 277 to be generated in a specific format. This is a highly customized parameter that is only useful in very specific circumstances.
-floose_src_check	Causes Response Generator to return 100 (success) even in cases where it can't create a response document from the source, such as a 997 from a 997 or a 277 from a non-837.
-fno_AIS	If present, the generated 277U will use the value in a ZZKPA custom record that has been inserted into the validation detail file via business rules. Typically, the business rule that generates this record would be in the 837's claim loop. It will put it in the 277U's REF02 at 2200D or 2200E (REF01=1K). If the detail file does not contain the ZZKPA record, this flag is invalid.
-fno_AK103	Suppresses output of the 997 AK103 segment. By default, AK103s are created if the source document is version 4060 or greater or if a 999 is requested. If this parameter is present, AK103 output is suppressed.
-fno_ung	Turns off functional group (UNG) data in EDIFACT CONTRL responses
-fsa824 and fsa277	Creates 824s and 277s from clean documents.
-fuse_highest_99x	Generates a 999 if available according to the source version. This option overrides any previously specified output (such as -o997 or -o999).
-fuseshortGS05	If present, the GS05 is set to four characters (HHMM), instead of the default six (HHMMSS).

-fver_pass — Causes Response Generator to pass the ISA and GS version numbers for the source document through unchanged. ST03/AK0103/AK0203s are generated based on the presence or absence of ST03 in the source document.

-fWriteSTC12 — When generating a 277CA, causes Response Generator to write the DTL record's more descriptive error text (EMSG) into the STC12 when STC01.01=A3 and STC01.02=21 is encountered.

Example: Default behavior provides error information that may be considered too generic to be useful.

```
21=Missing or invalid information
```

If -fWriteSTC12 is used, the more descriptive error message found in the Instream Detail file is written into the STC12 segment (when STC01.01=A3 and STC01.02=21 are both present).

```
STC*A3:21*20151203*U*1000*******Ambiguous
Segment placement - multiple Loops/Groups
exist - using definition at Segment HI
(Principal Diagnosis) at 2-2310, unable to
determine correct Loop/Group for use~
```

Examples

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt"
-f997_group_only
```

```
RespGen -i "HV_Results.txt" -o277 "EDI_277.txt" -fca277
```

-ge (Generate interchange and functional group)

Optional

Generate interchange and functional group envelopes (the TA1 response does not need functional groups).

The **-di** options control various envelope elements. If no Sender and/or Receiver IDs are specified on the command line, Response Generator will use the Sender/Receiver IDs from the inbound envelope, reversing them for the outbound transaction.

Format of Parameters

-ge

Example

```
-i "C:\Files\HV_Results.txt" -o824 "C:\Files\EDI_824.txt"  -
o997 "C:\Files\EDI_997.txt"  -o277 "C:\Files\EDI_277.txt"  -ge
-dic 9001 -dac 1001 -s
```

-gTA1 (Generate TA1 segment)

Optional

Generate a TA1 segment within the 997 or 999. This determines the conditions under which a TA1 segment will be created.

Important: If you use **-**gTA1, use -o997, -o999, and/or -oTA1 to tell Response Generator where to put the TA1 segment. Otherwise, the -gTA1 is ignored.

If -gTA1 is omitted, but -oTA1 is used, then option 2 (Always) is assumed.

See TA1 Parameter Combinations on page 51.

Format of Parameters

-gTA1 *n*

Where:

n One of the following:

0 = Never (default).

1 = Generate TA1 acknowledgement for interchanges if the ISA14=1. This works if the ISA1 SVALU record is present in the results.

If the SVALU is not present, Response Generator outputs the TA1 by default.

2 = Always (default). If -gTA1 is omitted, but -oTA1 is used, then option 2 (Always) is assumed.

3 = If envelope errors occurred.

4 = If envelope errors occurred, the only segment between the ISA and IEA will be the TA1. No 997 or 999 data will be generated for that interchange.

5 = If envelope errors occurred, the only segment between the ISA and IEA will be the TA1. No 997 or 999 data will be generated for that interchange and no other output (824, 277, custom report, etc.) will be generated for that interchange.

6 = Output is controlled by the contents of the ISA14 as shown on page 51.

Example

See TA1 Parameter Combinations on page 51.

-i (Input File)

Required

Specifies the Instream validation detail results file to be used as input by Response Generator. This input file and a specified output file are the two required parameters for RespGen. All other parameters are optional.

Format of Parameters

-i *filename*

Where:

filename The validation results file to be used as input, including path and filename.

Example

Using the file "Validation_Results.txt" as input, output an 824 response document and save it to "EDI_824.txt":

```
RespGen -i "C:\Files\Validation_Results.txt" -o824
"C:\Files\EDI_824.txt"
```

-k (Lenient AK501)

Optional

Specifies that a 997 or 999 AK501 and AK509 can contain only A or E:

- If there are no errors, then the AK501 = A.
- If there are errors, regardless of type or number of claims, then the AK501 = E.

Important: The option -k overrides -pt.

Format of Parameters

-k

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -k
```

-l (Delimiters)

Optional

Note: See -lsource for an alternative option regarding delimiters.

Segment, element, composite subelement, and element repetition delimiters to use in the outbound EDI file. Each delimiter may be the actual character, or the ASCII number representing the character.

This ASCII number may be hexadecimal, by starting the number with 0x, octal by starting the number with a zero, or decimal.

If a new-line sequence is to be used as a segment terminator, set the Segment Delimiter to zero (0), and do NOT use the -n option below.

Default delimiters are:

Segment: ~

Element: *

Subelement: :
Element repetition ^

Format of Parameters

-l "segdelim,elemdelim,compdelim,elemrep"

See the examples for clarification of "segdelim,elemdelim,compdelim,elemrep".

Example

To change the segment terminator to the character |, and the element separator to the character #, use this parameter:

```
-l "|,#"
```

Segment terminator = |, element separator = \, and composite subelement separator = #

```
-l "|,\,#,&"
```

Segment terminator = | and element separator = \ (In the absence of a subelement separator, use two slashes to distinguish it from the escape character slash)

```
-l "|,\\"
```

Segment terminator = the RS control character (ASCII 30), composite sub-element separator= the character ^

```
-l "30,,^"
```

Instead of 30 for segment terminator, we could have used the hexadecimal equivalent 0x1E, or the octal equivalent 037.

-lsource (Use Source delimiters)

Optional

Directs RespGen to use the source document's delimiters for the response document delimiters.

Important: This requires the GEN 10223 record to be present.

Format of Parameters

-lsource

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -lsource
```

-n (No New-Line)

Optional

No new-line sequence follows each segment delimiter in the EDI output.

Format of Parameters

-n

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -n
```

-np (No Partial Functional Group)

Optional

No partial functional group acceptance is permitted when generating a 997 or 824 response. If any transaction is rejected, the entire functional group is rejected.

Default behavior is to partially accept a functional group if any transaction set is accepted, and reject a functional group only if all transaction sets are rejected.

Format of Parameters

-np

Example

```
RespGen -i "HV_Results.txt" -o824 "EDI_824.txt" -np
```

-nz (No Zero Length Files)

Optional

No zero length files are to be generated. If any generated files are 0 bytes long, they are deleted.

Format of Parameters

-nz

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -nz
```

-o (Output Type)

Required

Specifies the type of output to be generated from the Instream validation detail results input file and the file where the output should be saved. The input file (**-i**) and the specified output type and destination file are the two required parameters for RespGen. All other parameters are optional.

Format of Parameters

-o<*type*> *filename*

Where:

type Is any one of the available response types. See **Output Type Options** below.

filename The validation results file to be used as input, including path and filename.

Output Type Options

Any of these output types can be used in the *-o*type *filename* parameter:

277 Output either a 277U or a 277CA with filename *yyy*.

- For 277U: Do not use the -fca277 parameter. The 277U will contain output only if errors are encountered in the 837 input file, and version will be according to the Output Version rules on page 2.
- For 277CA: Include the -fca277 parameter. The 277CA will be generated regardless of errors in the input 837, and will include all claims. Version will be 5010.

824 Output an 824 with filename *yyy*. The 824 reports on errors only, so if the input contains no errors, then the 824 output will be empty.

997 Output a 997 with filename *yyy*. You cannot generate both a 997 and a 999.

999 — Output a 999 with filename *yyy*. You cannot generate both a 997 and a 999.

The type of 999 generated is based on published standards adopted for X12 999 implementation.

The following 4010 and 5010 HIPAA documents result in a 999 5010X231 or 5010X231A1 response:

269 005010X187	277 005010X212	837I 005010X223
270 005010X203	278 005010X217	837D 005010X224
271 005010X203	820 005010X218	
274 005030X209	834 005010X220	
275 005040X254	835 005010X221	
276 005010X212	837P 005010X222	

This type of 999 includes:

- 5010X231 or 5010X231A1 in the GS08
  Example:
  `GS*FA*IND*DC0*20160302*130553*1*X*`**`005010X231A1`**`~`
- CTX segments, if they appear in the DTL file.

All other 4010 and 5010 documents, such as 824, 850, etc., result in a 5010 response.

This type of 999 includes:

- 5010 in the GS08
  Example:
  `GS*FA*DC0*IND*20160302*125531*1*X*`**`005010`**`~`
- **No** CTX segments, even if they appear in the DTL file.

ctl — Output an EDIFACT CONTRL message with filename *yyy*.

msa — Output an HL7 MSA (Message acknowledgement segment) with filename *yyy*.

TA1 — Output a file containing a TA1 segment surrounded with ISA and IEA, instead of putting the TA1 in the 997 or 999 file. (See also -gTA1.) Note that this parameter produces a TA1 **even if syntactically incorrect data means the TA1 is invalid**.
Example: -oTA1 TA1file.txt

If -gTA1 is not used, a TA1 value of 2 (Always) is assumed. If -gTA1 is 0, a warning is issued and no TA1 is created.

Use -fdostrictTA1 to produce a TA1 only when the TA1 is valid according to the base X12 5010 TA1 syntactical requirements.

See TA1 Parameter Combinations on page 51.

text	Used to generate custom report output. See "-otext (Output Custom Report)".

Example

Using the specified input file, output an 824 response document and save it to "EDI_824.txt":

```
RespGen -i "C:\Files\Validation_Results.txt" -o824
"C:\Files\EDI_824.txt"
```

-otext (Output Custom Report)

Optional

Generates text, formats the text using the specified template, and outputs a custom report. You can only output one custom report at a time (the command line cannot have two **-otext** parameters). The input EDI that was validated can be 837, 834, 835, 276 or 277u.

See page 65 for details about custom reports and templates.

Format of Parameters

-otext -tpl *tplfilename* (-te *n* -tel *nn* -tev *nn*) *filename*

Where:

tpl — Required. Specifies a template (***tplfilename***) to be used when creating the custom report.

te — Optional. Specifies the contents of the custom report, where *n* =

0	All
1	Errors only (default)
2	Error-free only
3	Claims with errors, but only if the errors are not envelope-related

tel — Optional. Specifies which error types are included in the report.
Example: `-tel 123`

If **-tel** is omitted, all error types are included in the report.

tev — Optional. Specifies which severity levels are included in the report.
Example: `-tev 34`

If **-tev** is omitted, all severity levels are included in the report.

filename — Required. The name of the file in which the -otext output should be saved.

See page 65 for details about custom reports.

Example

Using the specified input file, output a text response document that uses the template 824Custom(-tpl) shows all claims (-te 1) and includes errors of types 1, 2, and 3 (-tel 123), all severity levels (-tev option is omitted) and save it to "Custom_824.txt":

```
RespGen -i "C:\Files\Validation_Results.txt" -otext -tpl
(C:\Templates\824Custom" -te 1 -tel 123
"C:\Files\Custom_824.txt"
```

-pt (Partial transaction set acceptance)

Optional

Partial transaction set acceptance. A transaction set can be accepted with errors if you use Response Generator's -pt parameter. You can then run the EDI through Docsplitter and process the valid claims.

This parameter can be used when responding to 834, 835, and 837 transactions.

If a transaction set contains errors, the 997/999 created by Response Generator will partially accept the transaction set by placing an E in the AK501.

Important: The option -k overrides -pt. If you use both, -pt is ignored.

Note: This parameter is useful when you are using Docsplitter. It tells your partner that some claims in the transaction were good and others were not.

A variation is -pt p, which causes a P to be placed in the AK501 if the transaction set has errors.

	AK501 Default	AK501 -pt	AK501 -pt p
Clean set	A	A	A
Errors in set	R	E	P
Warnings in set	E	E	E
Errors in all sets	R	R	R

Format of Parameters

-pt (p)

Where:

p — Optional. Causes a P to be placed in the AK501 if the transaction set has errors. (P is not a standard code and should be used only with the agreement of your trading partners and your translator.)

-pt824 (Partial transaction set acceptance for 824 responses)

Optional

The -pt824 parameter allows partial transaction set acceptance when generating an 824 response. You can then run the EDI through Docsplitter and process the valid claims.

This parameter can be used when responding to any transaction.

If one or more transaction sets (but not all) contains errors, the 824 created by Response Generator will partially accept the transaction set by placing a TE in the OT101.

Note: This parameter is useful when you are using Docsplitter. It tells your partner that some claims in the transaction were good and others were not.

	OT101 Default	OT101 -pt824
Clean set	TA	TA
Errors in set	TR	TE
Warnings in set	TE	TE
Errors in all sets	TR	TR

Format of Parameters

-pt824

-s (Strict Processing)

Optional

Use strict 997 or 999 response processing in which:

- 997s or 999s will only show error types 1 and 2, regardless of whether there is also an 824 generated.
- 824s will only show types 3-7 regardless of whether there is also a 997 or 999 generated.

This parameter overrides the default behavior, which is as follows:

If only one of these types of output transactions are generated …

997/999

824

… then ALL errors, regardless of type, are put in the one type generated.

For details, see Appendix B: Combinations of X12 Response Documents on page 119.

Format of Parameters

-s

Example

```
-i "C:\Files\HV_Results.txt" -o824 "C:\Files\EDI_824.txt"  -s
```

-stc (STC override file)

Optional

STC override file for 277CA. See Appendix G: 277CA STC Override on page 175.

Format of Parameters

-stc

Example

```
RespGen -i "Current_Results.txt" -o277CA "EDI_277CA.txt" -
fca277 -stc c:\Foresight\Instream\Bin\stcoverride.csv
```

-TPA (Invoke TPA File)

Optional

Note: Not available when running Response Generator from an API.

Identifies a Trading Partner Automation lookup file that specifies which trading partners are to use which Response Generator setup files. The TPA lookup file can replace any or all command-line options except the input file.

When using -TPA, do not use other command line parameters.

Before using -TPA:

1. Put all desired options into Response Generator setup file(s).
2. Be sure all trading partners are assigned a settings file in the TPA CSV file.
3. Use -TPA to invoke the TPA lookup file.

See **TIB_fsp-instream_*<n.n>*_tpa.pdf**. For an example Response Generator setup file that is appropriate for TPA, see TPA_ResponseGen.ini in Instream's DemoData directory.

Format of Parameters

-TPA "*file_path*"

Where:

File_path The file path/name of a Trading Partner Automation lookup file.

Example

```
RespGen -i "C:\Files\HV_Results.txt" -o824
"C:\Files\EDI_824.txt" -TPA
"C:\lookups\SampleTPA_DS_RG.csv"
```

-u (use $Dir.ini)

Optional

Use $Dir.ini to find configuration files ErrRespXref.txt and ErrMsgTrans.txt. Response Generator will look for $Dir.ini in its current folder, read the BASEROOT value, and look in the \bin directory under BASEROOT for the configuration files.

Important: -cd and -u cannot be used at the same time. If neither is used, RespGen.exe will look in its own directory for these configuration files.

Format of Parameters

-u "***dir_path"***

Where:

dir_path Path to follow to find $Dir.ini.

Example

Directs RespGen to look in\\Server1\Instream\bin.

```
RespGen -i "C:\Files\HV_Results.txt" -o824
"C:\Files\EDI_824.txt" -u \\Server1\Instream
```

-v (Verbosity)

Optional

Set the output verbosity level.

Note: If a file can't be found, associated "File Not Found" errors are displayed despite the verbosity (-v) setting.

Format of Parameters

-v *n*

Where:

n Verbosity level:

0 = No Output

1 = Errors Only

2 = Warnings and Errors

3 = Info, Warnings (default)

9 = Debug (lots of output)

Example

```
RespGen -i "C:\Files\HV_Results.txt" -o824
"C:\Files\EDI_824.txt" -v 3
```

-version (Display Version)

Optional

Displays Response Generator version (see Response Generator Version below). Do not use this qualifier with other qualifiers, including -i/-o.

Note: This is the only parameter that can be used alone.

Format

-version

Example

Display the Response Generator version:

```
"C:\Foresight\Instream\Bin\RespGen.exe" -version
```

-w (Wait)

Optional

Waits for user to 'press any key' upon completion or cancellation. This option is most useful for batch file processing.

Format

-w

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -w
```

-y (Overwrite OK flag)

Optional

Overwrite OK Flag. The EDI output file, if it exists, will be overwritten without prompting. This option is most useful for batch file processing.

Important: This option will not work if the append option (**-a**) is also specified.

Format

-y

Example

```
RespGen -i "HV_Results.txt" -o997 "EDI_997.txt" -y
```

-z (Use Setup File)

Optional

Use a Response Generator setup file. (See TPA-Parameter Setup File Contents on page 126 and RED Segments on page 49).

Format

-z"file_path"

Where:

file_path Is the file path/name of a Response Generator setup file.

Example

```
"C:\FS\Bin\respgen.exe"
-i"C:\FS\output\MY837IA_Results.txt"
-o824 "C:\FS\Output\824.txt" -o997 "C:\FS\Output\997.txt"
-z"C:\FS\BIN\RG.INI"
```

Example Response Generator Commands

Assume these file names:

Validation detail results file	HV_Results.txt
997 Output	EDI_997.txt
824 Output	EDI_824.txt
277 Output	EDI_277.txt

The example command lines below use all defaults except:

- Interchange and functional group envelopes will be generated (-ge)
- Interchange Control Numbers start with 9001 (-dic)
- Functional Group Control Numbers start at 1001 (-dac)
- Strict mode will be used (-s)

Windows Example

All files are in the **/Files** directory.

```
"C:\Foresight\Instream\Bin\respgen.exe"
-i ”C:\Files\HV_Results.txt” -o824 ”C:\Files\EDI_824.txt”
-o997 ”C:\Files\EDI_997.txt”  -o277 ”C:\Files\EDI_277.txt”
-ge -dic 9001 -dac 1001 -s
```

UNIX Example

All files are in the **/Files** directory.

```
/HVInStream/bin/RespGen -i "/Files/HV_Results.txt"
-o824 ”/Files/EDI_824.txt”  -o997 ”/Files/EDI_997.txt”
-o277 ”/Files/EDI_277.txt”  -ge -dic 9001 -dac 1001 -s
```

Trading Partner Automation for Response Generator

You can set up trading partner automation to specify different Response Generator options, depending on certain values in the ISA and GS. This can replace any or all command-line options except the input file.

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

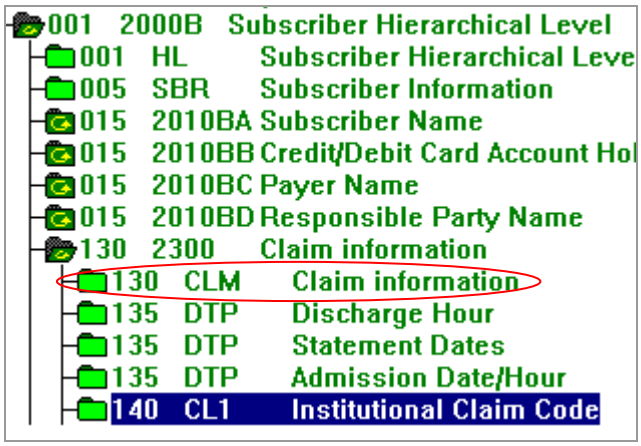
RED Segments

See Example Response Generator Commands on page 48 for options affecting the contents of the RED01 element.

Default behavior

- **If an error occurs within a claim**, the 824 response will include a TED segment explaining the error followed by a RED segment giving the claim ID.

  Example: An error in the CLM segment will result in a TED segment followed by a RED segment that contains the Claim ID:



  The 824 data might look like this:

```
TED*024*X7 E 30393 2300 The Facility Code 00 is not …
RED*1*AI~
```

- **If an error occurs above the claim level**, the 824 response will include a TED segment explaining the error followed by a RED segment giving a generic message.

  Example: An error in the 2010BA Subscriber Name loop will result in a TED segment followed by a RED segment containing a generic message in the RED01.

  The 824 data might look like this:

```
TED*024*X7 E 30373 2010BD The Zip code 99999 was …
RED*Not Claim Specific*AI~
```

Alternative RED Segment Usage

When an error occurs above the claim level, you will normally get one generic RED segment like this:

```
TED*024*X7 E 30373 2010BD The Zip code 99999 was …
RED*Not Claim Specific*AI~
```

You can instead get a RED segment for each claim that is under the level with the error, with the RED01 containing the claim ID:

```
TED*024*X7 E 30373 2010BD The Zip code 99999 was …
RED*1*AI~
RED*2*AI~
RED*3*AI~
```

To do this, create an INI file containing:

```
[824]
Display837ClaimID=1
```

If you want errors in Table 1 and at the top of Table 2, before the 2000 loop, to also trigger the additional RED segments, add this line below Display837IclaimID:

```
ShowWhenErrorAbove2000=1
```

Run Response Generator with the z command line parameter that points to the setup file:

```
"C:\FS\Bin\respgen.exe" -i "C:\FS\output\MY837IA_Results.txt"
-o824 "C:\FS\Output\824.txt" -o997 "C:\FS\Output\997.txt"
-z"C:\FS\BIN\RG.INI"
```

See Appendix D: Z-Parameter Setup File on page 141.

TA1 Parameter Combinations

Two command line parameters work together to determine TA1 output:

-gTA1 determines the conditions under which a TA1 segment will be created. If omitted, but -oTA1 is used, then TA1 of 2 (always) is assumed.

-oTA1 causes TA1 output to go to a separate file containing only ISA, TA1, and IEA. If omitted, the TA1 output goes in the 997/999 specified with the o997 or o999 parameter. If -oTA1 and o997/o999 are both omitted, no TA1 is created.

If you specify a file name for the TA1 output (-oTA1 "file"), and use a gTA1 option of **0** (Never), then a warning message appears and no TA1 is created.

Example

Definitions used in table below

TA1+env.	Full 997/999	Full 997/999 with TA1 seg.
ISA	ISA	ISA
TA1	GS	TA1
IEA	ST	GS
	AKx	ST
	SE	AKx
	GE	SE
	IEA	GE
		IEA

This example uses an EDI input file with these three ISAs, in order:

1 ISA1 - No errors

2 ISA2 – Error on ISA

3 ISA3 – Error below ISA

Combinations of command line parameters for TA1

	Error location	When command line has -o997 or -o999 Only	When command line has both -o997 and -oTA1 or both -999 and -oTA1	
-gTA1 option		**997 or 999 output**	**997 or 999 output**	**TA1 output**
0 (never)	No errors in ISA	Full 997 or 999	Full 997 or 999	None
	Error in ISA	Full 997 or 999	Full 997 or 999	None
	Error below ISA	Full 997 or 999	Full 997 or 999	None
1 (when ISA14=0)	No errors in ISA	Full 997 or 999	Full 997 or 999	None
	Error in ISA	Full 997 or 999	Full 997 or 999	None
	Error below ISA	Full 997 or 999	Full 997 or 999	None
1 (when ISA14=1)	No errors in ISA	Full 997 or 999 with TA1 seg.	Full 997 or 999	TA1+env
	Error in ISA	Full 997 or 999 with TA1 seg.	Full 997 or 999	TA1+env
	Error below ISA	Full 997 or 999 with TA1 seg.	Full 997 or 999	TA1+env
2 (always)	No errors in ISA	Full 997 or 999 with TA1 seg.	Full 997 or 999	TA1+env
	Error in ISA	Full 997 or 999 with TA1 seg.	Full 997 or 999	TA1+env
	Error below ISA	Full 997 or 999 with TA1 seg.	Full 997 or 999	TA1+env
3 (ISA errors)	No errors in ISA	Full 997 or 999	Full 997 or 999	None
	Error in ISA	Full 997 or 999 with TA1 seg.	Full 997 or 999	TA1+env
	Error below ISA	Full 997 or 999	Full 997 or 999	None
4 (ISA errors: TA1+env, no 997)	No errors in ISA	Full 997 or 999	Full 997 or 999	None
	Error in ISA	TA1+env	None	TA1+env
	Error below ISA	Full 997 or 999	Full 997 or 999	None
5 (ISA errors: TA1+env, no other output)	No errors in ISA	Full 997 or 999 with TA1 seg.	Full 997 or 999	TA1+env
	Error in ISA	TA1+env	None	TA1+env
	Error below ISA	Full 997 or 999 with TA1 seg.	Full 997 or 999	TA1+env
6 (when ISA14=0)	No errors in ISA	Full 997 or 999	Full 997 or 999	None
	Error in ISA	TA1+env	None	TA1+env
	Error below ISA	Full 997 or 999	Full 997 or 999	None
7 (when ISA14=0)	No errors in ISA/GS	Full	Full	None
	Error in ISA/GS	TA1+env	None	None
	Error below ISA/GS	Full	Full	None
7 (when ISA14=1)	No errors in ISA/GS	Full (no TA1?)	Full	None
	Error in ISA/GS	TA1+env	None	TA1+env
	Error below ISA/GS	Full (no TA1?)	Full	None

TA105 Error Codes

To use your own criteria for putting a value into the TA105, use a business rule to generate these error numbers in the detail results file.

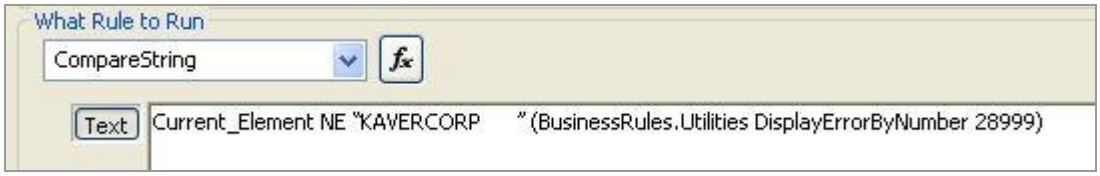
To generate this TA105 value …	Generate this error number
005 - Invalid Interchange ID Qualifier for Sender	28996
006 - Invalid Interchange ID for Sender	28997
007 - Invalid Interchange ID Qualifier for Receiver	28998
008 - Invalid Interchange ID for Receiver	28999
025 - Duplicate interchange control number	29000

To get any of these values, write a business rule specifying the error condition. You can attach it to:

- The ISA segment
- An element in the ISA
- The IEA segment
- An element in the IEA

See **BusinessRules.pdf** for details on writing business rules.

Example: This business rule on the ISA08 will cause error number 28999 if the ISA08 does not equal "KAVERCORP ".



Assume that you use this rule to validate this ISA08:

```
ISA*00*          *00*          *01*9012345720000  *01*9088877320000  *030212 …
```

The Instream validation detail results file will contain this error message:

```
DTL          1       ISA I07            1 8          0                    28999 3    01
```

Response Generator will create this TA1:

```
TA1*000000112*030212*0848*R*008~
```

Note
If Response Generator detects more than one error on the ISA - IEA, it reports the first one.

AK3 and IK301 Elements

The 997 AK301 and the 999 IK301 are mandatory elements that contain the tag of the segment containing the error. If the AK301 or IK301 refer to a segment with a problem tag, Response Generator handles it this way:

Problem with Tag	Contents of AK301 / IK30l	Example original segment tag	Corresponding AK301/ IK30l
missing tag	three question marks	<empty>	???
tag too short	tag is padded with trailing question mark	T	T?
tag too long	tag is truncated	TEST	TES

Multiple 997, 999, TA1, and Custom Report Files

Response Generator can create multiple 997, 999, TA1, and custom report files for each interchange or functional group file. To do this, include a variable name, surrounded by pound signs, in the output filename:

Variable	Contents of Variable
ICount	Counter that increments at each interchange (ISA) in the input file.
GCount	Counter that increments at each functional group (GS) in the input file. This increments within the interchange and resets to 1 with each new ISA. This variable cannot be used in a filename if the file might contain a TA1 segment. Therefore, don't use GCount in: ▪ a TA1 file name ▪ a 997 or 999 file name when -gTA1 is > 0 *and* -oTA1 is NOT used

To create a unique identifier, GCount should be used in combination with ICount, since GCount is the count WITHIN the current interchange.

If a file exists with the same name, it is overwritten.

Example custom report:

```
-otext "C:\Dir1\MyReport-#ICount#-#GCount#.txt"
```

Example 997:

```
-o997 "C:\Dir1\Test997-#ICount#-#GCount#.txt"
```

When the file is created, #ICount# and #GCount# are replaced with the interchange counter and group counter, creating these files:

Test997-1-1.txt	First functional group in first interchange.
Test997-1-2.txt	Second functional group in the first interchange
Test997-2-1.txt	First functional group in the second interchange

Response Generator Version

The **-version** command-line parameter displays the Response Generator version, as in this example batch file:

```
"C:\Foresight\Instream\Bin\RespGen.exe" -version
pause
```

To see the version and a list of command-line parameters, run Response Generator without any parameters, as in this example batch file:

```
"C:\Foresight\Instream\Bin\RespGen.exe"
pause
```

You can also run the **Version** script in Instream's Scripts directory.

3 Acceptance or Rejection Criteria

Enveloping Errors

Response Generator cannot build a good response document if the input data has severe enveloping errors. It handles enveloping errors this way:

- Interchange (ISA-IEA) errors:

  Create a TA1 if it is the only response document requested. Otherwise, it will stop processing.

- Severe Functional Group (GS-GE) error such as missing GS, or so invalid that Response Generator cannot create a valid response GS from it:

  Create a TA1 if it is the only response document requested. Otherwise, it will stop processing.

- Other Functional Group errors such as wrong transaction count or functional group control number is invalid:

  997/999 with the error reported in the AK9

Errors that always cause rejection

Certain enveloping errors cause Response Generator to reject, regardless of the severity assigned to these errors in the APF file. These are errors that prevent proper validation, and include these error numbers:

15090: Bad Partner ID Lookup (Force start of IC!)
10001 Segment/Element terminator is suspicious
10007 Segment *xxx* terminated by control character
10201 Could not interpret functional group header segment
10220: Missing Functional Group Header
10614 Missing Mandatory

10900: Functional Group Count *nnn* Incorrect - should be *nnn*

10903: Incorrect *xxx* Count *nnn* - should be *nnn*

10911: Functional Group Ctl No. xx in GE doesn't match GS

10912: Interchange Ctl No. xx in IEA doesn't match ISA

10913: Transaction Set Ctl No. xx in SE doesn't match ST

10917: Transaction Set segment total *nnn* Incorrect - should be *nnn*

10918: Incorrect *xxx* Count *nnn* - should be *nnn*

11003: Can't find Transaction Set *xxx*

11206: Functional Group Ctl. No. n not properly terminated

11207: Interchange Ctl. No. 000000001 not properly terminated

11208 Missing SE

11209 Missing GE

33333 Missing ST

Transaction Sets

Response Generator determines its response based upon its analysis of the Instream detail file's DTL and GEN records. Each Instream message has a severity level:

0 = Ignore

1 = Informational

2 = Warning

3 = Error

4 = Fatal Error

5 = User Error #1

6 = User Error #2

Response Generator has an Error Limit and a Warning Limit, which is set to one of these severities. These determine if a validation message should result in the rejection of a transaction set, an acceptance of the transaction set with errors, or a full acceptance of the transaction set.

Error Limit

The Error Limit is a number from 0 through 6 corresponding to a severity level. Errors at or above that setting cause rejection of the transaction set.

Example: With an Error Limit of 3, any error with a severity of 3, 4, 5, or 6 will cause rejection of the transaction set.

If validation found an error with a severity equal to or greater than the Error Limit, then Response Generator rejects the transaction set and generates the appropriate 997 AK3/AK4, 999 IK3/IK4, and/or 824 OTI/TED segments.

At the end of the transaction set processing, Response Generator will generate a 997 AK5-01 with an Acknowledgement Code of **R** for **Rejected**, a 999 IK5-01 of R, and/or an 824 OTI Application Acknowledgement Code of **TR** for **Transaction Set Reject**.

The default Error Limit is 3 (Error), but this can be overridden with the **-er** command-line option, or the **SetRejectSeverity()** API function

Warning Limit

The Warning Limit is also a number from 0 through 6 corresponding to a severity level. It determines what happens to the severities *not covered by* the Error Limits. It should always be less than or equal to the Error Limit.

If validation found a severity equal to or greater than the Warning Limit, but less than the Error Limit, then Response flags the transaction set as **Accepted with Errors**.

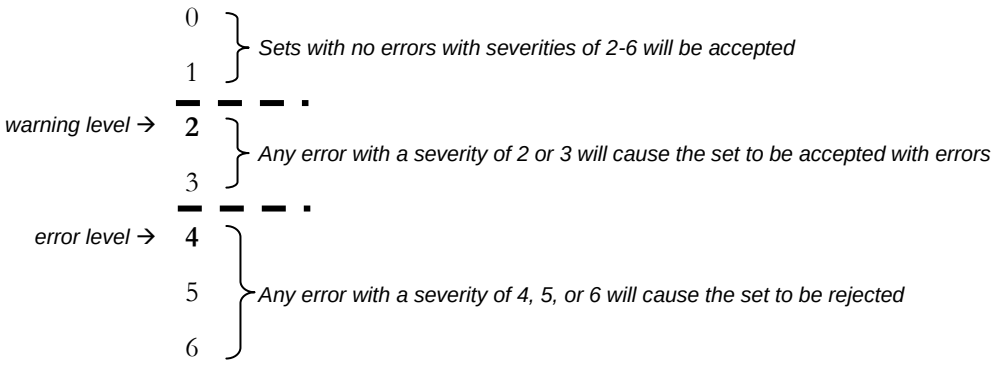
This causes a warning flag to be set for the transaction set, and generation of the appropriate 824 OTI/TED segments.

At the end of the transaction set processing, if the transaction set did not contain an error, Response Generator will generate an AK5-01 or IK5-01 with an Acknowledgement Code of **E** for **Accepted But Errors Were Noted**, and/or an 824 OTI Application Acknowledgement Code of **TE** for **Transaction Set Accept with Error**.

The default Warning Limit is 2 (Warning), but it can be overridden with the **-ew** command-line option, or the **SetWarningSeverity()** API function.

How Warning Limits and Error Limits Work Together – Example

If you run Response Generator with a Warning Limit of 2 and an Error Limit of 4:



If -er is …	and -ew is …	These severities cause Rejection	These severities cause Acceptance with Error
4	2	4, 5, 6	2, 3
3	2	3, 4, 5, 6	2
3	3	3, 4, 5, 6	none

-pt Parameter

A transaction set can be accepted with errors if you used Response Generator's -pt parameter. You can then run the EDI through Docsplitter and process the valid claims.

If you use the -pt parameter, you will probably need to communicate which claims are not getting paid by generating a 277 with the -o277 parameter.

Functional Groups

When Response Generator reaches the end of a functional group in the Instream detail results file, it then decides whether to accept, partially accept, or reject the functional group, and generates an appropriate 997 or 999 AK9 (Functional Group Response Trailer) segment:

- If all transaction sets were rejected, then the functional group is rejected (997 or 999 AK9 Acknowledgement Code of **R**).
- If at least one transaction set was rejected, but not all, then the functional group is partially accepted (AK9 Acknowledgement Code of **P**)
- If all transaction sets were accepted or accepted with errors noted, then the functional group is accepted (AK9 Acknowledgement Code of **A**)
- If all transaction sets were accepted with errors noted, then the functional group is accepted with errors (AK9 Acknowledgement Code of E) if the -fAK901E flag is set.

If you don't wish to allow a functional group to be partially accepted, use the command-line option -np, or the API function **SetNoPartialFGAcceptFlag()**. In this mode, Response Generator will reject a functional group if **any** transaction set is rejected.

4 Enveloping Format

ISA

Element			Type / Length	Contents
ISA01	I01	Authorization Information Qualifier	ID 2	**00** no authorization information present
ISA02	I02	Authorization Information	AN 10	Blank (ten spaces)
ISA03	I03	Security Information Qualifier	ID 2	**00** no security information present
ISA04	I04	Security Information	AN 10	Blank (ten spaces)
ISA05	I05	Interchange ID Qualifier	ID 2	From initiating transaction's ISA07: Interchange ID Qualifier (Receiver)
ISA06	I06	Interchange Sender ID	AN 15	From initiating transaction's ISA08: Interchange Receiver ID
ISA07	I05	Interchange ID Qualifier	ID 2	From initiating transaction's ISA05: Interchange ID Qualifier (Sender)
ISA08	I07	Interchange Receiver ID	AN 15	From initiating transaction's ISA06: Interchange Sender ID
ISA09	I08	Interchange Date	DT 6	Current date in YYMMDD format
ISA10	I09	Interchange Time	TM 4	Current time in HHMM format
ISA11	I10	Interchange Control Standards Identifier	ID 1	**U** US EDI Community of ASC X12, TDCC, and UCS
ISA12	I11	Interchange Control Version Number	ID 5	**00401** for 997 **00501** for 999
ISA13	I12	Interchange Control Number	N0 9	Specified interchange control number, zero-filled
ISA14	I13	Acknowledgment Requested	ID 1	**0** No acknowledgment requested
ISA15	I14	Usage Indicator	ID 1	**P** Production Data
ISA16	I15	Component Element Separator	AN 1	Sub-element separator character in effect (Default ':')

IEA

Element			Type and Length	Contents
IEA01	I16	Number of Included Functional Groups	N0 1/5	Number of functional groups included in interchange
IEA02	I12	Interchange Control Number	N0 9	Specified interchange control number, zero-filled. Same as ISA13

GS

Response Generator will optionally generate:

- A new interchange envelope whenever a new interchange envelope is seen in the Instream output.
- A new functional group at the beginning of the interchange, and then when functional groups are seen in the Instream output with different Sender (GS02) or Receiver (GS03) IDs.

Normally, Response Generator uses the Sender and Receiver IDs from the EDI data, and reverses them, since the response is going back. You can override the Sender and/or Receiver IDs on the command-line. These overrides remain in effect for the entire run.

Element			Type and Length	Contents
GS01	479	Functional Identifier Code	ID 2	**FA** functional acknowledgment
GS02	142	Application Sender's Code	AN 2/15	From initiating transaction's GS03: Application Receiver's Code
GS03	124	Application Receiver's Code	AN 2/15	From initiating transaction's GS02: Application Sender's Code
GS04	373	Date	DT 8	Current date in YYYYMMDD format
GS05	337	Time	TM 4/8	Current time in HHMMSS format
GS06	28	Group Control Number	N0 1/9	Specified Group Control Number
GS07	455	Responsible Agency Code	ID 1/2	**X** Accredited Standards Committee X12

Element			Type and Length	Contents
GS08	480	Version/Release/Industry Identifier Code	AN 1/12	This assumes that it isn't overridden on the command line with -fdo_gs08_passthrough or -dav. **997** 005010 unless the source document is 4010, then 004010. **999** 005010X231A1. **824** 004050X166 if -fdo824x166 is set. 005010X186 if -fdo82x186 is set and source document is 5010. 004010 if the source document is 4010. 005010 if none of the above are true. **277** 004010H01 if -fhm277 is set. 005010X214 if -fca277 is set and source is 5010. 003070X070A1 if -fdo277x070 is set. 00404X167 if -fdo277x167 is set. 004010 if the source document is 4010. 005010X212 if none of the above are true. **864** 005010 unless the source document is 4010, then 004010.

GE

Element			Type and Length	Contents
GE01	97	Number of transaction sets Included	N0 1/6	Number of transaction sets included in functional group
GE02	28	Group Control Number	N0 1/9	Specified group control number. Same as GS06.

TA1

Element			Type and Length	Contents
TA101	I12	Interchange Control Number	N0 9	From initiating transaction's ISA13: Interchange Control Number
TA102	I08	Interchange Date	DT 6	From initiating transaction's ISA09: Interchange Date
TA103	I09	Interchange Time	TM 4	From initiating transaction's ISA10: Interchange Time
TA104	I17	Interchange Acknowledgement Code	ID 1	**A** or **R**
TA105	I18	Interchange Note Code	ID 3	Code representing error

5 Custom Output

Overview

You can create a customized report from the following transactions processed by Response Generator:

- 270
- 276
- 277CA, 277U (4010 only), and 277X228
- 278
- 820x218 and 820x306
- 834
- 835
- 837

It can contain literal text, values from the data, and error information.

This is done by:

- creating a custom report template specifying the information you want to see
- directing Response Generator to generate a text report (using -otext) and apply the desired template (using -tpl) to the output.

Custom Report Example 1 - Plain Text

```
Error Report on: 02/12/2006 for Interchange Number 000000386
-----------------------------------------------------------------

LastName (Insured): SMITH
FirstName (Insured): MUFFY
LastName (Patient):
FirstName (Patient):
ClaimNumber: 1
TotalAmt: $100.00
StatementDate: 20030212-20030213
Error: :

LastName (Insured): SMITH
FirstName (Insured): MUFFY
LastName (Patient):
FirstName (Patient):
ClaimNumber: 10
TotalAmt: $1000.00
StatementDate: 20030212-20030213
Error: 10613: Svc: Element SV204 (D.E. 355) at col. 26 is missing,
though marked "Must Be Used"
       10613  Svc: Element SV205 (D.E. 380) at col. 26 is missing,
though marked "Must Be Used"
```

Custom Report Example 2 - Columns

```
Claim #      Claim Amount   Err #   Error Description
----------   ------------   -----   -------------------------

2                 $200.00   10811   Clm: Missing Segment DTP (Stat
                                    ement Dates) at 2-135, though
                                    marked "Must Be Used"

4                 $400.00   30354   Svc: The Previous Claim does n
                                    ot balance. Formula used CLM02
                                     (400.00) = sum of SV203 (665.
                                    75)

6                 $600.00   10605   Clm: Code Value "NDDDGDN" not
                                    found in the dictionary code l
                                    ist for PWK01 (D.E. 755) at co
                                    l. 5
```

Sample Scripts

Custom Report Example Scripts

Go to Instream's **Scripts** directory and execute some Response Generator scripts to see some of its output capabilities. The scripts with RG in the name will include Response Generator, like this:

`V_RG_277_4010_textrpt_all`

See **Demo_Index.pdf** for a complete list of scripts.

Errors and Claim Rejection

An 837 (and 277) has these levels:

- Interchange
  - Functional Group
    - Transaction Set
      - Trading Partner and Payer
        - Provider
          - **Subscriber**
            - Claim
              - Services
            - Dependent
              - Claim
                - Services

An error at a particular level usually causes rejection of all claims subordinate to it. For example, an error at the Subscriber level will cause all of the subscriber's claims to be rejected, even if the claims themselves have no errors.

Response Generator checks the **-er** command line parameter to see what severity is to be considered as an "error." Default is 3.

Command Line for Custom Reports

These command-line options request a custom report:

Option	Explanation
-otext "*output file name*"	*Mandatory if you want to generate a custom report.* Path and filename to use for the custom report. Example: `-otext "C:\Foresight\Instream\Output\textout.txt"`
-tpl "*template file name*"	*Mandatory if you want to generate a custom report.* Path and filename for the template. Example: `-tpl "C:\Foresight\Instream\Scripts\RGtemplate.txt"`
-te *n*	*Optional. Default is 1.* Specifies what claims are to be included in the custom report: **`-te 0`** – all claims go in the report. **`-te 1`** – claims with errors go in the report. **`-te 2`** – claims with no errors go in the report. **`-te 3`** – claims with non-envelope errors go in the report.
-tel *nn*	Claims with the error types specified go in the report. Example: `-tel 1237` puts claims with errors of types 1, 2, 3, and 7 in the report. If **-tel** is omitted, all error types go in the report.
-tev *nn*	Claims with the severity levels specified go in the report. Example: `-tev 345` puts claims with errors of severity 3, 4, and 5 in the report. If **-tev** is omitted, all severity levels go in the report.

Example Windows command line for a custom report

```
"C:\Foresight\Instream\Bin\respgen.exe" -i "C:\Files\Validation_Results.txt"
-otext "C:\Files\textout.txt" -tpl "C:\Files\RGtemplate837P.txt" -te 0 -s
```

Part of Command	Explanation
`"C:\Foresight\Instream\Bin\respgen.exe"`	Run Response Generator
`-i "C:\Files\Validation_Results.txt"`	Input validation results file
`-otext "C:\Files\textout.txt"`	Requests custom report output
`-tpl "C:\Files\RGtemplate837P.txt"`	Template for formatting the report
`-te 0`	Include all claims, whether they have errors or not

Example Windows command line for a custom report and a 997

```
"C:\Foresight\Instream\Bin\respgen.exe" -i "C:\Files\Validation_Results.txt"
-o997 "C:\Files\997.txt" -ge -dic 9001 -dac 1001 -otext "C:\Files\textout.txt"
-tpl "C:\Files\RGtemplate837P.txt" -te 0 -s
```

Example UNIX command line for a custom report

```
/HVInStream/bin/RespGen
-i"/HVInStream/output/Tutorial837IA_Results.txt"
-otext "/HVInStream/output/Tutorial837IA_textout.txt"
-tpl "/HVInStream/DemoData/RGtemplate837I.txt" -te 0 -s -y
```

Command	Explanation
`/HVInStream/bin/RespGen`	Run Response Generator
`-i"/HVInStream/output/ Tutorial837IA_Results.txt"`	Input is the results file from validation
`-otext "/HVInStream/output/ Tutorial837IA_textout.txt"`	Requests custom report output, specifies filename
`-tpl "/HVInStream/DemoData/ RGtemplate837I.txt"`	Specifies the template to use for formatting the report
`-te 0`	Include all claims, whether they have errors or not
`-s`	Use the strict option
`-y`	Overwrite output files without prompting

Example Unix command line for a custom report and a 997

```
/HVInStream/bin/RespGen
 -i"/HVInStream/output/Tutorial837IA_Results.txt"
-o997 "/HVInStream/output/EDI_997.txt" -ge -dic 9001 -dac 1001
-otext "/HVInStream/output/Tutorial837IA_textout.txt"
-tpl "/HVInStream/DemoData/RGtemplate837I.txt" -te 0 -s -y
```

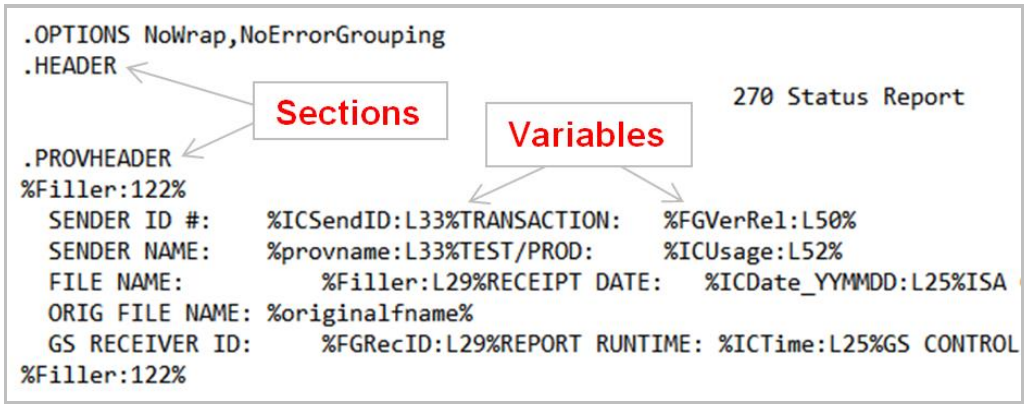
Report Templates

A report template is a text file that contains formatting information used by Response Generator to create a custom report.

Templates are made up of *sections* which contain *variables*.

- **Sections** provide structure for the report by grouping types of information such as provider information, receiver information, and so forth.
- **Variables** provide data by importing specific information into the report.

As shown here, template sections are preceded with a dot (.) (e.g, **.**PROVHEADER) and variables are preceded with a % sign (e.g., **%**ICUsage).



Templates also contain user-specified formatting options (.OPTIONS) and user-specified literal text (e.g., 270 Status Report).

Sample Templates

For sample templates, see the files that start with **RGtemplate** in Instream's DemoData directory. They are used by scripts with filenames containing **textrpt** (example: V_RG_837P_4010_**textrpt**_1).

See the **Instream's DemoData Directory** section of **Demo_Index.pdf** for a list and description.

Template Sections

General information about template sections:

- All template sections are optional.
- You can use a header without its trailer and vice versa.

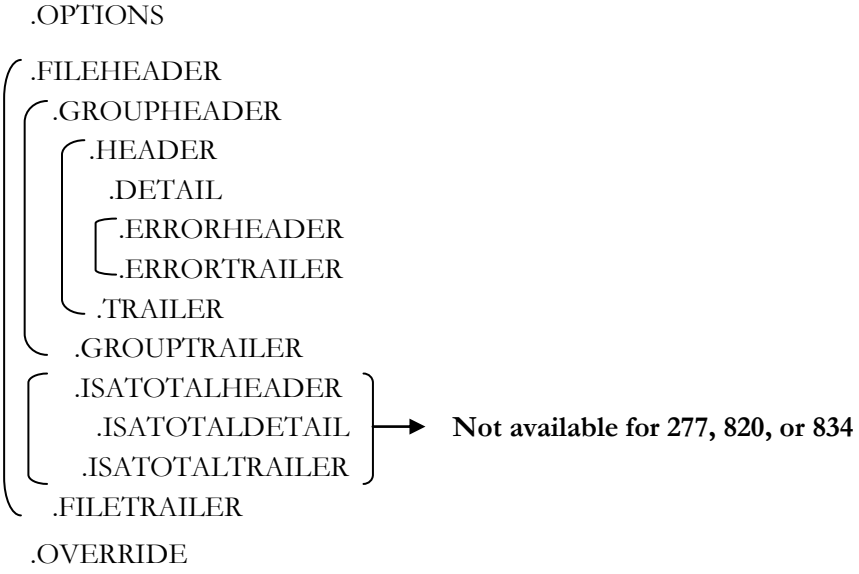
  Example: You can have .PROVHEADER without .PROVTRAILER.
- Template header and trailer lines are not case sensitive.
- When a template section is encountered, its variables are cleared and reloaded.

  Example: If you use a .SOURCEHEADER section, values that were set for the first provider are cleared when the second provider is encountered.
- Values set in a higher section can be used in lower sections, but not vice versa.

  Example: the variable %PayerName% gets set/reset at each 2000A/.SOURCEHEADER section. If you use it at the .HEADER section, it will be blank. You will only get output from %PayerName% at or below the .SOURCEHEADER section.

All Templates

All templates can have these sections:



OPTIONS Section

This section is optional and appears once at the beginning of the output file. It consists of one of these:

`.OPTIONS NoWrap`	Truncate information rather than wrap it to a new line.
`.OPTIONS RepeatText`	Repeats constant information on each error line.

`.OPTIONS NoWrap,RepeatText`

FILEHEADER and FILETRAILER Sections

In the output file, the FILEHEADER section appears once per ISA and begins with a line containing only:

```
.FILEHEADER
```

Put information here if you want it to appear once for each file.

The FILETRAILER section appears once per ISA and begins with a line containing only:

```
. FILETRAILER
```

Put information here if you want it to appear once for each file, after the file's information has been processed.

GROUPHEADER and GROUPTRAILER Sections

In the output file, the GROUPHEADER section appears once per GS and begins with a line containing only:

```
. GROUPHEADER
```

Put information here if you want it to appear once for each group.

The GROUPTRAILER section appears once per GS and begins with a line containing only:

```
. GROUPTRAILER
```

Put information here if you want it to appear once for each group, after the group's information has been processed.

HEADER and Trailer Sections

The Header appears once for each GS in the input file. If an OPTIONS section is included, the Header section follows it. It begins with a line containing only:

```
.HEADER
```

Put information here if you want it to appear once for each functional group. Examples include information from the ISA.

The TRAILER section is optional and appears once at the end of the output file, and begins with a line containing only:

```
.TRAILER
```

DETAIL Section

In the output file, the DETAIL section appears for each claim and begins with a line containing only:

```
.DETAIL
```

Put information here if you want it to appear once for each claim. Examples include claim number and amount.

ERRORHEADER and ERRORTRAILER Sections

In the output file, the ERRORHEADER section appears for each DETAIL section and begins with a line containing only:

```
.ERRORHEADER
```

The ERRORTRAILER section appears once for each DETAIL section and begins with a line containing only:

```
.ERRORTRAILER
```

ISATOTALHEADER and ISATOTALTRAILER Sections

Note: This section is not available for 277, 820, or 834 reports.

In the output file, the ISATOTALHEADER section appears once at the end of the output file, and begins with a line containing only:

```
. ISATOTALHEADER
```

Put information here to provide ISA summary information for the file. Examples include totals for data such as number of claims and dollar value.

The ISATOTALTRAILER section appears once at the end of the output file, and begins with a line containing only:

```
. ISATOTALTRAILER
```

Put information here if you want it to appear once to provide an end for ISA summary information for the file.

OVERRIDE Section

The OVERRIDE section is optional and can appear anywhere in the template. It allows you to override the default display information for a particular error (including its number and associated data) with information of your choosing. It consists of a line containing the following:

```
.OVERRIDE errno|loop|segelmdata|error_ref|error_message|error_data
```

Where:

errno	The error number to be overridden. This field must be populated.
|	Separator. This symbol must be used between line values.
loop	Alternate location information to be provided instead of the default (Loop ID).
segelmdata	Alternate segment/element error data to be provided instead of the default.
error_ref	User-defined field intended to provide additional reference information. There is no default text for this field.
error_message	Alternate error message text to be provided instead of the default.
error_data	Alternate error data to be provided instead of the default (%ErrData%).

All fields are required, although only errno must be populated. If a value is left empty, the default value is used. (This excludes errno, which can't be left empty.)

Note that there is a difference between leaving a field empty and entering a blank (or space) in the field. An empty field will display the default system value. A field with a space will display "" (a blank).

You can use a custom report variable in any field except errno. However, if the variable's value is empty, the original value will be replaced with "" (a blank).

Examples:

For error 23456, override the default loop location information that would typically be displayed (such as "2000B") and show the text "Provider" instead. Allow all other default information for the error to display:

```
.OVERRIDE 23456|Provider|||||
```

For error 60001, override the default loop location information and show the text "2010BA" instead. Use the default information for the segment/element error data. Use "NM109" as additional error reference information. Display the default error message text and, finally, display the contents of the variable %SubCustomRecord1% as additional error data:

```
.OVERRIDE 60001|2010BA||NM109||%SubCustomRecord1%
```

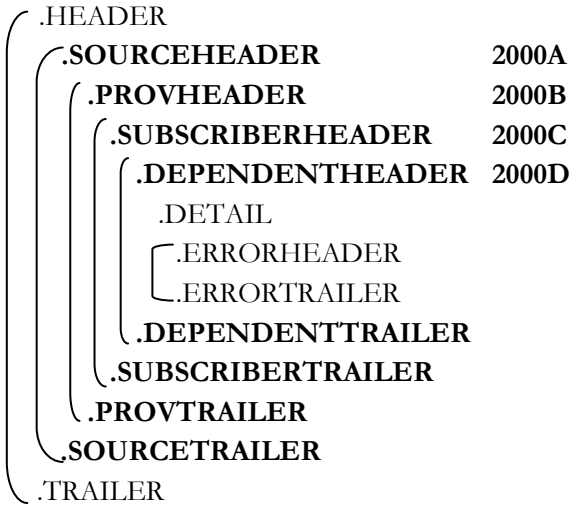
For error 60002, override the loop location information and show the text "2000B" instead. Display a blank for the segment/element error data. Use "SBR09" as additional error reference information. Display a blank instead of error message text and, finally, display the contents of the variable %SubCustomRecord0% as additional error data:

```
.OVERRIDE 60002|2000B| |SBR09| |%SubCustomRecord0%
```

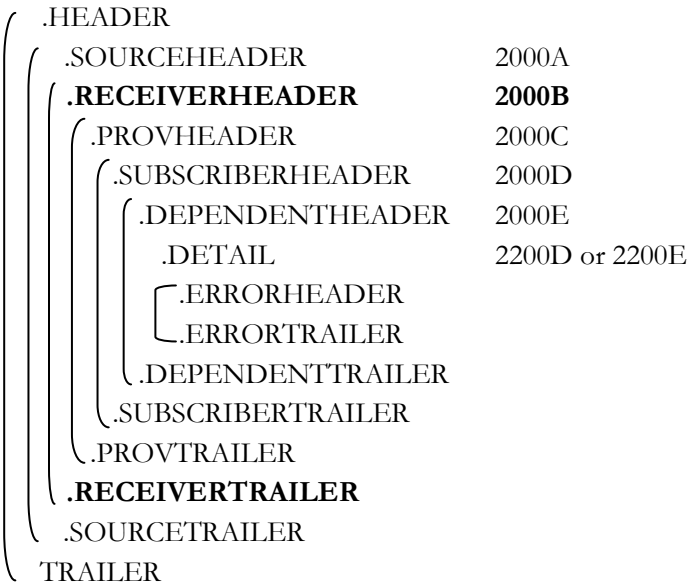
(*note space*) (*note space*)

270, 276, and 277 Templates

270, 276, and 277 templates can have additional sections for source, provider, subscriber, and dependent:



Additionally, 276 and 277 templates can have sections for receiver information:



SOURCEHEADER and SOURCETRAILER Sections

In the output file, the SOURCEHEADER section appears once for each 2000A loop in 270, 276, and 277 transactions. It begins with a line containing only:

```
.SOURCEHEADER
```

Put information here if you want it to appear once for each information source.

The SOURCETRAILER section appears once for each information source in 270, 276, and 277 transactions. It begins with a line containing only:

```
.SOURCETRAILER
```

Put information here if you want it to appear once for each information receiver, after the source's information has been processed.

RECEIVERHEADER and RECEIVERTRAILER Sections

In the output file, the RECEIVERHEADER section appears once for each 2000B loop in 276 and 277 transactions. It begins with a line containing only:

```
.RECEIVERHEADER
```

The RECEIVERTRAILER section appears once for each receiver in 276 and 277 transactions. It begins with a line containing only:

```
.RECEIVERTRAILER
```

Put information here if you want it to appear once for each information receiver, after the receiver's information has been processed.

PROVHEADER and PROVTRAILER Sections

In the output file, the PROVHEADER section appears once for each provider in 270, 276, and 277 transactions. It begins with a line containing only:

```
.PROVHEADER
```

Put information here if you want it to appear once for each provider. Examples include provider and patient information. Note that, for these transactions, "Provider" refers to information in the 2000C.

Put information here if you want it to appear once for each information receiver.

The PROVTRAILER section appears once for each provider in 270, 276, and 277 transactions. It begins with a line containing only:

```
.PROVTRAILER
```

Put information here if you want it to appear once for each provider, after the provider information has been processed. Examples include claim totals and counts by provider.

SUBSCRIBERHEADER and SUBSCRIBERTRAILER Sections

In the output file, the SUBSCRIBERHEADER section appears once for each subscriber in 270, 276, and 277 transactions. It begins with a line containing only:

```
.SUBSCRIBERHEADER
```

Put information here if you want it to appear once for each subscriber.

The SUBSCRIBERTRAILER section appears once for each subscriber in 270, 276, and 277 transactions. It begins with a line containing only:

```
.SUBSCRIBERTRAILER
```

Put information here if you want it to appear once for each subscriber, after the subscriber's information has been processed.

DEPENDENTHEADER and DEPENDENTTRAILER Sections

In the output file, the DEPENDENTHEADER section appears once for each dependent in 270, 276, and 277 transactions. It begins with a line containing only:

```
.DEPENDENTHEADER
```

Put information here if you want it to appear once for each dependent.

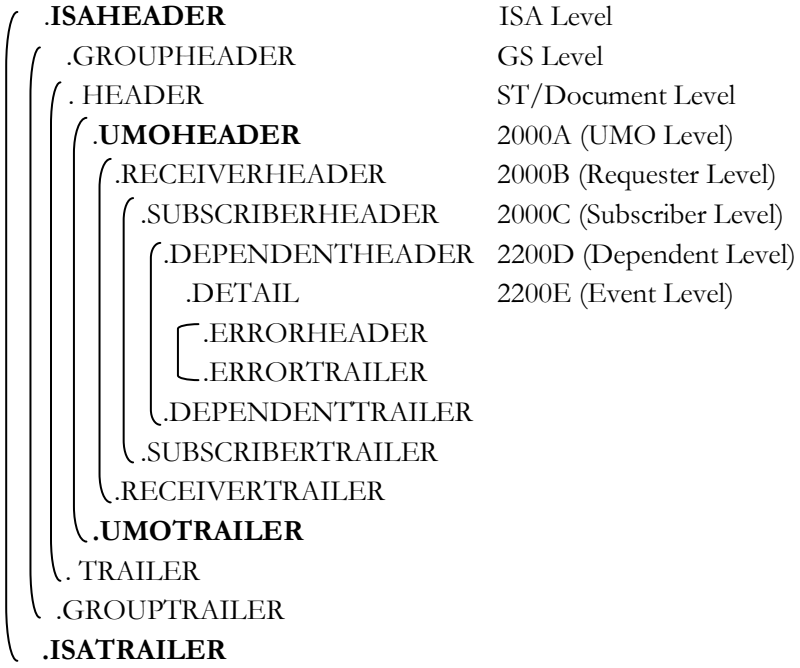
The DEPENDENTTRAILER section appears once for each dependent in 270, 276, and 277 transactions. It begins with a line containing only:

```
. DEPENDENTTRAILER
```

Put information here if you want it to appear once for each subscriber, after the dependent's information has been processed.

278 Templates

278 templates can have additional sections for interchange and utilization management organization information:



ISAHEADER and ISATRAILER Sections

In the output file, the ISAHEADER section appears once for each ISA Level loop in 278 transactions. It begins with a line containing only:

```
.ISAHEADER
```

Put information here if you want it to appear once for each information source.

The ISATRAILER section appears once for each ISA Level loop in 278 transactions. It begins with a line containing only:

```
.ISATRAILER
```

Put information here if you want it to appear once for each information receiver, after the receiver's information has been processed.

UMOHEADER and UMOTRAILER Sections

In the output file, the UMOHEADER section appears once for each 2000A UMO level loop in 278 transactions. It begins with a line containing only:

```
.UMOHEADER
```

Put information here if you want it to appear once for each information source.

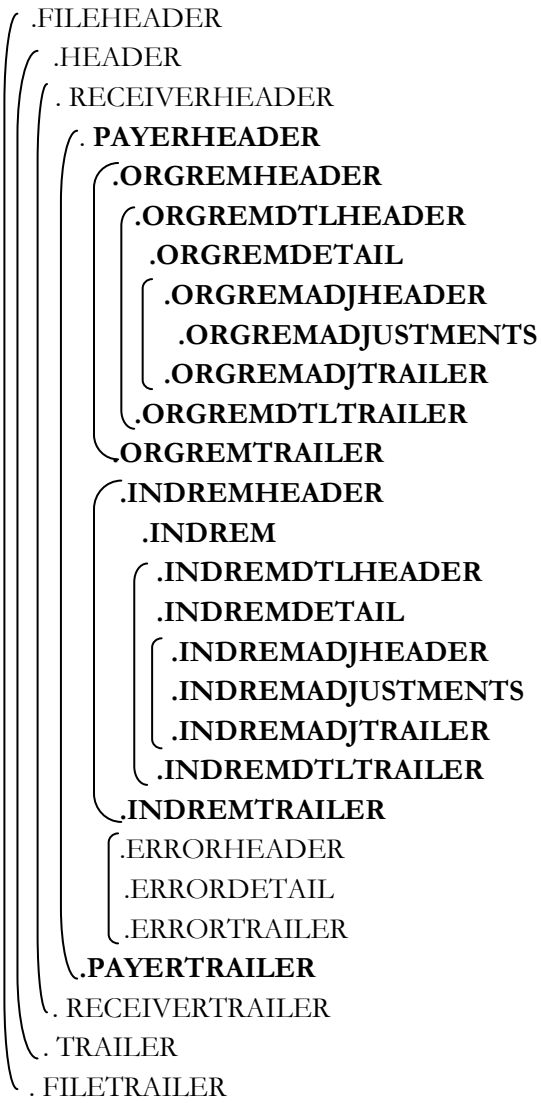
The UMOTRAILER section appears once for each 2000A UMO level loop in 278 transactions. It begins with a line containing only:

```
.UMOTRAILER
```

Put information here if you want it to appear once for each information receiver, after the receiver's information has been processed.

820 Templates

820 templates can have additional sections for payers, including information pertaining to remittance:



PAYERHEADER and PAYERTRAILER Sections

In the output file, the PAYERHEADER section appears once per transaction set. It begins with a line containing only:

```
. PAYERHEADER
```

Put information here if you want it to appear once for each transaction set. Examples include provider reference ID, fiscal period date, and adjustment reason codes.

The PAYERTRAIL section appears once per transaction set. It begins with a line containing only:

. PAYERTRAIL

Put information here if you want it to appear once per transaction set, after the provider level adjustment information has been processed.

ORGREMHEADER and ORGREMTRAILER Sections

In the output file, the ORGREMHEADER section appears once per transaction set, and begins with a line containing only:

. ORGREMHEADER

Put information here to provide information about organization remittance.

The ORGREMTRAILER section appears once per transaction set, and begins with a line containing only:

. ORGREMTRAILER

Put information here if you want it to appear once per transaction set, after the organization remittance information has been processed.

ORGREMDTLHEADER and ORGREMDTLTRAILER Sections

In the output file, the ORGREMDTLHEADER section appears once per transaction set, and begins with a line containing only:

.ORGREMDTLHEADER

Put information here to provide information about organization remittance details.

The ORGREMDTLTRAILER section appears once per transaction set, and begins with a line containing only:

. ORGREMDTLTRAILER

Put information here if you want it to appear once per transaction set, after the organization remittance detail information has been processed.

ORGREMDETAIL Section

In the output file, the ORGREMDETAIL section appears once per transaction set, and begins with a line containing only:

. ORGREMDETAIL

This section provides the organization remittance detail.

ORGREMADJHEADER and ORGREMADJTRAILER Sections

In the output file, the ORGREMADJHEADER section appears once per transaction set, and begins with a line containing only:

. ORGREMADJHEADER

Put information here to provide information about organization remittance adjustments.

The ORGREMADJTRAILER section appears once per transaction set, and begins with a line containing only:

. ORGREMADJTRAILER

Put information here if you want it to appear once per transaction set, after the organization remittance adjustment information has been processed.

INDREMHEADER and INDREMTRAILER Sections

In the output file, the INDREMHEADER section appears once per transaction set, and begins with a line containing only:

. INDREMHEADER

Put information here to provide information about individual remittance.

The INDREMTRAILER section appears once per transaction set, and begins with a line containing only:

. INDREMTRAILER

Put information here if you want it to appear once per transaction set, after the individual remittance information has been processed.

INDREM Section

In the output file, the INDREM section appears once per transaction set, and begins with a line containing only:

. INDREM

Put information here to provide information individual remittance.

INDREMDTLHEADER and INDREMDTLTRAILER Sections

In the output file, the INDREMDTLHEADER section appears once per transaction set, and begins with a line containing only:

. INDREMDTLHEADER

Put information here to provide information about individual remittance details.

The INDREMDTLTRAILER section appears once per transaction set, and begins with a line containing only:

. INDREMDTLTRAILER

Put information here if you want it to appear once per transaction set, after the individual remittance detail information has been processed.

INDREMDETAIL Section

In the output file, the INDREMDETAIL section appears once per transaction set, and begins with a line containing only:

. INDREMDETAIL

Put information here to provide information about individual remittance detail.

INDREMADJHEADER and INDREMADJTRAILER Section

In the output file, the INDREMADJHEADER section appears once per transaction set, and begins with a line containing only:

. INDREMADJHEADER

Put information here if you want it to appear once per transaction set, to provide information about individual remittance adjustments.

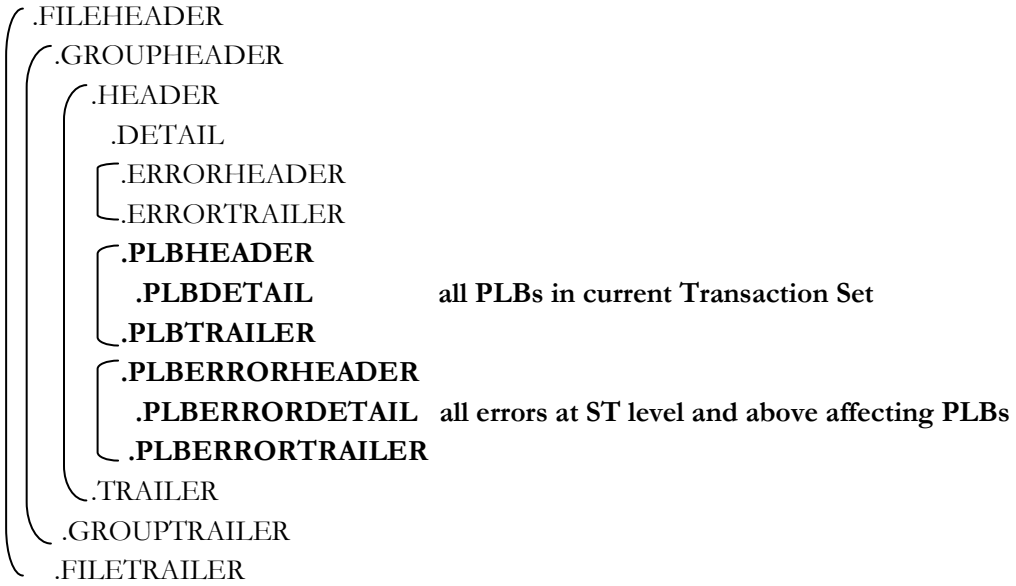
The INDREMADJTRAILER section appears once per transaction set, and begins with a line containing only:

. INDREMADJTRAILER

Put information here if you want it to appear once per transaction set, after the individual remittance adjustment information has been processed.

835 Templates

835 templates can have additional sections about provider level adjustments and errors affecting them.



PLBHEADER and PLBTRAILER Sections

In the output file, the PLBHEADER section appears once per transaction set. It begins with a line containing only:

. PLBHEADER

Put information here if you want it to appear once for each transaction set to provide information about provider level adjustments.

The PLBTRAILER section appears once per transaction set. It begins with a line containing only:

. PLBTRAILER

Put information here if you want it to appear once per transaction set, after the provider level adjustment information has been processed.

PLBERRORHEADER and PLBERRORTRAILER Sections

In the output file, the PLBERRORHEADER section appears once per transaction set, and begins with a line containing only:

. PLBERRORHEADER

Put information here to provide information about all errors at the ST level and above affecting provider level adjustments.

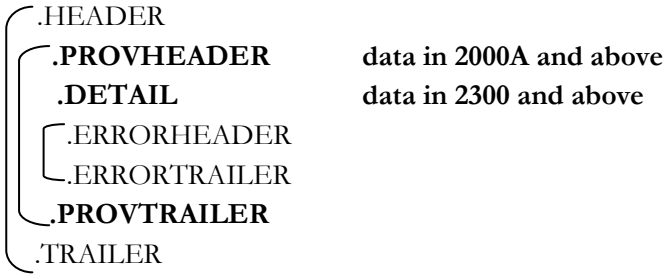
The PLBERRORTRAILER section appears once per transaction set, and begins with a line containing only:

```
. PLBERRORTRAILER
```

Put information here if you want it to appear once per transaction set, after the provider level adjustment error information has been processed.

837 Templates

837 templates can have additional sections for provider and interchange total information.



PROVHEADER and PROVTRAILER Sections

In the output file, the PROVHEADER section appears once for each provider in 837 transactions. It begins with a line containing only:

```
.PROVHEADER
```

Put information here if you want it to appear once for each provider. Examples include provider and patient information. Note that, for this transaction, "Provider" refers to information in the 2000A and above.

The PROVTRAILER section appears once for each provider in 837 transactions. It begins with a line containing only:

```
.PROVTRAILER
```

Put information here if you want it to appear once for each provider, after the provider information has been processed. Examples include claim totals and counts by provider.

Literals and Variables

The template has both literal text and variables. In the example above, everything not surrounded by % is a literal. Everything surrounded by % is a variable that will be replaced when the custom report is generated. For example:

ClaimNumber: %CLAIMID%
TotalAmt: %CLAIMAMT%

In the report, these lines might generate something like this for each claim:

ClaimNumber: 10
TotalAmt: $1000.00

If you have tables in your HTML template, put entire <TR>...</TR> onto one line of the template file. Add the statement '.OPTIONS RepeatText' at the beginning of the template file, as shown in the template file RGtemplate837I_h.txt in Instream's DemoData directory.

Variables are pre-defined by Response Generator (see page 89) and are not case sensitive: **CLAIMAMT** is the same as **ClaimAmt**.

In the template, variables may be accompanied by column width and justification information - especially important when producing a column report.

Format	Result in Report	Example
%*varname*%	*varname* is the name of the variable to be output (see page 89). The report displays the value, taking up as many spaces as needed.	**% PatName%** displays the patient last, first, and middle names.
%*varname*:w% ↑ (*note colon*)	***w*** is the field width.	**%PatName:25%** displays the patient last, first, and middle names in a column that is 25 characters wide.. If the name has less than 25 characters, blanks are added. If the name has more than 25 characters, it wraps. Note the colon that separates the variable name from the width.

Format	Result in Report	Example
%*varname*:*jw*%	***j*** is an optional field justification of either L or R. You must use a width if you use justification.	**%PatName:R25%** right-justifies up to 25 characters of the patient last, first, and middle names. For numeric fields, you can right-justify and fill with leading zeros. To do this, put a 0 right after the R. Example: R010 defines a right-justified field of 10 characters, with leading zeros. If the value is 23, then this field will show 0000000023 in the custom report. Default is left justification. The justification can come before or after the width: R25 and 25R have the same result.

Template Variables

Template variables are not case-sensitive.

General Template Variables

These variables can be used in any template.

ClaimCnt_byGS	Number of claims encountered so far in this functional group set. Resets at the beginning of each functional group set.
ClaimCntGood_byGS	Number of good claims encountered so far in this functional group set. Resets at the beginning of each functional group set.
ClaimCntBad_byGS	Number of bad claims encountered so far in this functional group set. Resets at the beginning of each functional group set.
ClaimCustomRecord [0-9]	Claim Custom Record as determined by the corresponding ZREC at the specified level. This variable can be used multiple times using 0-9 (for example, ClaimCustomRecord0, ClaimCustomRecord1). This variable corresponds to ZZCR*x*, where *x* is also 0-9 and is a 1:1 match. See **TIB_fsp-instream_*<n.n>*_usersguide.pdf**, the section "Z Custom Data Record". Example: ZZCR2 at the 2300 loop level corresponds to %ClaimCustomRecord2%.
ClaimPercentGood_byGS	Percent of claims that are good in this functional group set; includes one decimal place. Example: `60.0`.
ClaimPercentBad_byGS	Percent of claims that are bad in this functional group set; includes one decimal place. Example: `40.0`.
ContactPhone	Contact phone number as passed through a ZZWPC custom record.
Date	Current date in format MM/DD/YYYY.
Date_YYMMDD	Current date in format YYMMDD.
Date_MMDDYYYY	Current date in format MM-DD-YYYY.
DATE_YYYY_MM_DD	Current date in format YYYY-MM-DD.

DepCustomRecord[0-9]	Dependent Custom Record as determined by the corresponding ZREC at the specified level. This variable can be used multiple times using 0-9 (for example, DepCustomRecord0, DepCustomRecord1). This variable corresponds to ZZCR*x*, where *x* is also 0-9 where *x* is also 0-9 and is a 1:1 match. See **TIB_fsp-instream_ <n.n>_usersguide.pdf**, the section "Z Custom Data Record". Example: ZZCR2 at the 2300 loop level corresponds to %DepCustomRecord2%.
EdiFileName	Name of the EDI file. STRT record must be in the results file. Example output: `837P_098Dep_11.txt`
EdiFilePath	Path to the EDI file. STRT record must be in the results file. Example output: `C:\Systest\Data\`
EDILine	Line number in EDI file.
Filler:*n*	Insert *n* spaces. Example: `Filler:5`
LineCount	Number of lines so far in the report.
NEWLINE	Line break.
OriginalFname	Original Filename as passed through a ZZWRO custom record.
PayerCustomRecord[0-9]	Payer Custom Record as determined by the corresponding ZREC at the specified level. This variable can be used multiple times using 0-9 (for example, PayerCustomRecord0, PayerCustomRecord1). This variable corresponds to ZZCR*x*, where *x* is also 0-9 where *x* is also 0-9 and is a 1:1 match. See **TIB_fsp-instream_ <n.n>_usersguide.pdf,** the section "Z Custom Data Record". Example: ZZCR2 at the 2300 loop level corresponds to %PayerCustomRecord2%.
PayerLineCount	Placed in one of the types of trailer records, this shows the number of lines displayed for the current payer up to this point.

ProvCustomRecord[0-9]	Provider Custom Record as determined by the corresponding ZREC at the specified level. This variable can be used multiple times using 0-9 (for example, ProvCustomRecord0, ProvCustomRecord1). This variable corresponds to ZZCR*x*, where *x* is also 0-9 where *x* is also 0-9 and is a 1:1 match. See **TIB_fsp-instream_ <n.n>_usersguide.pdf**, the section "Z Custom Data Record". Example: ZZCR2 at the 2300 loop level corresponds to %ProvCustomRecord2%.
SubCustomRecord[0-9]	Submitter Custom Record as determined by the corresponding ZREC at the specified level. This variable can be used multiple times using 0-9 (for example, SubCustomRecord0, SubCustomRecord1). This variable corresponds to ZZCR*x*, where *x* is also 0-9 where *x* is also 0-9 and is a 1:1 match. See **TIB_fsp-instream_ <n.n>_usersguide.pdf**, the section "Z Custom Data Record". Example: ZZCR2 at the 2300 loop level corresponds to %SubCustomRecord2%.
Time	Current Time in 24hr format HH:MM.
TPCustomRecord[0-9]	Trading Partner Custom Record as determined by the corresponding ZREC at the specified level. This variable can be used multiple times using 0-9 (for example, TPCustomRecord0, TPCustomRecord1). This variable corresponds to ZZCR*x*, where *x* is also 0-9 where *x* is also 0-9 and is a 1:1 match. See **TIB_fsp-instream_ <n.n>_usersguide.pdf**, the section "Z Custom Data Record". Example: ZZCR2 at the 2300 loop level corresponds to %TPCustomRecord2%.
UserGSControlNum	User-specified GS Control Number, as set via the **-dcg** command.
WrappedText	Display the text that was truncated from the end of a variable that was specified with a fixed width. For example, using %WrappedText% after the variable %ErrMsg:128% results in the display of all text from positions 129 on.

Error/Message Template Variables

Error variables really represent lists of values, since a claim may have more than one. If so, subsequent occurrences appear on another line.

These variables can be used in any template.

ErrData	Data causing error.
ErrElmPos	Position of the erroneous element with the segment.
ErrID	Error number.
ErrLoop	ID of the loop with the error.
ErrMsg	Error text – TIBCO Foresight flavor.
ErrMsg_ADA	Error text – ADA flavor.
ErrMsg_COBA	Error text – COBA flavor. For errors not defined in ErrMsgTrans.txt, the report will have an error number of 009999. This will generally happen with your custom error messages that you set up with your own business rules.
ErrMsg_Foresight	Error text – TIBCO Foresight flavor.
ErrMsg_HCFA	Error text – HCFA flavor.
ErrMsg_NonTech	Error text – Non-technical flavor if an ELOC record is associated with the DTL record for an error in the 10000-29999 range. For a demo, see **V_RG_837P_4010_textrpt_ELOC** in Instream's Scripts directory.
ErrMsg_NSF	Error text – NSF flavor.
ErrMsg_UB92	Error text – UB92 flavor.
ErrRefDes	ErrSegData+ErrElemPos+ErrSubElmPos.
ErrSeg	ID of the segment with the error.
ErrSegData	Entire segment causing or containing the error (if present in data).
ErrSev	Error severity number.
ErrSubElmPos	Position of the subelement within the composite.
ErrType	Error type number.

Envelope Template Variables

These variables can be used in any template.

Interchange

ICSendID	Sender ID (ISA-06).
ICSendIDQual	Sender ID qualifier (ISA-05).
ICRecID	Receiver ID (ISA-08).
ICRecIDQual	Received ID qualifier (ISA-07).
ICCtlNoStart	Control number passed with the Response Generator command line parameter -dic.
ICDate	Date (ISA-09) in format MM/DD/YYYY.
ICDate_MMDDYYYY	Date (ISA09) in format MM-DD-YYYY.
ICDate_YYMMDD	Date (ISA-09) in format YYMMDD.
ICDATE_YYYY_MM_DD	Date (ISA-09) in format YYYY-MM-DD.
ICTime	Time (ISA-10) in format HH:MM.
ICCtlNo	Interchange control number (ISA-13).
ICDate_YYYYMMDD	ISA date in YYYYMMDD format.
ICUsage	Usage (ISA-15)

Functional Group

FGSendID	Sender ID (GS-02).
FGRecID	Receiver ID (GS-03).
FGDate	Date (GS-04) in format MM/DD/YYYY.
FGDate_YYMMDD	Date (GS-04) in format YYMMDD.
FGDate_MMDDYYYY	Date (GS-04) in format in MM-DD-YYYY.
FGTime	Time (GS-05) in format HH:MM.
FGCtlNo	Control number (GS-06).
FGVerRel	Version/Release/Industry Identifier Code (GS-08).
FGType	HIPAA type of the source document (Example: 837 D 5010).
COBA_837Type	Returns “Part A” or “Part B”.

Transaction Set

SetCtlNo	Set control number (ST-02).

ISATOTAL Summary Variables

These variables can be used in any template except 277, 820, and 834.

%FileClaimCntTotals% - total number of claims in that ISA.

%FileClaimCntGoodTotals% - total number of good claims in that ISA.

%FileClaimCntBadTotals% - total number of good claims in that ISA.

%FileClaimAmtTotals% - total dollar value of claims in that ISA.

%FileClaimAmtGoodTotals% - total dollar value of good claims in that ISA.

%FileClaimAmtBadTotals% - total dollar value of bad claims in that ISA.

%FileClaimTotalIndex% - the claim being reported on.

270-Specific Report Variables

These variables can be used only for a 270 custom report template.

Note: Payer and Provider are the generic values. If not listed, use the generic Subscriber and Dependent codes.

Detail Section

TSRefID	Transaction set reference ID (BHT03).
TSTypeCpde	Transaction type code (BHT06).

2000C

SubTraceTypeCode	Type of transaction being referenced (TRN01).
SubTraceID	Transaction trace number (TRN02).
SubOrigTraceID	Originating company identifier (TRN03).
SubOrigTraceSuppID	Originating company supplemental identifier (TRN04).

2110C

SubServiceTypeCode	Service type classification code (EQ01).
SubServiceIDQual	Service Id qualifier (EQ0201).
SubServiceID	Service Id (EQ0202).
SubProcedureMod1	Service/procedure modifier code #1 (EQ0203).
SubProcedureMod1	Service/procedure modifier code #2 (EQ0204).
SubProcedureMod1	Service/procedure modifier code #3 (EQ0205).
SubProcedureMod1	Service/procedure modifier code #4 (EQ0206).
SubCoverageLevelCode	Coverage level provided (EQ03).

2000D

DepTraceTypeCode	Type of transaction being referenced (TRN01).
DepTraceID	Transaction trace number (TRN02).
DepOrigTraceID	Originating company identifier (TRN03).
DepOrigTraceSuppID	Originating company supplemental identifier (TRN04).

2110D

DepServiceTypeCode	Service type classification code (EQ01).
DepServiceIDQual	Service Id qualifier (EQ0201).
DepServiceID	Service Id (EQ0202).
DepProcedureMod1	Service/procedure modifier code #1 (EQ0203).
DepProcedureMod1	Service/procedure modifier code #2 (EQ0204).
DepProcedureMod1	Service/procedure modifier code #3 (EQ0205).
DepProcedureMod1	Service/procedure modifier code #4 (EQ0206).
DepCoverageLevelCode	Coverage level provided (2110 EQ03).

276-Specific Report Variables

These variables can be used only for a 276 custom report template

Detail Section

2200D

SubPayerClaimControlNumber	Payer Claim Control Number (REF*1K).
SubSVCDate	Service date (DTP03).

2200E

DepPayerClaimControlNumber	Payer Claim Control Number (REF*1K).
DepSVCDate	Service date (DTP03).

2210D/E

SVCMonAmt	SVC monetary amount (SVC02).
SVCNUBCID	SVC NUBC Id (SVC04).

277CA, 277U, and 277X228-Specific Report Variables

These variables can be used only for a 277CA, 277U, or 277X228 custom report template.

Note: The 277U custom report is available for 4010 only.

HEADER Section

ProcessAllFor277 — If the input file contained multiple errors or service lines per claim, only the last error or service line information appears in the custom report. ProcessAllFor277 causes all errors and service line information to appear:

```
.HEADER
.PROCESSALLFOR277

Health Care Claim Status Response
=================================


.SOURCEHEADER

File Creation Date = %ICDate:25%
Transaction Set Control Number = %SetCtlNo:10%
```

SOURCEHEADER Section

2100A

PayerName	Payer last name (NM103).
PayerIDQual	Payer ID qualifier (NM108).
PayerID	Payer ID (NM109).

RECEIVERHEADER Section

2100B

ReceiverLName	Information receiver last name (NM103).
ReceiverFName	Information receiver first name (NM104).
ReceiverMName	Information receiver middle name (NM105).
ReceiverName	Information receiver full name (NM103, NM104, NM105).
ReceiverIDQual	Information receiver ID qualifier (NM108).
ReceiverID	Information receiver ID (NM109).

PROVHEADER Section

2100C

ProvIDCode	Provider entity code (NM101).
ProvLName	Provider last name (NM103).

ProvFName	Provider first name (NM104).
ProvMName	Provider middle name (NM105).
ProvName	Provider full name (NM103, NM104, NM105).
ProvIDQual	Provider ID qualifier (NM108).
ProvID	Provider entity code (NM109).

SUBSCRIBERHEADER Section

2000D

SubBirthDate	Subscriber birth date (DMG02).
SubGender	Subscriber gender (DMG03).

2100D

PatLName	Patient last name (NM103 (NM101=QC)).
PatFName	Patient first name (NM104 (NM101=QC)).
PatMName	Patient middle name (NM105 (NM101=QC)).
PatName	(NM103, NM104, NM105 (NM101=QC)).
PatIDQual	Patient ID code qualifier (NM108 (NM101=QC)).
PatID	Patient ID (NM109 (NM101=QC)).
SubIDCode	Subscriber qualifier (NM101 (NM101=IL)).
SubLName	Subscriber last name (NM103 (NM101=IL)).
SubFName	Subscriber first name (NM104 (NM101=IL)).
SubMName	Subscriber middle name (NM105 (NM101=IL)).
SubName	Subscriber full name (NM103, NM104, NM105 (NM101=IL)).
SubIDQual	Subscriber ID code qualifier (NM108 (NM101=IL)).
SubID	Subscriber ID (NM109 (NM101=IL)).

DEPENDENTHEADER Section

2000E

PatBirthDate	Patient birth date (DMG02).
PatGender	Patient gender (DMG03).

2100E

PatLName	Patient last name (NM103).
PatFName	Patient first name (NM104).

PatMName	Patient middle name (NM105).
PatName	Patient full name (NM103, NM104, NM105).
PatIDQual	Patient ID code qualifier (NM108).
PatID	Patient ID (NM109).

DETAIL Section

2200D

BillTypeQual	Bill type qualifier (REF01 (REF01=BLT)).
BillTypeID	Bill type ID (REF02 (REF01=BLT)).
ClaimID	Claim submitter trace number (TRN02).
DateOfService	Claim service period (DTP03).
PayerClaimQual	Payer claim ID qualifier (REF01 (REF01=1K)).
PayerClaimID	Payer claim control number (REF02 (REF01=1K)).
ClaimCategoryCode	Claim level status category code (STC01.01).
ClaimStatusCode	Claim level status code (STC01.02).
PlaceOfService	Place of service code (STC01.03).
PmtAmt	Original submitted charges (STC04).
PmtMethod	Claim payment method code (STC07).
PmtDate	Payment issue date (STC08).
CheckNum	Check number (STC09).

2220D

SvcIDQual	Medical procedure ID qualifier (SVC01.01).
SvcID	Medical procedure ID (SVC01.02).
SvcMod1	Medical procedure modifier code (SVC01.03).
SvcMod2	Medical procedure modifier code (SVC01.04).
SvcMod3	Medical procedure modifier code (SVC01.05).
SvcMod4	Medical procedure modifier code (SVC01.06).
SvcChgAmt	Submitted service charge (SVC02).
SvcQty	Original submitted units of service (SVC07).

2200E

BillTypeQual	Bill type ID qualifier (REF01 (REF01=BLT)).
BillTypeID	Bill type ID (REF02 (REF01=BLT)).
DateOfService	Claim service date (DTP03).
PayerClaimID	Payer claim ID qualifier (REF01 (REF01=1K)).
PayerClaimQual	Payer claim control number (REF02 (REF01=1K)).
ClaimID	Claim submitter trace number (TRN02).
ClaimCategoryCode	Claim industry category (STC01.01).
ClaimStatusCode	Claim status code (STC01.02).
PlaceOfService	Service entity (STC01.03).
PmtAmt	Amount of original submitted charges (STC04).
PmtMethod	Payment method (STC07).
PmtDate	Check issue date (STC08).
CheckNum	Check identification number (STC09).

2220E

SvcIDQual	Procedure code qualifier (SVC01.01).
SvcID	Procedure code (SVC01.02).
SvcMod1	Procedure modifier (SVC01.03).
SvcMod2	Procedure modifier (SVC01.04).
SvcMod3	Procedure modifier (SVC01.05).
SvcMod4	Procedure modifier (SVC01.06).
SvcChgAmt	Submitter service charge (SVC02).
SvcQty	Original submitted units of service (SVC07).

278-Specific Report Variables

(Including 278x215 and 278x216 transactions)

These variables can be used only for a 278 custom report template. This includes

2000A

%PayerName%	Last name or organization name (NM103).
%PayerIDQual%	Identification code qualifier (NM108).
%PayerID%	Identification code (NM109).

2000B

%ReceiverNameL%	Last name or organization name (NM103).
%ReceiverNameF%	First name (NM104).
%ReceiverNameF%	Middle name (NM105).
%ReceiverName%	Combined first name, last name, and middle initial.

2000C

%SubNameL%	Last name or organization name (NM103).
%SubNameF%	First name (NM104).
%SubNameF%	Middle name (NM105).
%SubName%	Combined first name, last name, and middle initial.
%SubTraceTypeCode%	Trace number type (TRN01).
%SubTrace%	Trace number (TRN02).

2000D

%DepNameL%	Last name or organization name (NM103).
%DepNameF%	First name (NM104).
%DepNameF%	Middle name (NM105).
%DepName%	Combined first name, last name, and middle initial
%DepTraceTypeCode%	Trace number type (TRN01).
%DepTrace%	Trace number (TRN02).

2000E

%PatientName%	Last name (If 2000D exists, 2000D NM103 is used, otherwise 2000C NM103 is used.)
%PatientTraceTypeCode%	Trace number type (If 2100D exists, 2100D NM108 is used, otherwise 2100C NM108 is used.)
%PatientTrace%	Trace number (If 2100D exists, 2100D NM109 is used, otherwise 2100C NM109 is used.)
%EventTraceTypeCode%	Trace number type (TRN01).
%EventTrace%	Trace number (TRN02).
%EventDate%	2000E Event Date - DTP*AAH DTP03 (YYYYMMDD).

820x218-Specific Report Variables

These variables can be used only for an 820 custom report template.

HEADER Section

TransCode	Transaction Handling Code (BPR01)
MonetaryAmt	Monetary Amount (BPR02)
PayMethod	Payment Method Code (BPR04)
PaymentDate	Date (BPR16)
ReassociationKey	Reassociation Key (TRN02)

1000A

ReceiverName	Premium Receiver Name (1000A, N102)
ReceiverID	Premium Receiver ID (1000A, N104)

1000B

PayersName	Premium Payer's Name (1000B, N102)
PayersID	Premium Payer's ID (1000B, N104)

2000B

OrgAssignedNumber	Organization Summary Remittance (2000A, ENT01)
OrganizationID	Organization ID (2000A, ENT04)

2300A

OrgRMRID	RMR Organization Reference ID (2300A, RMR02)
OrgRMRAmt	RMR Organization Monetary Amount (2300A, RMR04)

2320A

OrgADXAmt	Organization Monetary Amount (2320A, ADX01)
OrgADXAdjCode	Organization Adjustment Reason Code (2320A, ADX02)

2000B

IndAssignedNumber	Individual Summary Remittance (2000B, ENT01)
IndividualID	Individual ID (2000B, ENT04)

2100B

IndName	Individual Name (2100B, NM103)
IndID	Individual ID (2100B, NM109)

2300B

IndRMRID	RMR Individual Reference ID (2300B, RMR02)
IndRMRAmt	RMR Individual Monetary Amount (2300B, RMR04)

2320B

IndADXAmt	Individual Monetary Amount (2320B, ADX01)
IndADXAdjCode	Individual Adjustment Reason Code (2320B, ADX02)

820x306-Specific Report Variables

These variables can be used only for an 820 custom report template.

HEADER Section

TransCode	Transaction Handling Code (BPR01)
MonetaryAmt	Monetary Amount (BPR02)
PayMethod	Payment Method Code (BPR04)
PaymentDate	Date (BPR16)
ReassociationKey	Reassociation Key (TRN02)

1000A

PayeeName	Payee Name (1000A, N102)
PayeeID	Payee ID (1000A, N104)

1000B

PayersName	Premium Payer's Name (1000B, N102)
PayersID	Premium Payer's ID (1000B, N104)

2000

RemittanceInfo	Remittance Information (2000, ENT01)

2100

IndName	Individual Name (2100, NM103)
IndID	Individual ID (2100, NM109)

2300

IndRMRID	RMR Reference ID (2300, RMR02)
IndRMRAmt	RMR Monetary Amount (2300, RMR04)

834-Specific Variables

These variables can be used only for an 834 custom report template.

HEADER Section

TSPurposeCode	Transaction set purpose code (BGN-01).
CreationDate	Transaction set creation date (BGN-03).
ActionCode	Action code (BGN-08).

1000A

SponsorName	Plan sponsor name (N1-02).
SponsorIDQual	Sponsor identification code qualifier (N1-03).
SponsorID	Sponsor identifier (N1-04).

1000B

PayerName	Payer (insurer) name (N1-02).
PayerIDQual	Payer (insurer) identification code qualifier (N1-03).
PayerID	Payer (insurer) identification code (N1-04).

1000C

TPAName	TPA or broker name (N1-02).
TPAQual	TPA or broker identification code qualifier (N1-03).
TPAID	TPA or broker identification code (N1-04).

SUBSCRIBERHEADER Section

2000

SubDepIndicator	Subscriber or insured indicator (INS-01).
MbrRelCode	Individual relationship code (INS-02).
MbrMaintTypeCode	Maintenance type code (INS-03).
MbrMaintTypeCodeText	Description of the 834 Member MTC (Example: "Change" - Code 001).
MbrBenefitStatus	Benefit status code (INS-05).
SubID	Subscriber identifier (where REF-01 = OF/ REF-02).

MbrPolicyNum	Insured group or policy number (where REF01 = 1L/REF-02).
MbrLevelQual	Applicable dates qualifier (DTP-01).
MbrLevelDate	Status information effective date (DTP-03).

2100A

MbrIDCode	Entity identifier code (NM1-01).
MbrLName	Subscriber last name (NM1-03).
MbrFName	Subscriber first name (NM1-04).
MbrMName	Subscriber middle name (NM1-05).
MbrName (Last, First Middle)	MbrLName, MbrFName MbrMName (Example: JOHNSON, BARBARA T).
MbrIDQual	Identification code qualifier (NM1-08).
MbrID	Subscriber identifier (NM1-09).
MbrBirthDate	Member birth date (DMG-02).
MbrGenderCode	Gender code (DMG-03).

DETAIL Section

2300

CvrgMaintCode	Maintenance type code (HD-01).
CvrgInsurCode	Insurance line code (HD-03).
CvrgLevelQual	Maintenance date/time qualifier (DTP-01).
CvrgLevelDate	Coverage period (DTP-03).
CvrgPolicyNumQual	Coverage policy reference identification qualifier (REF-01).
CvrgPolicyNum	Insured group or policy number (REF-02).

2310

ProvIDCode	Provider entity identifier code (NM1-01).
ProvLName	Provider last name or organization name (NM1-03).
ProvFName	Provider (individual) first name (NM1-04).
ProvMName	Provider (individual) middle name or initial (NM1-05).
ProvName	(Last, First Middle) ProvLName, ProvFName ProvMName (Example: JOHNSON, BARBARA T).
ProvIDQual	Provider identification code qualifier (NM1-08).
ProvID	Provider identifier (NM1-09).

2320

COBResponsibility	Payer responsibility sequence number code (COB-01).
COBPolicyNum	Insured group or policy number (COB-02).
COBBenefitCode	Coordination of benefits code (COB-03).
COBName	Insurer name (N1-02).
COBID	Provider organizational ID (N1-04).
COBLevelQual	Coordination of benefits date/time qualifier (DTP-01).
COBPolicyDate	Coordination of benefits date (DTP-03).

835-Specific Report Variables

These variables can be used only for an 835 custom report template.

HEADER Section

TransHandleCode	Transaction Handling Code (BPR-01).
TotalPmtAmt	Total Payment Amount (BPR-02).
CreditDebitFlag	Credit/Debit Flag Code (BPR-03).
PmtMethod	Payment Method Code (BPR-04).
PmtFormat	Payment Format Code (BPR-05).
OrigDFIQual	Originating DFI Qualifier (BPR-06).
OrigDFIID	Originating DFI ID Number (BPR-07).
OrigAcctQual	Originating Acct Qualifier (BPR-08).
OrigAcctID	Originating Acct Number (BPR-09).
OrigCoID	Originating Company ID (BPR-10).
OrigCoSupplID	Originating Company Suppl Code (BPR-11).
RecvDFIQual	Receiving DFI Qualifier (BPR-12).
RecvDFIID	Receiving DFI ID Number (BPR-13).
RecvAcctQual	Receiving Acct Qualifier (BPR-14).
RecvAcctID	Receiving Acct Number (BPR-15).
PmtDate	Payment Date (BPR-16).
TraceType	Trace Type Code (TRN-01).
TraceID	Ref ID (TRN-02).
OrigCoIDTrn	Originating Company ID (TRN-03).
OrigCoSupplIDTrn	Originating Company Suppl Code (TRN-04).

RecvRefQual	Receiver Reference ID Qual (REF-01).
RecvRefID	Receiver Reference ID (REF-02).

1000A

PayerName	Payer Name (N1-02).
PayerID	Payer ID (N1-04).

1000B

PayeeName	Payee Name (N1-02).
PayeeID	Payee ID (N1-04).
PayeeAddlRefQual	Payee Additional Ref ID Qual (REF-01).
PayeeAddlRefID	Payee Additional Ref ID (REF-02).

1000B (Receiver Name)

ReceiverName	Receiver Name (1000B/NM103)
ReceiverID	Receiver ID (1000B/NM109)
ReceiverIDQual	Receiver ID Qualifier (1000B/NM108)

2000

PmtProvID	Provider ID (TS3-01).
FacType	Facility Type Code (TS3-02).
FiscalEndDate	Fiscal Period End Date (TS3-03).
TotalDRGAmt	Total Diagnosis Related Group (DRG) Amount (TS2-01).
TotalFedAmt	Total Federal Specific Amount (TS2-02).
TotalHospAmt	Total Hospital-specific Amount (TS2-03).
TotalDispAmt	Total Disproportionate Amount (TS2-04).
TotalCapAmt	Total Capital Amount (TS2-05).
TotalEduAmt	Total Indirect Medical Education Amount (TS2-06).
TotalOutlAmt	Total Day Outlier Amount (TS2-08).
TotalDayCount	Total Cost Report Day Count (TS2-12).
TotalMSPPassAmt	Total MSP Pass Through Amount (TS2-15).
TotalPPSFedDRGAmt	Total PPS Capital, Federal-specific portion DRG Amount (TS2-17).
TotalPPSHospDRGAmt	Total PPS Capital, Hospital-specific portion DRG Amount (TS2-18).

TotalPPSDispHospDRGAmt
: Total PPS Disproportionate Share, Hospital DRG Amount (TS2-19).

2010BB (Payer Name)

PayerName2010BB
: Payer Name (2010BB/NM103)

PayerNameID
: PayerNameID (2010BB/NM109)

PayerNameSecondaryIDQual
: Payer Name Secondary ID Qualifier (2010BB/REF02, where REF01=2U, EI, FY, or NF)

2100

PatientControlID
: Patient Control Number (CLP-01).

ClaimStatusCode
: Claim Status Code (CLP-02).

ClaimChgAmt
: Total Claim Charge Amount (CLP-03).

ClaimPmtAmt
: Claim Payment Amount (CLP-04).

ClaimPatRespAmt
: Patient Responsibility Amount (CLP-05).

ClaimFilingCode
: Claim Filing Indicator Code (CLP-06).

ClaimControlID
: Payer Claim Control Number (CLP-07).

ClaimFacType
: Facility Type Code (CLP-08).

ClaimFreq
: Claim Frequency Code (CLP-09).

ClaimDRGCode
: Diagnosis Related Group (DRG) Code (CLP-11).

ClaimDRGWt
: Diagnosis Related Group (DRG) Weight (CLP-12).

ClaimDischFrac
: Discharge Fraction (CLP-13).

2100 CAS

ClaimAdjTotalAmt
: Sum of Adjustment Amounts (CAS-03+CAS-06+CAS-09+CAS-12+CAS-15+CAS-18).

ClaimLifeResDays
: Lifetime Reserve Days (MIA-02).

ClaimDRGAmt
: Diagnosis Related Group (DRG) Amount (MIA-04).

ClaimDispAmt
: Disproportionate Share Amount (MIA-06).

ClaimMSPPassAmt
: Medicare Secondary Payer (MSP) Pass-through Amount (MIA-07).

ClaimPPSCapAmt
: Total Prospective Payment System (PPS) Capital Amount (MIA-08).

ClaimPPSFedDRGAmt
: Prospective Payment System (PPS) Federal DRG Amount (MIA-09).

ClaimPPSHospDRGAmt
: Prospective Payment System (PPS) Hospital DRG Amount (MIA-10).

ClaimPPSDispHospDRGAmt
: Prospective Payment System (PPS) Hospital Disproportionate Share DRG Amount (MIA-11).

ClaimDayCount
: Cost Report Days (MIA-15).

ClaimFedDRGAmt
: Federal DRG Amount (MIA-16).

ClaimIndTeachAmt
: Indirect Teaching Amount (MIA-18).

2110

SvcProcCode
: Procedure Code (SVC-0102).

SvcChgAmt
: Line Item Charge Amount (SVC-02).

SvcPmtAmt
: Line Item Paid Amount (SVC-03).

2110 CAS

SvcAdjTotalAmt
: Sum of Adjustment Amounts (CAS-03, 06, 09, 12, 15, 18).

PLB

Note: The PLB section follows the current 835 Custom Report Claim (CLP) and Error sections. To display the PLB segment information and the errors affecting it, you must use, at minimum, the .PLBDETAIL and .PLBERRORDETAIL layout directives. If these layout directives are not used, the PLB variables are left blank.

TotalNonClaimAdj
: Non-Claim Adjustment Total (Sum PLB-04, 06, 08, 10, 12, 14).

%PLBReferenceID%
: Provider reference ID.

%PLBDate%
: Fiscal period date, in CCYYMMDD.

%PLBAdjustmentCode1%
: Adjustment reason code 1.

%PLBAdjustmentRefID1%
: Adjustment reference ID 1.

%PLBMonetaryAmount1%
: Amount of adjustment 1.

%PLBAdjustmentCode2%
: Adjustment reason code 2.

%PLBAdjustmentRefID2%
: Adjustment reference ID 2.

%PLBMonetaryAmount2%
: Amount of adjustment 2.

%PLBAdjustmentCode3%
: Adjustment reason code 3.

%PLBAdjustmentRefID3%
: Adjustment reference ID 3.

%PLBMonetaryAmount3%
: Amount of adjustment 3.

%PLBAdjustmentCode4%
: Adjustment reason code 4.

%PLBAdjustmentRefID4%
: Adjustment reference ID 4.

%PLBMonetaryAmount4%
: Amount of adjustment 4.

%PLBAdjustmentCode5%
: Adjustment reason code 5.

%PLBAdjustmentRefID5%
: Adjustment reference ID 5.

%PLBMonetaryAmount5%
: Amount of adjustment 5.

%PLBAdjustmentCode6%
: Adjustment reason code 6.

%PLBAdjustmentRefID6%
: Adjustment reference ID 6.

%PLBMonetaryAmount6%
: Amount of adjustment 6.

837-Specific Report Variables

These variables can be used only for an 837 custom report template.

HEADER Section

TransTypeIDQ	Transmission type ID qualifier (REF-01 in Table 1, position 015).
TransTypeID	Transmission type ID (REF-02 in Table 1, position 015).
BusAppId	Contents of the BHT-03.
BusAppDate	Contents of the BHT-04 in MM-DD-YYYY format.
BusAppTime	Contents of the BHT-05 in HH:MM format.

DETAIL or TRAILER Section

BCBSClaimID	BCBS Claim ID (REF02 when REF01 = "D9").
ClaimCnt_byTS	Number of claims encountered so far in this transaction set. Resets at the beginning of each transaction set.
ClaimCntGood_byTS	Number of good claims encountered so far in this transaction set. Resets at the beginning of each transaction set.
ClaimCntBad_byTS	Number of bad claims encountered so far in this transaction set. Resets at the beginning of each transaction set.
ClaimAmt_byTS	Total claim amount (CLM-02) in dollars and cents so far in this transaction set. Resets at the beginning of each transaction set.
ClaimAmtGood_byTS	Total good claim amount (CLM-02) in dollars and cents so far in this transaction set. Resets at the beginning of each transaction set.
ClaimAmtBad_byTS	Total bad claim amount (CLM-02) in dollars and cents so far in this transaction set. Resets at the beginning of each transaction set.
ClaimPercentGood_byTS	Percent of claims that are good in this transaction set; includes one decimal place. Example: `60.0`.
ClaimPercentBad_byTS	Percent of claims that are bad in this transaction set; includes one decimal place. Example: `40.0`.
ClaimBillType	Concatenated CLM0501 and CLM0503. This is a customized bill type code.
BCBSClaimID	BCBS Claim ID (REF02 when REF01 = "D9").
NonBCBSClaimID	Non-BCBS Claim ID (REF03 when REF01 = "D9").

1000B (Payer)

PayerName	Name (1000B/NM1-03).
PayerID	ID (1000B/NM1-09).
PayerIDQual	ID qualifier (1000B/NM1-08).

1000B (Receiver Name)

ReceiverName	Receiver Name (1000B/NM103)
ReceiverID	Receiver ID (1000B/NM109)
ReceiverIDQual	Receiver ID Qualifier (1000B/NM108)

Secondary Payer

SecPayerName	Secondary Payer name.
SecPayerID	Secondary Payer ID.
SecPayerIDQual	Secondary Payer ID Qualifier.

2010BB (Payer Name)

PayerName2010BB	Payer Name (2010BB/NM103)
PayerNameID	PayerNameID (2010BB/NM109)
PayerNameSecondaryIDQual	Payer Name Secondary ID Qualifier (2010BB/REF02, where REF01=2U, EI, FY, or NF)

Secondary Payer

SecPayerID	Secondary Payer ID (2010BB/REF01)
SecPayerIDQual	Secondary Payer ID Qualifier (2010BB/NM108)

Payer Routing Indicator

PayerRI	Payer Routing Indicator.

1000A (Trading Partner)

TPName	Name (1000A/NM1-03).
TPID	ID (1000A/NM1-09).
TPIDQual	ID qualifier (1000A/NM1-08).

2000A (Provider)

Provider identification information (variables starting with PROV) comes from the Pay-to Provider loop (2010AB) if the 2010AB NM109 contains an ID. Otherwise, it comes from the 2010AA loop.

ProvLName	Last name (2010AB or 2010AA/NM1-03).
ProvFName	First name (2010AB or 2010AA/NM1-04).
ProvMName	Middle name (2010AB or 2010AA/NM1-05).
ProvName	Provider name in 'Last, First Middle' format.
ProvID	ID (2010AB or 2010AA/NM1-09).
ProvIDQual	ID qualifier (2010AB or 2010AA/NM1-08).
ProvSecID	Secondary ID (2010AB or 2010AA/REF-02).
ProvSecIDQual	Secondary ID qualifier (2010AB or 2010AA/ REF-01).

ClaimAmt_byProv	Total in CLM-02 for all claims in this 2000A loop. Put this in the PROVTRAILER section, before the TRAILER section.
ClaimAmtBad_byProv	Total in CLM-02 for bad claims in this 2000A loop. Put this before the PROVTRAILER section.
ClaimAmtGood_byProv	Total in CLM-02 for good claims in this 2000A loop. Put this before the PROVTRAILER section.
ClaimCnt_byProv	Number of claims in this 2000A loop. Put this in the PROVTRAILER section.
ClaimCntGood_byProv	Number of good claims in this 2000A loop. Put this in the PROVTRAILER section.
ClaimCntBad_byProv	Number of bad claims in this 2000A loop. Put this in the PROVTRAILER section.
ClaimPercentGood_byProv	Percent of claims that are good in this 2000A loop. Put this in the PROVTRAILER section.
ClaimPercentBad_byProv	Percent of claims that are bad in this 2000A loop. Put this in the PROVTRAILER section.

2010AA (Billing Provider)

BillProvLName	Last name (2010AA/NM1-03).
BillProvFName	First name (2010AA/NM1-04).
BillProvMName	Middle name (2010AA/NM1-05).
BillProvName	Billing Provider name in 'Last, First Middle' format.
BillProvID	ID (2010AA/NM1-09).
BillProvIDQual	ID qualifier (2010AA/NM1-08).
BillProvSecID	Secondary ID (2010AA/REF-02).
BillProvSecIDQual	Secondary ID qualifier (2010AA/REF-01).

2010AB (Pay-To Provider)

PayProvLName	Last name (2010AB/NM1-03).
PayProvFName	First name (2010AB/NM1-04).
PayProvMName	Middle name (2010AB/NM1-05).
PayProvName	Pay-To Provider name in 'Last, First Middle' format.
PayProvID	ID (2010AB/NM1-09).
PayProvIDQual	ID qualifier (2010AB/NM1-08).

PayProvSecID	Secondary ID (2010AB/REF-02).
PayProvSecIDQual	Secondary ID qualifier (2010AB/REF-01).

2010BA (Subscriber)

SubLName	Last name (2010BA/NM1-03).
SubFName	First name (2010BA/NM1-04).
SubMName	Middle name (2010BA/NM1-05).
SubName	Subscriber name in 'Last, First Middle' format.
SubID	ID (2010BA/NM1-09).
SubIDQual	ID qualifier (2010BA/NM1-08).
SubPropProvID	Proprietary Provider ID (2010BB REF for Billing Provider Secondary Identification)

2010CA (Dependent)

DepLName	Last name (2010CA/NM1-03).
DepFName	First name (2010CA/NM1-04).
DepMName	Middle name (2010CA/NM1-05).
DepName	Dependent name in 'Last, First Middle' format.
DepID	ID (2010CA/NM1-09).
DepIDQual	ID qualifier (2010CA/NM1-08).

Patient

The patient is the subscriber or the dependent, depending on where the claim occurred

PatLName	Last name (2010xA/NM1-03).
PatFName	First name (2010xA/NM1-04).
PatMName	Middle name (2010xA/NM1-05).
PatName	Patient name in 'Last, First Middle' format.
PatID	ID (2010xA/NM1-09).
PatIDQual	ID qualifier (2010xA/NM1-08).
PatBday	Patient Birth Day (2010xA/DMG-02).
PatGender	Patient Gender (2010xA/DMG-03).
PatRelship	Patient Relationship (SBR-02).

2300

ClaimID	Claim number (CLM-01).
ClaimAmt	Total Amount (CLM-02).
ClaimCHID	Claim Identification Number for Clearing Houses and Other Transmission Intermediaries (REF-02 when REF-01=D9).
ClaimICDN	ICDN (REF-02 when REF-01 = F8).

ClaimMRN	Medical Record number (REF-02 when REF-01 = EA).
ClaimDate	Claim date.
ClaimStatus	Claim contains one of these: `Good` no errors or warnings `Error` contains at least one error `Warning` contains at least one warning, but no errors See Acceptance or Rejection Criteria on page 57 for definitions of errors and warnings.
2310A	*(For 837P and 837D, this is Referring Provider. For 837I, this is Attending Physician. The following variables work for all three types of 837)*
ReferProvName	Referring/attending name in 'Last, First Middle' format.
ReferProvLName	Last name (2310A/NM1-03).
ReferProvFName	First name (2310A/NM1-04).
ReferProvMName	Middle name (2310A/NM1-05).
ReferProvID	ID (2310A/NM1-09).
ReferProvIDQual	ID qualifier (2310A/NM1-08).
ReferProvSecID	Secondary ID qualifier (2310A/REF-01).
ReferProvSecIDQual	Secondary ID (2310A/REF-02).
2330A	
ClaimOtherNameID	NM1-09 in loop 2330A Other Subscriber Name.
OtherSubNameID	Other Subscriber Name ID (2330A/NM109).
2330B	
ClaimOtherPayerSecId	REF-02 in loop 2330B Other Payer Name. See also CobaClaimOtherPayerSecId.
CobaClaimOtherPayerSecId	REF-02 in loop 2330B Other Payer Secondary Identifier-only if REF-01=F8.
OtherPayerControlNumber	Other Payer Control Number (2330B/REF02, where REF01=F8)
OtherPayerNameSecID	Other Payer Name (2330B/REF02, where REF01= 2U, EI, FY or NF) OtherPayerControlNumber (2330B/REF02, where REF01=F8)
Service	

Service variables really represent lists of values, since a claim may have more than one. If so, subsequent occurrences appear on another line.

SvcID	Service ID (2400/SV1-01.02 (P), 2400/SV2-01 (I), 2400/SV3-01.02 (D)).
SvcIDQual	ID qualifier.
SvcLine	Service Line number.
SvcChgAmt	Line Charge Amount.
SvcQty	Quantity.
SvcMod1	Modifier 1.
SvcMod2	Modifier 2.
SvcMod3	Modifier 3.
SvcMod4	Modifier 4.
SvcDateQual	Service date qualifier.
SvcDate	Service date.
ClaimFirstSrvDate	The earliest data in a service line for this claim (2400/DTP-03).
ClaimLastSrvDate	The latest data in a service line for this claim (2400/DTP-03).

ISATRAILER Section

These values can be used in the .ISATRAILER section. Values may be uninitialized if used in other report sections. Multiple values are presented in table format.

ClaimAmt_ByISARI	Total amount for all claims, per unique RI.
ClaimAmtBad_ByISARI	Claim amount totals for all bad claims, collated by unique RI.
ClaimAmtGood_ByISARI	Claim amount totals for all good claims, collated by unique RI.
ClaimCnt_ByISARI	Total count of all claims, per unique RI.
ClaimCntBad_ByISARI	Number of bad claims per unique RI.
ClaimCntGood_ByISARI	Number of good claims per unique RI.
ClaimPctBad_ByISARI	Percent of bad claims vs. total claims, per unique RI.
ClaimPctGood_ByISARI	Percent of good claims vs. total claims, per unique RI.
ISARI	List of all Routing Indicators seen within the ISA.

GROUPTRAILER section

Group values

These group values can be used in the 837 .GROUPTRAILER section. Values may be uninitialized if used in other report sections. Multiple values are presented in table format.

ClaimAmt_ByGroupRI Total amount for all claims, per unique RI.

ClaimAmtBad_ByGroupRI
Claim amount totals for all bad claims, collated by unique RI.

ClaimAmtGood_ByGroupRI
Claim amount totals for all good claims, collated by unique RI.

ClaimCnt_ByGroupRI Total count of all claims, per unique RI.

ClaimCntBad_ByGroupRI
Number of bad claims per unique RI.

ClaimCntGood_ByGroupRI
Number of good claims per unique RI.

ClaimPctGood_ByGroupRI
Percent of good claims vs. total claims, per unique RI.

ClaimPctBad_ByGroupRI
Percent of bad claims vs. total claims, per unique RI.

GroupRI List of all Routing Indicators seen within the ISA.

TRAILER section

Transaction Set values

These Transaction Set values can be used in the 837 .TRAILER section. Values may be uninitialized if used in other report sections. Multiple values are presented in table format.

BCBSClaimID BCBS Claim ID (REF02 when REF01 = "D9").

ClaimAmt_BySetRI Total amount for all claims, per unique RI.

ClaimAmtBad_BySetRI Claim amount totals for all bad claims, collated by unique RI.

ClaimAmtGood_BySetRI
Claim amount totals for all good claims, collated by unique RI.

ClaimCnt_BySetRI Total count of all claims, per unique RI.

ClaimCntBad_BySetRI Number of bad claims per unique RI.

ClaimCntGood_BySetRI
Number of good claims per unique RI.

ClaimPctBad_BySetRI	Percent of bad claims vs. total claims, per unique RI.
ClaimPctGood_BySetRI	Percent of good claims vs. total claims, per unique RI.
NonBCBSClaimID	Non-BCBS Claim ID (REF03 when REF01 = "D9").
SetRI	List of all Routing Indicators seen within the ISA.

Troubleshooting Custom Reports

The most common causes for custom report failures:

Symptom: **No custom report is generated**

Probable cause: The guideline used for validation was not a GuidelinePlus and did not create the necessary SVALU records. See **ForesightHIPAAguidelinelist.pdf**.

Symptom: **No data for a variable**

Probable cause: The template variable was used before it was encountered. Move the affected variables down into the proper section. See List of Template Variables on page 89.

Enveloping Problems

When Response Generator cannot find or understand the ST or GS in the input EDI file, and you requested a custom report, you will get a report file containing error messages. The return code remains 100.

.

6 Appendix A: Return Codes

Return Code	Meaning
100	Response generation succeeded.
101 – 199	**NOTE:** Return codes **101** through **199** indicate serious or unusual Response Generator processing errors. These codes may be generated in response to back-end issues such as Instream detail file problems or lack of available memory on your system. If you receive a return code in this range, make note of the error number and contact TIBCO Foresight Technical Support.
101	Could not generate EDIFACT CONTRL document.
102	Mandatory SVALU record is missing.
105	Mandatory ISA1 SVALU record is missing.
106	Mandatory GSSG SVALU record is missing.
107	Tree has no child nodes.
108	Failed to Locate Mandatory GSSG SVALU Record.
109	EDIFACT CONTRL document generation failed.
110	Response generation failed.
111	RespGen could not generate response document after opening required files.
112	Could not parse X12 270 document.
113	Could not create 270 Payer loop.
114	270 Payer map corrupted.
115	Could not create 270 Provider loop.
116	270 Provider map corrupted.
117	Could not create 270 Subscriber loop.
118	270 Subscriber map corrupted.
119	270 SUBTRN SVALU found with no corresponding SBNM.
120	270 Subscriber map corrupted during 2100C loop parsing.
121	Could not create 270 Dependent loop.
122	270 Dependent map corrupted.

Return Code	Meaning
123	270 DEPTRN SVALU found with no corresponding DPNM.
124	270 Dependent map corrupted during 2100D loop parsing.
125	Could not parse X12 276 document.
126	276 Payer corrupted.
127	Lost 276 Transaction Header.
128	Lost 276 Payer Header.
129	276 Trading Partner corrupted.
130	Lost 276 Receiver. Document is missing The detail file is missing information responsegen needs to generate a response.
131	276 Provider corrupted.
132	Lost 276 Provider.
133	276 Subscriber corrupted.
134	Lost 276 Subscriber.
135	276 Dependent corrupted.
136	Lost 276 Dependent.
137	SBTRN SVALU found with no corresponding Subscriber.
138	SDOS found with no corresponding TRN segment.
139	SPCREF found with no corresponding TRN segment.
140	DPTRN SVALU found with no corresponding Dependent.
141	DDOS found with no corresponding TRN segment.
142	DPCREF found with no corresponding TRN segment.
143	277 Payer corrupted.
144	Lost 277 Payer.
145	277 Trading Partner corrupted.
146	Lost 277 Trading Partner.
147	Lost 277 Provider.
148	Corrupted 277 Provider.
149	Lost 277 Subscriber.
150	Could not parse X12 834 document.
151	834 Member corrupted.
152	Lost 837 Payer.

Return Code	Meaning
153	Lost 837 Trading Partner.
154	837 Payer corrupted.
155	Lost 837 2000A Payer or TP.
156	Lost 837 2000A Provider.
157	837 Trading Partner corrupted.
158	Lost 837 Subscriber.
159	Lost 837 Payer and Subscriber for Secondary Payer.
160	Lost 837 Subscriber Service Line.
161	Lost 837 Dependent Service Line.
162	Could not parse X12 278 document.
163	278 Payer corrupted.
164	278 Payer lost its Transmission structure.
165	278 Receiver corrupted.
166	278 Receiver lost its Payer.
167	278 Subscriber corrupted.
168	278 Subscriber lost its Receiver.
169	278 Dependent corrupted.
170	278 Dependent lost its Subscriber.
171	278 Event corrupted.
172	278 Event not associated with a Subscriber.
173	HL7 MSH loop corrupted.
174	Could not parse X12 820 document.
175	820 Receiver corrupted.
176	820 Payer corrupted.
177	820 OrgRemittance corrupted.
178	820 OrgRemittanceDetail corrupted.
179	820 Remittance Adjustment corrupted.
180	820 IndRemittance corrupted.
181	820 IndRemittanceDetail corrupted.
182	820 Remittance Adjustment corrupted.
185	When running validation, Docsplitter, and Response Generator together in the API, Docsplitter failed.

Return Code	Meaning
186	When running validation, Docsplitter, and Response Generator together in the API, Response Generator failed.
187	When running validation, Docsplitter, and Response Generator together in the API, Docsplitter and Response Generator both failed.
198	TA1 errors were encountered when attempting to create a TA1 file or segment.
199	Errors were encountered when attempting to create a Strict 999.
201	Could not open input file.
202	Could not open one or more output files.
203	Could not open the template file.
204	Requested an inappropriate response document for your original document, such as: ▪ 997 requested for an incoming 997 ▪ 824 requested for an incoming 824 or 997 ▪ 277 requested for any document type other than an 837.
205	STRUS and STRUE records in detail file are mismatched. This can occur when ISA-IEA, GS-GE, and ST-SE segments are not paired properly. Also, be sure that you are using STRUS=1 and STRUE=1 in the validation profile (APF) to create STRUS and STRUE records in the validation detail file.
206	Could not open the specified file; the file either doesn't exist or can't be opened. Check the file for read access permission. Response generator failed.

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.

7 Appendix B: Combinations of X12 Response Documents

The content of response documents varies, depending on:

- What other response documents were requested on the command line
- Whether the strict option was requested on the command line
- The presence of errors

Contents of 824, 997, and 999 997 and 999 always generated if requested 824 generated on errors of type 3+ if requested				
Output Files Generated	**997 or 999 contents**		**824 contents**	
	No Strict	**With Strict**	**No Strict**	**With Strict**
824	*<not generated>*	*<not generated>*	All error types	All error types
997	All error types	Types 1 and 2	*<not generated>*	*<not generated>*
997+824	All error types	Types 1 and 2	Types 3+	Types 3+
999	All error types	Types 1 and 2	*<not generated>*	*<not generated>*
999+824	All error types	Types 1 and 2	Types 3+	Types 3+

Contents of 277U (with / without other output) Generated only as a response to 837s with errors of severity 3+	
824, 997, or 999 Generated	**Rejected claims found?**
<has no effect>	Yes = 277 generated No = empty file generated

Contents of 277CA Generated only as a response to 837s
If requested, the 277CA is always generated, and always contains all claims

8 Appendix C: TPA Setup File

Invoking a TPA Setup File

When using **-TPA**, do not use other command line parameters. Put all settings into the setup file, and be sure that all trading partners will be assigned a settings file in the TPA CSV file.

To invoke a Response Generator trading partner automation setup file, use the -TPA command line parameter (see page 44).

This identifies a Trading Partner Automation lookup file that identifies which partners are to use which Response Generator setup files.

Entering Filenames in the Response Generator TPA Setup Files

In the setup file, items with "FilePath" in their names can use * to refer to the data file without the extension.

Example where the data file is ABC.EDI:

Text in setup file	Filename created by Response Generator
`EDI277`**`FilePath`**`=C:\Out\*_277.txt`	277 will be ABC_277.txt.
`EDI824`**`FilePath`**`=C:\Out\*824.edi`	824 will be ABC824.edi
`EDI997`**`FilePath`**`=C:\Out\`	Since no name is specified, 997 output goes to the default name ABC_997.txt Be sure to include the slash at the end of the path if no output filename is specified

TPA-Parameter Setup File Contents

For an example setup file, see **TPA_ResponseGen.ini** in Instream's DemoData directory.

Setup file text	Description
[824]	You can create an [824] section and include any parameters listed in Appendix D: Z-Parameter Setup File on page 141.
[CommandLine Option]	Section containing general options and items that map to command line options.
864Description	Specify the 864 description field. Equivalent to the command line option: -dcd
864MTIDescription	Specify the 864 MIT description field. Equivalent to the command line option: -dcm
864PurposeCode	Specify the 864 purpose code. Equivalent to the command line option: -dcp
997GroupOnly	Causes 997s and 999s to have AK1 and AK9 segments only (no AK2/AK3/AK4/AK5). Equivalent to the command line option: -f997_group_only
997NoST03	Suppresses the ST03 in cases where the response document is greater than 4060. Equivalent to the command line option: -f997_no_ST03
Append	Append the generated EDI data to the specified output file. Equivalent to the command line option: -a
ApplicationReceiverID	Specify an Application Receiver ID (GS03) for the outbound GS. Value is 2 to 15 alphanumeric characters; default is the first Application Sender ID (GS02) in the inbound EDI file Equivalent to the command line option: -dar Example: `ApplicationReceiverID=PURCH1256`
ApplicationSenderID	Specify an Application Sender ID (GS02) for the outbound GS. Value is 2 to 15 alphanumeric characters; default is the first Application Receiver ID (GS03) in the inbound EDI file Equivalent to the command line option: -das Example: `ApplicationSenderID=PURCH1256`
ApplicationVersion	Specify an Application Version (GS08) for the output EDI's GS08. If the transaction has an ST03, this will be populated with the same value. Value is a string of 1 to 12 characters; default is 004010 Equivalent to the command line option: -dav Example: `ApplicationVersion=005010`

Setup file text	Description
ConfigFile	Specify an optional configuration file (see Appendix D: Z-Parameter Setup File on page 141. Indicates the configuration files ErrMsgTrans.txt and ErrRespXref.txt are located in the specified directory. Do not use with **UseBaseroot**. Equivalent to the command line options: -z and -cd Example: `ConfigFile=c:\configs\RG824config.ini`
ContactInformation	Specifies a contact phone number, which is needed for L2 report trailers. Equivalent to the command line option: - dcc
CustomReportContents	Specifies the contents of the custom report; only use if GenerateCustomReport is enabled. Value must be 0, 1, 2, or 3; default is 1. Equivalent to the command line option: -te
CustomReportErrorTypes	Specifies which error types are included in the report. Equivalent to the command line option: -tel Example: `CustomReportErrorTypes=123`
CustomReportSeverityLevels	Specifies which severity levels are included in the report. Equivalent to the command line option: -tev Example: `CustomReportSeverityLevels=34`
CustomReportTemplate	Path to a custom report template; only use if GenerateCustomReport is enabled. Equivalent to the command line option: -tpl

DeleteZeroLengthResults	Should Response Generator remove zero length output files?
	Equivalent to the command line option: -nz
	Examples:
	`DeleteZeroLengthResults=true`
	`DeleteZeroLengthResults=false`

Do277x070	Create a 277 that conforms to the 3070x070A standard. Equivalent to the command line option: -fdo277x070
Do277x167	Create a 277 that conforms to the 4050x167 standard. Equivalent to the command line option: -fdo277x167
Do824x166	Create an 824 that conforms to the 824x166 standard. Equivalent to the command line option: -fdo824x166
Do824x186	Create an 824 that conforms to the 824x186 standard. Equivalent to the command line option: -fdo824x186
Do864Wrapper	This creates an 864 wrapper around a custom report. Equivalent to the command line option: -fdo864Wrapper

Setup file text	Description
Do999NonErrata	If set to true (1), a non-errata version of the 999 is generated. If set to false (0), an errata version is generated. Equivalent to the command line option: -fdo999ne
DoAK3OnWarnings	For warnings (severity of 2), generate: - AK3/AK4 values in 997s - IK3/IK4 values in 999s Equivalent to the command line option: -fak3
DoAK901E	Should Response Generator use E in the 997 AK901 to indicate Accepted with Errors when all AK501s generated for the functional group are set to A or E? (Default behavior is the AK901 contained an A under these circumstances.) Equivalent to the command line option: - fAK901E Example: `DoAK901E=1`
DoCtlNumPassThrough	If set to true, ISA, GS, and ST1 control numbers are passed through to the response document. Equivalent to the command line option: -fdo_ctrlnum_passthrough
DoGS06Passthrough	Passes GS06 through to the 864 and the MIT01. Equivalent to the command line option: -fdo_gs06_passthrough
DoGS08Passthough	For created 997, GS08 mirrors the GS08 of the source document. This option is incompatible with the -dav (ApplicationVersion) option, and Response Generator fails if both are set. Equivalent to the command line option: -fdo_gs08_passthrough
DoHIPAA824	If set to true (1), an 824 is generated that conforms to either the 824x166 or 824x186 standard, based upon the version of the source document. Equivalent to the command line option: -fdo_HIPAA_824
DoLoopRollup	If set to true (1), 277CA, 277H, and 277U responses will have one loop per provider, even if the incoming document had multiple 2000A loops for a single provider. Default is false (0). Equivalent to the command line option: -fdo_loop_rollup. Note that setting this value to true (1) can cause differences in the interpretation of bad claims between the response document and DocSplitter/Importer.

Setup file text	Description
EDI277CA	If set to any of these, EDI277FilePath creates a 277CA: EDI277CA=1 EDI277CA=true EDI277CA=yes EDI277CA=on Default is for EDI277FilePath to create a 277U Requires EDI277FilePath Equivalent to the command line option: -0277 with -fca277
EDI277FilePath	Output a 277 By default, this creates a 277U. To create a 277CA, see EDI277CA Equivalent to the command line option: -o277 Example: `EDI277FilePath=C:\Output\*_277.txt`
EDI824FilePath	Output an 824 Equivalent to the command line option: -o824 Example: `EDI824FilePath=C:\Output\*824.edi`
EDI997FilePath	Output a 997 Equivalent to the command line options: -o997 and -octl Example: `EDI997FilePath=C:\Output\`
EDI999FilePath	Output a 999 Equivalent to the command line option: -o999 Example: `EDI999FilePath=C:\Output\`
EDIFACTAssociationCode	Specify a new UNH02.05 association agency code value. Equivalent to the command line option: -dea
EDIFACTControllingAgency	Specify a new UNH02.04 controlling agency code value. Equivalent to the command line option: -dec
EDIFACTGenerateUNA	Generate a UNA segment in the EDIFACT CONTRL response. Equivalent to the command line option: -fedifact_gen_una
EDIFACTICOnly	Cause EDIFACT CONTRL document to generate a response for the status of the Interchange only. Equivalent to the command line option: -fedifact_ic_only
EDIFACTNoUCM	Suppress generation of UCM segments in EDIFACT CONTRL documents. Equivalent to the command line option: -fedifact_no_ucm
EDIFACTReleaseNumber	Specify a new UNH02.03 release number value. Equivalent to the command line option: -der
EDIFACTVersionNumber	Specify a new UNH02.02 version number value. Equivalent to the command line option: -dev

Setup file text	Description
EDITA1FilePath	Output a TA1 Equivalent to the command line option: -oTA1
ElementDelimiter	Specify the element delimiter for the output files. This can be an actual character surrounded by double quotes, or an ASCII number representing the character. An ASCII number may be hexadecimal, by starting the number with 0x, octal by starting the number with a zero or decimal. The default delimiter is *. Equivalent to the command line option: -l Examples: `ElementDelimiter="^"`
GenerateAIS	If this is set to 1, the generated 277u will use the value in a ZZKPA custom record that has been inserted into the validation detail file via business rules. Typically, the business rule that generates this record would be in the 837's claim loop. It will put it in the 277u's REF02 at 2200D or 2200E (REF01=1K). If the detail file does not contain the ZZKPA record, this setting is ignored. Equivalent to the command line option: -fno_AIS Example: GenerateAIS=1
GenerateCustomReport	Output a custom report; requires other settings too Equivalent to the command line option: -otext Example: `GenerateCustomReport=C:\Output\textout.txt`
GenerateInterchangeAndFunctionalGroup	Specify whether to generate interchange and functional groups (the TA1 response does not need functional groups). Value is True or False; default is True. Equivalent to the command line option: -ge Example: `GenerateInterchangeAndFunctionalGroup=false`
GenTA1	This determines the conditions under which a TA1 segment will be created. See gTA1 Command section. Value is 0, 1, 2, 3, or 4; default is 0. Equivalent to the command line option: -gTA1 Example: `GenTA1=3`
GS05_HHMM	If set to true (1), the format of the GS05 is set to four characters (HHMM). If set to false (0), the default value is used (HHMMSS). Equivalent to the command line option: -fdo_gs05_hhmm Example: `GS05_HHMM=1`

Setup file text	Description
GS05_HHMMSS	If set to true (1), the format of the GS05 is set to six characters (HHMMSS). If set to false (0) and no other format is specified, this remains the default value. To use a different format, set one of these options to true (1): GS05_HHMM, GS05_HHMMSSD, or GS05_HHMMSSDD Equivalent to the command line option: -fdo_gs05_hhmmss Example: `GS05_HHMMSS=1`
GS05_HHMMSSD	If set to true (1), the format of the GS05 is set to seven characters (HHMM). If set to false (0), the default value is used (HHMMSSD). Equivalent to the command line option: -fdo_gs05_ hhmmssd Example: `GS05_HHMMSSD=1`
GS05_HHMMSSDD	If set to true (1), the format of the GS05 is set to eight characters (HHMMSSDD). If set to false (0), the default value is used (HHMMSSD). Equivalent to the command line option: -fdo_gs05_hhmmssdd Example: `GS05_HHMMSSDD=1`
GSControlNumber	Allows you to specify a GS Control Number. This user-specified GS Control Number should be thought of as a GS-level trace number; it is not the same as the GS06 number. Equivalent to the command line option: -dcg<gs control number value>

Setup file text	Description

InterchangeControlControlPath	**Note:** InterchangeControlControlPath and InterchangeControlNumberStart are both used to specify a starting interchange control number. Review both parameters before deciding which best suits your needs.
	
	InterchangeControlControlPath allows you to use a file to specify a starting interchange control number. This method allows Response Generator to “remember” the last interchange control number used and increment by one each time it encounters an ISA, no matter how many files are processed. There is no maximum number.
	
	Example:
	
	File 1 is run through Response Generator – The first ISA within the file is assigned interchange control number 000000001, the second is assigned 000000002, the third is assigned 000000003.
	
	File 2 is run through Response Generator – Response Generator continues the numbering sequence that began with File 1. The first ISA within the file is assigned interchange control number 000000004, the second is assigned 000000005, the third is assigned 000000006, and so on.
	
	To use this setting:
	
	1. Deactivate/comment out the following settings by adding a colon (:) to the beginning of each line:
	
	`: InterchangeControlNumberStart=1`
	`: InterchangeControlNumberMin=1`
	`: InterchangeControlNumberMax=100`
	`: InterchangeControlNumberStart`
	
	2. Activate InterchangeControlControlPath by either
	
	(a) typing the entry (if the setting does not exist in the file), or (b) deleting the colon (:) in front of the entry.
	
	3. Specify the path to a filename that contains a 9-digit starting interchange control number, such as:
	
	`000000001`
	`000000500`
	`000002000`
	
	Equivalent to the command line option: -dic <*filename*>
	
	Example:
	`InterchangeControlControlPath=my_control_filename.txt`
	
	Where the file `my_control_filename.txt` contains a nine-digit number such as 000000100.

Setup file text	Description
InterchangeControlNumberMax	Specify the maximum interchange control number. When the maximum number is reached, the interchange control number count will reset to the minimum interchange control number. Requires use of InterchangeControlNumberMin parameter (see InterchangeControlNumberMin). Default = 999999999. Equivalent to the command line option: -dic Example: `InterchangeControlNumberMax=100`
InterchangeControlNumberMin	Specify the minimum interchange control number. Requires use of InterchangeControlNumberMax parameter. The interchange control number count will start with the number specified with this parameter and increment by one until the maximum number is reached (see InterchangeControlNumberMax). Default = 000000001. Equivalent to the command line option: -dic Example: `InterchangeControlNumberMin=10`

Setup file text	Description

InterchangeControlNumberStart	**Note:** InterchangeControlNumberStart and InterchangeControlControlPath are both used to specify a starting interchange control number. Review both parameters before deciding which best suits your needs.
	
	InterchangeControlNumberStart allows you to specify a 9-digit starting interchange control number to be used each time Response Generator runs a file. This value should be between InterchangeControlNumberMin and InterchangeControlNumberMax, if they are specified.
	
	Examples:
	`000000001`
	`000000500`
	`000002000`
	
	Numbering increments within **that file only** and numbering restarts with each file if multiple files are processed.
	
	Example:
	
	File 1 is run through Response Generator – The first ISA within the file is assigned interchange control number 000000001, the second is assigned 000000002, and the third is assigned 000000003.
	
	File 2 is run through Response Generator – the count is restarted using the specified starting interchange control number. The first ISA within the file is assigned interchange control number 000000001, the second is assigned 000000002, the third is assigned 000000003, and so on.
	
	Note: This parameter cannot be used when InterchangeControlControlPath is active. If it exists, you must deactivate/comment out the setting by adding a colon (:) to the beginning of the line:
	
	`: InterchangeControlControlPath`
	
	Equivalent to the command line option: -dic <*starting ICN*>
	
	Example: `InterchangeControlNumberStart=000000010`

InterchangeReceiverID	Specify an interchange receiver ID to use in the outbound ISA08. Value is 2 to 15 alphanumeric characters; If less than 15 characters, it will be automatically padded on the right with spaces to make it 15 characters Default is the Interchange Receiver ID (ISA08) in the validation detail report Equivalent to the command line option: -dir Example: `InterchangeReceiverID=PARTNERCORP`

Setup file text	Description
InterchangeResponseRequested	Sets a flag in the ISA14 element to notify the receiver that the sender is expecting a TA1 to be returned for the document. Value is on or off; default is off. If value is **on**, the ISA14 is set to '1', indicating the sender is requesting a TA1 be returned for the document. If value is **off**, the ISA14 is set to '0', indicating the sender is not requesting a TA1 for the document. Equivalent to the command line option: -diq Example: `InterchangeResponseRequested=ON`
InterchangeSenderID	Specify an interchange sender ID to use in the outbound ISA06. Value is 2 to 15 alphanumeric characters; if less than 15, it is automatically padded on the right with spaces to make it 15 characters Default is the Interchange Sender ID (ISA06) in the validation detail report Equivalent to the command line option: -dis Example: `InterchangeSenderID=KAVERCORP`
InterchangeUsageIndicator	Specify the interchange usage indicator (ISA15) for the outbound ISA. Value is I (Information), P (production), T (testing) or X (use the value from the input file's ISA15); default is P. Equivalent to the command line option: -diu Example: `InterchangeUsageIndicator=P`
LenientAK501	If `LenientAK501= true`, `yes`, `on`, or `1`, then a 997 or 999 AK501 and AK509 can contain only A or E: ▪ If there are no errors, then the AK501 = A. ▪ If there are errors, regardless of type or number of claims, then the AK501 = E. All other settings cause LenientAK501 to be false. Equivalent to the command line option: -k
LooseSourceCheck	Return 100 (success) even when a response document cannot be created from the source (e.g., as a 277 from a non-837). Equivalent to the command line option: -floose_src_check
NewLines	Should a new-line sequence follow each segment delimiter in the EDI output? Value is True or False; default is True. Equivalent to the command line option: -n Example: `NewLines=true`

Setup file text	Description
NoAK103	Suppress output of the 997 AK103 segment. By default, AK103s are created if the source document is version 4060 or greater or if a 999 is requested. If this switch is present (NoAK103 = true), AK103 output is suppressed. Value is True or False; default is False. Equivalent to the command line option: -fno_AK103
NoUNG	Turns off functional group (UNG) data in EDIFACT CONTRL responses. Equivalent to the command line option: -no_ung
OriginalFilename	Specifies the original filename, which is needed for L2 report headers. Equivalent to the command line option: -dco<original filename>
OverwriteOK	Allows overwriting of the EDI output file without prompting (if it already exists). This option cannot be used if the Append option is set. If true, existing output files are overwritten. Example: `OverwriteOK=true` Equivalent to the command line option: -y
PartialFunctionalGroupAcceptance	Should partial functional group acceptance be allowed when generating a 997 or 824 response? If not, when any transaction is rejected, the entire functional group is rejected. Value is True or False; default is True. Default (True) is to partially accept a functional group if any transaction set is accepted, and reject a functional group only if all transaction sets are rejected. Equivalent to the command line option: -np Example: `PartialFunctionalGroupAcceptance=true`

Setup file text	Description
PartialTransactionSetAcceptance	Should partial transaction set acceptance be allowed when generating a 997 or 999 response? This parameter is useful when you are using Docsplitter. If omitted, default behavior is used: transaction set is rejected if there are any errors. If set to “p”, this is equivalent to the command line option: `-pt p` If set to ***anything else*** (true, false, 0, 1, etc.), it is equivalent to the command line option: `-pt` Examples: `PartialTransactionSetAcceptance="p"` ← on `PartialTransactionSetAcceptance ="t"` ← off `PartialTransactionSetAcceptance ="1"` ← off `PartialTransactionSetAcceptance ="on"` ← off
PartialTransactionSetAcceptance824	Should partial transaction set acceptance be allowed when generating an 824 response? This parameter is useful when you are using Docsplitter. If omitted, default behavior is used: transaction set is rejected if there are any errors. If set to anything (true, false, 0, 1, etc.), it is equivalent to the command line option: `-pt824` Example: `PartialTransactionSetAcceptance824 ="true"`
RejectErrorSeverity	Set the error severity that will result in a rejected transaction set. Value is 0, 1, 2, 3, 4, 5, or 6; default is 3. Equivalent to the command line option: -er Example: `RejectErrorSeverity=2`
RepetitionSeparator	Specify the repetition separator for the output files. This can be an actual character surrounded by double quotes, or an ASCII number representing the character. An ASCII number may be hexadecimal, by starting the number with 0x, or octal by starting the number with a zero, or decimal. The default separator is ^. Equivalent to the command line option: -I Example: RepetitionSeparator=”^” RepetitionSeparator=94 (Decimal value) RepetitionSeparator=0x5E (Hexadecimal value) RepetitionSeparator=0136 (Octal value)

Setup file text	Description
SegmentDelimiter	Specify the segment delimiter for the output files. This can be an actual character surrounded by double quotes, or an ASCII number representing the character. An ASCII number may be hexadecimal, by starting the number with 0x, octal by starting the number with a zero or decimal. If a new-line sequence is to be used as a segment terminator, set the Segment Delimiter to zero (0). The default delimiter is ~. Equivalent to the command line option: -l Example: `SegmentDelimiter="!"` `SegmentDelimiter=42` (Decimal value) `SegmentDelimiter=0x2A` (Hexadecimal value) `SegmentDelimiter=052` (Octal value)
SeveritylevelFilter	Specify the severity levels to go into the report. By default, all severity levels go in the report. Equivalent to the command line option: -tev Example (put claims with errors of severity 3, 4, and 5 in the report): `SeveritylevelFilter=345`
SeveritylevelFilter	Specify the error types to go into the report. By default, all error types go in the report. Equivalent to the command line option: -te1 Example (put claims with errors of type 1, 2, 3, and 7 in the report): `SeveritylevelFilter=1237`
ShowAllClaims	Create 824s and 277s from clean documents. Equivalent to the command line option: -fsa
StartingGS06	Specify a starting group control number (GS06) for the outbound GS. Value is a number greater than 0 and no more than 999999999; default is 1 This value automatically increments with each group until it reaches 999999999 or is reset with another StartingGS06 or -dac Equivalent to the command line option: -dac Example: `StartingGS06=1000`
StartingST02	Specify a starting transaction set control number (ST02) for the output EDI file. Value is a number greater than 0; default is 1 Equivalent to the command line option: -c Example: `StartingST02=500`

Setup file text	Description
STCTablePath	Name of a STC override file, if used. See Appendix G: 277CA STC Override on page 175. Example `STCTablePath=C:\setup\STCoverride.txt`
STCTablePath	Specifies table to STC override file for 277C. See Appendix G: 277CA STC Override on page 175. Equivalent to the command line option: -stc
Strict997	Should strict 997 or 999 response processing be used? Strict means: ▪ 997s and 999s will only show error types 1 and 2 ▪ 824s will only show application error types (usually 3 -7) By default, if only one of these types of output transactions are generated: 997s, 999s, or 824s … then ALL errors, regardless of type, are put in the one type generated. Equivalent to the command line option: -s Example: `Strict997`
SubelementDelimiter	Specify the subelement delimiter for the output files. This can be an actual character surrounded by double quotes, or an ASCII number representing the character. An ASCII number may be hexadecimal, by starting the number with 0x, octal by starting the number with a zero or decimal. The default delimiter is : (a colon). Equivalent to the command line option: -l Example: `SubelementDelimiter="#"`
UseBaseroot	Specifies where to find configuration files ErrMsgTrans.txt and ErrRespXref.txt. Do not use with **ConfigFile**. Equivalent to the command line option: -u
UseHighest99x	Generates a 999 if available according to the source version, overriding any previously specified output. Equivalent to the command line option: -fuse_highest_99x
UseShortGS05	If present, the GS05 is set to four characters (HHMM), instead of the default six (HHMMSS). Equivalent to the command line option: -fuseshortGS05
UseSourceDelims	Use the source document's delimiters for the response document delimiters. (Requires the GEN 10223 record to be present.) Equivalent to the command line option: -lsource

Setup file text	Description
Verbosity	Set the output verbosity level: 0 = No Output, 1 = Errors Only 2 = Warnings and Errors (default) 3 = Info, Warnings 9 = Debug (lots of output) Equivalent to the command line option: -v Example: `Verbosity=3`
VersionOnly	Displays Response Generator version. Do not use with other options. Equivalent to the command line option: -version
VersionPassthrough	Causes the GS08 for the created 997 to mirror the GS08 of the source document. Equivalent to the command line option: -fver_pass
WarnErrorSeverity	Set the error severity that will result in an 'Accepted but Errors were Noted' acknowledgement for the transaction set. Value is 0, 1, 2, 3, 4, 5, or 6; default is 2. Equivalent to the command line option: -ew Example: `WarnErrorSeverity=2`
WriteSTC12	When generating a 277CA, causes the DTL record's more descriptive error text (EMSG) to be written into the STC12 when A3:21 is encountered. WriteSTC12=true (1) means the feature is enabled. WriteSTC12=false (0) means the feature is disabled (default) Example: `WriteSC12="true"` Equivalent to the command line option: -fWriteSTC12

9 Appendix D: Z-Parameter Setup File

To invoke a Response Generator z-parameter setup file, use the command-line **z** option.

For an example setup file, see **ResponseGen_z.ini** in Instream's DemoData directory. The only options that can be in the file are in the table below. They have no equivalent command line options.

[824]	**Section containing items referring to an 824**
Display837ClaimID=*n*	In an 824, do you want an error to generate a RED01 with the claim ID (the CLM01) for each claim under it in the hierarchy, regardless of whether the error was inside the claim or above it? (See RED Segments on page 49 for more information). Display837ClaimID=0 no (default) Display837ClaimID=1 yes See ShowWhenErrorAbove2000 below for information about where the error must be located to trigger a RED01 in each underlying claim.
ShowWhenErrorAbove2000=*n*	Requires Display837ClaimID=1 This determines whether errors must be in the 2000 loop or below to cause a RED01 for each underlying claim: ShowWhenErrorAbove2000=0 (default) *(errors in the 2000 loop or below cause a RED01 for each underlying claim)* ShowWhenErrorAbove2000=1 *(errors in Table 1, Table 2 above the 2000 loop, or within the 2000 loop cause a RED01 for each underlying claim)*
UseDCNAsClaimID=*n*	In an 824, do you want every RED01 element to contain an original reference number (from the claim's REF02 when the REF01 is F8)? UseDCNAsClaimID=0 no (default) UseDCNAsClaimID=1 yes

10 Appendix E: Response Document Contents

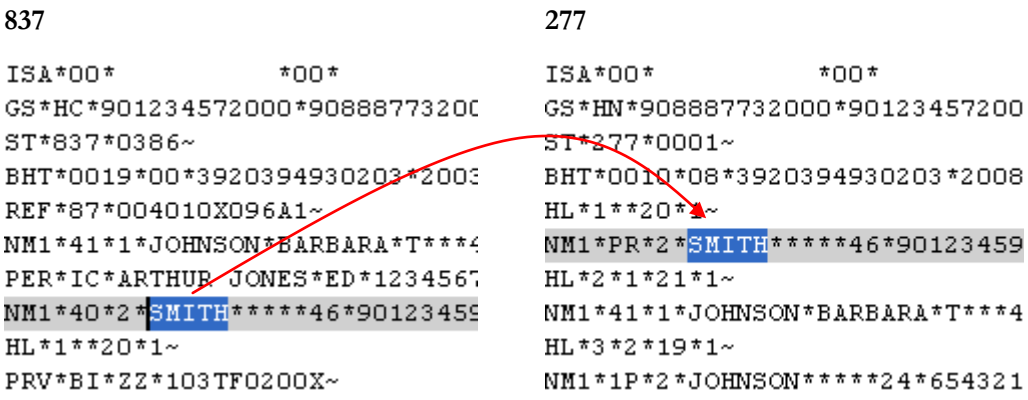
277CA Structure and Data Sources

Guideline 277CA-X214 shows the structure of the 277CA created by Response Generator. You can see this guideline in HIPAA Validator® Desktop's Library.

See also Appendix G: 277CA STC Override File on page 175.

277U Structure and Data Sources

This chart shows the structure of a 277U generated by Response Generator. The Source column shows the source of the values in the 277U. Many come from the originating 837:



This table describes the 277U created by Response Generator

Loop ID	Item	Name	Max Use	Base Req	Elem Type	Elem Length Min-Max	Source of Value	Notes
	ISA	**Interchange Control Header**	**1**	**O**				
	ISA01	Authorization Information Qualifier	1	M	ID	2-2	**00** no authorization information present	
	ISA02	Authorization Information	1	M	AN	10-10	Blank (ten spaces)	
	ISA03	Security Information Qualifier	1	M	ID	2-2	**00** no authorization information present	
	ISA04	Security Information	1	M	AN	10-10	Blank (ten spaces)	Command Line: -dis
	ISA05	Interchange ID Qualifier	1	M	ID	2-2	From initiating transaction's **ISA07**	Command Line: -dis
	ISA06	Interchange Sender ID	1	M	AN	15-15	From initiating transaction's **ISA08**	Command Line: -dir
	ISA07	Interchange ID Qualifier	1	M	ID	2-2	From initiating transaction's **ISA05**	Command Line: -dir
	ISA08	Interchange Receiver ID	1	M	AN	15-15	From initiating transaction's **ISA06**	
	ISA09	Interchange Date	1	M	DT	6-6	Current date in YYMMDD format	
	ISA10	Interchange Time	1	M	TM	4-4	Current time in HHMM format	
	ISA11	Repetition Separator	1	M	ID	1-1	**^** - Default	
	ISA12	Interchange Control Version Number	1	M	ID	5-5	**00501**	Command Line: -dic
	ISA13	Interchange Control Number	1	M	N0	9-9	Specified interchange control number, zero-filled, starting at "000000001", and increments by 1 for each Control Number	
	ISA14	Acknowledgment Requested	1	M	ID	1-1	**0** - No acknowledgment requested	Command Line: -diu
	ISA15	Usage Indicator	1	M	ID	1-1	**P** - Production Data	
	ISA16	Component Element Separator	1	M	AN	1-1	Sub-element separator character in effect (Default '**:**')	

	GS	**Functional Group Header**	**1**	**O**				
	GS01	Functional Identifier Code	1	M	ID	2-2	**HN**	
	GS02	Application Sender's Code	1	M	AN	2-15	From initiating transaction's GS03	Command Line: -das
	GS03	Application Receiver's Code	1	M	AN	2-15	From initiating transaction's GS02	Command Line: -dar
	GS04	Date	1	M	DT	8-8	Current date in YYYYMMDD format	
	GS05	Time	1	M	TM	4-8	Current time in HHMMSS format	
	GS06	Group Control Number	1	M	N0	1-9	Specified Group Control Number	
	GS07	Responsible Agency Code	1	M	ID	1-2	**X** Accredited Standards Committee X12	
	GS08	Version / Release / Industry Identifier Code	1	M	AN	1-12	See page 63.	Command Line: -dav
	ST	**Transaction Set Header**	**1**	**M**				
	ST01	Transaction Set Identifier Code	1	M	ID	3-3	**277**	
	ST02	Transaction Set Control Number	1	M	AN	4-9	Specified transaction control number, zero-filled, starting at "0001", and increments by 1 for each transaction control number.	
	ST03	Implementation Convention Reference	1	O	AN	1-35	See page 63.	
	BHT	**Beginning of Hierarchical Transaction**	**1**	**M**				
	BHT01	Hierarchical Structure Code	1	M	ID	4-4	**0010**	
	BHT02	Transaction Set Purpose Code	1	M	ID	2-2	**08**	
	BHT03	Reference Identification	1	O	AN	1-30	See ST02 definition	
	BHT04	Date	1	O	DT	8-8	Current date in YYYYMMDD format	
	BHT05	Time	1	O	TM	4-8	Not Used	
	BHT06	Transaction Type Code	1	O	ID	2-2	TH	

2000A	**HL**	**Information Source Level**		**M**				
	HL01	Hierarchical ID Number	1	M	AN	1-12	Generated internally by Response Generator	
	HL02	Hierarchical Parent ID Number	1	O	AN	1-12	Not used in 2000A (parent loop)	
	HL03	Hierarchical Level Code	1	M	ID	2-2	**20** (Information Source)	
	HL04	Hierarchical Child Code	1	M	ID	1-1	**1** (2000A loop has child nodes)	
2100A	**NM**	**Payer Name**		**M**				
	NM101	Entity Identifier Code	1	M	ID	2-3	**PR** (Payer)	
	NM102	Entity Type Qualifier	1	M	ID	1-1	**2** (Non-Person Entity)	
	NM103	Name (Last) or Organization Name	1	O	AN	1-35	837 Receiver NM103. Detail file ZRP record refers to NM1 segment passed along in 1000B.	
	NM108	Identification Code Qualifier	1	X	ID	1-2	837 Receiver NM108. Detail file ZRP record refers to NM1 segment passed along in 1000B.	
	NM109	Identification Code	1	X	AN	2-80	837 Receiver NM109. Detail file ZRP record refers to NM1 segment passed along in 1000B.	
2000B	**HL**	**Information Receiver Level**		**M**				
	HL01	Hierarchical ID Number	1	M	AN	1-12	Generated internally by Response Generator	
	HL02	Hierarchical Parent ID Number	1	O	AN	1-12	Generated internally by Response Generator	
	HL03	Hierarchical Level Code	1	M	ID	2-2	**21** (Information Receiver)	
	HL04	Hierarchical Child Code	1	M	ID	1-1	**1** (2000B loop has child nodes)	

2100B	HL	Receiver Name		M				
	NM101	Entity Identifier Code	1	M	ID	2-3	**41** (Submitter)	
	NM102	Entity Type Qualifier	1	M	ID	1-1	837 Submitter NM102. Detail file ZRT record refers to NM1 segment passed along in 1000A.	
	NM103	Name (Last) or Organization Name	1	O	AN	1-35	837 Submitter NM103. Detail file ZRT record refers to NM1 segment passed along in 1000A.	
	NM104	Name First	1	O	AN	1-25	837 Submitter NM104. Detail file ZRT record refers to NM1 segment passed along in 1000A.	
	NM105	Name Middle	1	O	AN	1-25	837 Submitter NM105. Detail file ZRT record refers to NM1 segment passed along in 1000A.	
	NM106	Name Prefix	1	O	AN	1-25	837 Submitter NM106. Detail file ZRT record refers to NM1 segment passed along in 1000A.	
	NM107	Name Suffix	1	O	AN	1-25	837 Submitter NM107. Detail file ZRT record refers to NM1 segment passed along in 1000A.	
	NM108	Identification Code Qualifier	1	X	ID	1-2	837 Submitter NM108. Detail file ZRT record refers to NM1 segment passed along in 1000A.	
	NM109	Identification Code	1	X	AN	2-80	837 Receiver NM109. Detail file ZRT record refers to NM1 segment passed along in 1000A.	
2000C	**HL**	**Service Provider Level**		**M**				
	HL01	Hierarchical ID Number	1	M	AN	1-12	Generated internally by Response Generator	
	HL02	Hierarchical Parent ID Number	1	O	AN	1-12	Generated internally by Response Generator	
	HL03	Hierarchical Level Code	1	M	ID	2-2	**19** (Provider of Service)	
	HL04	Hierarchical Child Code	1	M	ID	1-1	**1** (2000C loop has child nodes)	

2100C	NM1	Provider Name		M				
	NM101	Entity Identifier Code	1	M	ID	2-3	**1P** (Provider)	
	NM102	Entity Type Qualifier	1	M	ID	1-1	837 Provider NM102	See Provider values on page 158
	NM103	Name (Last) or Organization Name	1	O	AN	1-35	837 Provider NM103	
	NM104	Name First	1	O	AN	1-25	837 Provider NM104	
	NM105	Name Middle	1	O	AN	1-25	837 Provider NM105	
	NM107	Name Prefix	1	O	AN	1-25	837 Provider NM106	
	NM107	Name Suffix	1	O	AN	1-25	837 Provider NM107	
	NM108	Identification Code Qualifier	1	X	ID	1-2	837 Provider NM108	
	NM109	Identification Code	1	X	AN	2-80	837 Provider NM109	
2000D	**HL**	**Subscriber Level**		**M**				
	HL01	Hierarchical ID Number	1	M	AN	1-12	Generated internally by Response Generator	
	HL02	Hierarchical Parent ID Number	1	O	AN	1-12	Generated internally by Response Generator	
	HL03	Hierarchical Level Code	1	M	ID	2-2	**22** (Subscriber)	
	HL04	Hierarchical Child Code	1	M	ID	1-1	0 if no Dependent loop exists, 1 if one does exist	
	DMG	**Subscriber Demographic Information**	**1**	**O**				
	DMG01	Date Time Period Format Qualifier	1	X	ID	2-3	837 Subscriber DMG01. Detail file ZRSG record refers to DMG segment passed along in 2010BA.	Standard specifies **D8** for 277, but value is passed through from 837
	DMG02	Date Time Period	1	X	AN	1-35	837 Subscriber DMG02. Detail file ZRSG record refers to DMG segment passed along in 2010BA.	
	DMG03	Gender Code	1	O	ID	1-1	837 Subscriber DMG03. Detail file ZRSG record refers to DMG segment passed along in 2010BA.	

2100D	NM	Subscriber Name		M				
	NM101	Entity Identifier Code	1	M	ID	2-3	**IL** (Subscriber/Insured)	
	NM102	Entity Type Qualifier	1	M	ID	1-1	837 Subscriber NM102. Detail file 0003 record refers to NM1 segment passed along in 2010BA.	
	NM103	Name (Last) or Organization Name	1	O	AN	1-35	837 Subscriber NM103. Detail file 0003 record refers to NM1 segment passed along in 2010BA.	
	NM104	Name First	1	O	AN	1-25	837 Subscriber NM104. Detail file 0003 record refers to NM1 segment passed along in 2010BA.	
	NM105	Name Middle	1	O	AN	1-25	837 Subscriber NM105. Detail file 0003 record refers to NM1 segment passed along in 2010BA.	
	NM106	Name Prefix	1	O	AN	1-25	837 Subscriber NM106. Detail file 0003 record refers to NM1 segment passed along in 2010BA.	
	NM107	Name Suffix	1	O	AN	1-25	837 Subscriber NM107. Detail file 0003 record refers to NM1 segment passed along in 2010BA.	
	NM108	Identification Code Qualifier	1	X	ID	1-2	837 Subscriber NM108. Detail file 0003 record refers to NM1 segment passed along in 2010BA.	
	NM109	Identification Code	1	X	AN	2-80	837 Subscriber NM109. Detail file 0003 record refers to NM1 segment passed along in 2010BA.	
2200D	**TRN**	**Claim Submitter Trace Number**		**M**	***Claim information is currently written only if there are errors in either the claim itself, its parent loops, or its subordinate service lines.***			
	TRN01	Trace Type Code	1	M	ID	1-2	**2** (Referenced Transaction Trace Numbers)	
	TRN02	Reference Identification	1	M	AN	1-30	837 Subscriber Claim CLM01. Detail file S009 record refers to CLM segment passed along in Subscriber 2300.	

	STC	**Claim Level Status Information**	**1**	**M**				
	STC01.01	Industry Code	1	M	AN	1-30	A3	See 277U STC description on page 158
	STC01.02	Industry Code	1	M	AN	1-30	The STC code value from the DTL record. Default is 21.	
	STC01.03	Entity Identifier Code	1	O	ID	2-3	Not used	
	STC02	Date	1	O	DT	8-8	Current date in YYYYMMDD format	
	STC03	Not used.						
	STC04	Monetary Amount	1	0	R	1-10	837 Subscriber Claim CLM02. Detail file S009 record refers to CLM segment passed along in Subscriber 2300.	Submitted charges
	STC05	Monetary Amount	1	0	R	1-10	**0**	Paid charges
	STC06	Date	1	O	DT	8-8	Not used	Date of denial of charges
	STC07	Payment Method Code	1	O	ID	3-3	Not used	
	STC08	Date	1	O	DT	8-8	Not used	Date payment issued
	STC09	Check Number	1	O	AN	1-16	Not used	
	STC10.01	Industry Code	1	M	AN	1-30	A3	See 277U STC description on page 158
	STC10.02	Industry Code	1	M	AN	1-30	Generated by ResponseGen from 837 Subscriber Claim detail file error records. Used if a second claim status is needed.	
	STC10.03	Entity Identifier Code	1	O	ID	2-3	Not used	
	STC11.01	Industry Code	1	M	AN	1-30	A3	See 277U STC description on page 158
	STC11.02	Industry Code	1	M	AN	1-30	Generated by ResponseGen from 837 Subscriber Claim detail file error records. Used if a third claim status is needed.	
	STC11.03	Entity Identifier Code	1	O	ID	2-3	Not used	
	REF	**Payer Claim Identification Number**	**1**	**O**				
	REF01	Reference Identification Qualifier	1	M	ID	2-3	**1K** (Payor's claim number)	
	REF02	Reference Identification	1	X	AN	1-30	If 837 Subscriber Claim REF F8 exists, its REF02 is used, otherwise CLM01 (ClaimID) is used. Detail file record ZRSI refers to the Subscriber Claim REF F8 segment.	

	REF	**Institutional Bill Type Identification**	**1**	**O**				
	REF01	Reference Identification Qualifier	1	M	ID	2-3	**BLT** (Billing Type)	
	REF02	Reference Identification	1	X	AN	1-30	837 Subscriber Claim CLM05.1 and CLM05.3 concatenated together as a single value. Detail file record S009 refers to the Subscriber Claim CLM segment.	
	REF	**Medical Record Identification**	**1**	**O**				
	REF01	Reference Identification Qualifier	1	M	ID	2-3	**EA** (Medical Record Identification Number)	
	REF02	Reference Identification	1	X	AN	1-30	837 Subscriber Claim REF EA REF02. Detail file record ZRSM refers to the Subscriber Claim REF EA segment.	
	REF	**Claim ID Number for Clearinghouses and Other Transmission Intermediaries**	**1**	**O**				
	REF01	Reference Identification Qualifier	1	M	ID	2-3	**D9** (Claim Number)	
	REF02	Reference Identification	1	X	AN	1-30	REF02 where REF01=D9	
	DTP	**Claim Service Date**	**1**	**O**				
	DTP01	Date/Time Qualifier	1	M	ID	3-3	**232** (Claim Statement Period Start)	
	DTP02	Date Time Period Format Qualifier	1	M	ID	2-3	**RD8** (Range of dates expressed in YYYYMMDD-YYYYMMDD format)	
	DTP03	Date Time Period	1	M	AN	1-35	837 Subscriber Claim Service Date range from DTP 435. Detail file record S010 refers to the Subscriber Claim DTP 435 segment.	
2220D	**SVC**	**Service Line Information**		**O**	***Service line segments are written only if there is an error in the service line itself***			
	SVC01	Composite Medical Procedure Identifier	1	M	C003		Read from 837 Subscriber Claim SV1 or SV2 segments. Detail file record S011 refers to the Subscriber Claim SV1 or SV2 segment.	Value is identical to that in the 837 SV101 or SV202
	SVC02	Monetary Amount	1	M	R	1-10	Read from 837 Subscriber Claim SV102 or SV203. Detail file record S011 refers to the Subscriber Claim SV1 or SV2 segment.	Submitted charge.
	SVC03	Monetary Amount	1	O	R	1-10	**0**	Amount paid on service line item.

	SVC04	Product/Service ID	1	O	AN	1-48	Read from 837 Subscriber Claim SV1 or SV2 segments. Detail file record S011 refers to the Subscriber Claim SV1 or SV2 segment.	SV101.02 or SV202.02.
	SVC05	Not used						
	SVC06	Not used						
	SVC07	Quantity	1				Read from 837 Subscriber Claim SV1 or SV2 segments. Detail file record S011 refers to the Subscriber Claim SV1 or SV2 segment.	SV104 or SV205.
	STC	**Service Line Status Information**	**1**	**M**				
	STC01.01	Industry Code	1	M	AN	1-30	A3	See 277U STC description on page 158
	STC01.02	Industry Code	1	M	AN	1-30	Generated by ResponseGen from 837 Subscriber Claim detail file error records	
	STC01.03	Entity Identifier Code	1	O	ID	2-3	Not used	
	STC02	Date	1	O	DT	8-8	Current date in YYYYMMDD format	
	STC03	Not used.						
	STC04	Monetary Amount	1	0	R	1-10	Read from 837 Subscriber Claim SV102 or SV203. Detail file record S011 refers to the Subscriber Claim SV1 or SV2 segment.	Submitted charges. Identical to SVC02 listed above
	STC05	Monetary Amount	1	0	R	1-10	**0**	Amount paid on service line item
	STC06	Date	1	O	DT	8-8	Not used.	Date of denial of charges
	STC07	Payment Method Code	1	O	ID	3-3	Not used.	
	STC08	Date	1	O	DT	8-8	Not used.	Date payment issued
	STC09	Check Number	1	O	AN	1-16	Not used.	
	STC10.01	Industry Code	1	M	AN	1-30	A3	See 277U STC description on page 158
	STC10.02	Industry Code	1	M	AN	1-30	Generated by ResponseGen from 837 Subscriber Claim detail file error records. Used if a second claim status is needed.	
	STC10.03	Entity Identifier Code	1	O	ID	2-3	Not used	
	STC11.01	Industry Code	1	M	AN	1-30	A3	See 277U STC description on page 158
	STC11.02	Industry Code	1	M	AN	1-30	Generated by ResponseGen from 837 Subscriber Claim detail file error records. Used if a third claim status is needed.	
	STC11.03	Entity Identifier Code	1	O	ID	2-3	Not used	

	REF	**Service Line Item Identification**	**1**	**O**				
	REF01	Reference Identification Qualifier	1	M	ID	2-3	**FJ** (Line Item Control Number)	
	REF02	Reference Identification	1	X	AN	1-30	837 Subscriber Claim Service Line LX01 value. Detail file record 0007refers to this LX segment	
	DTP	**Service Line Date**	**1**	**O**				
	DTP01	Date/Time Qualifier	1	M	ID	3-3	**472** (Service)	
	DTP02	Date Time Period Format Qualifier	1	M	ID	2-3	**RD8** (Range of dates expressed in YYYYMMDD-YYYYMMDD format)	
	DTP03	Date Time Period	1	M	AN	1-35	837 Subscriber Claim Service Line Date range from DTP 472. Detail file record ZRSV refers to this DTP segment.	
2000E	**HL**	**Dependent Level**		**M**				
	HL01	Hierarchical ID Number	1	M	AN	1-12	Generated internally by Response Generator	
	HL02	Hierarchical Parent ID Number	1	O	AN	1-12	Generated internally by Response Generator	
	HL03	Hierarchical Level Code	1	M	ID	2-2	**23** (Dependent Patient)	
	HL04	Hierarchical Child Code	1	M	ID	1-1	Not used (No subordinate loops)	
	DMG	**Dependent Demographic Information**	**1**	**O**				
	DMG01	Date Time Period Format Qualifier	1	X	ID	2-3	837 Dependent DMG01. Detail file ZRDG record refers to DMG segment passed along in 2010BA.	Standard specifies **D8** for 277, but value is passed through from 837
	DMG02	Date Time Period	1	X	AN	1-35	837 Dependent DMG02. Detail file ZRDG record refers to DMG segment passed along in 2010CA.	
	DMG03	Gender Code	1	O	ID	1-1	837 Dependent DMG03. Detail file ZRDG record refers to DMG segment passed along in 2010CA.	

2100E	NM	Dependent Name		M				
	NM101	Entity Identifier Code	1	M	ID	2-3	**QC** (Dependent)	
	NM102	Entity Type Qualifier	1	M	ID	1-1	837 Dependent NM102. Detail file 0006 record refers to NM1 segment passed along in 2010BA.	
	NM103	Name (Last) or Organization Name	1	O	AN	1-35	837 Dependent NM103. Detail file 0006 record refers to NM1 segment passed along in 2010BA.	
	NM104	Name First	1	O	AN	1-25	837 Dependent NM104. Detail file 0006 record refers to NM1 segment passed along in 2010BA.	
	NM105	Name Middle	1	O	AN	1-25	837 Dependent NM105. Detail file 0006 record refers to NM1 segment passed along in 2010BA.	
	NM106	Name Prefix	1	O	AN	1-25	837 Dependent NM106. Detail file 0006 record refers to NM1 segment passed along in 2010BA.	
	NM107	Name Suffix	1	O	AN	1-25	837 Dependent NM107. Detail file 0006 record refers to NM1 segment passed along in 2010BA.	
	NM108	Identification Code Qualifier	1	X	ID	1-2	837 Dependent NM108. Detail file 0006 record refers to NM1 segment passed along in 2010BA.	
	NM109	Identification Code	1	X	AN	2-80	837 Dependent NM109. Detail file 0006 record refers to NM1 segment passed along in 2010BA.	
2200E	**TRN**	**Claim Submitter Trace Number**		**M**	***Claim information is currently written only if there are errors in either the claim itself or its subordinate service lines***			
	TRN01	Trace Type Code	1	M	ID	1-2	**2** (Referenced Transaction Trace Numbers)	
	TRN02	Reference Identification	1	M	AN	1-30	837 Dependent Claim CLM01. Detail file P009 record refers to CLM segment passed along in Dependent 2300.	

	STC	**Claim Level Status Information**	**1**	**M**				
	STC01	Health Care Claim Status	1	M	CO43	N/A	Generated by ResponseGen from 837 Dependent Claim detail file error records	See 277U STC description on page 158
	STC02	Date	1	O	DT	8-8	Current date in YYYYMMDD format	
	STC03	Not used.						
	STC04	Monetary Amount	1	0	R	1-10	837 Dependent Claim CLM02. Detail file P009 record refers to CLM segment passed along in Dependent 2300.	Submitted charges
	STC05	Monetary Amount	1	0	R	1-10	**0**	Paid charges
	STC06	Date	1	O	DT	8-8	Not used	Date of denial of charges
	STC07	Payment Method Code	1	O	ID	3-3	Not used	
	STC08	Date	1	O	DT	8-8	Not used	Date payment issued
	STC09	Check Number	1	O	AN	1-16	Not used	
	STC10	Health Care Claim Status	1	O	C043	N/A	Generated by ResponseGen from 837 Dependent Claim detail file error records. Used if a second claim status is needed.	See 277U STC description on page 158
	STC11	Health Care Claim Status	1	O	C043	N/A	Generated by ResponseGen from 837 Dependent Claim detail file error records. Used if a third claim status is needed.	See 277U STC description on page 158
	REF	**Payer Claim Identification Number**	**1**	**O**				
	REF01	Reference Identification Qualifier	1	M	ID	2-3	**1K** (Payor's claim number)	
	REF02	Reference Identification	1	X	AN	1-30	If 837 Dependent Claim REF F8 exists, its REF02 is used, otherwise CLM01 (ClaimID) is used. Detail file record ZRPI refers to the Dependent Claim REF F8 segment.	
	REF	**Institutional Bill Type Identification**	**1**	**O**				
	REF01	Reference Identification Qualifier	1	M	ID	2-3	**BLT** (Billing Type)	
	REF02	Reference Identification	1	X	AN	1-30	837 Dependent Claim CLM05.1 and CLM05.3 concatenated together as a single value. Detail file record P009 refers to the Dependent Claim CLM segment.	

	REF	**Medical Record Identification**	**1**	**O**				
	REF01	Reference Identification Qualifier	1	M	ID	2-3	**EA** (Medical Record Identification Number)	
	REF02	Reference Identification	1	X	AN	1-30	837 Dependent Claim REF EA REF02. Detail file record ZRPM refers to the Dependent Claim REF EA segment.	
	DTP	Claim Service Date	1	O				
	DTP01	Date/Time Qualifier	1	M	ID	3-3	**232** (Claim Statement Period Start)	
	DTP02	Date Time Period Format Qualifier	1	M	ID	2-3	**RD8** (Range of dates expressed in YYYYMMDD-YYYYMMDD format)	
	DTP03	Date Time Period	1	M	AN	1-35	837 Dependent Claim Service Date range from DTP 435. Detail file record P010 refers to the Dependent Claim DTP 435 segment.	
2220E	**SVC**	**Service Line Information**		**O**	***Service line segments are written only if there is an error in the service line itself***			
	SVC01	Composite Medical Procedure Identifier	1	M	C003		Read from 837 Dependent Claim SV1 or SV2 segments. Detail file record P011 refers to the Dependent Claim SV1 or SV2 segment.	Value is identical to that in the 837 SV101 or SV202
	SVC02	Monetary Amount	1	M	R	1-10	Read from 837 Dependent Claim SV102 or SV203. Detail file record P011 refers to the Dependent Claim SV1 or SV2 segment.	Submitted charge
	SVC03	Monetary Amount	1	O	R	1-10	**0**	Amount paid on service line item
	SVC04	Product/Service ID	1	O	AN	1-48	Read from 837 Dependent Claim SV1 or SV2 segments. Detail file record P011 refers to the Dependent Claim SV1 or SV2 segment.	SV101.02 or SV202.02
	SVC05	Not used						
	SVC06	Not used						
	SVC07	Quantity	1				Read from 837 Dependent Claim SV1 or SV2 segments. Detail file record P011 refers to the Dependent Claim SV1 or SV2 segment.	SV104 or SV205

	STC	**Service Line Status Information**	**1**	**M**				
	STC01	Health Care Claim Status	1	M	CO43	N/A	Generated by ResponseGen from 837 Dependent Claim detail file error records.	See 277U STC description on page 158
	STC02	Date	1	O	DT	8-8	Current date in YYYYMMDD format.	
	STC03	Not used.						
	STC04	Monetary Amount	1	0	R	1-10	Read from 837 Dependent Claim SV102 or SV203. Detail file record P011 refers to the Dependent Claim SV1 or SV2 segment.	Submitted charges. Identical to SVC02 listed above
	STC05	Monetary Amount	1	0	R	1-10	**0**	Amount paid on service line item
	STC06	Date	1	O	DT	8-8	Not used	Date of denial of charges
	STC07	Payment Method Code	1	O	ID	3-3	Not used	
	STC08	Date	1	O	DT	8-8	Not used	Date payment issued
	STC09	Check Number	1	O	AN	1-16	Not used	
	STC10	Health Care Claim Status	1	O	C043	N/A	Generated by ResponseGen from 837 Dependent Claim detail file error records. Used if a second claim status is needed	See 277U STC description on page 158
	STC11	Health Care Claim Status	1	O	C043	N/A	Generated by ResponseGen from 837 Dependent Claim detail file error records. Used if a third claim status is needed.	See 277U STC description on page 158
	REF	**Service Line Item Identification**	**1**	**O**				
	REF01	Reference Identification Qualifier	1	M	ID	2-3	**FJ** (Line Item Control Number)	
	REF02	Reference Identification	1	X	AN	1-30	837 Dependent Claim Service Line LX01 value. Detail file record 0005 refers to this LX segment.	
	DTP	**Service Line Date**	**1**	**O**				
	DTP01	Date/Time Qualifier	1	M	ID	3-3	**472** (Service)	
	DTP02	Date Time Period Format Qualifier	1	M	ID	2-3	**RD8** (Range of dates expressed in YYYYMMDD-YYYYMMDD format)	
	DTP03	Date Time Period	1	M	AN	1-35	837 Dependent Claim Service Line Date range from DTP 472. Detail file record ZRPV refers to this DTP segment.	
	SE	**Transaction Set Trailer**	**1**	**M**				
	SE01	Number of Included Segments	1	M	N0	1-10	Count of Segments in output transaction	
	SE02	Transaction Set Control Number	1	M	AN	4-9	Matches ST02 of output transaction	

	GE	**Functional Group Trailer**	**1**	**O**				
	GE01	Number of Transaction Sets Included	1	M	N0	1-6	Count of ST/SEs in output transaction	
	GE02	Group Control Number	1	M	N0	1-9	Matches GS06 of output transaction	
	IEA	**Interchange Control Trailer**	**1**	**O**				
	IEA01	Number of Included Functional Groups	1	M	N0	1-5	Count of GS/GEs in output transaction	
	IEA02	Interchange Control Number	1	M	N0	9-9	Matches ISA13 of output transaction	

Provider values

- If the Pay-To Provider (2010AB) ID (NM109) contains a value, its NM1 values are used.
- If not, the Bill-To Provider's (2010AA) NM1 values are used.
- The 2010AA and 2010AB NM1 values are in detail file records 0001 and 0002 respectively.

277U STC

- An error encountered during 837 validation may have an STC01.02 Health Care Claim Status code specified in the validation APF file. If so, the code is recorded in the validation detail file's DTL record and it is used in the 277U.
- Otherwise, the value "21" is assigned to the error.
- The STC01 composite element at the 277 2200D or 2200E level (Subscriber or Dependent Claim) is populated as follows:
- If errors were encountered in the 2300 level itself, the status codes associated with only the first three errors are added to the STC01/10/11 composites.
- If no errors were found in the 2300 level, any errors in the parent loop closest to the 2300 are reported. It then continues up the hierarchy until it finds three errors or finishes. The first three errors above it are used to generate the STC01/10/11 composites.
- When generating an STC segment at the 277 2220D or 2220E level (Service Line):
- Only errors encountered at the Service Line level (2400) are used to generate the STC.
- Errors from higher loops in the 2220D/E STCs are not considered.
- If user-defined status codes are provided in the validation APF file, these will be used. Otherwise, "21" is assigned to the error.
- The first subelement (Industry Code) for these composites will always be "A3".

See the **TIB_fsp-instream_ <*n.n*>_usersguide.pdf** and **APF.pdf** documents for details on how to specify user-defined status codes, as well as for details on how these codes are passed along through the Instream detail file.

997 Structure and Data Sources

This chart shows the structure of a 997 generated by Response Generator. The Source column shows the source of the values in the 997. Many come from the originating transaction:

837

```
ISA*00*          *00*
*030212*0848*U*00401*000000111
GS*HC*901234572000*90888773200
ST*837*0111~
BHT*0019*18*3920394930203*2003
REF*87*004010X096A1~
NM1*41*1*ALBRIGHT*AMELIA*T***
PER*IC*ARTHUR JONES*ED*1334567
NM1*40*2*SMITH*****46*9012345
HL*1**20*1~
```

997

```
ISA*00*          *00*
*090107*1701*U*00401*00000900
GS*FA*908887732000*9012345720
ST*997*0001~
AK1*HC*111~
AK2*837*0111~
AK3*DTP*76*2300*8~
AK4*2*1250*7*QQ~
AK4*3*1251*8*1230~
AK3*DTP*96*2300*8~
```

This table describes the 997 created by Response Generator

Loop ID	I tem	Name	Max Use	Base Req	Elem Type	Elem Length Min-Max	Source of Value	Notes
	ISA	**Interchange Control Header**	**1**	**O**				
	ISA01	Authorization Information Qualifier	1	M	ID	2-2	**00** no authorization information present	
	ISA02	Authorization Information	1	M	AN	10-10	Blank (ten spaces)	
	ISA03	Security Information Qualifier	1	M	ID	2-2	**00** no authorization information present	
	ISA04	Security Information	1	M	AN	10-10	Blank (ten spaces)	Command Line: -dis
	ISA05	Interchange ID Qualifier	1	M	ID	2-2	From initiating transaction's ISA07	Command Line: -dis
	ISA06	Interchange Sender ID	1	M	AN	15-15	From initiating transaction's ISA08	Command Line: -dir
	ISA07	Interchange ID Qualifier	1	M	ID	2-2	From initiating transaction's ISA05	Command Line: -dir
	ISA08	Interchange Receiver ID	1	M	AN	15-15	From initiating transaction's ISA06	
	ISA09	Interchange Date	1	M	DT	6-6	Current date in YYMMDD format	
	ISA10	Interchange Time	1	M	TM	4-4	Current time in HHMM format	
	ISA11	Interchange Control Standards Identifier	1	M	ID	1-1	**U** US EDI Community of ASC X12, TDCC, and UCS	

Loop ID	I tem	Name	Max Use	Base Req	Elem Type	Elem Length Min-Max	Source of Value	Notes
	ISA12	Interchange Control Version Number	1	M	ID	5-5	**00401**	Command Line: -dic
	ISA13	Interchange Control Number	1	M	N0	9-9	Specified interchange control number, zero-filled, starting at “000000001”, and increments by 1 for each Control Number	
	ISA14	Acknowledgment Requested	1	M	ID	1-1	**0** No acknowledgment requested	Command Line: -diu
	ISA15	Usage Indicator	1	M	ID	1-1	**P** production Data	
	ISA16	Component Element Separator	1	M	AN	1-1	Sub-element separator character in effect (Default ‘:’)	
	TA1	**Interchange Acknowledgment**	**1**	**O**				**Command Line: -gTA1**
	TA101	Interchange Control Number	1	M	N0	9-9	From initiating transaction’s ISA13	
	TA102	Interchange Date	1	M	DT	6-6	Current date in YYMMDD format	
	TA103	Interchange Time	1	M	TM	4-4	Current time in HHMM format	
	TA104	Interchange Acknowledgment Code	1	M	ID	1-1	**A** or **R**	
	TA105	Interchange Note Code	1	M	ID	3-3	Codes Generated by Response Generator based on encountered Errors	
	GS	**Functional Group Header**	**1**	**O**				
	GS01	Functional Identifier Code	1	M	ID	2-2	**FA** functional acknowledgment	
	GS02	Application Sender's Code	1	M	AN	2-15	From initiating transaction’s GS03	Command Line: -das
	GS03	Application Receiver's Code	1	M	AN	2-15	From initiating transaction’s GS02	Command Line: -dar
	GS04	Date	1	M	DT	8-8	Current date in YYYYMMDD format	
	GS05	Time	1	M	TM	4-8	Current time in HHMMSS format	
	GS06	Group Control Number	1	M	N0	1-9	Specified Group Control Number	
	GS07	Responsible Agency Code	1	M	ID	1-2	**X** Accredited Standards Committee X12	
	GS08	Version / Release / Industry Identifier Code	1	M	AN	1-12	See page 63.	Command Line: -dav

Loop ID	I tem	Name	Max Use	Base Req	Elem Type	Elem Length Min-Max	Source of Value	Notes
	ST	**Transaction Set Header**	**1**	**M**				
	ST01	Transaction Set Identifier Code	1	M	ID	3-3	**997**	
	ST02	Transaction Set Control Number	1	M	AN	4-9	Specified transaction control number, zero-filled, starting at "0001", and increments by 1 for each transaction control number	
	AK1	**Functional Group Response Header**	**1**	**M**				
	AK101	Functional Identifier Code	1	M	ID	2-2	From initiating transaction's ST01	
	AK102	Group Control Number	1	M	N0	1-9	From initiating transaction's ST02	
AK2	**AK2**	**Transaction Set Response Header**	**1**	**O**				
AK2	AK201	Transaction Set Identifier Code	1	M	ID	3-3	From initiating transaction's ST01	
AK2	AK202	Transaction Set Control Number	1	M	AN	4-9	From initiating transaction's ST02	
AK3	**AK3**	**Data Segment Note**	**1**	**O**				
AK3	AK301	Segment ID Code	1	M	ID	2-3	Mapped from DTL record; reporting "Segment ID" of where error is reported	
AK3	AK302	Segment Position in Transaction Set	1	M	N0	1-6	Segment count offset from last ST to error segment	
AK3	AK303	Loop Identifier Code	1	O	AN	1-6	Mapped from DTL record; reporting "Loop" of where error is reported	
AK3	AK304	Segment Syntax Error Code	1	O	ID	1-3	Mapped from DTL record; reporting "Syntax Error Code" of where error is reported	Error Profile (*.apf) configurable
AK3	**AK4**	**Data Element Note**	**99**	**O**				
AK3	AK401.1	Element Position in Segment	1	M	N0	1-2	Mapped from DTL record; reporting "Element Position" of where error is reported	
AK3	AK402.1	Component Data Element Position in Composite	1	O	N0	1-2	Mapped from DTL record; reporting "Sub-Element Position" of where error is reported	
AK3	AK402	Data Element Reference Number	1	O	N0	1-4	Mapped from DTL record; reporting "Element ID" of where error is reported	
AK3	AK403	Data Element Syntax Error Code	1	M	ID	1-3	Mapped from DTL record; reporting "Data Element Syntax Error Code" of where error is reported	Error Profile (*.apf) configurable
AK3	AK404	Copy of Bad Data Element	1	O	AN	1-99	Mapped from DTL record; reporting "Element Data" of where error is reported	

Loop ID	I tem	Name	Max Use	Base Req	Elem Type	Elem Length Min-Max	Source of Value	Notes
AK2	**AK5**	**Transaction Set Response Trailer**	**1**	**M**				
AK2	AK501	Transaction Set Acknowledgment Code	1	M	ID	1-1	**A, E, P** or **R** based on configuration options	Command Line: -pt -LenientAK501
AK2	AK502	Transaction Set Syntax Error Code	1	O	ID	1-3	Set to **1** if an 11003 (Set Not Supported) error is encountered, otherwise empty	
AK2	AK503	Transaction Set Syntax Error Code	1	O	ID	1-3	Set to **1** if an 11208 (Set Trailer Not Found) error is encountered, otherwise empty	
AK2	AK504	Transaction Set Syntax Error Code	1	O	ID	1-3	Set to **1** if a 10913 (Set Control Number Mismatch) error is encountered, otherwise empty	
AK2	AK505	Transaction Set Syntax Error Code	1	O	ID	1-3	Set to **1** if a 10917 (Bad Segment Count) error is encountered, otherwise empty	
AK2	AK506	Transaction Set Syntax Error Code	1	O	ID	1-3	Set to **5** if any error occurs in the set	
	AK9	**Functional Group Response Trailer**	**1**	**M**				
	AK901	Functional Group Acknowledge Code	1	M	ID	1-1	**A**, **P** or **R** based on configuration options	Command Line: -np
	AK902	Number of Transaction Sets Included	1	M	N0	1-6	Number of Transaction Sets Included	
	AK903	Number of Received Transaction Sets	1	M	N0	1-6	Number of Received Transaction Sets	
	AK904	Number of Accepted Transaction Sets	1	M	N0	1-6	Number of Accepted Transaction Sets	
	AK905	Functional Group Syntax Error Code	1	O	ID	1-3	Generated by Response Generator based on specific error codes encountered at the Group level. Possible values are 2, 3, 4, 5, 14, 19, 21, 24.	
	AK906	Functional Group Syntax Error Code	1	O	ID	1-3	Not used	
	AK907	Functional Group Syntax Error Code	1	O	ID	1-3	Not used	
	AK908	Functional Group Syntax Error Code	1	O	ID	1-3	Not used	
	AK909	Functional Group Syntax Error Code	1	O	ID	1-3	Not used	

Loop ID	I tem	Name	Max Use	Base Req	Elem Type	Elem Length Min-Max	Source of Value	Notes
	SE	**Transaction Set Trailer**	**1**	**M**				
	SE01	Number of Included Segments	1	M	N0	1-10	Count of Segments in output transaction	
	SE02	Transaction Set Control Number	1	M	AN	4-9	Matches ST02 of output transaction	
	GE	**Functional Group Trailer**	**1**	**O**				
	GE01	Number of Transaction Sets Included	1	M	N0	1-6	Count of ST/SEs in output transaction	
	GE02	Group Control Number	1	M	N0	1-9	Matches GS06 of output transaction	
	IEA	**Interchange Control Trailer**	**1**	**O**				
	IEA01	Number of Included Functional Groups	1	M	N0	1-5	Count of GS/GEs in output transaction	
	IEA02	Interchange Control Number	1	M	N0	9-9	Matches ISA13 of output transaction	

999 Structure and Data Sources

If the input data contains Type 1, Type 2, or implementation errors, these will be reported in the IK3 (segment errors) and IK4 (element errors).

The 5010 Errata 999 may contain CTX records under certain conditions. See Appendix H: CTX Segments in Response Generator 999s on page 183.

Response Generator's 999 might include:

IK5 Transaction set acceptance/rejection and why (types 1-2)

AK9 Functional group acceptance/rejection and why (types 1-2)

IK3 Segment with a type 1-8 error.

IK4 Element containing the IK3 error (if the error involves an element).

This example is a typical 999 generated by Response Generator.

```
ISA*00*          *00*          *01*9088877320000  *01*9012345720000
*090706*1448*^*00501*000000001*0*P*:~
GS*FA*908887732000*901234572000*20090706*144828*1*X*005010X231~
ST*999*0001*005010X231~
AK1*HC*370~
AK2*837*0370~
IK3*N4*9*2010AA*8~
IK4*1*19*7~
IK4*4*26*I12~
IK3*CLM*31*2400*I6~
IK3*NM1*36*2310A*8~
IK4*3*1035*7~
IK3*HL*40*2400*I6~
```

CTX: Context Record Segments in 999s

In previous releases, Response Generator created a CTX segment in the 999 if the IK304 and/or IK403 started with "I" for Implementation syntax errors. Now, CTX generation is under your control. You can choose:

- No CTX segments
- TIBCO Foresight-supplied CTX segments
- Your own custom CTX segments

See Appendix H: CTX Segments in Response Generator 999s on page 183.

This table describes the 999 created by Response Generator

Loop ID	I tem	Name	Max Use	Base Req	Elem Type	Elem Len. Min-Max	Source of Value	Alternate configuration
	ISA	**Interchange Control Header**	**1**	**O**				
	ISA01	Authorization Information Qualifier	1	M	ID	2-2	00 no authorization information present	
	ISA01	Authorization Information Qualifier	1	M	ID	2-2	00 no authorization information present	
	ISA02	Authorization Information	1	M	AN	10-10	Blank (ten spaces)	
	ISA03	Security Information Qualifier	1	M	ID	2-2	00 no authorization information present	
	ISA04	Security Information	1	M	AN	10-10	Blank (ten spaces)	dis
	ISA05	Interchange ID Qualifier	1	M	ID	2-2	From initiating transaction's ISA07	dis
	ISA06	Interchange Sender ID	1	M	AN	15-15	From initiating transaction's ISA08	dir
	ISA07	Interchange ID Qualifier	1	M	ID	2-2	From initiating transaction's ISA05	dir
	ISA08	Interchange Receiver ID	1	M	AN	15-15	From initiating transaction's ISA06	
	ISA09	Interchange Date	1	M	DT	6-6	Current date in YYMMDD format	
	ISA10	Interchange Time	1	M	TM	4-4	Current time in HHMM format	
	ISA11	Interchange Control Standards Identifier	1	M	ID	1-1	U US EDI Community of ASC X12, TDCC, and UCS	
	ISA12	Interchange Control Version Number	1	M	ID	5-5	00501	
	ISA13	Interchange Control Number	1	M	N0	9-9	Specified interchange control number, zero-filled, starting at "000000001", and increments by 1 for each Control Number	
	ISA14	Acknowledgment Requested	1	M	ID	1-1	0 No acknowledgment requested	diu
	ISA15	Usage Indicator	1	M	ID	1-1	P production Data	
	ISA16	Component Element Separator	1	M	AN	1-1	Sub-element separator character in effect (Default ':')	
	GS	**Functional Group Header**	**1**	**O**				
	GS01	Functional Identifier Code	1	M	ID	2-2	FA functional acknowledgment	
	GS02	Application Sender's Code	1	M	AN	2-15	From initiating transaction's GS03	das
	GS03	Application Receiver's Code	1	M	AN	2-15	From initiating transaction's GS02	dar
	GS04	Date	1	M	DT	8-8	Current date in YYYYMMDD format	

Loop ID	I tem	Name	Max Use	Base Req	Elem Type	Elem Len. Min-Max	Source of Value	Alternate configuration
	GS05	Time	1	M	TM	4-8	Current time in HHMMSS format	
	GS06	Group Control Number	1	M	N0	1-9	Specified Group Control Number	
	GS07	Responsible Agency Code	1	M	ID	1-2	X Accredited Standards Committee X12	
	GS08	Version / Release / Industry Identifier Code	1	M	AN	1-12	See page 63.	dav
	ST	**Transaction Set Header**	**1**	**M**				
	ST01	Transaction Set Identifier Code	1	M	ID	3-3	999	
	ST02	Transaction Set Control Number	1	M	AN	4-9	Specified transaction control number, zero-filled, starting at "0001", and increments by 1 for each transaction control number	
	ST03	Implementation Convention Reference	1	M	AN	1-35		
	AK1	**Functional Group Response Header**	**1**	**M**				
	AK101	Functional Identifier Code	1	M	ID	2-2	From initiating transaction's GS01	
	AK102	Group Control Number	1	M	N0	1-9	From initiating transaction's GS06	
	AK103	Vers/Rel/Ind ID Code	1	M	AN	1-12	From initiating transaction's GS08	
AK2	**AK2**	**Transaction Set Response Header**	**1**	**O**				
AK2	AK201	Transaction Set Identifier Code	1	M	ID	3-3	From initiating transaction's ST01	
AK2	AK202	Transaction Set Control Number	1	M	AN	4-9	From initiating transaction's ST02	
AK2	AK203	Implementation Convention Reference	1	O	AN 1-35		From initiating transaction's ST03	
IK3	**IK3**	**Data Segment Note**	**1**	**O**				
IK3	IK301	Segment ID Code	1	M	ID	2-3	Mapped from DTL record; reporting "Segment ID" of where error is reported	
IK3	IK302	Segment Position in Transaction Set	1	M	N0	1-6	Segment count offset from last ST to error segment	
IK3	IK303	Loop Identifier Code	1	O	AN	1-4	Mapped from DTL record; reporting "Loop" of where error is reported	
IK3	IK304	Segment Syntax Error Code	1	O	ID	1-3	Mapped from DTL record; reporting "Syntax Error Code" of where error is reported	Error Profile (*.apf) configurable

Loop ID	I tem	Name	Max Use	Base Req	Elem Type	Elem Len. Min-Max	Source of Value	Alternate configuration
IK3	**CTX**	**Segment Context**	**10**	**S**			**Required when the IK3 above it was triggered by a situational business rule.**	**Only issued on error of WEDI type 4 (Situational)**
IK3	CTX01.1	Context Name	1	M	AN	1-35	SITUATIONAL TRIGGER	
IK3	CTX01.2	Context Reference	1	O	ID	2-3	Not used	
IK3	CTX02	Segment ID Code	1	O	ID	2-3	Mapped from DTL record; reporting "Segment ID" of first segment in the situational relationship	
IK3	CTX03	Segment Position in Transaction Set	1	O	N0	1-6	Segment count offset from last ST to first segment in the situational relationship	
IK3	CTX04	Loop Identifier Code	1	O	AN	1-6	Mapped from DTL record; reporting "Loop" of first segment in the situational relationship	
IK3	CTX05.1	Element Position in Segment	1	M	N0	1-2	Mapped from DTL record; reporting "Element Position" in first segment in the situational relationship	
IK3	CTX05.2	Component Data Element Position in Composite	1	O	N0	1-2	Mapped from DTL record; reporting "Sub-Element Position" within first segment in the situational relationship	
IK3	CTX05.3	Repeating Data Element Position	1	O	N0	1-4	Not used	
IK3	CTX06.1	Reference in Segment	1	O	N0	1-4	Reference number of first segment in the situational relationship	Mandatory if CTX06 is used.
CTX	CTX05.2	Reference in Composite	1	O	N0	1-4	Reference number of an element in a composite	
IK3	**CTX**	**Business Unit Identifier**	**1**	**S**			**Required when the IK3 above it was triggered by a situational business rule and the business unit identifier is known by RespGen.**	
CTX	CTX01.1	Context Identification	1	M	AN	1-35	Name or tag of value below.	See note below.
CTX	CTX01.2	Context Reference	1	O	AN	1-35	Identifier value, generated from various entries from specific loops in the source HIPAA documents	
IK4	**IK4**	**Data Element Note**	**99**	**O**				
IK4	IK401.1	Element Position in Segment	1	M	N0	1-2	Mapped from DTL record; reporting "Element Position" of where error is reported	
IK4	IK402.1	Component Data Element Position in Composite	1	O	N0	1-2	Mapped from DTL record; reporting "Sub-Element Position" of where error is reported	
IK4	IK402	Data Element Reference Number	1	O	N0	1-4	Mapped from DTL record; reporting "Element ID" of where error is reported	

Loop ID	I tem	Name	Max Use	Base Req	Elem Type	Elem Len. Min-Max	Source of Value	Alternate configuration
IK4	IK403	Data Element Syntax Error Code	1	M	ID	1-3	Mapped from DTL record; reporting "Data Element Syntax Error Code" of where error is reported	Error Profile (*.apf) configurable
IK4	IK404	Copy of Bad Data Element	1	O	AN	1-99	Mapped from DTL record; reporting "Element Data" of where error is reported	
IK4	**CTX**	**Element Context**	**10**	**S**			**Required when the IK4 above it was triggered by a situational business rule.**	**Only issued on error of WEDI type 4 (Situational)**
IK4	CTX01.1	Context Name	1	M	AN	1-35	SITUATIONAL TRIGGER	
IK4	CTX01.2	Context Reference	1	O	ID	2-3	Not used	
IK4	CTX02	Segment ID Code	1	O	ID	2-3	Mapped from DTL record; reporting "Segment ID" of first segment in the situational relationship	
IK4	CTX03	Segment Position in Transaction Set	1	O	N0	1-6	Segment count offset from last ST to first segment in the situational relationship	
IK4	CTX04	Loop Identifier Code	1	O	AN	1-6	Mapped from DTL record; reporting "Loop" containing first segment in the situational relationship	
IK4	CTX05.1	Element Position in Segment	1	M	N0	1-2	Mapped from DTL record; reporting "Element Position" within first segment in the situational relationship	
IK4	CTX05.2	Component Data Element Position in Composite	1	O	N0	1-2	Mapped from DTL record; reporting "Sub-Element Position" within first segment in the situational relationship	
IK4	CTX05.3	Repeating Data Element Position	1	O	N0	1-4	Not used.	
IK4	CTX06.1	Reference in Segment	1	O	N0	1-4	Reference number of first segment in the situational relationship	Mandatory if CTX06 is used.
IK4	CTX06.2	Reference in Composite	1	O	N0	1-4	Reference number of an element in a composite.	
AK2	**IK5**	**Transaction Set Response Trailer**	**1**	**M**				
AK2	IK501	Transaction Set Acknowledgment Code	1	M	ID	1-1	A, E, P or R based on configuration options	pt, LenientIK501
AK2	IK502	Transaction Set Syntax Error Code	1	O	ID	1-3	Mapped from DTL record; reporting "Element Position" of where error is reported	
AK2	IK503	Transaction Set Syntax Error Code	1	O	ID	1-3	Mapped from DTL record; reporting "Sub-Element Position" of where error is reported	
AK2	IK504	Transaction Set Syntax Error Code	1	O	ID	1-3	Mapped from DTL record; reporting "Element ID" of where error is reported	

Loop ID	I tem	Name	Max Use	Base Req	Elem Type	Elem Len. Min-Max	Source of Value	Alternate configuration
AK2	IK505	Transaction Set Syntax Error Code	1	O	ID	1-3	Mapped from DTL record; reporting “Data Element Syntax Error Code” of where error is reported	
AK2	IK506	Transaction Set Syntax Error Code	1	O	ID	1-3	Mapped from DTL record; reporting “Element Data” of where error is reported	
	AK9	**Functional Group Response Trailer**	**1**	**M**				
	AK901	Functional Group Acknowledge Code	1	M	ID	1-1	A, P or R based on configuration options	np
	AK902	Number of Transaction Sets Included	1	M	N0	1-6	Number of Transaction Sets Included	
	AK903	Number of Received Transaction Sets	1	M	N0	1-6	Number of Received Transaction Sets	
	AK904	Number of Accepted Transaction Sets	1	M	N0	1-6	Number of Accepted Transaction Sets	
	AK905	Functional Group Syntax Error Code	1	O	ID	1-3	Generated by Response Generator based on specific error codes encountered at the Group level. Possible values are 2, 3, 4, 5, 14, 21, 24.	
	AK906	Functional Group Syntax Error Code	1	O	ID	1-3	Not used.	
	AK907	Functional Group Syntax Error Code	1	O	ID	1-3	Not used.	
	AK908	Functional Group Syntax Error Code	1	O	ID	1-3	Not used.	
	AK909	Functional Group Syntax Error Code	1	O	ID	1-3	Not used.	
	SE	**Transaction Set Trailer**	**1**	**M**				
	SE01	Number of Included Segments	1	M	N0	1-10	Count of Segments in output transaction	
	SE02	Transaction Set Control Number	1	M	AN	4-9	Matches ST02 of output transaction	
	GE	**Functional Group Trailer**	**1**	**O**				
	GE01	Number of Transaction Sets Included	1	M	N0	1-6	Count of ST/SEs in output transaction	
	GE02	Group Control Number	1	M	N0	1-9	Matches GS06 of output transaction	

Loop ID	I tem	Name	Max Use	Base Req	Elem Type	Elem Len. Min-Max	Source of Value	Alternate configuration
	IEA	**Interchange Control Trailer**	**1**	**O**				
	IEA01	Number of Included Functional Groups	1	M	N0	1-5	Count of GS/GEs in output transaction	
	IEA02	Interchange Control Number	1	M	N0	9-9	Matches ISA13 of output transaction	

Source of CTX-01-01 Business Unit Identifier Source

Transaction	Value in CTX-01-01
269	TRN02
270	TRN02
271	TRN02
274	NM109
275	PATIENT NAME NM109
276	TRN02
277	TRN02
278	SUBSCRIBER NAME NM109
820	ENT01
834	SUBSCRIBER NUMBER REF02
835	TRN02
837	CLM01

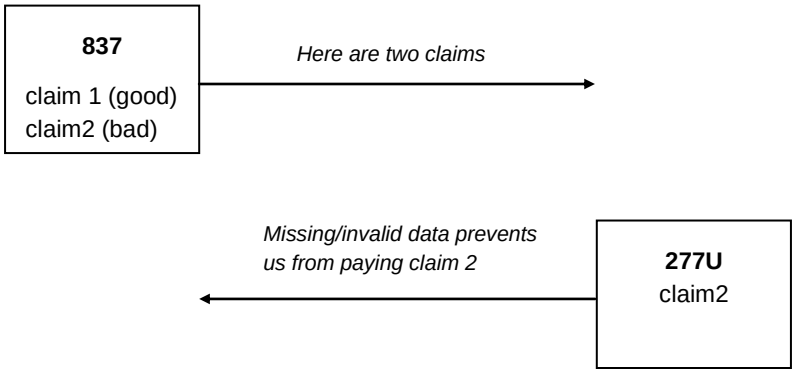
11 Appendix F: 277 Comparison

Comparison Chart

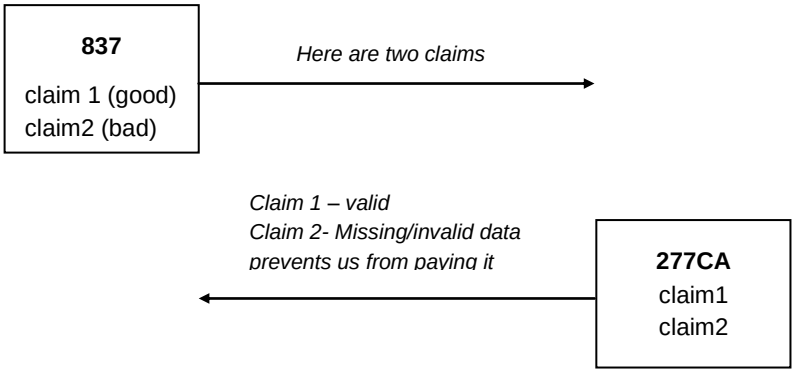
	277S	277U	277CA	277X167	277X070
Purpose	Respond to 276 inquiry of claim status Includes any claims specified in the 276	Response to 837 Reports status of invalid claims (-fsa command line parameter will include good and bad claims)	Response to 837 Reports status of all claims	Response to 837 Reports status of all claims	Response to 837 Reports status of all claims
HIPAA Implementation guide	Mandated X212_276_277.pdf	Not mandated No HIPAA IG	Not mandated, but strongly encouraged X214_277CA.pdf		
X12 versions	4010 and 5010	4010 and 5010	5010	4040+	3070
GS08 value	004010X093A1 005010X212	004010X093A1 005010X212	005010X214	004040X167	003070X070A1
Validated by Instream	Yes	Yes	Yes		
Created by Response Generator	No	Yes – 4010 and 5010 -o277 with no -fca	Yes – 5010 -o277 with -fca277	Yes - 4040 -o277 with -fdo277X167	Yes - 3070 -o277 with -fdo277x070
TIBCO Foresight guidelines (ForesightHIPAA guidelineList.pdf)	All types	All types	All types		
Maximum number of errors Resp. Gen reports per claim	n/a Response Gen does not create it	Up to 3 errors per claim	All errors in each claim but does not repeat error number within claim	All errors in each claim	Up to 3 errors per claim

277 Diagrams

277U

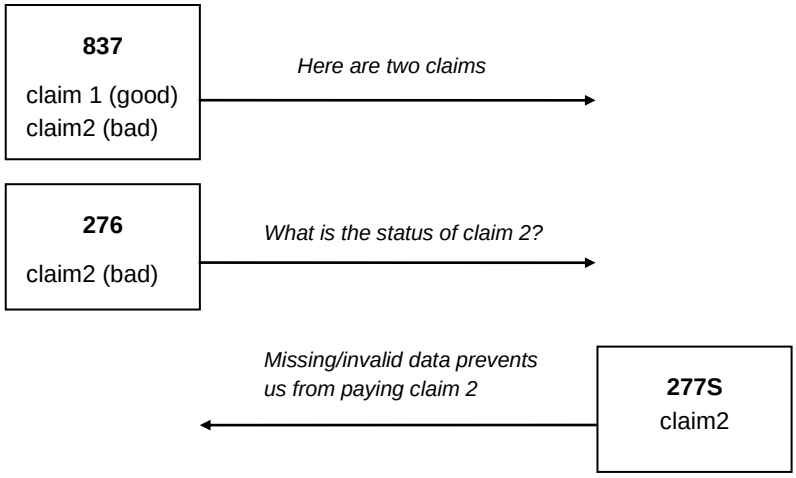


277CA



277S

Response Generator does not create a 277S. This diagram shows how your back end application might handle it.



12 Appendix G: 277CA STC Override File

Default 277CA STC Behavior

The 277CA created by Response Generator contains STC segments as follows:

- Each error is reported in a STC segment
- The STC01.01 (category code) will contain A3
- The STC01.02 (status code) is controlled by the Warning Levels section of the APF
- The STC10 and STC11 can contain information about another error at the same level

Changing STC Behavior

Depending on the error number, different and sometimes multiple STC codes are needed in the 277CA.

You can create an STC override file containing error numbers and other information for Response Generator to use when creating STC segments and then invoke by:

- the command line **-stc** parameter, or
- the STCTablePath setup file option when using partner automation.

NOTE: Adding an error number to the STC override file automatically flags it as an "error" in Response Generator. Therefore, warning messages will appear in 277CAs created by Response Generator even when they are not included in other responses.

Verifying Results

There is no confirmation in the detail file that an STC override file has been used. To verify your results, manually compare the results against the STC override file.

STC Override File Format

See SampleSTCOverride.csv in Instream's Bin directory.

The override file is a comma delimited file containing error numbers and information for Response Generator to use when creating STC segments.

When omitting values in a line, include the commas to maintain correct positioning.

Each line contains these, in order:

Error#	Error number that should have new STC behavior. To create multiple STCs for an error, add additional lines in the CSV file starting with the error number.
LoopID	Apply the STC override to this Loop ID. If this value is omitted, the override will apply to ALL Loop IDs.
Segment	Future use. Currently, the STC override will apply to the error number in all segments. If you will want to have the override apply to a certain segment only, you can fill out the segment tag now to save yourself future effort.
Element	Future use. Currently, the STC override will apply to the error number in any element. If you will want to have the override apply to a certain element only, you can fill out the element's location in the segment now to save yourself future effort. (Example: 2 refers to the second element in the segment.)
CategoryCode	Supply the value to be used for the STC01.01. If omitted, this element will be empty in the 277CA.
StatusCode	Supply the value to be used for the STC01.02. If omitted, this element will be empty in the 277CA.
EntityIdentifierCode	Supply the value to be used for the STC01.03. If omitted, this element will be empty in the 277CA.
InsertWithShuffler	This is ignored by Response Generator. It is used by the Shuffler component provided with TIBCO Foresight® Instream® MAC Adaptor; users should refer to Instream® MAC Adaptor documentation for information.

13 Appendix H: Response Generator Overrides File

Overview

NOTE: Response Generator Overrides for CTX segments in 999s have a different format and usage. Refer to Appendix H: CTX Segments in Response Generator 999s on page 183 for complete information.

You can use Response Generator Overrides to substitute a custom value when creating certain segments in the detail file. This allows you to generate response documents with more customized information about errors encountered in the data.

For example:

- You have created a set of customized errors specifically for your trading partners and you want your custom error number inserted instead of the TIBCO Foresight-supplied default value.
- You want a specific code value inserted when a certain error number is encountered in a specific location.

Segments that can be Overridden

In the specified transaction sets, you can override segments as shown here.

In this transaction set…	You can override this segment…
824 – Application Advice	RED (Related Data) Element RED06 only
997 – Functional Acknowledgement	AK9 (Functional Group Response Trailer) (con't)

In this transaction set…	You can override this segment…
999 – Implementation Acknowledgement	AK9 (Functional Group Response Trailer)
	CTX (Context) **Note:** Refer Appendix H: CTX Segments in Response Generator 999s for complete information on CTX usage.
	IK3 (Implementation Data Segment Note)
	IK4 (Implementation Data Element Note)
	IK5 (Implementation Transaction Set Response Trailer)
TA1	TA1 (Interchange Acknowledgement)

Invoking Overrides

Overrides are invoked when Response Generator encounters a section within a file that begins with the header [RespGen Overrides]. There are two ways to do this:

- add RespGen Overrides to your Profile (.apf file)
- create a separate RespGen Overrides file.

Add RespGen Overrides to the APF

1. Copy $fsdeflt.apf, located in Instream's Bin directory, to a new filename.
2. Open the new .apf file.

   Find the [RespGen Overrides] section, which is provided by default for generation of CTX segments for a 999 (See Appendix H: CTX Segments in Response Generator 999s on page 183).

   ```
   [RespGen Overrides]
   CTX00001=CTX,41233,SBR,2000,2,,,1069
   CTX00002=CTX,41657,SBR,2000,2,,,1069
   CTX00003=CTX,41236,NM1,1000,2,,,1065
   ```

3. Edit the new .apf file to include additional overrides.
4. Save your changes.
5. Specify this profile with Response Generator's **-apf** command.

   For example:

   ```
   RespGen -i "HV_Results.txt" -o997 "EDI_997.txt"
   -apf "C:\profiles\custom.apf"
   ```

Make a Separate RespGen Overrides File

1. Make your own RespGen Overrides file in .csv format. Be sure the file begins with `[RespGen Overrides]`.

   For example:

   ```
   [RespGen Overrides]
   ```

2. Edit the new .csv file to include overrides as desired.

   For example:

   ```
   [RespGen Overrides]
   02004=IK5,11409,,ST,2,23
   02005=IK5,ANY,,ST,2,7
   02006=IK5,ANY,,ST,3,I6
   03011=TA1,ANY,,GS,7,024
   03012=TA1,ANY,,IEA,1,021
   ```

3. Save your changes.

4. Specify this file with Response Generator's **-apf** command.

   ```
   RespGen -i "HV_Results.txt" -o997  "EDI_997.txt"
   -apf "C:\overrides\respgenoverrides.csv"
   ```

Formatting Override Entries

The RespGen Overrides section contains comma delimited information.

- Enter values in the order shown here.
- When omitting values in a line, include the commas to maintain correct positioning.
- Lines starting with an asterisk ('*') and blank lines are ignored.

The order of the Override lines is important; specific entries should come before broader entries. See File Example with Explanation on page 181.

Format of RespGen Overrides Lines

Each line contains these, in order:

For the segments **AK9, IK3, IK4, IK5, and TA1**

n=TargSegCode,error,loopID,segID,elemID,Value#1{,Value#2,…}

For the segment **RED06**

n=TargSegCode,error,segID,loopID,Value#1

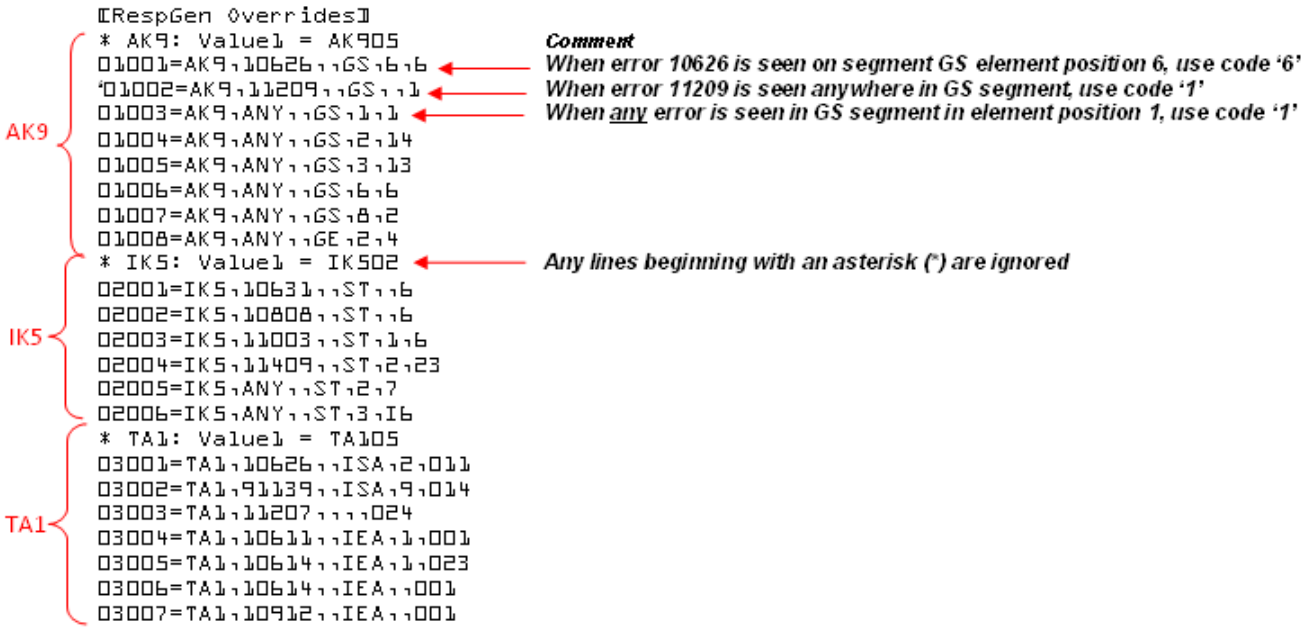
For the segment **CTX**

Refer to Appendix H: CTX Segments in Response Generator 999s on page 183 for complete information.

Value	Description	Example
n=TargSegCode	*n* is a unique numeric ID, such as a counter. Use this number to identify the override you are specifying. *TargSegCode* is one of the following: AK9 IK3 IK4 IK5 TA1 CTX RED06	Examples: 01001=AK9
Error	Error number to be overridden or `ANY` to represent any error. Each error has a unique ID number. Their ranges and locations are shown in **ErrorMessageNumbers.pdf**.	Examples: 01001=AK9,ANY
LoopID	Loop ID to be searched for error. (Optional)	Example: 01001=AK9,ANY,,
SegmentID	Segment ID to be searched for error. (Optional)	Examples: 01001=AK9,ANY,,GS
ElementID	Element position to be searched for error. (Optional) (Not used for RED06 overrides)	Examples: 02001=IK5,10631,,ST,,
Value#1{,Value#2, …}	Literal value(s) to use in target.	Examples: 01001=AK9,10626,,GS,6,6 RED01=RED06,12006,CLM,2300,E122

File Example with Explanation

The following example shows entries for AK9, IK5, and TA1 segments.



More specific entries should come before broader entries. In the example above, AK9 GS segment overrides for specific error numbers are listed before overrides for any errors on the segment.

Specific: When error 10626 is seen on segment GS element position 6, use code '6'

Specific: When error 11209 is seen anywhere in GS segment, use code '1'

Broad: When any error is seen in SpecificGS segment in element position 1, use code '1'

File Entry Examples

Desired Behavior…	File Entry…
If, on the RED06 element, the TIBCO Foresight-supplied default error number "12006" is encountered in the CLM segment, 2300 loop, display the custom error number "E122" instead.	`RED0612006=RED06,12006,CLM,2300,E122`
If an AK9 record with error number 10626 is encountered on any loop, the GS segment, element 6, insert the literal value "6" instead of the default AK9 value.	`01001=AK9,10626,,GS,6,6`
When error 11207 is seen anywhere in interchange segments on a TA1 record, use code "024"	`03003=TA1,11207,,,,024`

Demo Example

In the Instream's **Scripts** directory, the demo **V_RG_837P_5010_999_rgoverride** illustrates the use of Response Generator overrides.

The demo validates a 5010-837P and generates a 999 response document. Using a customized .apf, we've specified that, if error 10626 is encountered on the IK4 CLM02, the text "Invalid Character(s) in Amount" should be displayed instead of the default error message.

The sample **Respgenoverride.apf**, found in Instream's **Demodata** directory, contains the override file entry.

```
[RespGen Overrides]
01001=IK4,10626,2300,CLM,2,Invalid Character(s) in Amount
```

The demo script validates an 837P that contains the error (see **837P_5010_H_2Sub_BadCLM02_results.txt** in Instream's Output directory**)**:

```
DTL          31  2300 CLM 782    28 2     1    1  8  5    10626 3
```

The demo script then generates a 999 that replaces the 10626 value with our desired text (see **837P_5010_H_2Sub_BadCLM02.999**).

```
IK4*2*782*Invalid Character(s) in Amount*$460.00~
```

To run the demo:

1. Go to Instream's **Scripts** directory and double-click on **V_RG_837P_5010_999_rgoverride**.
2. Go to Instream's **Output** directory and look at the files created by Instream and Response Generator: **837P_5010_H_2Sub_BadCLM02_results.txt** and **837P_5010_H_2Sub_BadCLM02.999** in Instream's Output directory.

14 Appendix H: CTX Segments in Response Generator 999s

Overview

CTX segments appear in 999s in two flavors:

- Business Unit Identifier CTS: Errors in the "business unit" section of the data. The business unit is what TIBCO Foresight calls the application document – a claim, for instance. See **ApplicationDocTables.pdf**.
- Segment Context CTX: Information about situational relationships between segments involving specific errors.

Business Unit Identifier CTX

Response Generator automatically puts these in the 999 if an error occurs in the business unit section of the input data. The APF setting for CTX does not control whether this type of CTX appears in the 999.

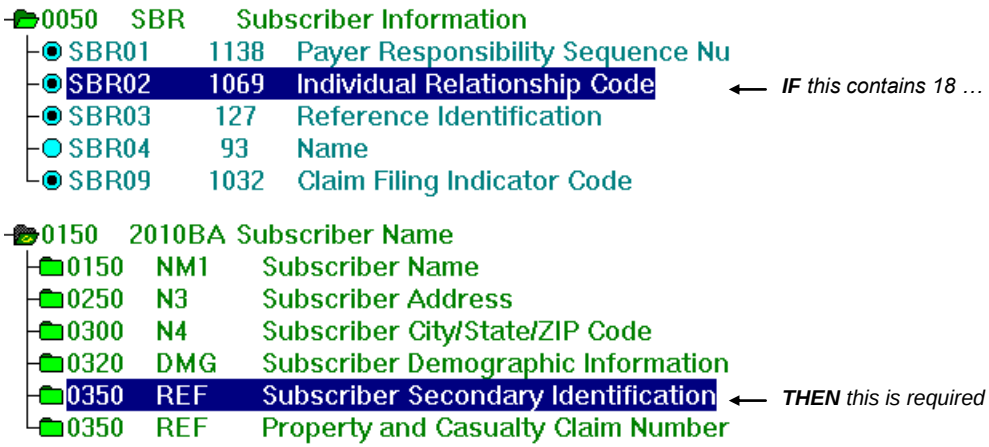
Example `CTX*CLM01:123456789~`

This segment automatically appears in the 999 when an 837 claim contains an error. The value 123456789 is the contents of the input CLM01.

Segment Context CTX

These are under your control. You can generate these or not, and can control which error numbers in the input data cause the CTX to appear in the 999.

For example, you might have a situational rule where this REF is required if the SBR02 contains 18:



You could set up a typical business rule to display an error message on the REF segment during validation if this relationship is not honored. This gives you information about the REF not being there when needed.

If you also wanted information about the SBR02, the "if" part of the relationship, then you could use a CTX segment in the 999 created by Response Generator.

You have these options for Segment Context CTX in a 999 created by Response Generator:

- No CTX segments
- TIBCO Foresight-Created CTX segments
- Your own custom CTX segments

No Segment Context CTX Segments

To generate no Segment Context CTX segments at all, be sure the validation APF file contains CTX=0:

```
[Detail Record Output]
STRT=1
END=1
ENDS=1
DTL=1
EMSG=1
EDAT=1
ESEG=1
VER=1
SVRTY=1
ETYPE=1
SVRTS=1
ETYPS=1
GEN=1
ZREC=1
CSEG=0
STRUS=1
STRUE=1
SVALU=1
EVALU=1
UID=0
ELOC=0
IDENT=0
CTX=0
```

TIBCO Foresight-Created Segment Context CTX Segments

Setup

To generate the TIBCO Foresight-defined CTX segments:

1. Set CTX=1 in the APF that you are using.
2. Be sure that your APF contains a [RespGen Overrides] section like this:

```
80010=,,,,,,,,,

[RespGen Overrides]
CTX00001=CTX,41233,SBR,,2,,,1069
CTX00002=CTX,41657,SBR,,2,,,1069
CTX00003=CTX,41236,NM1,,2,,,1065
CTX00004=CTX,41421,NM1,,2,,,1065
```

3. Use a 5010 Errata PDSA guideline that includes the CTX rules, or a guideline that is merged with one. These are listed in **ForesightHIPAAguidelinelist.pdf**.
4. On the Response Generator command line, use the **-apf** parameter and include the entire path to the APF, like this:

```
"C:\Foresight\Instream\Bin\respgen.exe"
-i "C:\Foresight\Instream\output\837Ifile_results.txt"
-o999 "C:\Foresight\Instream\Output\837Ifile_999.txt" -ge
-apf "C:\profiles\my.apf"
```

Results

- The validation detail file will contain CTX records like this if the situational rule is violated:

```
CTX          18|CTX02|14,32001
```

  TIB_fsp-instream_<*n.n*>_usersguide.pdf describes the format of the CTX record.

- The 999 will contain CTX segments like this if the situational rule is violated:

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

Example

If the situational rule is:

```
If the SBR02 contains 18, the N3 segment is required.
```

… then the violation of the rule will be discovered at the position where the N3 should have been (the THEN part of the relationship).

The CTX segment is saying that the IF part of the relationship was on the SBR segment at position 14 in the EDI data, the element is at position 2 in the SBR, and its element ID is1069.

Your own Custom Segment Context CTX Segments

Setup

To generate your own Segment Context CTX segment:

1. In the APF's [Detail Record Output] section, set CTX=1.

2. In the APF's [RespGen Overrides] section, add your own line for each CTX segment you intend to generate. See CTX Lines in the APF on page 188.

3. If you want to specify the contents of the IK304 or IK403 for the error number being used for this situational rule, do so in the APF's [Warning Levels] section. Find the line that corresponds to the error message and type the value that you want to use in the appropriate slot. See **APF.pdf** for details.

   Example

   Your situational rule uses error number 32001 and should generate a 3 in the IK304, like this:

   ```
   IK3*REF*16*2010*3~
   ```

   In the APF, add the 3 here:

   ```
   32000=3,8,,,,,,,
   32001=3,8,,,,,3,,
   ```

4. Create a guideline that generates these CTX records in the validation detail file. See Creating a Guideline that makes Segment Context CTX Records on page 190.

5. Merge with a 5010 Errata PDSA guideline that includes the CTX rule. These are listed in **ForesightHIPAAguidelinelist.pdf**.

6. Validate with the merged guideline.

7. On the Response Generator command line, use the **-apf** parameter and include the *entire path* to the APF, like this:

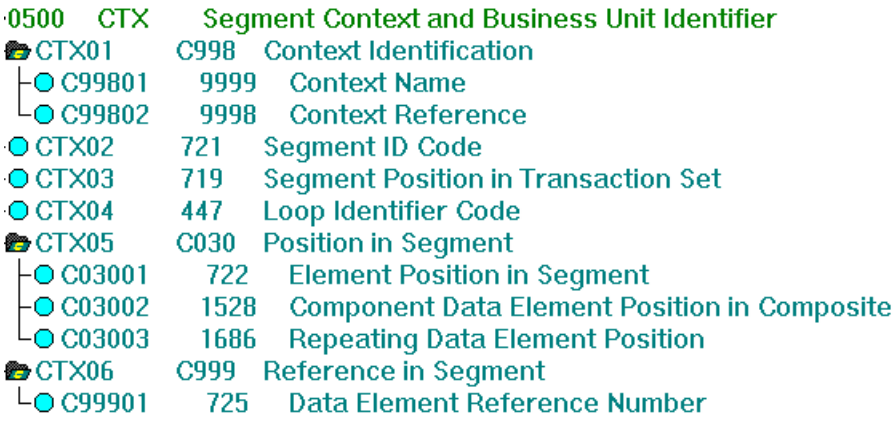
   ```
   "C:\Foresight\Instream\Bin\respgen.exe"
   -i "C:\Foresight\Instream\output\837Ifile_results.txt"
   -o999 "C:\Foresight\Instream\Output\837Ifile_999.txt" -ge
   -apf "C:\profiles\my.apf"
   ```

Results

The 999 will contain your customized CTX segment with the same structure as the ones created by TIBCO Foresight rules.

CTX Lines in the APF

The format of the CTX line in the APF corresponds closely to the structure of the 999 Segment Context CTX segment, which is:



The format in the APF is:

number=CTX,*errorNum*,*segID*,*loopID*,*posInSeg*,*PosinComp*,*Repeat*,*elemID*

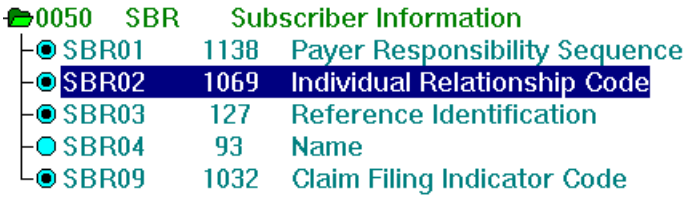
Where:

number=CTX	A unique number that starts with "CTX" followed by five digits followed by "=CTX". The number must be unique. Recommendation: Start with CTX50001 to avoid having your number overwritten by a new default TIBCO Foresight CTX number.
errorNum	This error's number in your custom errors file.
segID	ID of the first segment in the situational relationship.
loopID	LX loop ID. Only include this if the 999 CTX segment requires it.
posInSeg	In the first segment of the situational relationship, the position of the element or composite involved in the error. Omit if the error concerns the segment as a whole.
posInComp	In the first segment of the situational relationship, the position of the element within the composite. Omit if the element is not in a composite.
Repeat	In the first segment of the situational relationship, the number of the instance if this is a repeating element.
elemID	In the first segment of the situational relationship, the ID of the element, if any, containing the error.

To see the structure of the 999, see one of the 5010 Errata 999 guidelines listed in **ForesightHIPAAguidelinesList.pdf**.

Example 1

The first segment in the situational relationship is the SBR, and the situational rule involves the SBR02:



Example CTX entry in APF file:

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

Where:

`CTX50001` A unique number.

`=CTX` Literal text.

`32001` The error number is 32001 in the customer errors file.

`SBR` The first segment in the situational relationship is the SBR.

`,,` No LX loop ID.

`2` The error involves the SBR-02.

`,,` There is no subelement involved in the situational relationship.

`,,` There is no element repeat involved in the situational relationship.

`1069` The error involves the element with ID 1069.

Example 2

The first segment in the situational relationship is the TRN, and the situational rule involves the entire segment:

```
CTX50002=CTX,32002,TRN,,,,,
```

Example 3

The first segment in the situational relationship is the CLM and the situational rule involves the CLM-05 (a composite), subelement 1.

```
CTX50003=CTX,41079,CLM,,5,1,,
```

Creating a Guideline that makes Segment Context CTX Records

See **BusinessRules.pdf** for details on these rules, which are typically used to create Segment Context CTX records:

- SaveCurrentSegment
- GetValueFromSegment
- BuildString
- OutputCTX

This section demonstrates these rules with this example:

IF the SBR02=18
THEN the 2010BA REF for Subscriber Secondary Identification must be used

We want two results if this is violated:

- A validation error message
- A CTX record in Response Generator's 999

Strategy for Creating Segment Context CTX Rules

Files for this example are in TIBCO Foresight® Instream®'s DemoData\CTX directory.

Set up the IF part:

1. On the first segment in the situational relationship, use SaveCurrentSegment to capture the entire segment in a variable.

   Example

   On the SBR segment, save the SBR and clear a variable that will contain the contents of the SBR02:

   ```
   BusinessRules.Variable. SaveCurrentSegment 2000BSBRSeg
   BusinessRules.Variable Clear SBR02Var
   ```

2. If there is a value in the first segment that is involved in the situational relationship, capture it in a variable.

   Example

   On the SBR02, capture its contents in a variable:

   ```
   BusinessRules.Variable SetVar SBR02Var
   ```

Set up the THEN part:

1. Set up a test to see if the condition has been met and display a validation error message if needed.

   Example

   On the REF segment, if the segment doesn't exist, test the contents of the SBR02. If it contains 18, display a custom error message.

   ```
   BusinessRules.Run RunNoData (BusinessRules.Variable CompareString
   SBR02Var EQ "18" (BusinessRules.Utilities DisplayErrorByNumber
   32001))
   ```

2. Create the rules needed to put a CTX record in the validation detail results file.

   Example

   On the REF segment, if the segment doesn't exist, test the contents of the SBR02. If it contains 18, put the entire SBR segment into a variable.

   We already have the SBR segment in 2000BSBRSeg. We just need to get it from there into an ordinary variable, which we are calling SEGPOS2000BSBRSeg.

   ```
   BusinessRules.Run RunNoData (BusinessRules.Variable CompareString
   SBR02Var EQ "18" (BusinessRules.Variable GetValueFromSegment
   2000BSBRSeg POS -1 -1 SEGPOS2000BSBRSeg))
   ```

   We use this rule to build the CTX record for the validation detail file:

   ```
   BusinessRules.Run RunNoData (BusinessRules.Variable CompareString
   SBR02Var EQ "18" (BusinessRules.Utilities BuildString CTXOUTSTRING
   "" "CTX02" "|" SEGPOS2000BSBRSeg "," "32001"))
   ```

   And, finally, we output the CTX record:

   ```
   BusinessRules.Run RunNoData (BusinessRules.Variable CompareString
   SBR02Var EQ "18" (BusinessRules.Utilities OutputCTX CTXOUTSTRING))
   ```

3. Now merge it with a PDSA guideline and use the merged guideline for Instream® validation and Response Generator.