CHIAPAS EDI 2012
USER MANUAL

CHIAPAS EDI TECHNOLOGIES, INC.

TABLE OF CONTENTS

Contents

INTRODUCTION

Chiapas EDI 2012 was designed from the ground up to integrate HIPAA transactions with enterprise business processes as simply as possible. The 4010 and 5010 HIPAA transactions are highly complex, with hierarchical file formats that are anything but intuitive. Embedded within this file format is the actual business data that trading partners need to exchange for ongoing business – claims, eligibility, remittances, and so on. The core technology of Chiapas EDI focuses on making the business data within a transaction accessible with as little focus on the HIPAA syntax 'container' as possible.

To decode a HIPAA file into a form that can be queried and analyzed, an analyst need only complete a few simple operations within the included Chiapas Studio. Because Chiapas EDI uniquely assigns name to every business element within a HIPAA transaction, it can translate EDI directly to a database table without 'mapping' activities traditionally associated with EDI parsers. The analyst can choose to either create a new database table reflecting the incoming EDI file or merge data from many EDI files into a single destination, allowing analysis on a large group of business data.

Creating a new HIPAA file is a more complex task, but still within the reach of a SQL-aware business analyst. The sample database included with the Chiapas EDI distribution presents a small amount of mock claims and eligibility data rows, as well as a SQL View for each 5010 transactions. These views transform the mock healthcare data into a form Chiapas EDI can accept to project a HIPAA compliant EDI file. The well-documented transformations included within the sample SQL views provide a solid foundation for creating new outbound transactions.

Chiapas Studio is a powerful tool for developing and testing new trading partner transactions. Once it is time to go to production, Chiapas EDI contains a Windows Workflow 4 based automation system capable of creating a completely 'hands-free' process using a visual design language and event-driven triggers. Encrypted file uploads are supported via the included WinSCP SecureFTP client to ensure PHI information is safeguarded at all times.

To install Chiapas EDI to your 64-bit Windows PC and see the minimum system requirements, please see the Install Guide.

To get started using the features, please see the Quick Start Guide. This will show you how to use Chiapas Studio and introduce basic concepts related to decoding, encoding and automation.

For a thorough coverage of all of the functionality within Chiapas Studio and a thorough explanation of the 'Chiapas Gate Intermediate Format' that underlies the core translation engine, see the User Reference.

Finally, the Technical Reference gives in-depth coverage of the data structures and Windows Workflow 4 Activities used to develop complex new business automation and trading partner transactions

INSTALL GUIDE

REQUIREMENTS

Chiapas EDI 2012 has the following requirements:

- 64-Bit Windows Operating System: Windows 7, Windows Server 2008 R2, Windows 8 or Windows Server 2012
- .NET 4.0. Although installed by default on Windows 7 and 8, it is normally an optional Feature on Server operating systems.
- SQL Server 2008 R2 (Express or Standard versions)
- 8 GB of RAM
- Local Administrator rights when installing

INSTALLATION

Chiapas EDI 2012 is distributed as a single .MSI file. To run it, double-click on it and it will automatically install all components to the C: drive under the C:\chiapas directory.

Next, launch the executable located at:

```
    C:\chiapas\db
```

Follow the instructions to add the chiapasData SQL database to the SQL Server instance. This contains all of the tables necessary for Chiapas EDI's automation system as well as the mock healthcare data and SQL Views for creating new outbound transactions.

Finally, the automation background service needs to be configured:

Press Start -> Right Click on Computer, select 'Manage', Go to Configuration, Services, Right-click Properties on ChiapasService. Switch to the LogOn tab, then fill in the 'Log On As' information with login and password information for an account with Administrator permissions.

Return to the General Tab, press 'Start' to start running the automation service.

UPDATES

Your Chiapas EDI distribution may occasionally be updated as part of your ongoing maintenance agreement. All Chiapas EDI downloads are controlled via a single SecureFTP server accessed at:

```
sftp.chiapas-edi.org
```

Login information is provided to you along with your Chiapas EDI 2012 license.

If your corporation lacks the ability to log into Secure FTP sites, a well-established client can be found at http://winscp.net .

QUICK START GUIDE

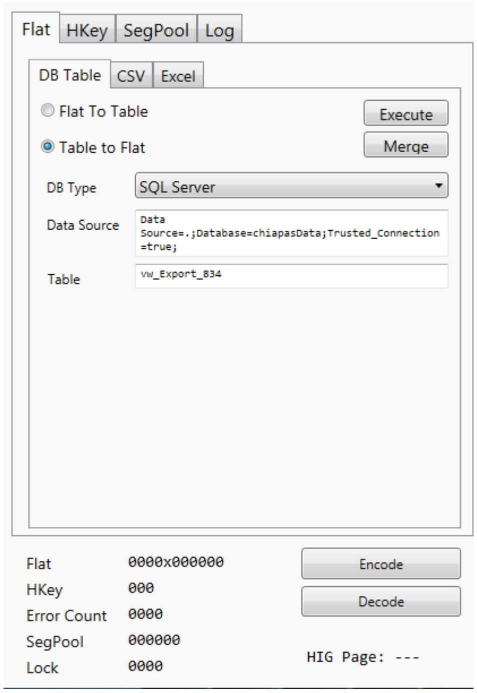
This Quick Start Guide is oriented to quickly accessing three core features of the Chiapas EDI 2012 product: Decoding, Encoding, and Automation. Decoding is focused on taking one of the included sample HIPAA files through the steps of converting it to a database table. Encoding focuses on converting some of the included mock healthcare data to a new HIPAA file. This covers the most basic usage patterns of Chiapas EDI. Both of these examples start with launching Chiapas Studio, located at the following path:

```
C:\chiapas\bin\ChiapasStudio.exe
```

The tab group on the left is called the Data I/O pane; the middle is called the Register panne; and the rightmost tab group is called the SegPool pane.

DECODING

A basic operation of Chiapas EDI 2012 is the ability to convert an arbitrary 4010 or 5010 EDI file to a database table. This section will guide you through the process step by step.



○ Flat To Table
● Table to Flat

DB Type: SQL Server

Data Source: Data Source=.;Database=chiapasData;Trusted_Connection=true;

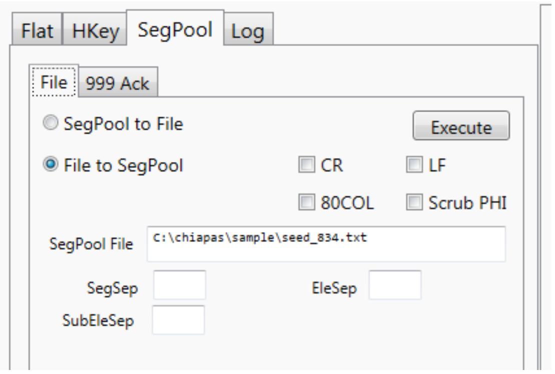
Table: vw_Export_834

Flat	0000x000000
HKey	000
Error Count	0000
SegPool	000000
Lock	0000

HIG Page: ---

When you first open Chiapas Studio, the left side of the screen is occupied by the Data I/O pane. This pane allows the user to directly control loading or saving information from the Chiapas Studio environment.

To start, press on the SegPool tab:



○ SegPool to File
● File to SegPool

□ CR □ LF
□ 80COL □ Scrub PHI

SegPool File: C:\chiapas\sample\seed_834.txt

SegSep

EleSep

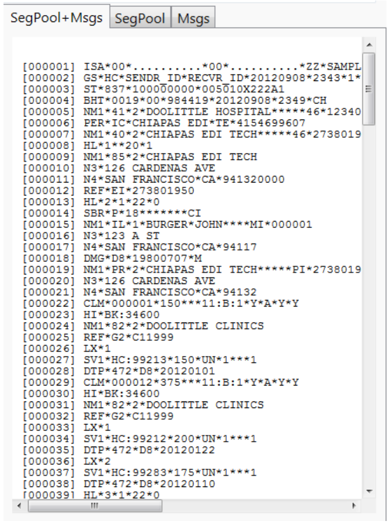
SubEleSep

On the SegPool pane, you will notice that the SegPool File box is already filled out with a default value. Double click the mouse cursor within the white area of the SegPool File area and select a new file:

```
C:\chiapas\sample\seed_837p.txt
```

Now, leaving the default 'File to SegPool' radio button selected, press the 'Execute' button.

On the right side of the screen, the 'SegPool+Msgs' window is filled in with the contents of the file. This is some of the mock healthcare data that is included with the distribution.

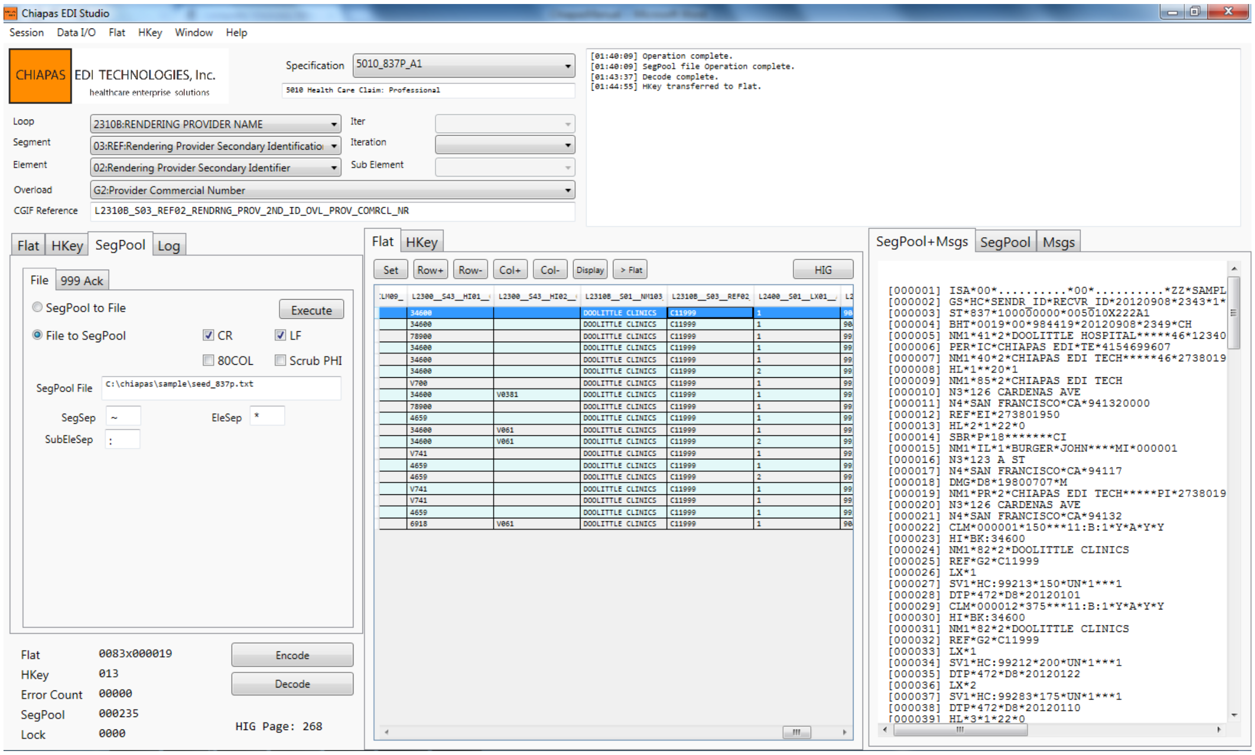


```
[000001] ISA*00*..........*00*..........*ZZ*SAMPL
[000002] GS*HC*SENDR_ID*RECVR_ID*20120908*2343*1*
[000003] ST*837*100000000*005010X222A1
[000004] BHT*0019*00*984419*20120908*2349*CH
[000005] NM1*41*2*DOOLITTLE HOSPITAL*****46*12340
[000006] PER*IC*CHIAPAS EDI*TE*4154699607
[000007] NM1*40*2*CHIAPAS EDI TECH*****46*2738019
[000008] HL*1**20*1
[000009] NM1*85*2*CHIAPAS EDI TECH
[000010] N3*126 CARDENAS AVE
[000011] N4*SAN FRANCISCO*CA*941320000
[000012] REF*EI*273801950
[000013] HL*2*1*22*0
[000014] SBR*P*18*******CI
[000015] NM1*IL*1*BURGER*JOHN****MI*000001
[000016] N3*123 A ST
[000017] N4*SAN FRANCISCO*CA*94117
[000018] DMG*D8*19800707*M
[000019] NM1*PR*2*CHIAPAS EDI TECH*****PI*2738019
[000020] N3*126 CARDENAS AVE
[000021] N4*SAN FRANCISCO*CA*94132
[000022] CLM*000001*150***11:B:1*Y*A*Y*Y
[000023] HI*BK:34600
[000024] NM1*82*2*DOOLITTLE CLINICS
[000025] REF*G2*C11999
[000026] LX*1
[000027] SV1*HC:99213*150*UN*1***1
[000028] DTP*472*D8*20120101
[000029] CLM*000012*375***11:B:1*Y*A*Y*Y
[000030] HI*BK:34600
[000031] NM1*82*2*DOOLITTLE CLINICS
[000032] REF*G2*C11999
[000033] LX*1
[000034] SV1*HC:99212*200*UN*1***1
[000035] DTP*472*D8*20120122
[000036] LX*2
[000037] SV1*HC:99283*175*UN*1***1
[000038] DTP*472*D8*20120110
[000039] HL*3*1*22*0
```

Next, press the 'Decode' button in the lower left part of the screen.

The middle area of the screen is divided into two tabbed areas – Flat and HKey. Click on the HKey tab now, and then press the 'Flat < HKey' button. Now, click on the 'Flat' tab again.

The screen at this point will look similar to this:



Specification: 5010_837P_A1

5010 Health Care Claim: Professional

```
[01:40:09] Operation complete.
[01:40:09] SegPool file Operation complete.
[01:43:37] Decode complete.
[01:44:55] HKey transferred to Flat.
```

Loop: 2310B:RENDERING PROVIDER NAME

Segment: 03:REF:Rendering Provider Secondary Identificatio

Element: 02:Rendering Provider Secondary Identifier

Iter:

Iteration:

Sub Element:

Overload: G2:Provider Commercial Number

CGIF Reference: L2310B_S03_REF02_RENDRNG_PROV_2ND_ID_OVL_PROV_COMRCL_NR

○ SegPool to File

● File to SegPool

☑ CR ☑ LF

☐ 80COL ☐ Scrub PHI

SegPool File: C:\chiapas\sample\seed_837p.txt

SegSep: ~ EleSep: *

SubEleSep: :

Flat 0083x000019

HKey 013

Error Count 00000

SegPool 000235

Lock 0000

HIG Page: 268

:LM09_	L2300__S43__HI01__	L2300__S43__HI02__	L2310B__S01__NM103	L2310B__S03__REF02	L2400__S01__LX01__	L2
	34600		DOOLITTLE CLINICS	C11999	1	90
	34600		DOOLITTLE CLINICS	C11999	1	90
	78900		DOOLITTLE CLINICS	C11999	1	99
	34600		DOOLITTLE CLINICS	C11999	1	99
	34600		DOOLITTLE CLINICS	C11999	1	99
	34600		DOOLITTLE CLINICS	C11999	2	99
	V700		DOOLITTLE CLINICS	C11999	1	99
	34600	V0381	DOOLITTLE CLINICS	C11999	1	99
	78900		DOOLITTLE CLINICS	C11999	1	99
	4659		DOOLITTLE CLINICS	C11999	1	99
	34600	V061	DOOLITTLE CLINICS	C11999	1	99
	34600	V061	DOOLITTLE CLINICS	C11999	2	99
	V741		DOOLITTLE CLINICS	C11999	1	99
	4659		DOOLITTLE CLINICS	C11999	1	99
	4659		DOOLITTLE CLINICS	C11999	2	99
	V741		DOOLITTLE CLINICS	C11999	1	99
	V741		DOOLITTLE CLINICS	C11999	1	99
	4659		DOOLITTLE CLINICS	C11999	1	99
	6918	V061	DOOLITTLE CLINICS	C11999	1	90

```
[000001] ISA*00*..........*00*..........*ZZ*SAMPL
[000002] GS*HC*SENDR_ID*RECVR_ID*20120908*2343*1*
[000003] ST*837*100000000*005010X222A1
[000004] BHT*0019*00*984419*20120908*2349*CH
[000005] NM1*41*2*DOOLITTLE HOSPITAL*****46*12340
[000006] PER*IC*CHIAPAS EDI*TE*4154699607
[000007] NM1*40*2*CHIAPAS EDI TECH*****46*2738019
[000008] HL*1**20*1
[000009] NM1*85*2*CHIAPAS EDI TECH
[000010] N3*126 CARDENAS AVE
[000011] N4*SAN FRANCISCO*CA*941320000
[000012] REF*EI*273801950
[000013] HL*2*1*22*0
[000014] SBR*P*18*******CI
[000015] NM1*IL*1*BURGER*JOHN****MI*000001
[000016] N3*123 A ST
[000017] N4*SAN FRANCISCO*CA*94117
[000018] DMG*D8*19800707*M
[000019] NM1*PR*2*CHIAPAS EDI TECH*****PI*2738019
[000020] N3*126 CARDENAS AVE
[000021] N4*SAN FRANCISCO*CA*94132
[000022] CLM*000001*150***11:B:1*Y*A*Y*Y
[000023] HI*BK:34600
[000024] NM1*82*2*DOOLITTLE CLINICS
[000025] REF*G2*C11999
[000026] LX*1
[000027] SV1*HC:99213*150*UN*1***1
[000028] DTP*472*D8*20120101
[000029] CLM*000012*375***11:B:1*Y*A*Y*Y
[000030] HI*BK:34600
[000031] NM1*82*2*DOOLITTLE CLINICS
[000032] REF*G2*C11999
[000033] LX*1
[000034] SV1*HC:99212*200*UN*1***1
[000035] DTP*472*D8*20120122
[000036] LX*2
[000037] SV1*HC:99283*175*UN*1***1
[000038] DTP*472*D8*20120110
[000039] HL*3*1*22*0
```

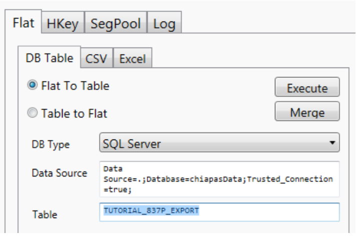
The middle pane represents the ‘Flat’ register, where the preceding actions have decoded the incoming 837 Professional EDI file and transformed it into a two dimensional table representation. By clicking on the various cells of the table, mapping information is projected onto the pulldown textboxes in the upper left.

Now, let’s get this information into a SQL database table, where the business information within this EDI file can be easily queried.

Click on the ‘Flat’ tab on the left Data I/O pane. Click on the ‘Flat to Table’ radio button, and then change the Table textbox to the following name:

```
TUTORIAL_837P_EXPORT
```

The pane should look like this:



● Flat To Table
○ Table to Flat
DB Type: SQL Server
Data Source: Data Source=.;Database=chiapasData;Trusted_Connection=true;
Table: TUTORIAL_837P_EXPORT

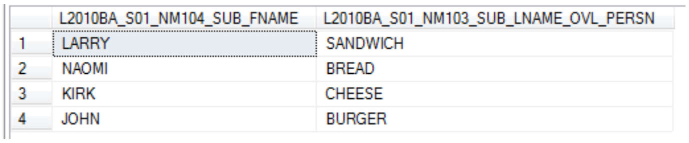
Now, press the ‘Execute’ button. The TUTORIAL_837P_EXPORT table will be created and populated with the data.

If you have a group of files of the same file format and you would like to merge the data into a single table, you can use the ‘Merge’ button instead of the ‘Execute’ button. Any business data where the mappings are common between the destination table and the loaded data will be inserted into the new table. Any mappings in the loaded data that are not present within the destination table will be flagged as a message.

For now, let’s proceed to querying the newly created data. Assuming that you have SQL Server Management Studio on the workstation, open it up and execute the following query against the chiapasData database:

```
SELECT          L2010BA_S01_NM104_SUB_FNAME,
                L2010BA_S01_NM103_SUB_LNAME_OVL_PERSN
FROM            chiapasData.dbo.TUTORIAL_837P_EXPORT
WHERE           L2400_S11_DTP03_SVC_DT_OVL_SVC_D8 > '20120120'
```

The results of this query show the first and last names of subscribers linked to a date of service past January 20th, 2012:

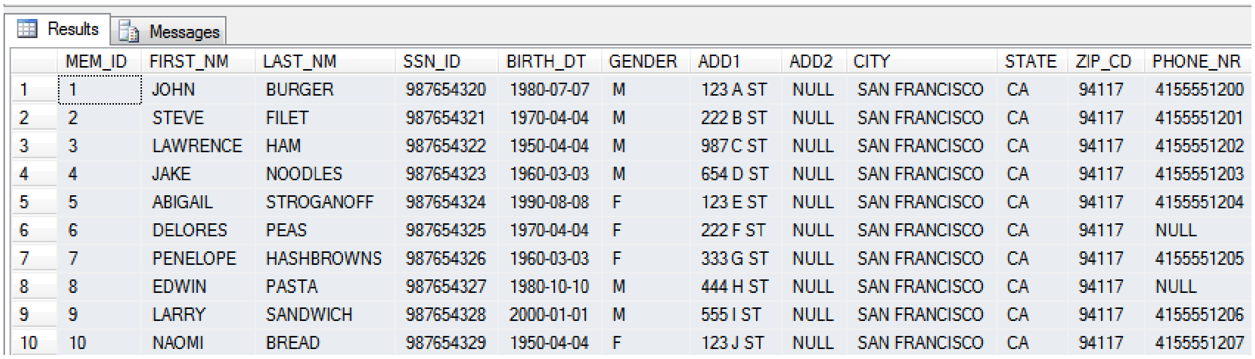


	L2010BA_S01_NM104_SUB_FNAME	L2010BA_S01_NM103_SUB_LNAME_OVL_PERSN
1	LARRY	SANDWICH
2	NAOMI	BREAD
3	KIRK	CHEESE
4	JOHN	BURGER

ENCODING

This example will demonstrate how to use Chiapas EDI to progress from some simple, mock healthcare information and transform it into a fully compliant EDI file. Let's start with examining some of the mock healthcare information. First, let's investigate the Member table. Using SQL Server Management Services (SSMS), connect to the chiapasData database and open a new query window. Run the following query:

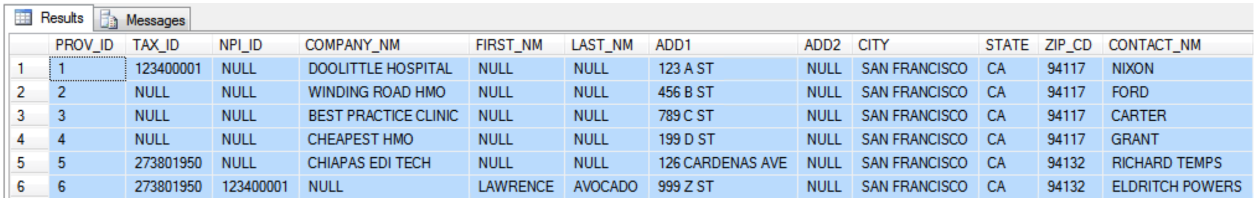
```
SELECT TOP 10 * FROM MEMBER
```



	MEM_ID	FIRST_NM	LAST_NM	SSN_ID	BIRTH_DT	GENDER	ADD1	ADD2	CITY	STATE	ZIP_CD	PHONE_NR
1	1	JOHN	BURGER	987654320	1980-07-07	M	123 A ST	NULL	SAN FRANCISCO	CA	94117	4155551200
2	2	STEVE	FILET	987654321	1970-04-04	M	222 B ST	NULL	SAN FRANCISCO	CA	94117	4155551201
3	3	LAWRENCE	HAM	987654322	1950-04-04	M	987 C ST	NULL	SAN FRANCISCO	CA	94117	4155551202
4	4	JAKE	NOODLES	987654323	1960-03-03	M	654 D ST	NULL	SAN FRANCISCO	CA	94117	4155551203
5	5	ABIGAIL	STROGANOFF	987654324	1990-08-08	F	123 E ST	NULL	SAN FRANCISCO	CA	94117	4155551204
6	6	DELORES	PEAS	987654325	1970-04-04	F	222 F ST	NULL	SAN FRANCISCO	CA	94117	NULL
7	7	PENELOPE	HASHBROWNS	987654326	1960-03-03	F	333 G ST	NULL	SAN FRANCISCO	CA	94117	4155551205
8	8	EDWIN	PASTA	987654327	1980-10-10	M	444 H ST	NULL	SAN FRANCISCO	CA	94117	NULL
9	9	LARRY	SANDWICH	987654328	2000-01-01	M	555 I ST	NULL	SAN FRANCISCO	CA	94117	4155551206
10	10	NAOMI	BREAD	987654329	1950-04-04	F	123 J ST	NULL	SAN FRANCISCO	CA	94117	4155551207

Now, here's to view the roster of providers included in the mock data:

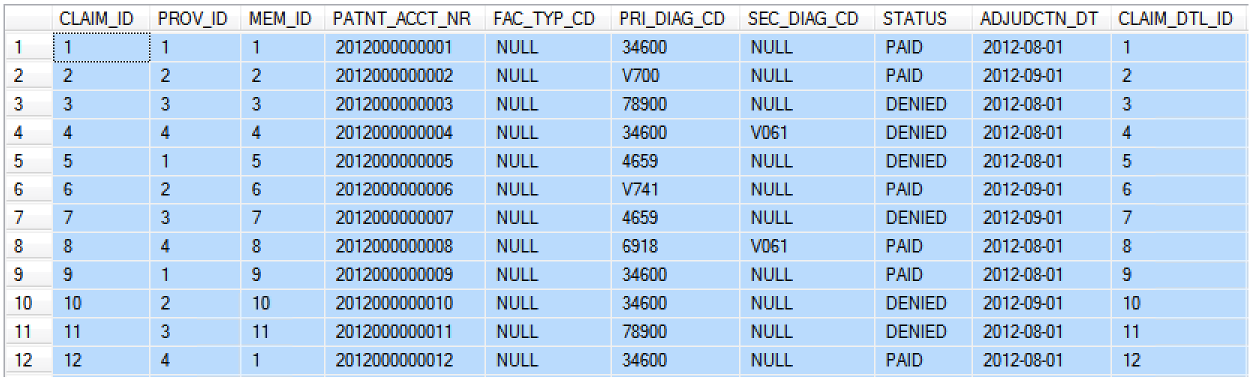
```
SELECT TOP 10 * FROM PROVIDER
```



	PROV_ID	TAX_ID	NPI_ID	COMPANY_NM	FIRST_NM	LAST_NM	ADD1	ADD2	CITY	STATE	ZIP_CD	CONTACT_NM
1	1	123400001	NULL	DOOLITTLE HOSPITAL	NULL	NULL	123 A ST	NULL	SAN FRANCISCO	CA	94117	NIXON
2	2	NULL	NULL	WINDING ROAD HMO	NULL	NULL	456 B ST	NULL	SAN FRANCISCO	CA	94117	FORD
3	3	NULL	NULL	BEST PRACTICE CLINIC	NULL	NULL	789 C ST	NULL	SAN FRANCISCO	CA	94117	CARTER
4	4	NULL	NULL	CHEAPEST HMO	NULL	NULL	199 D ST	NULL	SAN FRANCISCO	CA	94117	GRANT
5	5	273801950	NULL	CHIAPAS EDI TECH	NULL	NULL	126 CARDENAS AVE	NULL	SAN FRANCISCO	CA	94132	RICHARD TEMPS
6	6	273801950	123400001	NULL	LAWRENCE	AVOCADO	999 Z ST	NULL	SAN FRANCISCO	CA	94132	ELDRITCH POWERS

Now, to bring up the mock claims information:

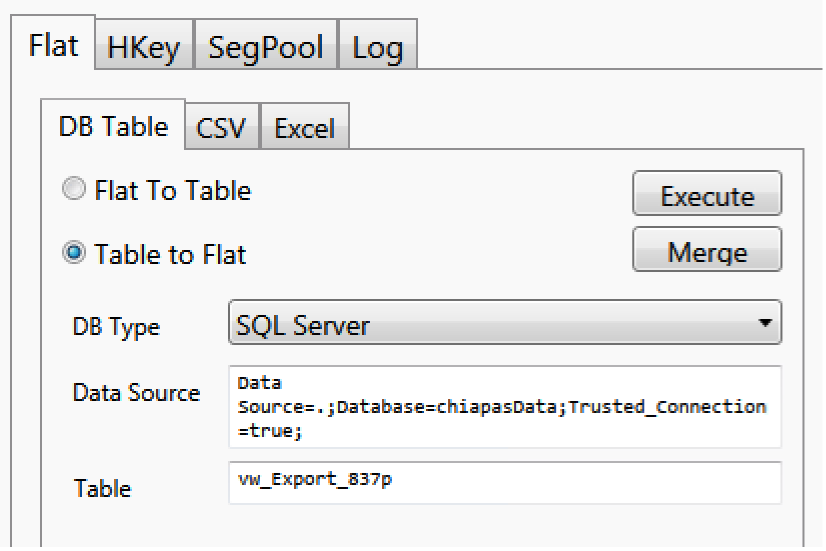
```
SELECT * FROM CLAIM_HDR HDR INNER JOIN CLAIM_DTL DTL ON HDR.CLAIM_ID =
DTL.CLAIM_ID
```



	CLAIM_ID	PROV_ID	MEM_ID	PATNT_ACCT_NR	FAC_TYP_CD	PRI_DIAG_CD	SEC_DIAG_CD	STATUS	ADJUDCTN_DT	CLAIM_DTL_ID
1	1	1	1	2012000000001	NULL	34600	NULL	PAID	2012-08-01	1
2	2	2	2	2012000000002	NULL	V700	NULL	PAID	2012-09-01	2
3	3	3	3	2012000000003	NULL	78900	NULL	DENIED	2012-08-01	3
4	4	4	4	2012000000004	NULL	34600	V061	DENIED	2012-08-01	4
5	5	1	5	2012000000005	NULL	4659	NULL	DENIED	2012-08-01	5
6	6	2	6	2012000000006	NULL	V741	NULL	PAID	2012-09-01	6
7	7	3	7	2012000000007	NULL	4659	NULL	DENIED	2012-09-01	7
8	8	4	8	2012000000008	NULL	6918	V061	PAID	2012-08-01	8
9	9	1	9	2012000000009	NULL	34600	NULL	PAID	2012-08-01	9
10	10	2	10	2012000000010	NULL	34600	NULL	DENIED	2012-09-01	10
11	11	3	11	2012000000011	NULL	78900	NULL	DENIED	2012-08-01	11
12	12	4	1	2012000000012	NULL	34600	NULL	PAID	2012-08-01	12

On the previous page, we saw member and provider information, and above we can see a mock claims table that will be encoded to a file. There is a single SQL View, vw_Export_837P, that will be used to transform this data into a form that Chiapas EDI can convert to EDI format. See the User Reference and Technical Reference for more information about how this works.

Within Chiapas Studio, click on the 'Flat' tab on the left Data I/O pane, and then select the 'Table to Flat' radio button and type in 'vw_Export_837p' within the Table textbox.



● Table to Flat

DB Type: SQL Server

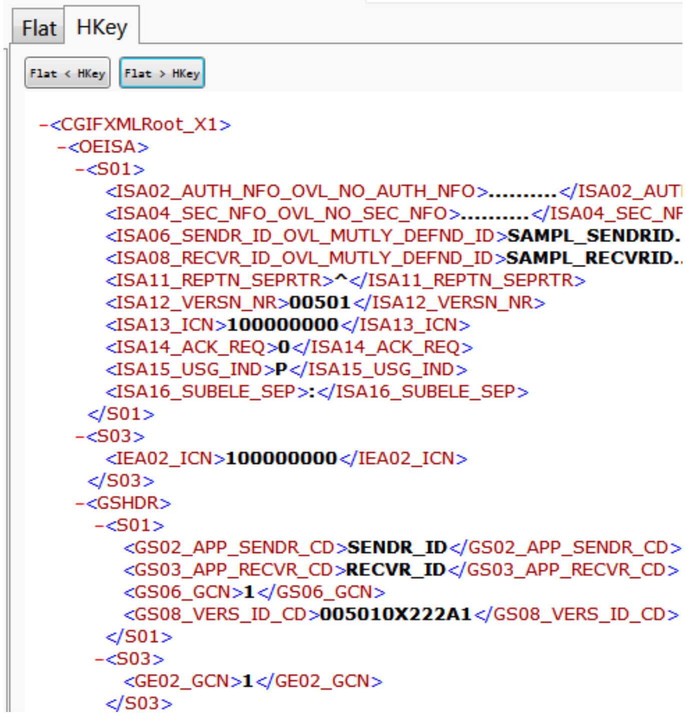
Data Source: Data Source=.;Database=chiapasData;Trusted_Connection=true;

Table: vw_Export_837p

Now, press the Execute button. Press the 'Flat' tab in the middle Register pane, and you will see the data loaded from the View successfully.

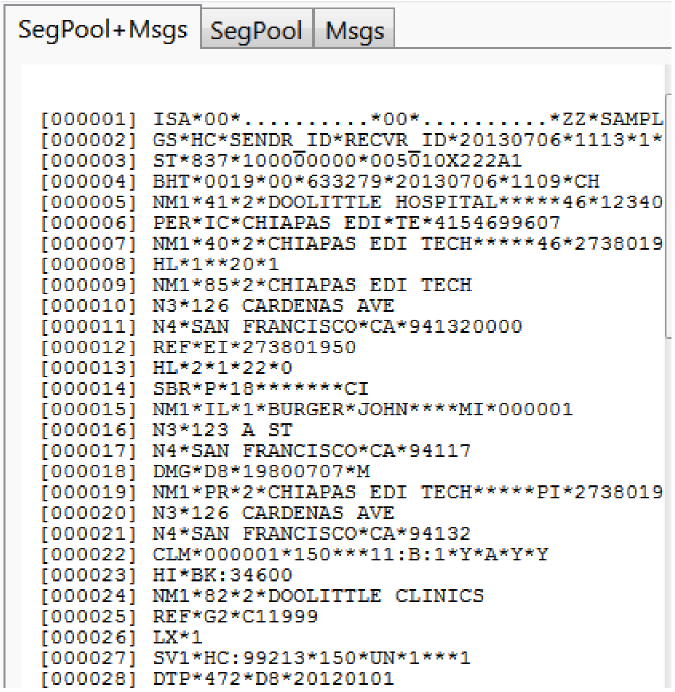
Now, select the HKey tab in the middle Register pane, then press the ‘Flat > HKey’ button.

The results will appear like this:



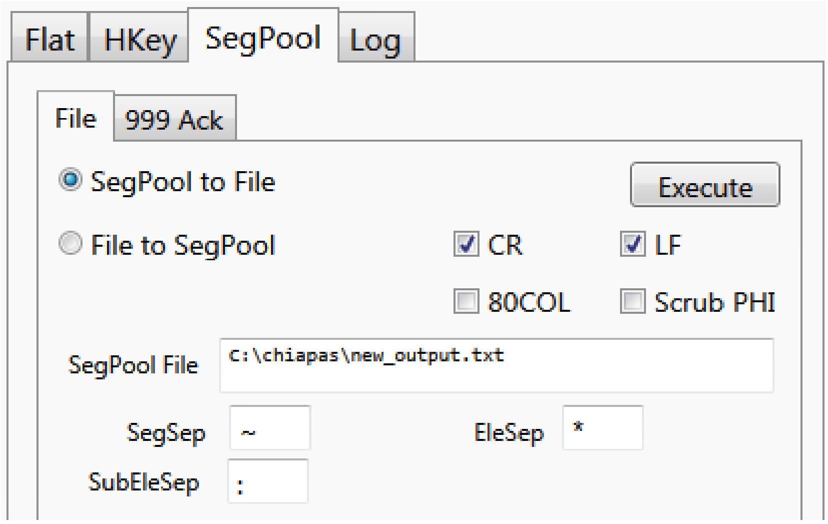
```
-<CGIFXMLRoot_X1>
  -<OEISA>
    -<S01>
        <ISA02_AUTH_NFO_OVL_NO_AUTH_NFO>..........</ISA02_AUT
        <ISA04_SEC_NFO_OVL_NO_SEC_NFO>..........</ISA04_SEC_NF
        <ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID>SAMPL_SENDRID.
        <ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID>SAMPL_RECVRID..
        <ISA11_REPTN_SEPRTR>^</ISA11_REPTN_SEPRTR>
        <ISA12_VERSN_NR>00501</ISA12_VERSN_NR>
        <ISA13_ICN>100000000</ISA13_ICN>
        <ISA14_ACK_REQ>0</ISA14_ACK_REQ>
        <ISA15_USG_IND>P</ISA15_USG_IND>
        <ISA16_SUBELE_SEP>:</ISA16_SUBELE_SEP>
      </S01>
    -<S03>
        <IEA02_ICN>100000000</IEA02_ICN>
      </S03>
    -<GSHDR>
      -<S01>
          <GS02_APP_SENDR_CD>SENDR_ID</GS02_APP_SENDR_CD>
          <GS03_APP_RECVR_CD>RECVR_ID</GS03_APP_RECVR_CD>
          <GS06_GCN>1</GS06_GCN>
          <GS08_VERS_ID_CD>005010X222A1</GS08_VERS_ID_CD>
        </S01>
      -<S03>
          <GE02_GCN>1</GE02_GCN>
        </S03>
```

Now, press the ‘Encode’ button on the lower left. On the right, the SegPool register will be loaded and appear like so:



```
[000001] ISA*00*..........*00*..........*ZZ*SAMPL
[000002] GS*HC*SENDR_ID*RECVR_ID*20130706*1113*1*
[000003] ST*837*100000000*005010X222A1
[000004] BHT*0019*00*633279*20130706*1109*CH
[000005] NM1*41*2*DOOLITTLE HOSPITAL*****46*12340
[000006] PER*IC*CHIAPAS EDI*TE*4154699607
[000007] NM1*40*2*CHIAPAS EDI TECH*****46*2738019
[000008] HL*1**20*1
[000009] NM1*85*2*CHIAPAS EDI TECH
[000010] N3*126 CARDENAS AVE
[000011] N4*SAN FRANCISCO*CA*941320000
[000012] REF*EI*273801950
[000013] HL*2*1*22*0
[000014] SBR*P*18*******CI
[000015] NM1*IL*1*BURGER*JOHN****MI*000001
[000016] N3*123 A ST
[000017] N4*SAN FRANCISCO*CA*94117
[000018] DMG*D8*19800707*M
[000019] NM1*PR*2*CHIAPAS EDI TECH*****PI*2738019
[000020] N3*126 CARDENAS AVE
[000021] N4*SAN FRANCISCO*CA*94132
[000022] CLM*000001*150***11:B:1*Y*A*Y*Y
[000023] HI*BK:34600
[000024] NM1*82*2*DOOLITTLE CLINICS
[000025] REF*G2*C11999
[000026] LX*1
[000027] SV1*HC:99213*150*UN*1***1
[000028] DTP*472*D8*20120101
```

To save the results to a new EDI file, go to the Data I/O pane on the left. Select the 'SegPool to File' radio button, click on the CR and LF checkboxes, and enter a filename under the SegPool File textbox. Then, press Execute, and a new file will be generated.



● SegPool to File

○ File to SegPool

☑ CR ☑ LF

□ 80COL □ Scrub PHI

SegPool File C:\chiapas\new_output.txt

SegSep ~ EleSep *

SubEleSep :

This concludes the Encode example.

USER REFERENCE

INTRODUCTION

This section will cover the essential functions of Chiapas EDI 2012, beginning from Chiapas Studio and proceeding through the core functions of scheduling and automation. It will cover the Chiapas Gate Intermediate Format, which is used by Chiapas EDI to convert between hierarchical element references and fixed column names. The main objective of this section is to give the user an understanding of the fundamentals of EDI translation and how Chiapas EDI can be used to integrate HIPAA transactions with enterprise systems, with a focus on SQL Server databases. Before we proceed into the details, one may ask: what is Chiapas EDI 2012?

In simplest terms, Chiapas EDI 2012 is a Windows-based ANSI X12 4010/5010 EDI translation software with automation components. EDI translation systems are focused on communicating internal business data with an external trading partner. Within a single business, there is no need for EDI because all of the information is integrated into the enterprise system. However, in the normal course of business, many healthcare entities enter into contractual agreements where they must regularly exchange certain types of data, like claims or eligibility. Trading partners are required to transform their internal business data into an EDI message and securely transmit it to the other party, who then extracts the business data from the message into a form they can use.

The 5010 format is actually over a dozen different transaction specifications that defines over ten thousand individually defined business elements. Even though this is a federally regulated specification, the exact description of the protocol needs to be licensed by the Washington Publishing Company, who publish the specifications on behalf of the ASC X12 Accredited Standards Committee who own the specifications. Chiapas EDI 2012 contains sufficient information about these specifications to translate any 5010 EDI message into a CSV, XML or database table; every business element is assigned to a unique name that fully qualifies both the data element and its position within the hierarchical EDI format. This format is called the *Chiapas Gate Intermediate Format*, or CGIF. This is covered in the first chapter of the User Reference.

CGIF provides a layer of abstraction against the complexities of the 5010 format, and is similar to working with a data dictionary associated with a database system. In this case, the CGIF data dictionary defines 23,730 unique data elements for the 4010 and 5010 specifications. Once a business analyst becomes familiar with associating CGIF elements with the corresponding HIPAA elements, the processes of decoding incoming EDI messages as well as creating them become much simpler.

Once the logistics of the actual decode or encode is worked out using the front-end *Chiapas Studio*, the developer can then use the library of included automation components to create completely automated, hands-free business processes. Chiapas Studio is covered in the second chapter of the User Reference, and the automation components are covered in the third chapter.

CHIAPAS GATE INTERMEDIATE FORMAT

To understand how this naming convention functions, it's first necessary to review the fundamental structure of any HIPAA transaction and how these structures are used in modern healthcare communications. At the highest level of organization, all HIPAA transactions are grouped by specifications. These specifications are defined in the HIPAA Implementation Guides are sold and published by the Washington Publishing Corporation (WPC) on behalf of the ASC X12 organization, and are divided into two categories, 4010 and 5010. Each of these categories defines a broad spectrum of healthcare communication specifications, such as claims, authorizations, enrollments and remittance advice. The 4010 set of specifications were finalized in 2002, whereas the 5010 specifications were finalized in 2008. Many of the transactions have one or two formally published Addenda that materially impact the specification.

Within each specification, information is grouped into hierarchical structures called loops. Loops are composed of segments, which are themselves made of elements. In certain cases, elements are further divided into composite elements. At every level, numeric and alphabetical text qualifiers are used to identify the information being described within the loops and segments. The specifications define every element possible with a given transaction, but trading partners generally need to exchange only a small fraction of these elements to meet their business requirements. Often, at least one partner will formalize the required mappings into a *Companion Guide*, which serves as an easy point of reference about what information is being exchanged.

Hierarchical protocols like XML and HIPAA are generally used to transmit groups of information in batches. However, within healthcare enterprise systems, healthcare data is rarely stored this way – most often, it lays embedded within relational databases where information is stored in columns and rows. Transforming relational healthcare information like claims and member information into a hierarchical format like 5010, and vice versa, is the primary focus of this software package.

The *Chiapas Gate Intermediate Format*, or CGIF, represents the syntax that Chiapas EDI uses to identify HIPAA elements and assign human-understandable names to them. Furthermore, it uses a number of prefixes to fix the location of an element to a specific specification, loop and segment. When Chiapas EDI is decoding a hierarchical EDI message structure to a two dimensional format, it assigns a new column to every unique data element encountered. Every row represents a new iteration of what is called the *deepest key* – meaning, the deepest loop containing repeating information within the transaction. For an 837 Professional or Institutional transaction, the deepest key would be the Claims Detail loop, where every row represents a new service line within a claim. Within Chiapas EDI, every loop defined within the HIPAA Implementation Guides that has 24 or more possible iterations is mapped *vertically* as new rows. Loops containing 23 or less possible iterations are mapped *horizontally* as additional columns. Therefore, the business elements sent via the Claim Adjudication Loop (2430, 5010 837 Institutional) will all be mapped as additional columns existing on every row because the HIPAA Implementation Guides defines that loop as having 15 iterations or less.

The columns associated with this deepest key loop will contain unique business information for every row within the transformed output, and columns associated to loops higher in the hierarchy would inherit the information given in the previous row. When converting this two-dimensional structure back to an EDI message format, each successive row is scanned for changes from the previous row and this would in turn trigger a new iteration of a loop. In this way, a two-dimensional tabular data format is completely interchangeable with a hierarchical data format. Because the two-dimensional format is amenable to normal SQL, all of the examples are oriented around interfacing with Chiapas EDI 2012 in this manner. For an example, see the following HIPAA EDI message and the equivalent CGIF flat table.

```
ISA*00*..........*00*..........*ZZ*SAMPL_SENDRID..*ZZ*SAMPL_RECVRID..*120906*
1013*^*00501*100000000*0*P*:
 GS*RA*SENDR_ID*RECVR_ID*20120906*1013*1*X*005010X218
  ST*820*100000000*005010X218
   BPR*I*0*C*CHK******1273801950******20120906
   TRN*3*753476
   N1*PE*DOOLITTLE HOSPITAL*FI*123400001
   N1*PR*CHIAPAS EDI TECH*FI*273801950
   ENT*1*2J*EI*987654324
    NM1*IL*1*STROGANOFF*ABIGAIL****N*00000005
    RMR*AZ*5**0
   ENT*2*2J*EI*987654325
    NM1*IL*1*PEAS*DELORES****N*00000006
    RMR*AZ*6**0
   ENT*3*2J*EI*987654326
    NM1*IL*1*HASHBROWNS*PENELOPE****N*00000007
    RMR*AZ*7**0
   ENT*4*2J*EI*987654327
    NM1*IL*1*PASTA*EDWIN****N*00000008
    RMR*AZ*8**0
   ENT*5*2J*EI*987654332
    NM1*IL*1*CHIPOTLE*AARON****N*00000013
    RMR*AZ*13**0
  SE*21*100000000
 GE*1*1
IEA*1*100000000
```

L1000A_S01_N102_LAST_ORG_NM	L1000A_S01_N104_PREM_RECVR_ID_CD_OVL_FED_TAX_ID_NR	L1000B_S01_N102_PAYR_NM	L1000B_S01_N104_PAYR_ID_OVL_FED_TAX_ID_NR	L2000B_S01_ENT01_ASGND_NR	L2000B_S01_ENT04_RECVR_INDVDL_ID_OVL_EMPLY_ID_NR	L2100B_IL_S01_NM103_INDVDL_LNAME	L2100B_IL_S01_NM104_INDVDL_FNAME	L2100B_IL_S01_NM109_INDVDL_ID_OVL_INSRD_UNQ_ID_NR	L2300B_S01_RMR02_INS_REMTNC_REF_NR_OVL_HEALTH_INS_POLCY_NR	L2300B_S01_RMR04_DETL_PREM_PMT_AMT
DOOLITTLE HOSPITAL	123400001	CHIAPAS EDI TECH	273801950	1	987654324	STROGANOFF	ABIGAIL	00000005	5	0
DOOLITTLE HOSPITAL	123400001	CHIAPAS EDI TECH	273801950	2	987654325	PEAS	DELORES	00000006	6	0
DOOLITTLE HOSPITAL	123400001	CHIAPAS EDI TECH	273801950	3	987654326	HASHBROWNS	PENELOPE	00000007	7	0
DOOLITTLE HOSPITAL	123400001	CHIAPAS EDI TECH	273801950	4	987654327	PASTA	EDWIN	00000008	8	0
DOOLITTLE HOSPITAL	123400001	CHIAPAS EDI TECH	273801950	5	987654332	CHIPOTLE	AARON	00000013	13	0

Now that we have an idea about how Chiapas EDI 2012 works at a high level to translate between EDI messages and database objects, we need to review the exact specifics of how CGIF works. One of the primary focuses for CGIF is to make a strong division between *metadata* and business data. Metadata, which belong to the elements highlighted in orange in the HIPAA sample, identifies the data being encoded and is part of the complex encoding standard defined in 5010. Chiapas EDI 2012 uses a special syntax within the column name itself in lieu of metadata to discretely define the data being encoded which defines the loop, element and possibly the composite element. Following that is a condensed version of the Implementation Name for that element, which is the real business purpose of that element. Not shown in this example is a special prefix for the very first column in any CGIF table that defines the exact specification being used – 4010 837P A1, 5010 278 Request, etc.

Here are four different examples that describe how Chiapas EDI 2012 associates a business element to a HIPAA element.

Scenario 1: Patient Account Number

In this example, we want to describe the first element within the Claim segment of the Claim Information loop within an 837 Institutional HIPAA transaction. In this scenario, the loop name is followed by an 'S' and the index to the specified segment, followed by the two to three digit segment code, the element number, and the name of the element.

```
L2300_S01_CLM01_PATNT_ACCT_NR
```

Spelled out, this describes a complete path to the element: Loop 2300, Segment 01, 'CLM' segment Element 01, 'Patient Account Number'.

Scenario 2: Claim Note Segment Iterations

The twenty-fourth segment in the 2300 loop is the Claim Note segment, which can repeat up to ten times. Furthermore, Claim Notes are divided into a number of categories, including Medications, Diagnosis Description, Durable Medical Equipment / Supplies and so on. In CGIF, any information that is qualified by a value in a preceding element is called an overload, and all overloads have their own discrete mapping. These mappings would reference the first three occurrences of a Claim Note segment containing information on Durable Medical Equipment / Supplies:

```
L2300_S24_NOT02_TEXT_OVL_DME_SUPLS

L2300_S24_NOT02_I02_TEXT_OVL_DME_SUPLS

L2300_S24_NOT02_I03_TEXT_OVL_DME_SUPLS
```

The I02 and I03 represent the second and third iterations, respectively, whereas the OVL represents 'Overload'. If these mappings were used to encode a HIPAA transaction and NULL values were stored in these columns (meaning, 'no information present'), then no segments would be generated.

Scenario 3: TPA/Broker Named Loops

The HIPAA Implementation Guides will often 'overload' entire loops, redefining a single loop to have a number of different meanings. CGIF independently addresses each named iteration by combining the loop name with the metadata element present within the header segment of that loop.

One such example is the TPA/Broker Name (Loop 1000C) within the 4010 834 Eligibility specification. It contains a definition for both 'Broker or Sales Office' (element 1 containing 'BO') or 'Third Party Administrator' (element 1 containing 'TV'). To define both the name and the required Broker Identification Code, the following four mappings are used:

```
L1000C_BO_S01_N102_BROKR_NM

L1000C_BO_S01_N104_TPA_BROKR_ID_OVL_FEDRL_TAXPYR_ID_NR

L1000C_TV_S01_N102_BROKR_NM

L1000C_TV_S01_N104_TPA_BROKR_ID_OVL_FEDRL_TAXPYR_ID_NR
```

Scenario 4: Other Subscriber Information Iterated Loops

In HIPAA transactions, new database rows are triggered by the deepest key loop. This is the deepest loop in the transaction containing 24 or more iterations, and is typically the most fundamental required data segment for that specification. For a Claims file, this is the Claim Line segment, and for eligibility transactions this would be a 2300 Health Coverage loop. This is similar to the logical arrangement of data in most databases and data warehouses.

In certain scenarios, loops will repeat to convey information associated with a claim and not with a specific claim line. Chiapas Gate maps these loops laterally, and assigns an iteration counter much like the one used to convey repeating segments. In this case however, the iteration counter will apply to both the loop and children loops. In this way, both a loop and its associated child loops are all bundled together.

One example where this can occur is with Loop 2320, Other Subscriber Information inside an 837 Institutional claim transaction. Each of the ten loop iterations can themselves have up to eight different single-iteration children loops. To encapsulate this complexity, an iteration counter is used to group the whole batch of loops and sub-loops together.

```
L2330A_I02_S03_N403_OTHR_INSRD_POSTL_ZON_ZIP_CD
```

This mapping represents the Other Subscriber ZIP Code belonging to a single instance of the 2330A Other Subscriber Name loop which is a child under the second iteration of its repeating parent loop, 2320 Other Subscriber Information.

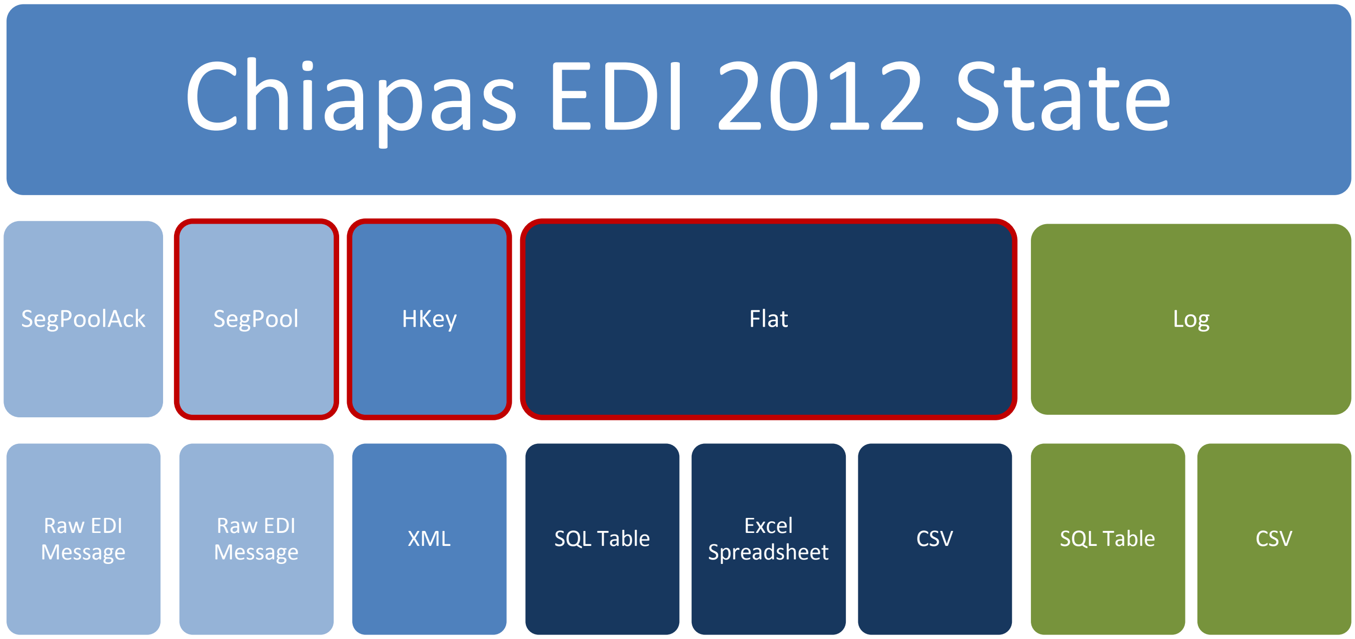
CHIAPAS REGISTERS

Internally, Chiapas EDI 2012 contains a 'State' object that is instanced every time Chiapas Studio is opened, or a new Automation activity is run. This State object contains five registers:

SegPool – This contains a representation of a raw 4010/5010 EDI message. It is a container for the elements and segments as well as basic information like line termination, element, and composite element and segment separators. When an EDI message is loaded, it will automatically set the values for the separators. When set by the user, these values will be ignored until the HKey Encode action is executed.

SegPoolAck – This is a container that contains the segments and elements for a 999 Acknowledgement transactions. It can be generated once a decode operation is complete to show the status of the decode operation, which can then be saved to a new 999 file and returned to a trading partner to show successful receipt of their file. It can also be loaded and used in Chiapas Studio to indicate the error messages within a file.

HKey – This is a hierarchical representation of the business data within an X12 message. Internally, the HKey register is divided by *loops*, with the mappings in each loop iteration aggregated together. Each iteration also has an index to the iteration of the *parent* loop. Thus, each HKey stores containers for the ISA envelope loop, the GS loop, the ST loop, and one for each discrete loop that appears in the transaction.



Flat – This is a tabular, column/row representation of business data. Each discrete mapped element is represented via a unique column that includes hierarchy positioning information embedded within the name. The first column name in any Flat key also contains an extra prefix that describes which HIPAA specification this data belongs to.

Log – This is an internal list of messages generating during all EDI parsing activities. It is an output-only register.

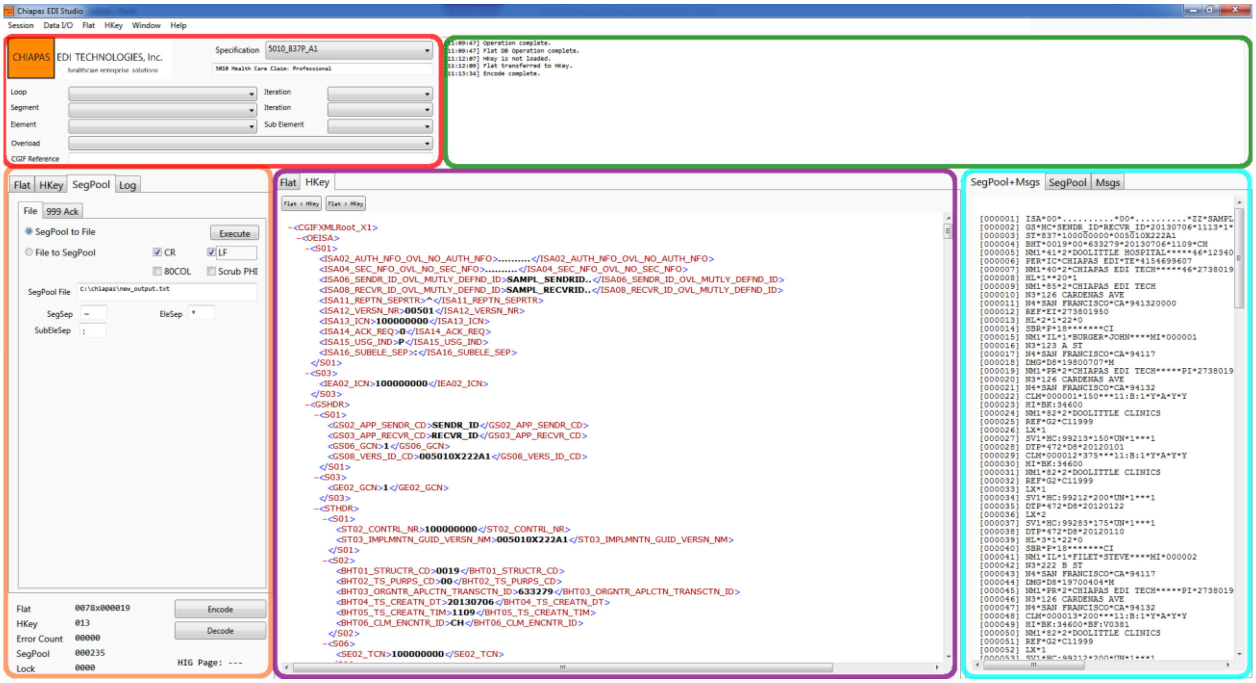
In the above diagram, the five registers are listed in the second row, with the next row showing the data formats usable with that register. The three highlighted registers can be transformed to a neighboring register. For example, the SegPool register can be loaded from a Raw EDI message, and then *decoded* into the HKey register. Transforming the HKey to the SegPool register is called *encoding*. Data within the HKey register can be shifted to the Flat register, where it can then be saved to a SQL database table, Excel spreadsheet or CSV file.

Chiapas EDI 2012 does not handle business-level translations – for example, transforming a raw eight-character date like '20120101' into a database Date column in a database. Instead, it presents a fairly straightforward CGIF interface where internal business data can easily be projected into a form that Chiapas EDI 2012 can transform into HIPAA files. When decoding incoming EDI files, the data can be put into a database form and queried very quickly. In this way, the business logic is isolated from the actual creation or decoding of a raw EDI file.

CHIAPAS STUDIO OVERVIEW

The purpose of Chiapas Studio is to provide a workspace for building a data flow from the desired source to destination. It's typically used in two ways. The first way is as a tool to quickly check the syntax and data integrity of production EDI files that have some anomaly. The second way is as a development tool to develop the mappings and ETL logic. With both Chiapas Studio and SQL Server Management Studio (SSMS) open, a developer can develop interfaces between enterprise systems and EDI files.

The screen is split into a number of different functional areas. The Data I/O Interface (outlined in pink) gives methods for loading and saving business data. The Registers Interface (outlined in purple) represents the two main registers within Chiapas Gate – the Flat and HKey registers. The SegPool Interface on the right (outlined in cyan) represents the SegPool object. In its default state, it will also display any integrity messages associated with those segments. The messages and segments can be viewed separately in different tabs. A log of the users' actions is kept in the upper right (outlined in cyan), and to the left of that is the Mapping Interface dealing with CGIF mappings (outlined in orange). The specification, Loop, Loop Iteration, Segment, Segment Iteration, Element, Sub-Element and Overload can all be individually selected. Selecting each combo box will re-populate the combo boxes below it in sequence, which allows any of the more than 17,000 mappings to be selected and mapped. At the bottom of the Data I/O Interface is the Information Display, which shows information



Specification: 5010_837P_A1

5010 Health Care Claim: Professional

```
[11:09:47] Operation complete.
[11:09:47] Flat DB Operation complete.
[11:12:07] HKey is not loaded.
[11:12:09] Flat transferred to HKey.
[11:13:34] Encode complete.
```

● SegPool to File ○ File to SegPool ☑ CR ☑ LF ☐ 80COL ☐ Scrub PHI

SegPool File: C:\chiapas\new_output.txt

SegSep: ~ EleSep: * SubEleSep: :

Flat	0078x000019
HKey	013
Error Count	00000
SegPool	000235
Lock	0000

HIG Page: ---

```
-<CGIFXMLRoot_X1>
  -<OEISA>
    -<S01>
       <ISA02_AUTH_NFO_OVL_NO_AUTH_NFO>..........</ISA02_AUTH_NFO_OVL_NO_AUTH_NFO>
       <ISA04_SEC_NFO_OVL_NO_SEC_NFO>..........</ISA04_SEC_NFO_OVL_NO_SEC_NFO>
       <ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID>SAMPL_SENDRID..</ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID>
       <ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID>SAMPL_RECVRID..</ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID>
       <ISA11_REPTN_SEPRTR>^</ISA11_REPTN_SEPRTR>
       <ISA12_VERSN_NR>00501</ISA12_VERSN_NR>
       <ISA13_ICN>100000000</ISA13_ICN>
       <ISA14_ACK_REQ>0</ISA14_ACK_REQ>
       <ISA15_USG_IND>P</ISA15_USG_IND>
       <ISA16_SUBELE_SEP>:</ISA16_SUBELE_SEP>
     </S01>
    -<S03>
       <IEA02_ICN>100000000</IEA02_ICN>
     </S03>
    -<GSHDR>
      -<S01>
         <GS02_APP_SENDR_CD>SENDR_ID</GS02_APP_SENDR_CD>
         <GS03_APP_RECVR_CD>RECVR_ID</GS03_APP_RECVR_CD>
         <GS06_GCN>1</GS06_GCN>
         <GS08_VERS_ID_CD>005010X222A1</GS08_VERS_ID_CD>
       </S01>
      -<S03>
         <GE02_GCN>1</GE02_GCN>
       </S03>
      -<STHDR>
        -<S01>
           <ST02_CONTRL_NR>100000000</ST02_CONTRL_NR>
           <ST03_IMPLMNTN_GUID_VERSN_NM>005010X222A1</ST03_IMPLMNTN_GUID_VERSN_NM>
         </S01>
        -<S02>
           <BHT01_STRUCTR_CD>0019</BHT01_STRUCTR_CD>
           <BHT02_TS_PURPS_CD>00</BHT02_TS_PURPS_CD>
           <BHT03_ORGNTR_APLCTN_TRANSCTN_ID>633279</BHT03_ORGNTR_APLCTN_TRANSCTN_ID>
           <BHT04_TS_CREATN_DT>20130706</BHT04_TS_CREATN_DT>
           <BHT05_TS_CREATN_TIM>1109</BHT05_TS_CREATN_TIM>
           <BHT06_CLM_ENCNTR_ID>CH</BHT06_CLM_ENCNTR_ID>
         </S02>
        -<S06>
           <SE02_TCN>100000000</SE02_TCN>
```

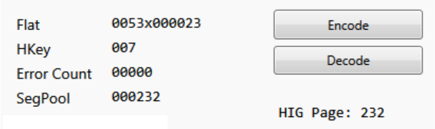
```
[000001] ISA*00*..........*00*..........*ZZ*SAMPL
[000002] GS*HC*SENDR_ID*RECVR_ID*20130706*1113*1*
[000003] ST*837*100000000*005010X222A1
[000004] BHT*0019*00*633279*20130706*1109*CH
[000005] NM1*41*2*DOOLITTLE HOSPITAL*****46*12340
[000006] PER*IC*CHIAPAS EDI*TE*4154699607
[000007] NM1*40*2*CHIAPAS EDI TECH*****46*2738019
[000008] HL*1**20*1
[000009] NM1*85*2*CHIAPAS EDI TECH
[000010] N3*126 CARDENAS AVE
[000011] N4*SAN FRANCISCO*CA*941320000
[000012] REF*EI*273801950
[000013] HL*2*1*22*0
[000014] SBR*P*18*******CI
[000015] NM1*IL*1*BURGER*JOHN****MI*000001
[000016] N3*123 A ST
[000017] N4*SAN FRANCISCO*CA*94117
[000018] DMG*D8*19800707*M
[000019] NM1*PR*2*CHIAPAS EDI TECH*****PI*2738019
[000020] N3*126 CARDENAS AVE
[000021] N4*SAN FRANCISCO*CA*94132
[000022] CLM*000001*150***11:B:1*Y*A*Y*Y
[000023] HI*BK:34600
[000024] NM1*82*2*DOOLITTLE CLINICS
[000025] REF*G2*C11999
[000026] LX*1
[000027] SV1*HC:99213*150*UN*1***1
[000028] DTP*472*D8*20120101
[000029] CLM*000012*375***11:B:1*Y*A*Y*Y
[000030] HI*BK:34600
[000031] NM1*82*2*DOOLITTLE CLINICS
[000032] REF*G2*C11999
[000033] LX*1
[000034] SV1*HC:99212*200*UN*1***1
[000035] DTP*472*D8*20120122
[000036] LX*2
[000037] SV1*HC:99283*175*UN*1***1
[000038] DTP*472*D8*20120110
[000039] HL*3*1*22*0
[000040] SBR*P*18*******CI
[000041] NM1*IL*1*FILET*STEVE****MI*000002
[000042] N3*222 B ST
[000043] N4*SAN FRANCISCO*CA*94117
[000044] DMG*D8*19700404*M
[000045] NM1*PR*2*CHIAPAS EDI TECH*****PI*2738019
[000046] N3*126 CARDENAS AVE
[000047] N4*SAN FRANCISCO*CA*94132
[000048] CLM*000013*200***11:B:1*Y*A*Y*Y
[000049] HI*BK:34600*BF:V0381
[000050] NM1*82*2*DOOLITTLE CLINICS
[000051] REF*G2*C11999
[000052] LX*1
[000053] SV1*HC:99212*200*UN*1***1
```

about currently loaded registers. It also has the Encode and Decode buttons. Encode will translate the HKey register to the SegPool register, and Decode will translate the SegPool register to the HKey register.

The green area is the Message Log, which keeps a running log of Chiapas Studio operations.

Data I/O Interface

This part of Chiapas Studio is focused on the import and export of data in a variety of data formats. The top row of tabs represent the Chiapas EDI registers, with the sub-tabs below them representing discrete interfaces for each specific register.



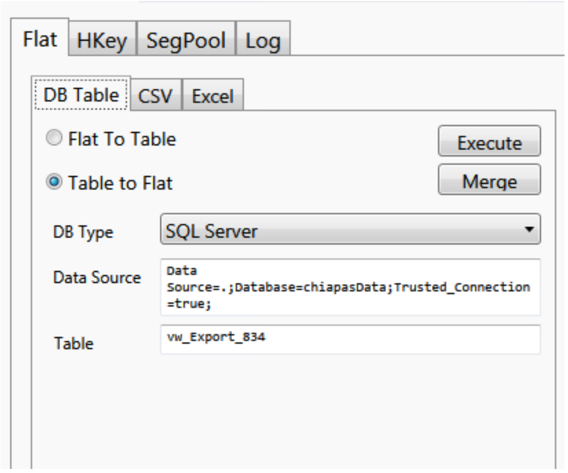
Flat 0053x000023
HKey 007
Error Count 00000
SegPool 000232
HIG Page: 232

At the bottom of the Data I/O Interface area is a small information pane showing the current state of the various Chiapas EDI 2012 registers. The Encode button will encode a loaded HKey register to the SegPool register. The Decode button will decode a loaded SegPool register to the HKey register.

Flat Register Tab

This pane presents interfaces for loading Flat data and exporting to one of the supported destinations.

DB Table



○ Flat To Table
● Table to Flat
DB Type: SQL Server
Data Source: Data Source=.;Database=chiapasData;Trusted_Connection=true;
Table: vw_Export_834

This pane enables the user to import data from or export data to a SQL database. Interfaces for ODBC, Ole, SQL Server and SQLite are presented. If the database type is SQLite or SQL Server, it will detect if a table exists and if not, automatically create it when exporting data, with all fields having the VARCHAR(200) data type.

The Data Source field must be populated with a valid connection string, with the table representing a table or view for importing, or a table name for export. For a good resource containing examples of valid SQL Server connection strings, visit http://connectionstrings.com .

Flat to Table – Exports loaded Flat register data to a database table.

Table to Flat – Loads Flat register from a database table. Note that the table must only contain CGIF compatible field names.

DB Type – Pull down to select the database connection type

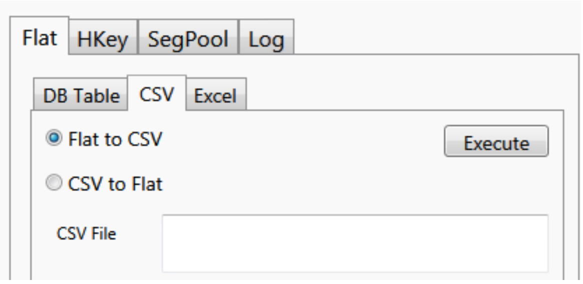
Data Source – A valid SQL Connection String

Table – For exports, this will contain the name of the database table to export records to; if it doesn't exist, a table will be created as described above. For data imports, this can be the name of a table, a view, or the SQL of a SELECT statement.

Execute Button – When all parameters are filled in, this will actually run the export or import. Any errors will be reported in the Message Log area.

Merge Button – This feature is available only for SQL Server or SQLite connection strings, and enables a special Export functionality where the loaded DataKey register will be merged into an existing database table. Only the CGIF fields common to both the source and destination will be inserted to the backend table; CGIF fields present on the source DataKey and not present within the destination table will be flagged as errors in the Error Log.

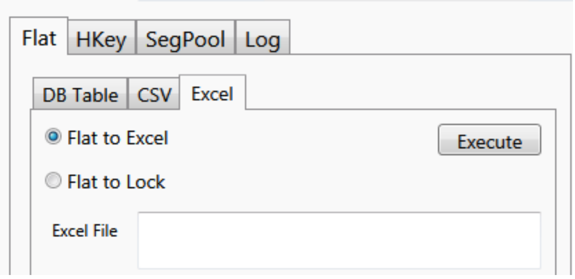
CSV



● Flat to CSV
○ CSV to Flat

This pane Exports a loaded DataKey to a headered CSV file, or loads it in from the same. The CSV File area can be double-clicked to bring up an Open File Dialog. Once a valid filename is selected, the Execute button will carry out the operation.

Excel



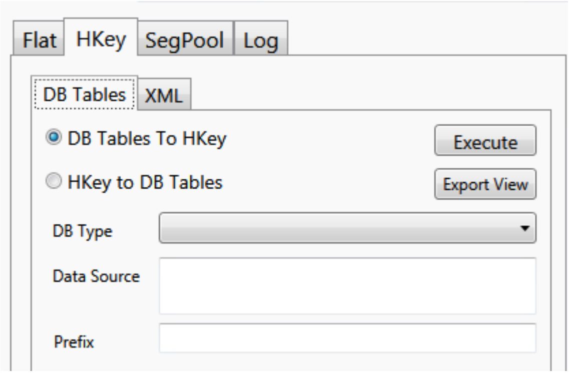
● Flat to Excel
○ Flat to Lock

Similar to the CSV Data I/O functionality, this tab will export a loaded DataKey to a new Excel spreadsheet. The extension chosen for the destination filename determines whether it is an Excel '97 or an Excel 2003 format, determined by either an .xls extension ('97) or an .xlsx (2003) extension.

HKey Register Tab

The HKey tab enables transport of the HKey register. The HKey register encodes hierarchical data that follows the HIPAA Implementation Guide specifications, and presents two transports for importing and exporting this data. These transports give developers alternatives to using the Flat register interface for working with HIPAA files.

DB Tables



This tab presents a transport between SQL Server database tables and the HKey register. When exporting a loaded HKey via the Hkey to DB Tables radio button, Chiapas EDI 2012 will create a number of tables with the specified prefix, one for each loop present within the transaction.

To see how this works in practice, let's see an example. To replicate this example, load in the include 'seed_834.txt' sample file into the SegPool register, press the Decode button, and then select 'HKey to DB Tables', fill in SQL Server as DB Type, a valid Data Source, and enter in 'PL' as the prefix. Then, press Execute.

The resulting exported tables will appear like this:

```
PLISA Table:
     T1_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO
     OEISA_S01_ISA04_SEC_NFO_OVL_PASSWD
     OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID
     OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID
     OEISA_S01_ISA09_DT
     OEISA_S01_ISA10_TM
     OEISA_S01_ISA11_REPTN_SEPRTR
     OEISA_S01_ISA12_VERSN_NR
     OEISA_S01_ISA13_ICN
     OEISA_S01_ISA14_ACK_REQ
     OEISA_S01_ISA15_USG_IND
     OEISA_S01_ISA16_SUBELE_SEP
     OEISA_S03_IEA01_GS_CT
     OEISA_S03_IEA02_ICN
     KEY_ID
     PAR_KEY_ID
```

```
PLGSHDR Table:
      GSHDR_S01_GS02_APP_SENDR_CD
      GSHDR_S01_GS03_APP_RECVR_CD
      GSHDR_S01_GS04_DT
      GSHDR_S01_GS05_TM
      GSHDR_S01_GS06_GCN
      GSHDR_S01_GS08_VERS_ID_CD
      GSHDR_S03_GE01_TS_CT
      GSHDR_S03_GE02_GCN
      KEY_ID
      PAR_KEY_ID

PLSTHDR Table:
      STHDR_S01_ST02_CONTRL_NR
      STHDR_S01_ST03_IMPLMNTN_CONVNTN_REFNC
      STHDR_S02_BGN01_TS_PURPS_CD
      STHDR_S02_BGN02_TS_REF_NR
      STHDR_S02_BGN03_TS_CREATN_DT
      STHDR_S02_BGN04_TS_CREATN_TIM
      STHDR_S02_BGN08_ACTN_CD
      STHDR_S10_SE01_SEG_CT
      STHDR_S10_SE02_TCN
      KEY_ID
      PAR_KEY_ID
```

In addition, the tables PL1000A, PL1000B, PL2000, and PL2100A are all created and populated. When we examine the 834 HIPAA Implementation Guide, we would see the following relationship between the loops:

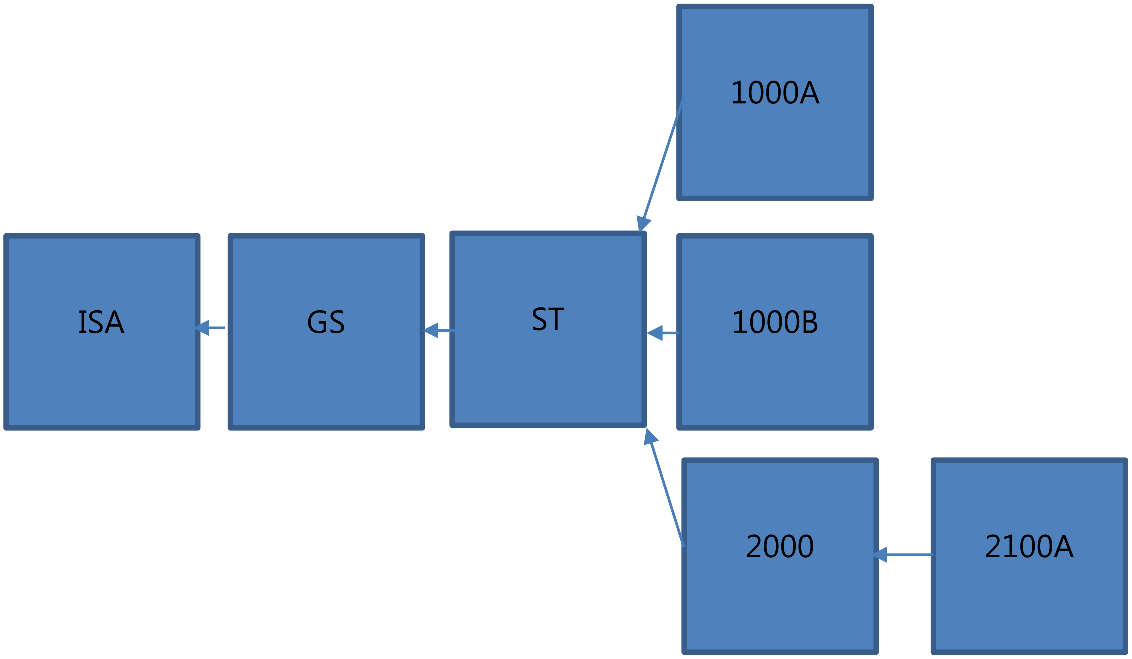
These relationships describe the 'PAR_KEY_ID' present within each row in each table, it being the KEY_ID of parent loop iteration. When there is a complicated transaction, however, it can be difficult to establish these relationships correctly. This is the function of the Export View button – it will create a view on the specified destination database called 'vw_HKey_<Prefix>'. This view projects the hierarchical data into a two dimensional format, the same as the Flat register.

DB Tables to HKey – This option tells Chiapas EDI to scan the source database for tables matching both the specified prefix and each of the loops that belong to the specification indicated in the first ISA loop mapping, and then load the HKey data from these tables.

HKey to DB Tables – As indicated above, Chiapas EDI storages all HKey register data to a number of database tables. All fields will have type VARCHAR(200), and every row contains both the KEY_ID (row index) and PAR_KEY_ID (parent row index).

DB Type – Only SQL Server and Sqlite are supported for these operations; other DB types will generate an error.

Data Source – Database connection string

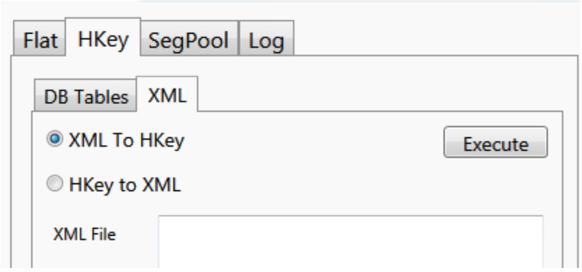


Prefix – Several letters that will prefix the database table names, as described above.

Execute Button – Runs the specified action against the database.

Export View – Creates a SQL View on the destination database that links all of the loaded HKey loop tables together to project a view of the table that can be loaded back into Chiapas EDI via the Flat key interface. This view could also be used by a developer as a guide to reformat various internal data sources needed for different loops to present a correct CGIF representation back to Chiapas EDI.

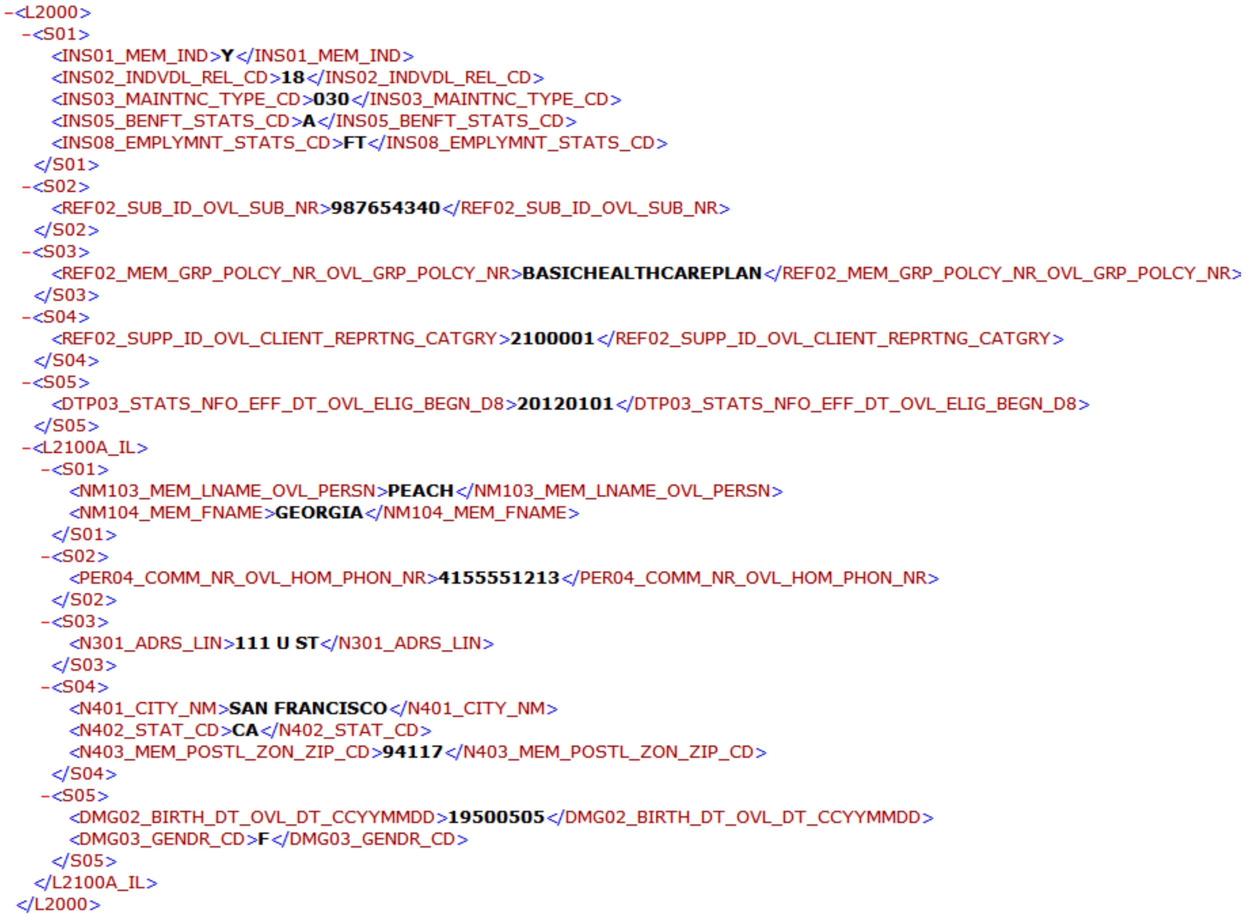
XML



● XML To HKey

○ HKey to XML

This interface gives an XML transport for the HKey register. By default, whenever the HKey register is loaded, an XML view is presented in the user interface. An example of an iteration of an 5010 834 eligibility loop follows:



```
-<L2000>
  -<S01>
     <INS01_MEM_IND>Y</INS01_MEM_IND>
     <INS02_INDVDL_REL_CD>18</INS02_INDVDL_REL_CD>
     <INS03_MAINTNC_TYPE_CD>030</INS03_MAINTNC_TYPE_CD>
     <INS05_BENFT_STATS_CD>A</INS05_BENFT_STATS_CD>
     <INS08_EMPLYMNT_STATS_CD>FT</INS08_EMPLYMNT_STATS_CD>
   </S01>
  -<S02>
     <REF02_SUB_ID_OVL_SUB_NR>987654340</REF02_SUB_ID_OVL_SUB_NR>
   </S02>
  -<S03>
     <REF02_MEM_GRP_POLCY_NR_OVL_GRP_POLCY_NR>BASICHEALTHCAREPLAN</REF02_MEM_GRP_POLCY_NR_OVL_GRP_POLCY_NR>
   </S03>
  -<S04>
     <REF02_SUPP_ID_OVL_CLIENT_REPRTNG_CATGRY>2100001</REF02_SUPP_ID_OVL_CLIENT_REPRTNG_CATGRY>
   </S04>
  -<S05>
     <DTP03_STATS_NFO_EFF_DT_OVL_ELIG_BEGN_D8>20120101</DTP03_STATS_NFO_EFF_DT_OVL_ELIG_BEGN_D8>
   </S05>
  -<L2100A_IL>
    -<S01>
       <NM103_MEM_LNAME_OVL_PERSN>PEACH</NM103_MEM_LNAME_OVL_PERSN>
       <NM104_MEM_FNAME>GEORGIA</NM104_MEM_FNAME>
     </S01>
    -<S02>
       <PER04_COMM_NR_OVL_HOM_PHON_NR>4155551213</PER04_COMM_NR_OVL_HOM_PHON_NR>
     </S02>
    -<S03>
       <N301_ADRS_LIN>111 U ST</N301_ADRS_LIN>
     </S03>
    -<S04>
       <N401_CITY_NM>SAN FRANCISCO</N401_CITY_NM>
       <N402_STAT_CD>CA</N402_STAT_CD>
       <N403_MEM_POSTL_ZON_ZIP_CD>94117</N403_MEM_POSTL_ZON_ZIP_CD>
     </S04>
    -<S05>
       <DMG02_BIRTH_DT_OVL_DT_CCYYMMDD>19500505</DMG02_BIRTH_DT_OVL_DT_CCYYMMDD>
       <DMG03_GENDR_CD>F</DMG03_GENDR_CD>
     </S05>
   </L2100A_IL>
 </L2000>
```

Note that this is not a complete file, but this snippet serves as a guide to how the names of CGIF elements are split into XML element names.

XML To HKey – This option causes Chiapas EDI to load the XML file into the HKey register. The XML elements must follow a specific naming convention as described in the example above, with loops, segments and elements as three different hierarchical levels.

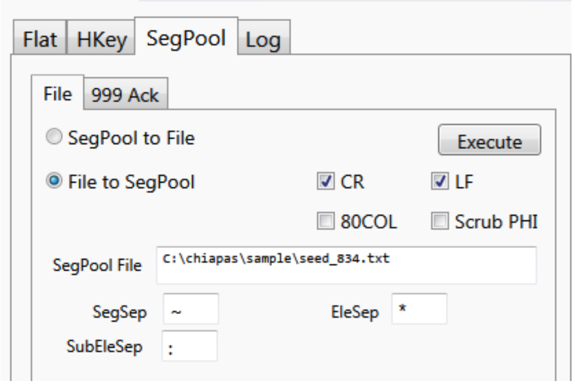
HKey to XML – Exports the HKey register to an XML file.

XML File – XML file to be loaded or saved.

SegPool Register Tab

The SegPool interface presents an interface for loading HIPAA transaction text files into the SegPool register, or saving a loaded SegPool register to a text file. By HIPAA rules, these files may contain Carriage Return / Line Feed information, different segment, element and composite element separators, and Chiapas EDI also allows a file to be exported with 80 column fixed line format. Finally, exported files may be ‘sanitized’ of Protected Healthcare Information using a simple algorithm.

File



○ SegPool to File
● File to SegPool
☑ CR
☑ LF
□ 80COL
□ Scrub PHI

SegPool File: C:\chiapas\sample\seed_834.txt
SegSep: ~
EleSep: *
SubEleSep: :

SegPool to File – Exports a loaded SegPool Register to a file with the given separator characters and formatting options.

File to SegPool – Loads the SegPool register from a specified file, and automatically fills in the formatting options according to the file contents.

CR – If checked, the file will have a Carriage Return (ASCII 13) at the end of each segment.

LF – If checked, the file will have a Line Feed (ASCII 10) at the end of each segment.

80 COL – Setting this checkbox will set the CR/LF line terminators at 80 character boundaries.

Scrub PHI - Setting this option before exporting will scrub it of Protected Healthcare Information. Dates, Names and Identifiers will be altered from the originals. Note that this is merely a ‘best guess’ to filter out the most commonly known pieces of publically identifiable information; the only way to guarantee a file contains no PHI is to scrub all identifiers manually.

SegPool File – Filename of the HIPAA text file to be created or loaded.

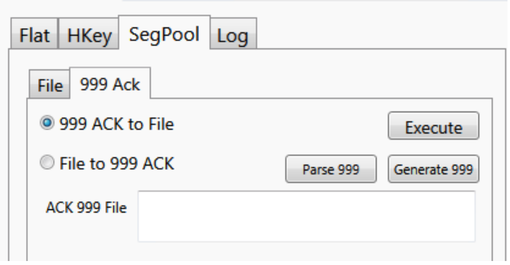
SegSep – A single character representing the Segment Separator that marks the boundary between segments.

EleSep – A single character representing the Element Separator that marks the boundary between elements.

SubEleSep – A single character representing the composite element separator that marks the boundary between composite elements. Not every transaction contains composite elements.

999 Ack

Chiapas EDI 2012 has limited built-in support for 999 transactions. A 999 transaction is generally a tool of clearinghouses to generate automated notifications notify of file rejection or integrity errors. Chiapas EDI 2012 can generate a 999 after a Decode event: either a simple 999 'Acceptance' record will be created if the file parsed without major errors, or it will be Rejected for a serious integrity violation. For incoming 999 messages, Chiapas EDI 2012 can apply the 999 against an existing loaded SegPool and generate messages that can be displayed alongside the segments in the SegPool display, allowing you to see which segments triggered the warnings that were listed within the 999.



● 999 ACK to File

○ File to 999 ACK

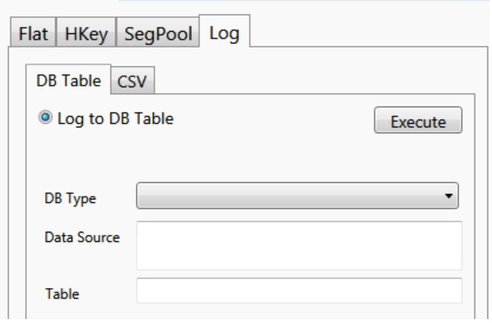
Parse 999 – This will load a 999 file and add messages to the active Message Log that will be displayed in the SegPool pane.

Generate 999 – This will generate a basic 999 acknowledgment file indicating 'Accepted' when a Decode event is able to parse a file, or Rejected in the case of a serious integrity error.

Execute – This will save a 999 file that was previously generated via the Generate 999 function, or it will load in 999 Transaction Acknowledgment file that was generated from a clearinghouse.

ACK 999 File – The file to be saved or loaded.

Log Register Tab



This Data I/O tab allows the internal Log register to be exported to a database table or a CSV file. This register tracks all messages and errors that can occur during an encode or decode process, including critical integrity errors or more minor syntax errors. The database format follows that of the MSGLOG table included in the ChiapasData database that is part of the distribution. If the table does not exist, it will be created. Note that unlike other Chiapas EDI registers, this is output-only and not a read register.

DB Table

Log to DB Table – The single option available for this read-only register.

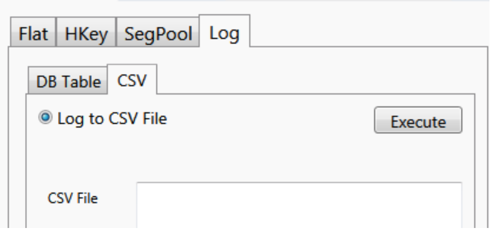
Execute – Exports the Log register.

DB Type – Database type to receive the Log register.

Data Source - Connection string to the receiving database.

Table – Name of the table to receive the Log register.

CSV



Log to CSV Register – The single option available for the read-only register.

Execute – Exports the Log register.

CSV File – CSV File to create when exporting the register.

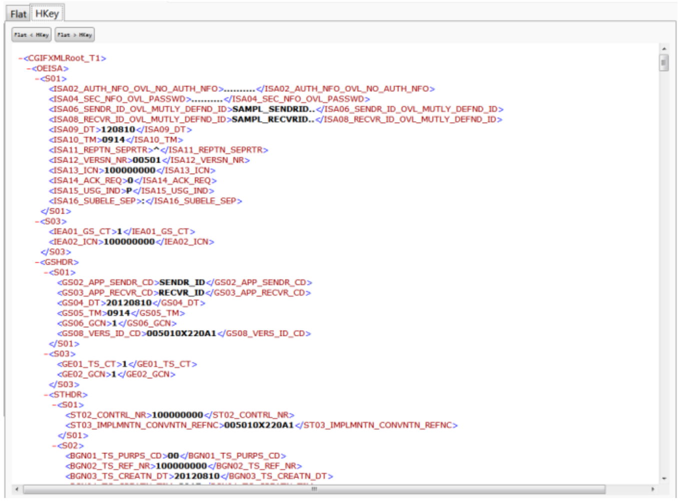
Registers Interface

The central two windows of Chiapas Studio are dedicated to providing a live view of the information contained within the Flat and HKey registers. Because these displays are memory intensive, they will not work to display data files larger than 10 MB, even if the register loads successfully.

<u>HKey Tab</u>

This tab displays information about a loaded HKey register. As described in a previous section, the HKey storages hierarchical business information and is an intermediary between the SegPool register (for interacting with HIPAA files) and the Flat register (for interacting with databases). The information is displayed on-screen in the form of an XML file. Furthermore, there are two controls that allow for translating the register to or from the Flat register.

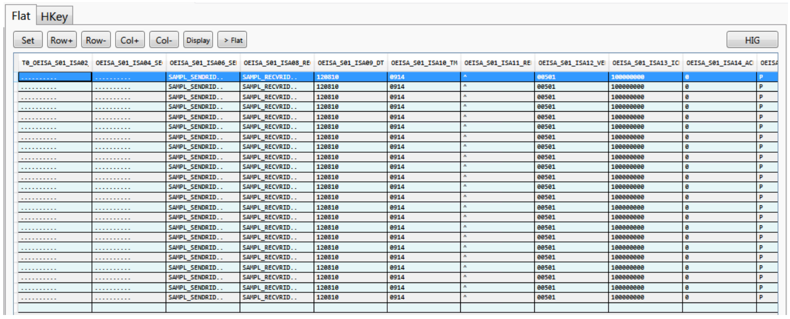
Flat < HKey – This button translates the HKey to the Flat register. Internally, Chiapas EDI will convert the HKey to a series of in-memory tables via SQLite and then ‘Flatten’ the data, and push the results to the Flat register. For a demonstration on how Chiapas EDI does this, refer to the documentation that covers Export View button within the Data I/O Interace.



```
-<CGIFXMLRoot_T1>
  -<OEISA>
    -<S01>
        <ISA02_AUTH_NFO_OVL_NO_AUTH_NFO>..........</ISA02_AUTH_NFO_OVL_NO_AUTH_NFO>
        <ISA04_SEC_NFO_OVL_PASSWD>..........</ISA04_SEC_NFO_OVL_PASSWD>
        <ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID>SAMPL_SENDRID..</ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID>
        <ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID>SAMPL_RECVRID..</ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID>
        <ISA09_DT>120810</ISA09_DT>
        <ISA10_TM>0914</ISA10_TM>
        <ISA11_REPTN_SEPRTR>^</ISA11_REPTN_SEPRTR>
        <ISA12_VERSN_NR>00501</ISA12_VERSN_NR>
        <ISA13_ICN>100000000</ISA13_ICN>
        <ISA14_ACK_REQ>0</ISA14_ACK_REQ>
        <ISA15_USG_IND>P</ISA15_USG_IND>
        <ISA16_SUBELE_SEP>:</ISA16_SUBELE_SEP>
      </S01>
    -<S03>
        <IEA01_GS_CT>1</IEA01_GS_CT>
        <IEA02_ICN>100000000</IEA02_ICN>
      </S03>
    -<GSHDR>
      -<S01>
          <GS02_APP_SENDR_CD>SENDR_ID</GS02_APP_SENDR_CD>
          <GS03_APP_RECVR_CD>RECVR_ID</GS03_APP_RECVR_CD>
          <GS04_DT>20120810</GS04_DT>
          <GS05_TM>0914</GS05_TM>
          <GS06_GCN>1</GS06_GCN>
          <GS08_VERS_ID_CD>005010X220A1</GS08_VERS_ID_CD>
        </S01>
      -<S03>
          <GE01_TS_CT>1</GE01_TS_CT>
          <GE02_GCN>1</GE02_GCN>
        </S03>
      -<STHDR>
        -<S01>
            <ST02_CONTRL_NR>100000000</ST02_CONTRL_NR>
            <ST03_IMPLMNTN_CONVNTN_REFNC>005010X220A1</ST03_IMPLMNTN_CONVNTN_REFNC>
          </S01>
        -<S02>
            <BGN01_TS_PURPS_CD>00</BGN01_TS_PURPS_CD>
            <BGN02_TS_REF_NR>100000000</BGN02_TS_REF_NR>
            <BGN03_TS_CREATN_DT>20120810</BGN03_TS_CREATN_DT>
```

Flat > HKey – This button loads the HKey from the Flat register – essentially a reverse of the process described above.

Flat Tab



T0_OEISA_S01_ISA02_	OEISA_S01_ISA04_SE	OEISA_S01_ISA06_SE	OEISA_S01_ISA08_RE	OEISA_S01_ISA09_DT	OEISA_S01_ISA10_TM	OEISA_S01_ISA11_RE	OEISA_S01_ISA12_VE	OEISA_S01_ISA13_IC	OEISA_S01_ISA14_AC	OEISA
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P
....	....	SAMPL_SENDRID..	SAMPL_RECVRID..	120810	0914	^	00501	100000000	0	P

This interface enables the user to view the Flat register interactively as well as modify both the mappings and the information contained within. This capability allows the user to 'test' different situations with data and mappings quickly. Note that generally, the Flat is loaded either from the Data I/O interface or from the HKey pane, and then this interface is updated with a snapshot of the data. Changing the data and mappings does not directly affect the Flat register; the > Flat and Set buttons are used to actually update the internal register.

Set - This button propagates the mapping indicated in the Mapping Interface to the currently selected column within the Flat interface.

+Row - This will clone the current row and insert a duplicate.

-Row - This will remove the currently selected row.

+Col - This will insert a new, 'Unmapped' column. You must manually 'Set' it to a valid mapping before this can be used with the Chiapas Gate engine for encoding.

-Col - Removes the currently selected column.

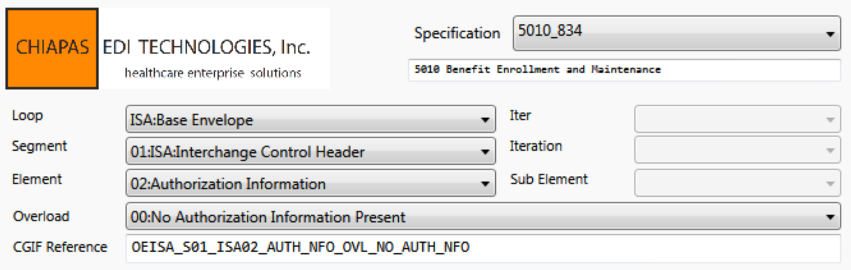
Display - This toggles the interface to display the entire mapping name in each column, or to keep them fixed length and display only the first 15 character or so of each column.

>Flat - This button updates the Flat register from the interface, so that the changes can be used with the rest of the system.

HIG - If the HIPAA Implementatio Guides have been properly set up, then this will launch Adobe Acrobat Reader and open the HIG to the correct page for the currently selected mapping within the Flat interface.

Mapping Interface

This section covers the functionality of the group of pull down menus in the upper-left corner of the Chiapas Studio screen. These menus are tied to the active mapping selected within the within the Flat Tab.



Specification: 5010_834

5010 Benefit Enrollment and Maintenance

Loop: ISA:Base Envelope

Segment: 01:ISA:Interchange Control Header

Element: 02:Authorization Information

Iter:

Iteration:

Sub Element:

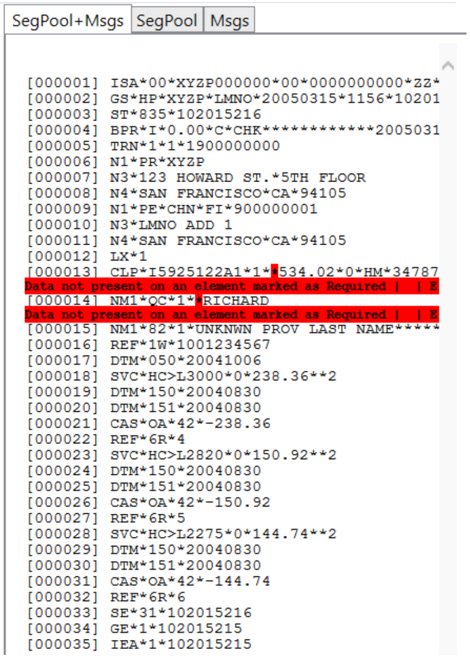
Overload: 00:No Authorization Information Present

CGIF Reference: OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO

The mapping interface has two functions. For existing mappings selected in the Flat or Lock panes, it displays verbose information about all the components of that mapping. Otherwise, it presents a straightforward interface for entering in new mappings. When entering new mappings, it's important to follow this sequence: Loop, Segment, Element, and Sub-Element. The selection of one combo box dynamically changes the contents in the next combo box in sequence, so this order is important.

When enough selections are made to generate a valid mapping, the CGIF name is populated in the CGIF Reference textbox.

Segpool Interface



```
[000001] ISA*00*XYZP000000*00*0000000000*ZZ*
[000002] GS*HP*XYZP*LMNO*20050315*1156*10201
[000003] ST*835*102015216
[000004] BPR*I*0.00*C*CHK************2005031
[000005] TRN*1*1*1900000000
[000006] N1*PR*XYZP
[000007] N3*123 HOWARD ST.*5TH FLOOR
[000008] N4*SAN FRANCISCO*CA*94105
[000009] N1*PE*CHN*FI*900000001
[000010] N3*LMNO ADD 1
[000011] N4*SAN FRANCISCO*CA*94105
[000012] LX*1
[000013] CLP*I5925122A1*1**534.02*0*HM*34787
Data not present on an element marked as Required |  | E
[000014] NM1*QC*1**RICHARD
Data not present on an element marked as Required |  | E
[000015] NM1*82*1*UNKNWN PROV LAST NAME*****
[000016] REF*1W*1001234567
[000017] DTM*050*20041006
[000018] SVC*HC>L3000*0*238.36**2
[000019] DTM*150*20040830
[000020] DTM*151*20040830
[000021] CAS*OA*42*-238.36
[000022] REF*6R*4
[000023] SVC*HC>L2820*0*150.92**2
[000024] DTM*150*20040830
[000025] DTM*151*20040830
[000026] CAS*OA*42*-150.92
[000027] REF*6R*5
[000028] SVC*HC>L2275*0*144.74**2
[000029] DTM*150*20040830
[000030] DTM*151*20040830
[000031] CAS*OA*42*-144.74
[000032] REF*6R*6
[000033] SE*31*102015216
[000034] GE*1*102015215
[000035] IEA*1*102015215
```

The SegPool Display is responsible for displaying both the segments within the SegPool object as well as the Message object. Normally, messages are associated with specific segments, so the default mode is to display both simultaneously, as shown here. However, the SegPool and Messages can be shown independently by switching to other tabs.

Menu Options

This section describes the menu bar options at the top of the window.

Session

- Load Session

- Save Session

This feature lets you save the settings within the Data I/O panel to a file that can later be restored. There is also a 'default.winstate' file within the Chiapas/bin directory that represents the default state of these settings; it can be overwritten so that every time Chiapas Studio is launched, these settings are loaded at the beginning of the session.

- Copy CGIF to Clipboard

Copies the currently selected CGIF mapping to the clipboard.

- Open HIG

If there is a currently selected mapping and the options have been set properly to link to a set of HIPAA Implementation Guide PDFs (available from Washington Publishing Company, http://wpc-edi.com), this will open the PDF to the page number describing the mapped element.

- Quit

This quits out of the application.

Data I/O, Flat, HKey

These menu options provide an alternate path to access commonly used Data I/O functions. The Flat menu offers one item not available on the rest of the interface: 'Flat Maps > Clipboard'. If there is a loaded Flat register, this will create a four column description of the mappings contained within that register and paste it to the clipboard. The Seed specifications at the end of this manual were created with the help of this function.

Window

- Studio Interface

- Workflow Interface

- Scheduler Interface

This menu option is available throughout each of the three Chiapas EDI 2012 user interfaces, and allows you to freely switch between them. The current interface is marked with a checkmark. Note that work is not preserved when switching interfaces.

- WinSCP Sessions

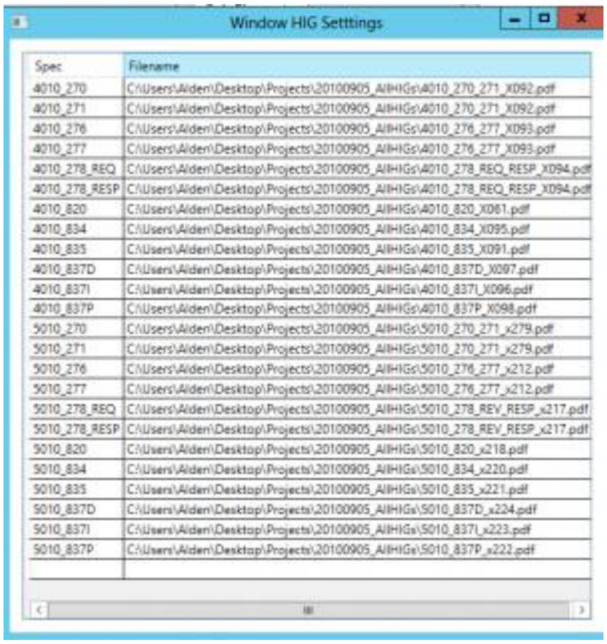
This launches the WinSCP Secure FTP client that comes bundled with Chiapas EDI 2012. The Chiapas Gate Workflow activities as well as the Scheduler system both communicate with this bundled client for remote Secure FTP operations. Note that when configuring WinSCP Sessions, the password needs to be saved. Sessions should be tested and the network key cached before the session name is referenced by Chiapas EDI 2012 components.

Help

- Help System

Brings up the integrated help system

- Set HIG Bindings



Spec	Filename
4010_270	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_270_271_X092.pdf
4010_271	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_270_271_X092.pdf
4010_276	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_276_277_X093.pdf
4010_277	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_276_277_X093.pdf
4010_278_REQ	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_278_REQ_RESP_X094.pdf
4010_278_RESP	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_278_REQ_RESP_X094.pdf
4010_820	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_820_X061.pdf
4010_834	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_834_X095.pdf
4010_835	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_835_X091.pdf
4010_837D	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_837D_X097.pdf
4010_837I	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_837I_X096.pdf
4010_837P	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\4010_837P_X098.pdf
5010_270	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_270_271_x279.pdf
5010_271	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_270_271_x279.pdf
5010_276	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_276_277_x212.pdf
5010_277	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_276_277_x212.pdf
5010_278_REQ	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_278_REV_RESP_x217.pdf
5010_278_RESP	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_278_REV_RESP_x217.pdf
5010_820	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_820_x218.pdf
5010_834	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_834_x220.pdf
5010_835	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_835_x221.pdf
5010_837D	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_837D_x224.pdf
5010_837I	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_837I_x223.pdf
5010_837P	C:\Users\Alden\Desktop\Projects\20100905_AllHIGs\5010_837P_x222.pdf

Chiapas Studio can link directly to the HIPAA Implementatio Guides to pull up a page associated with any mapped element. These guides are copyrighted, controlled materials and not included with the Chiapas EDI 2012 distribution; they must be purchased directly from Washington Publishing Corporation (wpc-edi.org). This feature works only with the initially published 4010 and 5010 PDFs that were not corrected with the Addenda. Type in the path to the file with each associated specification. In the picture at left, each PDF was renamed with a prefix indicating HIPAA version and specification; the original filenames for each specification is the four characters immediately to the left of the .PDF extension.

- GUI Options

```
1. Refresh Segment Pane on Encode
```

This checkbox controls whether the Segment Pane is displayed after an Encode operation. If you are working with extremely large files, this can prevent long freezes within the user interface.

```
2. Refresh HKey Pane on Decode
```

Whenever a HIPAA file is decoded, the XML within the HKey pane is automatically refreshed. If you are working with HIPAA files more than 5 MB in size, this can lead to user interface freezes. Deactivating the interface has no bearing on the underlying registers working properly.

```
3. Enable Flat Data Pane
```

Again, when working with large data sets, disabling the display refresh can increase user interface performance.

- Clear Notification Area

This will reset the notifications window in the upper right hand corner.

- Clear Messages

This will clear the accumulated error and message logs.

- Reset All

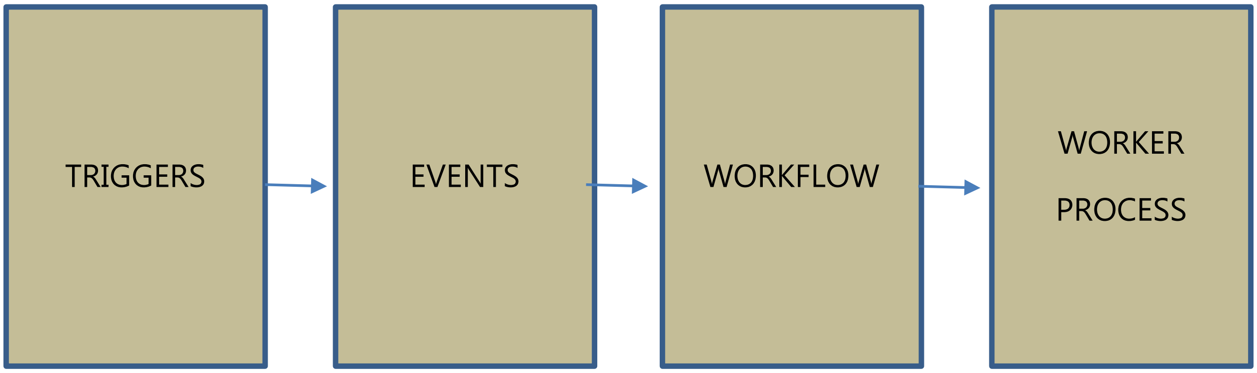
This will clear all data registers and windows, restoring the interface to a newly-opened state.

Workflow Interface

What is a workflow?

Chiapas Studio is useful to handle a lot of ad-hoc tasks – but it isn't an automation platform. For that, Chiapas EDI 2012 relies on a Microsoft technology called Windows Workflow Foundation 4.0. This technology consists of two parts – a design studio to design new workflows and an engine platform to run them. The workflows themselves are composed of 'Activities', which are similar in visual appearance to a flowchart box. Chiapas EDI 2012 includes 42 activities that cover all functionality present within Chiapas Studio as well as additional functions relating to shell commands, databases, SecureFTP and File I/O transfers. By stacking these in order, the developer can create workflows to implement a number of different business processes relating to HIPAA files.

Automation Architecture

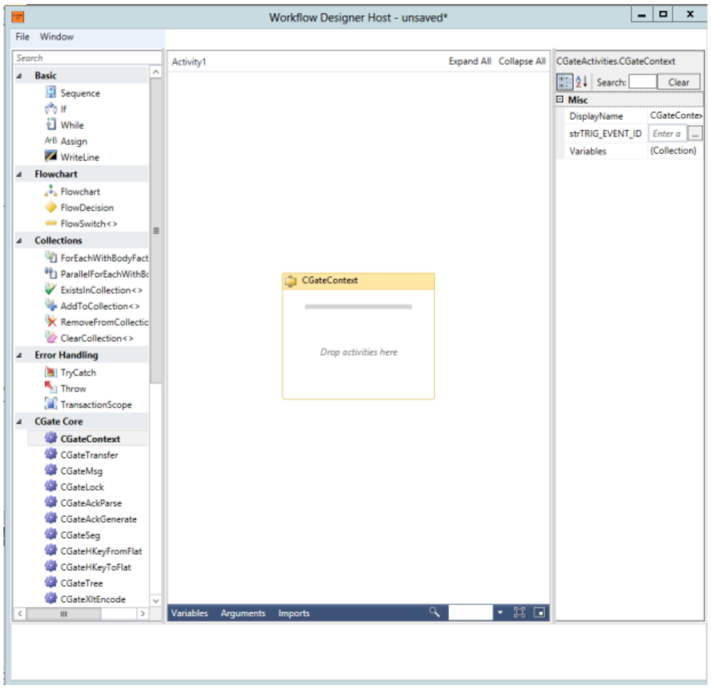


The Chiapas EDI 2012 installation sets up a system service that waits in the background and periodically polls the chiapasData database for active trigger conditions. When a trigger condition is fulfilled, a new Event is launched. In the background, a Worker process will take ownership of the new event, load in the associated workflow, execute it with the passed arguments passed to the workflow from the automation system, and then exit. The background service will refresh the worker process automatically. This ensures that no single event can 'hang' the whole automation system or consume all memory resources.

The criteria for launching new events include scheduled events, events triggered by the results of a SQL query, events triggered by files appearing in an upload directory, events triggered by new files appearing in an archive directory, files being uploaded to a remote SecureFTP server upload directory or new files being uploaded to a remote SecureFTP archive directory.

A single Chiapas EDI 2012 workflow is oriented to handling or creating a single, discrete file. Due to the memory intensive nature of the Encode / Decode engines, a single workflow should not be developed to process many files within a loop – this leads to memory fragmentation and reliability issues and is not a supported development model.

Using the Workflow Studio



DisplayName CGateConte
strTRIG_EVENT_ID Enter a
Variables (Collection)

The workflow studio has functionality for creating and debugging workflows, loading and saving workflows, and testing and debugging them. The left pane is the toolbox, and contains all of the activities supported within Chiapas EDI 2012 workflows.

The middle pane is the workspace where workflows are constructed. The CGateContext is a *sequence*, which in WWF parlance means that it is itself a container for other activities. All Chiapas EDI 2012 activities must be reside within a single CGateContext container, and these containers cannot be nested. It is possible, however, to ignore the default setup and place a CGateContext within a Sequence along with activities under Control Flow, Flowchart, Runtime, Primitives and Error Handling toolbox headers, as these are part of the Windows Workflow environment and do not need to be contained within a CGateContext container.

The right pane displays variables that are associated with any selected activity. For Chiapas activities that do not present individual designers interfaces, you must enter in values in this pane to interact with the activity.

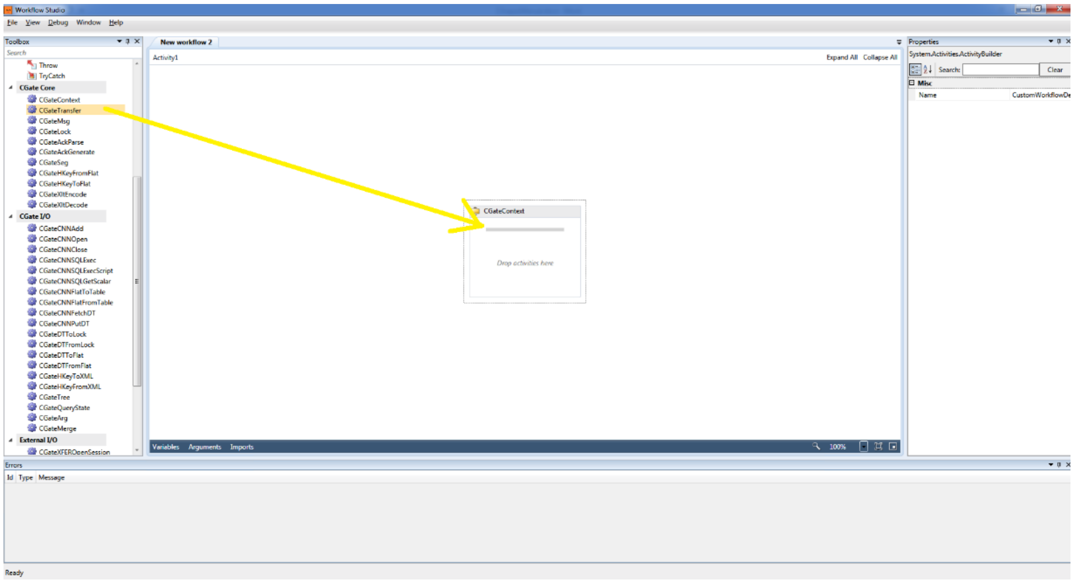
Workflow Tutorial

To really explore how to use the Workflow Studio, let's construct a simple workflow step-by-step from scratch. In this example, we will load in one of the sample file HIPAA files, decode it, and export the results to a CSV file in the C:\chiapas directory.

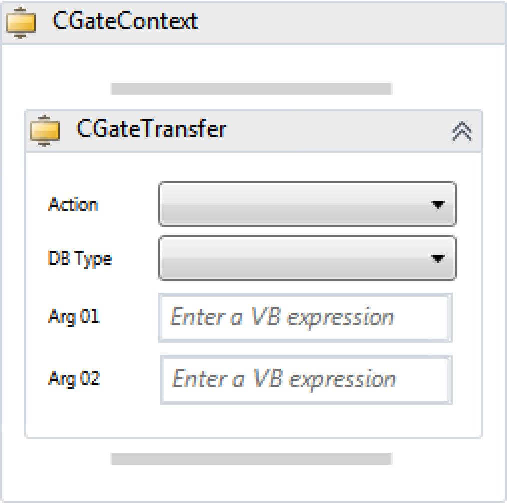
Open the Workflow Studio. When you first launch Chiapas Studio, select 'Window' from the top menu options and select, 'Workflow Interface'.

Create a new workspace. Select 'File' from the menu options, and then select 'New Workflow...'

Add a CGateTransfer activity.



Click on the CGateTransfer activity listed under the CGateCore category in the toolbox on the left, then hold and drag the cursor to the grey bar at the top of the CGateContext activity in the center of the screen. Release the mouse button. This will result in adding the CGateTransfer activity to the workflow:



Set the options on the CGateTransfer activity. Some Chiapas EDI 2012 workflows have *designers*, which are dynamic interfaces to the options within the activity. In this case, when the SegPool < File action is selected, unused options will disappear.

Action: `SegPool < File`

File Path: `”C:\chiapas\sample\seed_834.txt”`

Add the CGateXltDecode activity. This is also found within the CGateCore category of the toolbox; drag to to the gray bar beneath the CGateTransfer activity.

Add the CGateHKeyToFlat activity. Again, drag this to below the CGateXltDecode activity.

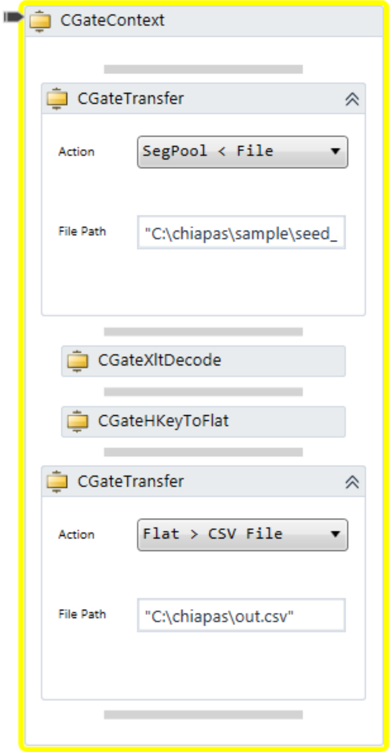
Add another CGateTransfer activity.

Set the options on the second CGateTransfer activity.

Action: `Flat > CSV File`

File Path: `”C:\chiapas\834_out.csv”`

Run the workflow. Under the Debug menu at the top of the screen, select ‘Start Debugging’.



Action `SegPool < File`
File Path "C:\chiapas\sample\seed_

Action `Flat > CSV File`
File Path "C:\chiapas\out.csv"

At this stage, you should now have a new file called '834_out.csv' in the C:\chiapas directory, that contains the 834 HIPAA file decoded to a flat-file format. The full workflow is shown here.

Variables

To create a variable, select the CGateContext activity and open the Variables window by pressing 'Variables' at the bottom of the screen. You may select a default value for the variable, and the variable is visible only within the scope you've selected.\

Imports

Like any .NET language, you will need to add library references for accessing certain functions. For example, to create a FileInfo object, first you will need to add a reference to System.IO in this window. Then, you will be able to create new object instances for objects defined within that library and use all public functions as well.

Scheduler Interface

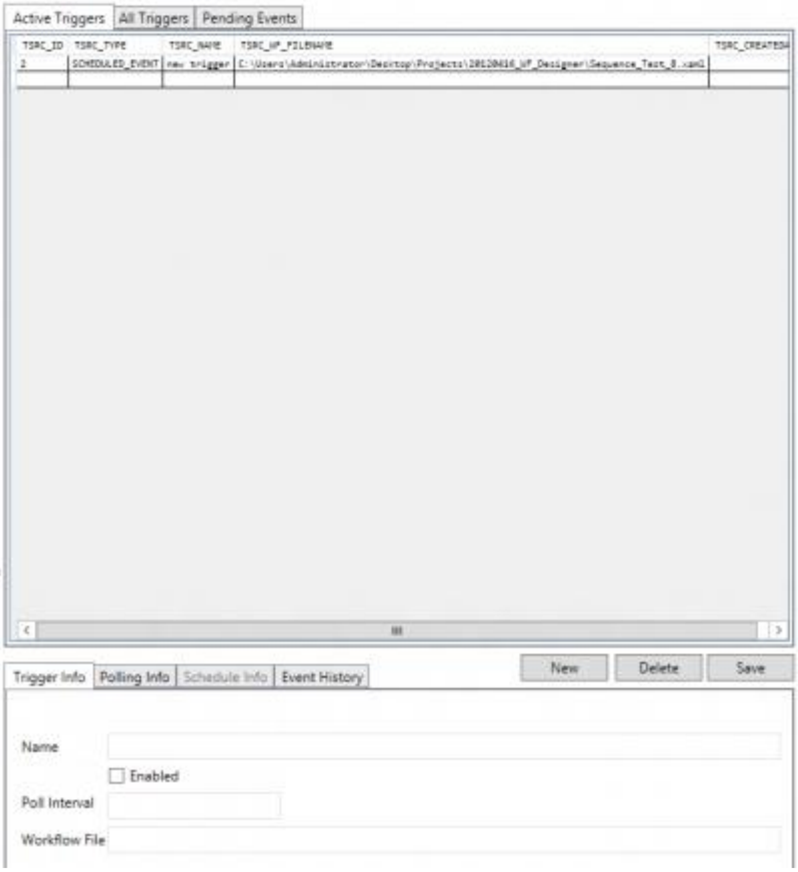
The scheduler interface allows the user to see the status of currently defined triggers and events, and configure new triggers. Also, it will show the event history associated with a trigger. The Scheduler Interface works with these SQL database tables within the chiapasData database:

TSRC – This table stores trigger information and which triggers are active.

EVENTS – This table stores every event triggered by the automation system.

MSGLOG – All messages generated by the events are stored here.

Trigger Info Tab



TSRC_ID	TSRC_TYPE	TSRC_NAME	TSRC_WF_FILENAME	TSRC_CREATEDA
2	SCHEDULED_EVENT	new trigger	C:\Users\Administrator\Desktop\Projects\20120416_WF_Designer\Sequence_Test_8.xaml	

Name

□ Enabled

Poll Interval

Workflow File

Parameters

Name - Name of the Trigger

Enabled - When this is checked, the trigger is actively polled by the background service

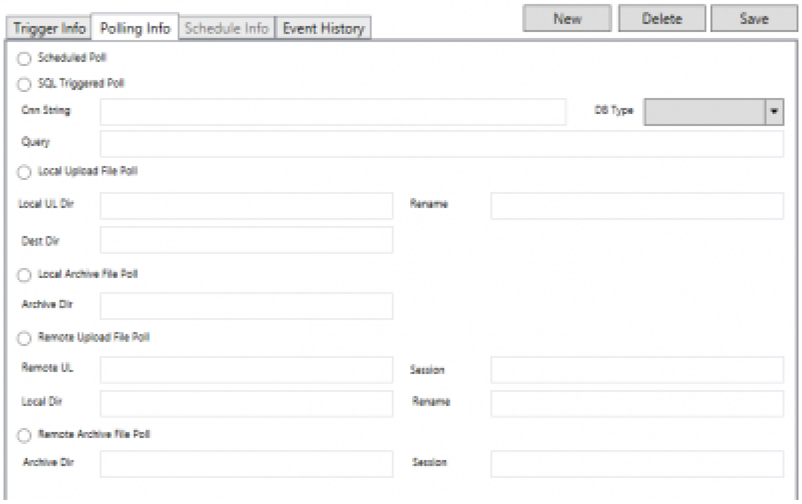
Poll Interval - This is the number of seconds between polling. Note for resource intensive operations like large trigger queries or Remote file checks, this should not be set to a very low value. A value of 600 (ten minutes) is usually sufficient for most business processes. This value can be set to 30 for processes being developer to allow the user to quickly see the results.

Workflow File - This is the name of the .XAML Workflow Activity created within the Workflow Interface. File triggered events will be able to supply information to the Workflow Activity through the use of the CGateArg activity.

Creating a new Trigger

First press the 'New' button, then type in the Trigger Name, Poll Interval and the .XAML Workflow file to be run when the event is fired. Do not click Enabled at this time. Then, press Save, and this will commit the new trigger to the backend system. Afterwards, select the trigger, click on the Enabled checkbox, and re-save the trigger. This will enable the trigger and add it for polling by the background CGateServer service.

Polling Info Tab



Chiapas EDI 2012 supports the following trigger types:

LOCAL_UPLOAD_DIRECTORY_EVENT – The background service will poll a specified directory for files. If any are found, the file is moved to an archive location and the trigger is fired.

LOCAL_ARCHIVE_DIRECTORY_EVENT – The background service will poll a specified directory for new files. If a new file is found, it is recorded in an internal table and a trigger is fired.

REMOTE_UPLOAD_DIRECTORY_EVENT – Here, the background service will use a pre-defined WinSCP session (complete with fixed username and password) and scan a specified directory for new files similar as above.

REMOTE_ARCHIVE_DIRECTORY_EVENT – The background service uses a pre-defined WinSCP session and scans a directory for new files, similar to the local version.

SCHEDULED_EVENT – The background service will wait until a specified time or date using a number of different criteria.

SQL_TRIGGERED_EVENT – If a given connection string and query return a '1' result, this fires the trigger.

Active Triggers Tab

This tab shows the triggers that are currently active and being polled by the background CGateService.

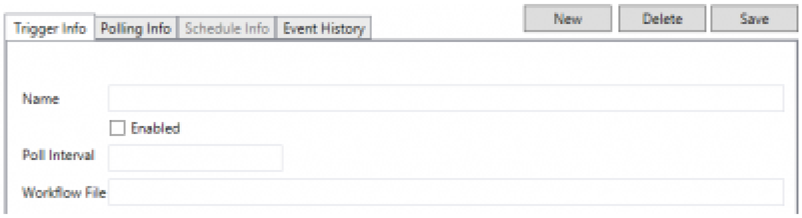
All Triggers Tab

This tab shows all triggers, active or not.

Pending Events

This shows the events that are currently being processed by the background service.

Schedule Info Tab



The Schedule Info tab allows a great deal in flexibility in when and how often a Scheduled trigger is fired. Note that whenever the schedule is updated, Chiapas EDI will reset the 'last polled date' to the current date/time.

Here is a breakdown of the various options:

Run Once Only – The trigger will execute immediately when it is enabled but not thereafter. This is useful for testing the trigger.

Second Interval – This specifies that the Interval window specifies the number of seconds between trigger fires. Note that the Polling Interval specified on the Trigger Info tab takes precedence over this value. A value of '0' in both the Polling Interval and here could possibly freeze the machine and make it unresponsive.

Minute Interval – Here, the Interval specifies how many minutes between trigger fires.

Hourly Interval – This specifies the number of hours between trigger fires.

Start Time / End Time – These should be values like "8:00 PM or 23:00:30". When populated, the triggers will be limited to firing during this time span. Be cautious when scheduling triggers between 1 AM and 2 AM, as shifts in Daylight Savings Time could lead to missed triggers or unintentionally running a trigger twice in one night.

Date Span – Clicking on the checkbox will enable to Calendar controls that allow you to specify a date range when the trigger can fire.

Day of Week – This checkbox will enable the Weekday checkboxes that allow you to limit the trigger to certain days of the week.

Months – This checkbox allows you to filter the trigger to operate only on specified calendar months.

Events History Tab

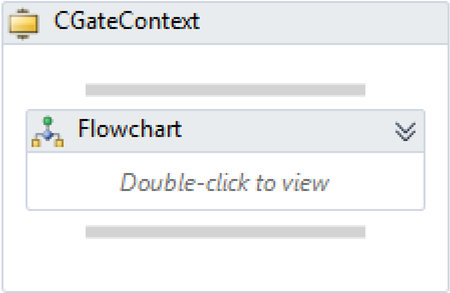
This shows the trigger history. Clicking on one will populate the log screen on the right, which shows all of the messages generated from the Workflow during the execution of the event.

Workflow Activity Reference

This is a reference of the Activities available within Chiapas EDI 2012. Some of them have a customized user interface where values can be plugged directly on the surface; others do not and will need their values entered into the Properties window on the right of the main window. One important note is that within Windows Workflow 4, expression text boxes need to be valid VB.NET expressions – which means all text strings need to be enclosed in quotes. There are several excellent resources and books on Windows Workflow 4 – these can help with getting the most out of this platform. This section is meant to supplement that knowledge to learn how to use Windows Workflow 4 to use the Chiapas Gate technology.

CGate Core Activities

CGateContext Activity

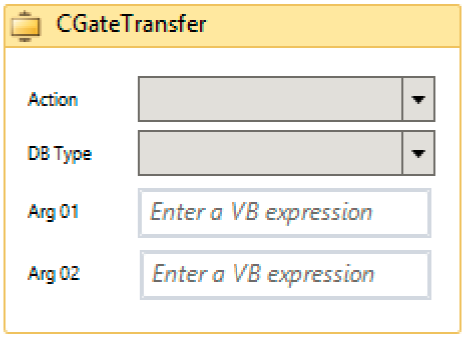


This activity is the main container for all other Chiapas EDI activities. It is defined as a special 'Sequence' activity – to add more Activities inside, they need to be dragged over the grey rectangles inside. CGateContext activities cannot be nested – you cannot place one CGateContext inside of another. Furthermore, all of the other defined Chiapas EDI activities must be placed within a CGateContext container – placing one outside of a CGateContext will result in an error.

The CGateContext storages the 'State' of the Chiapas engine, and holds a single instance of all of the registers. A single instance of the CGateContext activity can be used for a single Encode or Decode operation, or a Decode operation and the creation of a 999 Acknowledgment file. In order to carry out multiple operations within a single workflow, multiple CGateContext activities must be used.

Upon first getting called, it will automatically create and open a named database connection called 'TriggerDB' to the chiapasData database that is part of the automation system (the connection string is in a fixed location at C:\chiapas\bin\cnnstr.txt). If the automation system spawned this object, it will pass the EVENT_ID to the activity so that on completion, it can pass all logged results back to the automation system. It will also use this passed EVENT_ID to pull the four possible string arguments from the automation system.

CGateTransfer Activity



This activity functions similar to the Data I/O Interface within Chiapas Studio, and can import / export data from many different sources. The labels will change according to the Action that is selected.

<u>ErrLog > DB Table</u>

This will export the current error log to a database table using the specified DB Type, Connection String and Data Table arguments.

<u>ErrLog > CSV File</u>

Exports the error log to a CSV file specified in the File Path textbox.

<u>Flat > DB Table</u>

Exports a loaded Flat register to a database table. If the database type selected is SQLite or SQL Server, the destination table will be created if it doesn't exist. Then, all rows of the Flat object will be inserted. If the table previously exists and the schema of the Flat register doesn't match the destination database table, no rows will be inserted and an error will be logged. For this situation, the Merge activity is more appropriate.

Flat < DB Table

Loads the Flat register from a datababse table. Note that it is possible to use a SQL SELECT statement in lieu of a table name here; this is the only way to ensure the Encoder receives a sorted dataset if going from a raw database table.

Flat > CSV File
Flat < CSV File

Exports or imports the Flat register to a CSV file.

Flat > Excel File
Flat < Excel File

Exports or imports the Flat register to an Excel file. The filename should end with either '.xls' to use the Excel 2003 format, or '.xlsx' to use the Excel 2007 format.

HKey > XML File
HKey < XML File

Imports and exports the HKey register to an XML file.

HKey > DB Tables
HKey < DB Tables

Imports and exports the HKey register to a collection of database tables using an indicated prefix. Every loop that encodes business information is used for export. For importing, Chiapas Gate checks all the loops possible for the specification indicated within the mandatory Specification Tag within the first mapping in the ISA loop.

SegPool > File
SegPool < File

Imports and exports the SegPool register to an ASC X12 file. Note that the options set using the CGateSeg activity will need to be set *prior* to encoding.

SegPoolAck > File
SegPoolAck < File

Imports and exports the SegPoolAck register to a 999 Transaction Acknowledgment file. The register is loaded first via a CGateAckGenerate activity. The content of this file will depend on the results of a previous Decode operation. Chiapas EDI 2012 will return success unless there was a critical, parser-stopping HIPAA integrity error.

CGateMsg Activity

This activity is used to either clear or log messages and errors to the Log register. Upon completion of the workflow, these errors will automatically be logged to the automation system if the workflow was executed via a trigger.

Add Simple Message

This action will add a simple message to the Log.

Add Full Message

This action will add a complex message to the Log, including string data and two numeric data values.

Add Simple Error

Adds a single error message.

Add Full Error

Similar to the Add Full Message, but logging the message as an error.

Clear

Clears the message log.

CGateAckParse Activity

This Activity translates the SegPoolAck register into a series of human-readable error messages and writes them to the Log register. This can be used in conjunction with the acknowledged file to determine the cause of any integrity issues that may be occurring.

CGateAckGenerate Activity

This activity will load the SegPoolAck register with a basic 999 Acknowledgment transaction based on the results of a previously executed Decode operation. If there was no critical error, the 999 will show the transaction was accepted. If there was a critical integrity error, it will show that the transaction was rejected, and which segment generated the integrity error.

The SegPoolAck register can then be saved to a 999 Transaction Acknowledgment file via the CGateTransfer Activity.

CGateSeg Activity

This activity will set a number of options on the SegPool that will affect any file export operations. These options will set the Segment Separator, Element Separator, Sub-Element Separator, and on the right, Carriage Return at the end of a segment, Line Feed at the end of a segment, 80 columns split, and whether to scrub Protected Healthcare Information.

This activity needs to be present *prior* to encoding of the file for these formatting options to be applied on the final outgoing file.

CGateHKeyFromFlat Activity

This activity will translate the Flat register to the HKey register, transforming the two-dimensional column/row layout of the Flat register into the hierarchical format of the HKey register.

CGateHKeyToFlat

The counterpart to the CGateHKeyFromFlat activity, this will transform the HKey to the two dimensional Flat register. Internally, the HKey register is pushed into a series of SQLite memory tables and then transformed via a SQL view to a flat representation of the data; see the Export View function under the HKey Data I/O Interface tab for more information.

CGateXltEncode

This activity will transform a loaded HKey register into the SegPool register. If certain required elements are not supplied, the encoder will attempt to default them to certain values.

The Chiapas Gate Encoder can generate values for the following segments and elements:

Defaulted Element	Defaulted Value
ISA01	00
ISA02	Whitespace
ISA03	00
ISA04	Whitespace
ISA05	Sender Qual - Value set in the Encounter Data Activity
ISA06	Sender ID - Value set in the Encounter Data Activity
ISA07	Receiver Qual - Value set in the Encounter Data Activity
ISA08	Receiver ID - Value set in the Encounter Data Activity
ISA09	Current Datestamp
ISA10	Current Timestamp
ISA11	Value set in the Encounter Data Activity
ISA12	Value set in the Encounter Data Activity
ISA13	ICN - Value set in the Encounter Data Activity, right padded with zeros
ISA14	0
ISA15	P
ISA16	Value set in the SegPool
GS01	Value set according to specification
GS02	Sender Code - Value set in Encounter Data Activity

GS03	Receiver Code - Value set in Encounter Data Activity
GS04	Current Datestamp
GS05	Current Timestamp
GS06	GCN - Value set in the Encounter Data Activity
GS07	X
GS08	Value set according to specification
ST02	TCN - Value set in the Encounter Data Activity
ST03	Value set according to specification
BHT04	Current Datestamp
BHT05	Current Timestamp
SE01	Calculated Segment Count
SE02	TCN - Value set in the Encounter Data Activity
GE01	Calculated Transaction Count
GE02	GCN - Value set in the Encounter Data Activity
IEA01	Calculated Group Count
IEA02	ICN - Value set in the Encounter Data Activity
HL01	Indexed according to situation
HL02	Indexed according to situation
HL03	20,21,22, or 23
HL04	Chiapas Gate will fill in for a 1 for when HL03=20 and a 0 for when HL03=23. User must supply 0 or 1 according to the presence of children HL loops.
LX01	Incremented according to situation

This activity depends on a loaded SegPool register, and will try to decode it to the HKey register. If the decoder encounters any SNIP Type 1 and SNIP Type 2 integrity errors, they will be logged. A critical parsing error will stop decoding entirely and result in a critical error.

Data I/O Activities

CGateCNNAdd Activity

This activity registers a named data source with the Chiapas Gate engine state object associated with the CGateContext activity. Other CGate I/O activities can refer to this connection by name alone.

Arguments:

CNNDataSource

Connection string for the data source. Like all Workflow strings, they need to be encapsulated with quotes.

CNNDBType

This string indicates the type of database connection. It should be ‘None’, ‘Sql Server’, ‘ODBC’, ‘OleDB’, or ‘SQLite’. Note that this string is case-sensitive.

CNNName

The name to associate with this connection

CGateCNNOpen

This will open the specified connection.

Arguments:

CNNName

Name of the connection established with CGateCNNAdd

CGateCNNClose

This will close the specified connection. Every database connection that is opened via Windows Workflow activities should also be closed.

Arguments:

CNNName

Name of the connection

CGateSQLExec

This activity will execute a specified string on the indicated named connection.

Arguments:

CNNName - Name of the connection

CNNSQL - SQL string to execute

CGateSQLExecScript

This activity will load the specified file as a text SQL script to be immediately executed on the named connection.

Arguments:

CNNName - Name of the connection

CNNFileName - Full path to a text file containing SQL commands

CGateCNNSQLGetScalar

This activity will execute a SQL query against a connection and then store the result into the specified variable. The variable should be of type ‘object’. If the query returns a null result, the object variable will be set to:

System.DBNull.Value

Arguments:

CNNName - Name of the connection

CNNSQL - text of the SQL query to execute

Result - An L-Value object-type variable that will store the result of the query. The name of the variable should not be quoted.

CGateCNNFlatToTable

This activity will export the contents of the Flat register to a table on the specified connection. All data types will be set to VARCHAR(200) fields.

Arguments:

CNNName – Name of the connection

CNNTable – Name of the table the Flat register will be stored to

CNNReplace – if this string value is set to ‘true’, the table will be dropped if it exists and recreated.

CGateCNNFlatFromTable

This activity imports the Flat register from a specified SQL table on the indicated connection. The SQL table should consist of VARCHAR(200) fields without a primary key, and all fields should be named according to Chiapas Gate Intermediate Format naming convention.

Arguments:

CNNName - Name of the connection

CNNTable - Name of the table the Flat register is to be loaded from.

CGateCNNFetchDT

This activity will load a supplied datatable variable with the contents of a view or table on a specified named database connection.

Arguments:

CNNName - Name of the database connection

CNNTable - Name of the table or view

Result - A supplied DataTable variable that will be written to

CGateCNNPutDT

This activity will push a supplied DataTable variable to a database table. You may indicate whether to drop the receiving table first or not.

Arguments:

CNNName – name of the connection

DT – a DataTable variable

CNNTable – name of the table to push the DataTable to

Replace – if this string value is to ‘true’, the receiving database table will be recreated

CGateDTToFlat

This activity copies the contents of a DataTable to the Flat register.

Arguments:

DT – a supplied DataTable variable

CGateDTFromFlat

This activity copies the contents of the Flat register to a supplied DataTable variable

Arguments:

DT – a supplied DataTable variable

CGateHKeyToXML

This activity will copy the HKey register to a supplied string variable as XML.

Arguments:

HKeyXML – the string variable that will receive the HKey's XML text

CGateHKeyFromXML

This activity converts the supplied string variable to an XML document and loads the HKey register with it.

Arguments:

HKeyXML – the string variable containing CGIF XML that will get loaded to the HKey register

CGateQueryState

This activity allows you to directly query the state of Chiapas Gate. The value being queried is supplied in the incoming string, and the result is assigned to the supplied string variable.

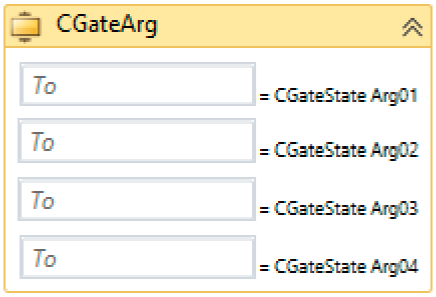
Arguments:

Query – This is one of several values indicating the information being requested.

```
“EVENT_ID”    – The Scheduler event ID that spawned this workflow
"SEGTERM"     – The segment terminator of the current SegPool object
"ELETERM"     – The element terminator of the current SegPool object
"SUBELETERM"  – The element terminator of the current SegPool object
"SPEC_SHRT_NM"– Short name of the currently loaded specification

"SPEC_LONG_NM"– Long name of the currently loaded specification
"CRIT_ERR"    – this returns the value ‘true’ if there was a critical integrity error encountered during decoding
"ERR_CT"      – returns the count of errors and messages in the Log register
"ERR_LOG"     - returns all messages in the log as a string
"FLAT_COL_CT" – returns the number of mappings within the Flat register
"FLAT_ROW_CT" – returns the number of rows within the Flat register
"LOCK_COL_CT" – returns the number of mappings within the Lock register
"SEG_CT"      – returns the number of segments within the SegPool object
"FLAT_SPEC_NM"
"HKEY_SPEC_NM"
"SEG_SPEC_NM" - Either one of these strings will cause the Chiapas Gate engine to interrogate the Flat, HKey or SegPool

                registers for information about the associated specification. The values returned are:

                "4010_270.BIN"               Health Care Eligibility Benefit Inquiry (270)
                "4010_270_A1.BIN"            Addenda 1
                "4010_271.BIN"               Health Care Eligibility Benefit Inquiry Response (271)
                "4010_271_A1.BIN"            Addenda 1
                "4010_276.BIN"               Health Care Claim Status Request (276)
                "4010_276_A1.BIN"            Addenda 1
                "4010_277.BIN"               Health Care Claim Status Response (277)
                "4010_277_A1.BIN"            Addenda 1
                "4010_278_REQ.BIN"           Health Care Services Review-Request for Review (278)
                "4010_278_REQ_A1.BIN"        Addenda 1
                "4010_278_RESP.BIN"          Health Care Services Review-Response (278)
                "4010_278_RESP_A1.BIN"       Addenda 1
                "4010_820.BIN"               Payroll Deducted and Other Group Premium Payment for Insurance Product
                                             (820)
                "4010_820_A1.BIN"            Addenda 1
                "4010_834.BIN"               Benefit Enrollment and Maintenance (834)
                "4010_834_A1.BIN"            Addenda 1
                "4010_835.BIN"               Health Care Claim Payment/ Advice (835)
                "4010_835_A1.BIN"            Addenda 1
                "4010_837D.BIN "             Health Care Claim: Dental (837)
                "4010_837D_A1.BIN "          Addenda 1
                "4010_837I.BIN"              Health Care Claim: Institutional (837)
                "4010_837I_A1.BIN "          Addenda 1
                "4010_837P.BIN "             Health Care Claim: Professional (837)
                "4010_837P_A1.BIN”           Addenda 1
                "5010_270.BIN"               Health Care Eligibility Benefit Inquiry (270)
                "5010_270_A1.BIN"            Addenda 1

                "5010_271.BIN"               Health Care Eligibility Benefit Inquiry Response (271)
                "5010_271_A1.BIN"            Addenda 1
                "5010_276.BIN"               Health Care Claim Status Request (276)
                "5010_277.BIN"               Health Care Claim Status Response (277)
                "5010_277CA.BIN"             Health Care Claim Status Response (277CA)

                 "5010_278_REQ.BIN"          Health Care Services Review-Request for Review (278)
                "5010_278_RESP.BIN"          Health Care Services Review-Response (278)
                "5010_820.BIN"               Payroll Deducted and Other Group Premium Payment for Insurance Products
                                             (820)
                "5010_820_A1.BIN"            Addenda 1
                "5010_834.BIN"               Benefit Enrollment and Maintenance (834)
                "5010_834_A1.BIN"            Addenda 1
                "5010_835.BIN"              Health Care Claim Payment/ Advice (835)
                "5010_837D.BIN"             Health Care Claim: Dental (837)
                "5010_837D_A1.BIN"          Addenda 1
                "5010_837D_A2.BIN"          Addenda 2
                "5010_837I.BIN"             Health Care Claim: Institutional (837)
                "5010_837I_A1.BIN"          Addenda 1
                "5010_837I_A2.BIN"          Addenda 2
                "5010_837P.BIN"             Health Care Claim: Professional (837)
                "5010_837P_A1.BIN"          Addenda 1
                "5010_999_A1.BIN"           Transaction Acknowledgment (999)
```

Result – The result of the query is stored in the supplied string variable

CGateArg



This activity is used when the Workflow is ready to make the transition a scheduled or triggered process, and propagates information sent to it by the Scheduler system. Up to four string variables are supplied, and these will be populated with the information specific for the type of trigger that fired this Workflow.

The information supplied depends on the trigger type, as follows:

1. SCHEDULED_EVENT

```
Arg 01: Current date and time
```

2. SQL_TRIGGERED_EVENT

```
No information is propagated by the trigger
```

3. LOCAL_UPLOAD_DIRECTORY_EVENT

```
Arg 01: The full path and filename to the file. Note that the trigger
only fires after the file has been moved to its’ new location.
```

4. LOCAL_ARCHIVE_DIRECTORY_EVENT

```
Arg 01: The full path and filename to the file.
```

5. REMOTE_UPLOAD_DIRECTORY_EVENT

```
Arg 01: The full path to the local filename, after it has been pulled
to the local filesystem, renamed, and then moved to the new remote
direction.
```

6. REMOTE_ARCHIVE_DIRECTORY_EVENT

```
Arg 01: The full path to the local filename, after it has been pulled
to the local filesystem.
```

CGateMerge

This activity allows a loaded to Flat register to be merged into an existing SQL Server database table. Any fields that exist in the Flat register that are not present within the table will be flagged with an error message; otherwise, all fields that are in common to both will be inserted into the table. This activity replaces all 'Lock' associated commands and activities.

It requires an open database connection that is first set up with the CGateCNNAdd and CGateCNNOpen commands. The loaded Flat register will then be converted into a series of SQL INSERT statements into the destination table.

If the source Flat register contains mappings not found within the destination table, a log message will be generated 'Unmerged Mapping: ' followed by the CGIF field name that is being dropped.

Note that if mappings are present within the destination table but are not present in the source, this will not generate any error messages. Furthermore, the destination table may contain non-CGIF column names like 'FileName' or 'CreateDate' so that after the merge operation is complete, the NULL default values of these columns can be updated to reflect the source file name and creation date of the record. If you were to merge the CGIF schemas for an 837I and 837P source file, it would be possible to have a single queryable database table containing both Professional and Institutional files.

Arguments:

DestCnn - Destination Connection. This is the name of an existing, open database connection.

DestTable - Destination Table. This must be the name of a table that already exists within the target database.

External I/O Activities

CGateXFEROpenSession

This activity will open a WinSCP session that has been previously configured with the bundled WinSCP application.

Arguments:

SessionName – name of the session as registered within the WinSCP Sessions interface

CGateXFERCloseSession

This activity closes the active WinSCP session.

CGateXFERPutFile

This activity requires an open WinSCP session, and will upload a local directory to the remote file system with the specified filename.

Arguments:

FileName – Local filename to upload RemoteFileName – The name of the upload when it is uploaded to the remote server

CGateXFERGetFile

This activity requires an open WinSCP session, and will get a single named file to the local filesystem.

Arguments:

FileName – Local path and filename to store the received file RemoteFileName – path and filename of the file to get

CGateXFERGetDirectoryFiles

This activity will store the list of files in a given directory of the current WinSCP session to a supplied string array variable.

Arguments:

RemoteDirectory – The remote directory containing the files

FileList – A supplied variable of type string[] that will store the resulting file list

CGateSendMail

This activity will open a connection to an SMTP (EMail) server and send out a message with the given sender, receiver, subject, body, and server login credentials.

Arguments:

MessageSubject - Subject line of the message

MessageBody - Message body

SenderEMail - EMail address of the sender

ReceiverEMail - EMail address of the receiver

SMTPServer - Server name of the SMTP host

SMTPUsername - Login username on the SMTP host

SMTPPassword - Login password on the SMTP host

SMTP_SSL - 'true' to connect to the SMTP host with SSL, 'false' otherwise

If an error occurs during any stage of the process of sending an outgoing Email, it will be logged.

File I/O Activities

CGateIOCopyFile

This will copy a specified file.

Arguments:

CopyFrom - Full path and name of the source file

CopyTo - Full path and name of the destination file

CGateIODeleteFile

Deletes a specified file.

Arguments:

DeleteFile - Full path and name of the file to be deleted

CGateIODirectory

This will create a list of files found in the specified directory and then place them in the variable given in the ResultFiles argument. The variable must be of type string[].

Arguments:

DirectoryPath - Full path to the directory to be scanned

ResultFiles - The variable the results will be recorded to

CGateRunShell Activity

This command will execute an arbitrary shell command and wait for its completion. If the command returns a non-zero return value, a message is added to the message log indicating the exit code.

Arguments:

ShellCmd - A string that represents the full path and filename of the shell command to execute

ShellArgs - A string storing the command line arguments, if any

Message Reference

Anytime an error occurs during an automation activity, it will be logged to the distribution database. During decoding operations, the segment or element reference number may be included with the error. The schema of the error log is covered in the Technical Reference chapter.

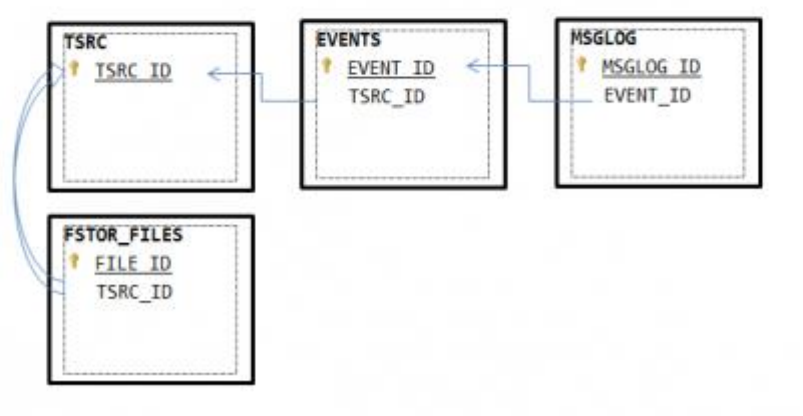
X12Decoder::ConsumeSegment	Data present on an element marked as Unused
	Element *x* (*element*) Mismatch with Data Type (*datatype*)
	Missing qualifier for an identifier
	This element is not a valid value for this element
	Data not present on an element marked as Required
X12Decoder::LoopParse	Premature End of File Detected
	Mandatory Segment *loop* / *Segent* Not Found
	Mandatory Loop *loop* Not Found
	Segment *loop / segment* Exceeded Maximum Iterations (*max iter.*)
	Loop *loop* Exceeded Maximum Iterations (*repeats*)
X12Decoder::DecodeX12File	Insufficient segments found within source X12 file.
	Insufficient elements in the Group Envelope GS segment. This is not a HIPAA 4010/5010 file.
	ChiapasGate could not determine the specification from the information within the file.
	Premature End of File
	Unknown specification encountered.
	Decode Failure
	Exception during parsing
X12Heir::VerifySegment	ERROR_H005: 'E' (either but not both) Requirement Condition violated (*condition*) at *segment name*
	ERROR_H006: 'R' (at least one needs to be present) Requirement Condition violated (*condition*) at *segment name*
	ERROR_H007: 'L' (if first present, 2nd or 3rd is required) Requirement Condition violated (*condition*) at *segment name*
	ERROR_H008: 'P' (if one is present, both are needed) Requirement Condition violated (*condition*) at *segment name*
	ERROR_H009: 'C' (if first is present, second is required) Requirement Condition violated (*condition*) at *segment name*
HKEY_FromFlat	Exception while transferring Flat to HKey

HKEY_ToFlat	Exception while transferring HKey to Flat
HKEY_FromXmlDoc	Exception while transferring XmlDocument to HKey
TREE_Set	The Map Tag was an invalid tree reference.
CGate_API::Tree_SetFromObject	Cannot set object - Flat is invalid
	Cannot set object - HKey is invalid
	Cannot set object - SegPool is invalid
	There was difficulty parsing the correct specification.
XLT_Decode	X12 Segments are not loaded, decode failure
	Tree not loaded, decode failure
	Exception during Decode operation
XLT_Encode	Hierarchical Key is not loaded with data, encode fail.

Technical Reference

This chapter will cover the important things developers need in order to be productive with Chiapas EDI 2012 Automation. This includes the Data Dictionary of all the objects used by the automation system, technical specifics on the scheduling system, and finally information about the pre-built decode / encode automation workflows built into Chiapas EDI 2012.

Core Object Schema



These represent the core objects needed to run the automation system within Chiapas EDI 2012. Chiapas EDI 2012 uses the file C:\chiapas\bin\cnnstr.txt as the SQL Server connection string to connect to the ChiapasData database that should have been installed as part of the Chiapas EDI 2012 installation. These tables store information about triggers, events, logged messages, and archive files.

EVENTS Table

This table stores events. Normally, events are generated by the background ChiapasService when it scans an active trigger that meets its firing conditions. When the CGateService background automation handler finds a triggering condition based off of the TSRC table, it inserts a row into this table with an EVENT_OWNER of -1. This will tell the WorkerProcesses maintained by the automation system that this is a new process, and they will take ownership of the event and execute it.

Column Name	Data Type	Purpose
EVENT_ID	INT, PK	Primary Key (Identity)
EVENT_OWNER	INT	This is refers to the numeric ID of the background worker process that took ownership and processed the event.
TSRC_ID	INT	Foreign key to the Trigger table of the trigger that spawned this event.
EVENT_DATA01	VARCHAR(200)	Event Data 01 (contents depend on trigger type)
EVENT_DATA02	VARCHAR(200)	Event Data 02 (contents depend on trigger type)
EVENT_DATA03	VARCHAR(200)	Event Data 03 (contents depend on trigger type)
EVENT_DATA04	VARCHAR(200)	Event Data 04 (contents depend on trigger type)
EVENT_CREATED	DATETIME	Timestamp of when the event was created
EVENT_PROCESS_BEGIN	DATETIME	Timestamp of when the event began processing
EVENT_PROCESS_COMPLETE	DATETIME	Timestamp of the completion of processing
EVENT_RESULT	VARCHAR(200)	Values: 'CRITICAL ERROR, SEE MESSAGE LOGS', 'CRITICAL ERROR, NO LOGS', 'COMPLETE WITH MESSAGE LOGS', 'COMPLETE, NO LOGS'

FSTOR_FILES Table

This table stores filenames associated with an archive directory. An archive directory is defined as a holding area that can receive new files, either as a local directory or a remote SecureFTP directory.

an UPLOAD trigger. This table stores events. Normally, events are generated by the background ChiapasService when it scans an active trigger that meets its firing conditions. More information about events is in the Scheduling System section.

Column Name	Data Type	Purpose
FILE_ID	INT, PK	Primary Key (Identity)
TSRC_ID	INT	Foreign key to the TSRC row associated with an archive trigger
FILE_NM	VARCHAR(250)	File name of the file. Note that as the trigger is linked to a specific directory (local or remote), this does *not* contain path information.

MSGLOG Tab

This table stores the Message Logs that are associated with a specific event. They can be created either by the internal Chiapas EDI systems, or added by the user via the MsgLog activity.

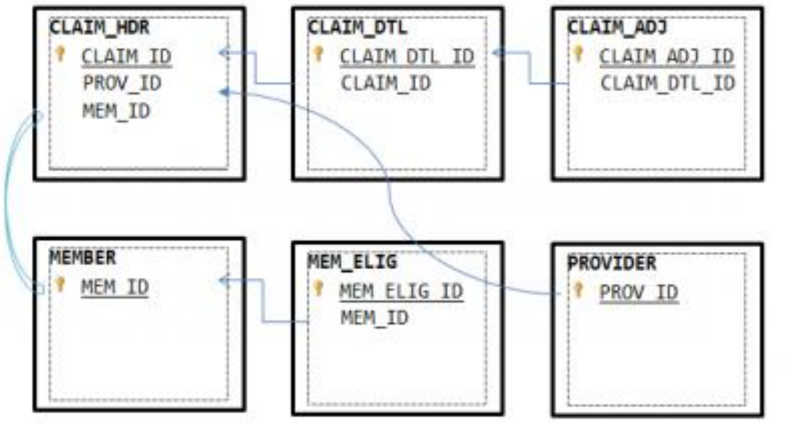
Column Name	Data Type	Purpose
MSGLOG_ID	INT, PK	Primary Key (Identity)
Origin	VARCHAR(200)	Chiapas EDI Module that created the message
Message	VARCHAR(200)	Error Message
MsgData_Str	VARCHAR(200)	Message Data - String
MsgData_Num1	VARCHAR(200)	Message Data - Num 1
MsgData_Num2	VARCHAR(200)	Message Data - Num 2
IsError	VARCHAR(200)	Flags whether this message is an error
TimeStamp	VARCHAR(200)	Timestamp of when the error was generated
EVENT_ID	VARCHAR(200)	ID of the triggering Event
MSGLOG_NR	VARCHAR(200)	Number indicating the sequence of the message within the event

TSRC Table

This table is responsible for storing active and inactive triggers. The different types of triggers are covered in the chapter on the Scheduling System.

Column Name	Data Type	Purpose
TSRC_ID	INT, PK	Primary Key (Identity)
TSRC_ENABLED	INT	0 = Trigger Disabled; 1 = Trigger Enabled
TSRC_TYPE	VARCHAR(50)	Indicates the type of trigger. Valid values are 'SCHEDULED_EVENT', 'SQL_TRIGGERED_EVENT', 'LOCAL_UPLOAD_DIRECTORY_EVENT', 'LOCAL_ARCHIVE_DIRECTORY_EVENT', 'REMOTE_UPLOAD_DIRECTORY_EVENT', 'REMOTE_ARCHIVE_DIRECTORY_EVENT'
TSRC_WF_FILENAME	VARCHAR(250)	Full path to the Workflow .XAML file that is executed as part of the event spawned by this trigger
TSRC_POLL_INTERVAL	INT	This is the number of seconds the background ChiapasService will wait between checks for this trigger. Caution should be exercised when setting this to a value less than 60.
TSRC_NAME	VARCHAR(50)	Name of the trigger
TSRC_DATA01	VARCHAR(4000)	Trigger Data 01
TSRC_DATA02	VARCHAR(4000)	Trigger Data 02
TSRC_DATA03	VARCHAR(4000)	Trigger Data 03
TSRC_DATA04	VARCHAR(4000)	Trigger Data 04
TSRC_DATA05	VARCHAR(4000)	Trigger Data 05
TSRC_CREATEDAT	DATETIME	Timestamp when the trigger was created

Sample Object Schema



In order to demonstrate the process of encoding different specifications, Chiapas EDI 2012 stores mock member, claims and provider data in this schema. As the generation of EDI files is linked to managed care, these tables were designed as a miniature managed care database. In this way, the twelve 5010 extract views that generate sample files are more closely aligned with the extracts developers will need to create in order to support their business processes.

Claim Header Table

This table represents Claim Header data for the mock claims in the distribution database.

Column Name	Data Type	Purpose
CLAIM_ID	INT, PK	Unique Claim Identifier
PROV_ID	INT	Foreign Key to Provider table
MEM_ID	INT	Foreign Key to Member table
PATNT_ACCT_NR	VARCHAR(50)	Patient Account Number
FAC_TYP_CD	VARCHAR(2)	Facility Type Code
PRI_DIAG_CD	VARCHAR(10)	Primary Diagnosis Code
SEC_DIAG_CD	VARCHAR(10)	Secondary Diagnosis Code
STATUS	VARCHAR(10)	Claim Adjudication Status
ADJUDCTN_DT	DATE	Claim Adjudication Date

Claim Detail Table

This table represents the Claim Detail data linked to the mock claims within the mock database. The Claim Details are linked many-to-one to the Claim Header table.

Column Name	Data Type	Purpose
CLAIM_DTL_ID	INT, PK	Unique Claim Detail Line Identifier
CLAIM_ID	INT	Foreign Key reference to the Claim Header table
CHG_AMT	MONEY	Claim Line Charge Amount
COPAY_AMT	MONEY	Claim Line Patient Copay Amount
PAID_AMT	MONEY	Claim Line Paid Amount
FROM_DT	DATE	Claim Line Service From Date
TO_DT	DATE	Claim Line Service To Date
SVC_CD	VARCHAR(10)	Claim Line Procedure Code
UNIT_CT	INT	Claim Line Unit Count

Claim Adjustment Table

This table represents adjustments made on Claim Line details.

Column Name	Data Type	Purpose
CLAIM_ADJ_ID	INT, PK	Unique Claim Detail Adjustment Identifier
CLAIM_DTL_ID	INT	Foreign Key reference to the Claim Detail table
ADJ_REASN	VARCHAR(2)	Adjustment Reason Code
ADJ_AMT	MONEY	Adjustment Amount
ADJ_SEQ	INT	Adjustment Sequence

Member Table

This table represents a mock membership that is used in the various sample transactions.

Column Name	Data Type	Purpose
MEM_ID	INT, PK	Unique Member Adjustment Identifier
FIRST_NM	VARCHAR(50)	First Name
LAST_NM	VARCHAR(50)	Last Name
SSN_ID	VARCHAR(20)	Social Security Number
BIRTH_DT	DATE	Birth Date
GENDER	VARCHAR(1)	Gender
ADD1	VARCHAR(75)	Address Line 1
ADD2	VARCHAR(75)	Address Line 2
CITY	VARCHAR(50)	City
STATE	VARCHAR(2)	State
ZIP_CD	VARCHAR(20)	Zip Code
PHONE_NR	VARCHAR(20)	Phone Number

Member Eligibility Table

This table is used to encode a Member’s eligibility history, and is used by the 834 Eligibility extract as a data source.

Column Name	Data Type	Purpose
MEM_ELIG_ID	INT, PK	Unique Member Eligibility Identifier
MEM_ID	INT	Foreign Key to the Member table
STATUS	VARCHAR(10)	Eligibility status
FROM_DT	DATE	Eligibility From Date
TO_DT	DATE	Eligibility To Date

Provider Table

This table storages a number of mock providers that are used through a number of the sample extracts.

Column Name	Data Type	Purpose
PROV_ID	INT, PK	Unique Provider Identifier
TAX_ID	VARCHAR(15)	Federal Tax ID
NPI_ID	VARCHAR(12)	National Provider Identifier
COMPANY_NM	VARCHAR(50)	Company Name
FIRST_NM	VARCHAR(20)	First Name
LAST_NM	VARCHAR(20)	Last Name
ADD1	VARCHAR(75)	Address Line 1
ADD2	VARCHAR(75)	Address Line 2
CITY	VARCHAR(50)	City
STATE	VARCHAR(2)	State
ZIP_CD	VARCHAR(20)	ZIP Code
CONTACT_NM	VARCHAR(50)	Contact Name
WORK_PH	VARCHAR(50)	Work Phone
EMAIL	VARCHAR(50)	Email

Tutorials

Chiapas EDI 2012 includes a number of pre-built workflows, including a full-fledged automated decoding system. In this section, we will tie the Scheduling system together with the workflow system to see full demonstrations of the encoder and decoder automation systems at work.

First, let's cover the inventory of the sample files and database objects included with the Chiapas EDI 2012 distribution:

File System:

File	Purpose
The sample directory contains 'seed' 5010 files for every specification. They are the result of the output of the included 5010 database views after they are fed into the Chiapas EDI 2012 encoder. Each of these files has undergone extensive testing to meet the full Implementation Guide syntax and business requirements; therefore, they can be used as the 'seed' of new transaction extracts.	
C:\chiapas\sample\seed_270.txt	Health Care Eligibility Benefit Inquiry
C:\chiapas\sample\seed_271.txt	Health Care Eligibility Benefit Response
C:\chiapas\sample\seed_276.txt	Health Care Claim Status Request
C:\chiapas\sample\seed_277.txt	Health Care Claim Status Response
C:\chiapas\sample\seed_278_req.txt	Health Care Services Review - Request for Review
C:\chiapas\sample\seed_278_resp.txt	Health Care Services Review - Response
C:\chiapas\sample\seed_820.txt	Payroll Deducted and Other Group Premium Payment for Insurance Products
C:\chiapas\sample\seed_834.txt	Benefit Enrollment and Maintenance
C:\chiapas\sample\seed_835.txt	Health Care Claim Payment/Advice
C:\chiapas\sample\seed_837d.txt	Health Care Claim: Dental
C:\chiapas\sample\seed_837i.txt	Health Care Claim: Institutional
C:\chiapas\sample\seed_837p.txt	Health Care Claim: Professional
The XAML directory contains all of the sample workflow files included with the distribution. When linked to the scheduling system, these represent out-of-the-box automation solutions for a number of scenarios.	
C:\chiapas\xaml\binDecoder.xaml	Workflow for the binDecoder automation example

`C:\chiapas\xaml\defaultWorkflow.xaml`	This is the default workflow opened whenever the 'New Workflow' menu option is selected within the Workflow Interface.
`C:\chiapas\xaml\simpleEncodePipeline.xaml`	This encode workflow was designed to work with a Scheduled automation trigger.

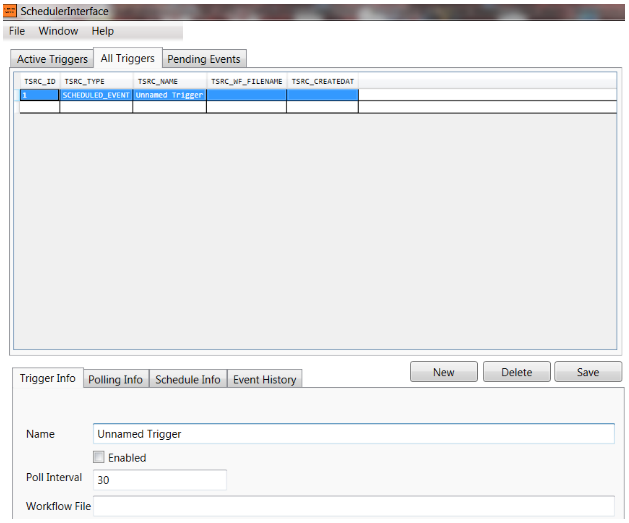
chiapasData Database:

Object	Purpose
`BIN_834_5`	The BIN objects are related to the binDecoder workflow and are covered in the following section.
`BIN_837D_5`	
`BIN_837I_5`	
`BIN_837P_5`	
`BIN_LOG`	
`vw_Export_270`	These views link to the mock data covered in the Mock Object Schema section to produce a fully HIPAA compliant for each of the twelve 5010 Implementation Guides. The views reformat the mock business data to the CGIF naming convention accepted by Chiapas EDI 2012. To see how this is done, right click on the view within SQL Server Management Studio and click "Create To…" to see a new window showing all of the documented SQL source code within these views. The Seed Specification Defintions chapter later in this manual covers the output in a way similar to how it would be documented in a Companion Guide.
`vw_Export_271`	
`vw_Export_276`	
`vw_Export_277`	
`vw_Export_278_REQ`	
`vw_Export_278_RESP`	
`vw_Export_820`	
`vw_Export_834`	
`vw_Export_835`	
`vw_Export_837D`	
`vw_Export_837I`	
`vw_Export_837P`	

Encoder Tutorial

In this section, we'll investigate using the Scheduler linked with the encoder example workflow to encode a database view to a HIPAA file on a periodic basis.

1. Launch C:\chiapas\bin\ChiapasStudio.exe
2. At the top of the screen, pull down the 'Window' menu option and select 'Scheduler Interface'.
3. Switch to the 'All Triggers' pane.
4. Press the 'New' button. Press OK.
5. Select the new trigger that was generated. The screen will look like this:



TSRC_ID	TSRC_TYPE	TSRC_NAME	TSRC_WF_FILENAME	TSRC_CREATEDAT
1	SCHEDULED_EVENT	Unnamed Trigger		

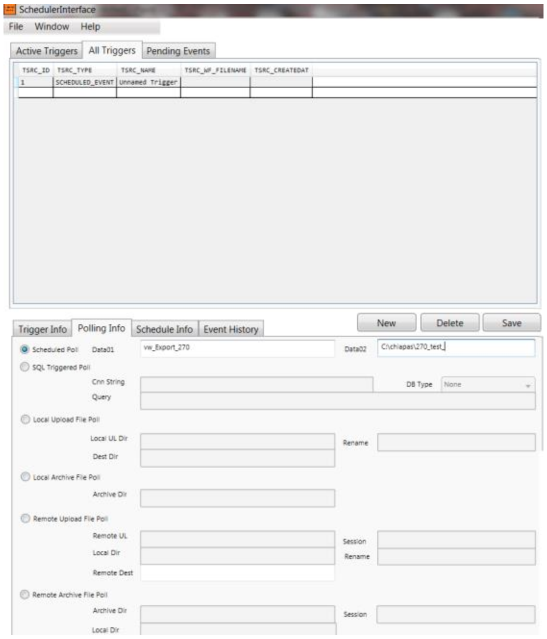
Name: Unnamed Trigger

□ Enabled

Poll Interval: 30

Workflow File:

6. Double click in the Workflow File field, and in the resulting Open File Dialog, select the file C:\chiapas\xaml\simpleEncodePipeline.xaml
7. Press the Polling Info pane.
8. Leaved the Scheduled Poll selected.
   a. In the first textbox, Data01, type in: vw_Export_270
   b. In the second textbox, Data02, type in: C:\chiapas\270_text_



TSRC_ID	TSRC_TYPE	TSRC_NAME	TSRC_WF_FILENAME	TSRC_CREATEDAT
1	SCHEDULED_EVENT	Unnamed Trigger		

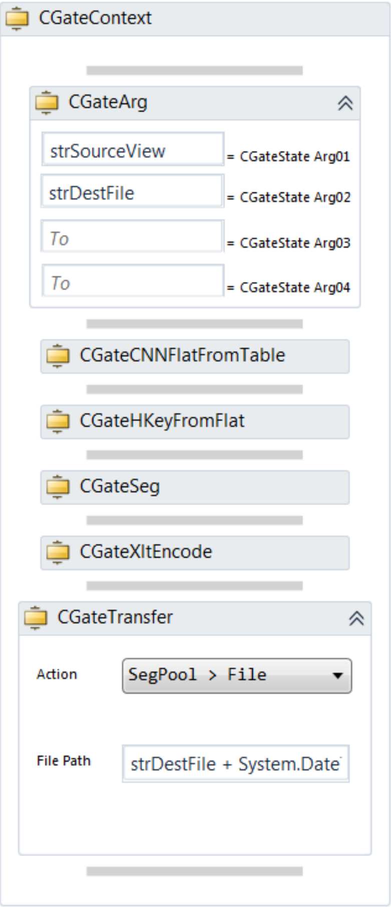
● Scheduled Poll Data01 vw_Export_270 Data02 C:\chiapas\270_test_

9. Press Save.
10. Return to the Trigger Info tab.
11. Set the 'Enabled' checkbox.
12. Press Save.

In the C:\chiapas directory, a new file should appear. Since in this configuration, Chiapas will create new files every 30 seconds, you should disable it by toggling the Enabled checkbox and Saving again.

Now that we see *what* it does, we should investigate *how* it does it. To see the next step, go to the Windows menu option and select Workflow Interface. Once there, select File -> Open and open C:\chiapas\xaml\simpleEncodePipeline.xaml

On the left, you will see a toolbar filled with valid Chiapas EDI workflow activities, and in the center you will see the stacked activities that comprise the workflow itself.



strSourceView = CGateState Arg01
strDestFile = CGateState Arg02
To = CGateState Arg03
To = CGateState Arg04

Action SegPool > File
File Path strDestFile + System.Date

From top to bottom:

CGateArg – The automation system allows the user to pass two user-defined variables to the events spawned by the trigger. In this case, the values we entered in Data01 and Data02 will populate to these two defined string variables.

CGateCNNFlatFromTable – Using the predefined connection ‘TriggerDB’, this opens up the view passed to the ‘strSourceView’ variables and loads into the Flat register.

CGateHKeyFromFlat – This transfers the information from the Flat register to the HKey register.

CGateSeg – This sets commonly used formatting options that will be used on the finalized output file. Note that Chiapas requires these options to be set prior to encoding.

CGateXltEncode – This encodes the HKey register to the SegPool register.

CGateTransfer – This is set to the action ‘SegPool > File’, which saves the SegPool register to an arbitrary file. The filename is described in a Visual Basic expression contained within the File Path box.

In order to go one step deeper, we would have to look at the next step in the chain – the vw_Export_270 view that we specified originally in the trigger setup.

If we open SQL Server Management Studio and right click on the ‘vw_Export_270’ view within the chiapasData distribution database and selected ‘Create To…’, we will see the source code of the view:

```
/*
vw_Export_270

This view is an extract that translates from a sample set of eligibility and provider
data to the format needed by Chiapas EDI 2012 to create a HIPAA-compliant 5010
270 eligibility inquiry file.  This represents a 'seed' extract that can help trading
partners while setting up new HIPAA transactions.
*/

CREATE VIEW [dbo].[vw_Export_270]
…
```

The view is responsible for the transformation of the raw mock table data used for this extract to a form Chiapas EDI can encode into HIPAA elements. The full list of possible mappings is located on C:\chiapas\docs\CGIF. For this extract, all of the possible CGIF mappings for the 270 specification is located in the file '5010_270.html'. Modifiers for the Addenda 1 specification are listed in '5010_270_A1.html', and is viewable with any web browser.

Decoder Tutorial

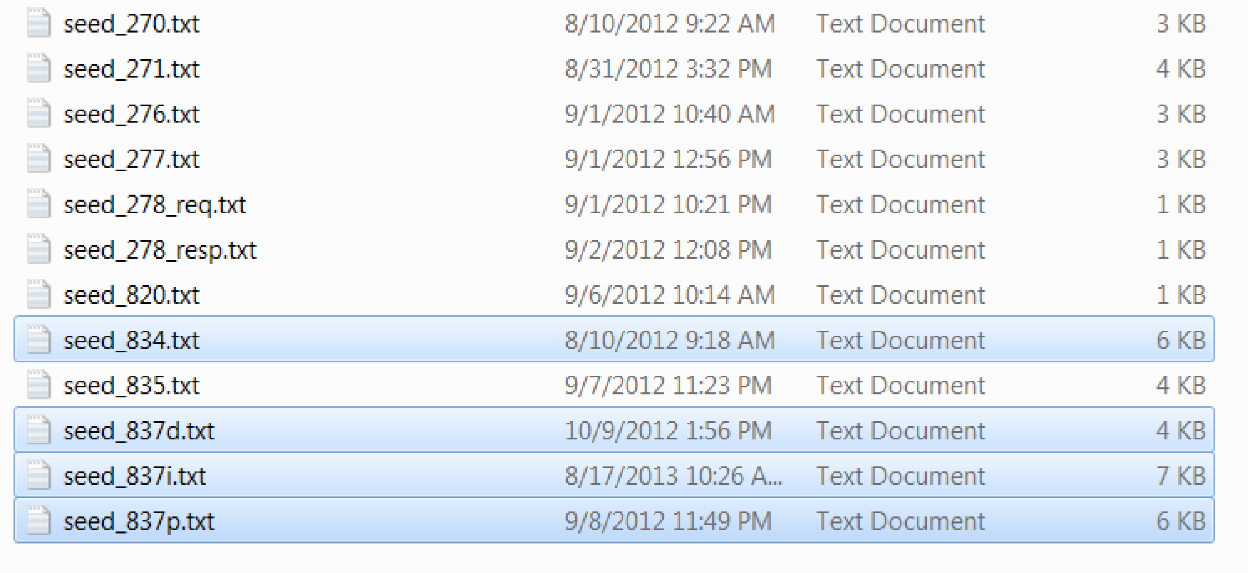
In this section, we'll investigate decoding arbitrary files placed into a directory and have them logged and storaged in one of four different tables.

Again, we'll start in the Scheduler interface and create a new trigger.

1. Launch C:\chiapas\bin\ChiapasStudio.exe
2. At the top of the screen, pull down the 'Window' menu option and select 'Scheduler Interface'.
3. Switch to the 'All Triggers' pane.
4. Press the 'New' button. Press OK.
5. Select the new trigger that was generated.
6. Double click in the Workflow File field, and in the resulting Open File Dialog, select the file C:\chiapas\xaml\binDecoder.xaml
7. Click on the Polling Info tab.
8. Select the 'Local Archive Poll' radio button.
9. For the 'Archive Dir' textbox, fill in: C:\chiapas\input
10. Press the 'Save' button.
11. Create a new folder called 'input' off of the C:\chiapas directory.
12. Switch to the Trigger Info tab, press 'Enabled', and press Save. Press OK.
13. Open SQL Server Management Studio and open up a new query window off of the chiapasData database.
14. Run the following queries:

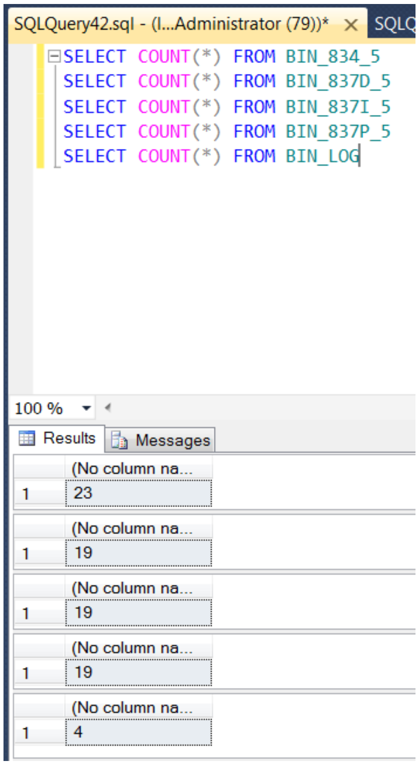
```sql
SELECT COUNT(*) FROM BIN_834_5
SELECT COUNT(*) FROM BIN_837D_5
SELECT COUNT(*) FROM BIN_837I_5
SELECT COUNT(*) FROM BIN_837P_5
SELECT COUNT(*) FROM BIN_LOG
```

15. The results should all be 0 (assuming this is the first time the example is run – if not, you can TRUNCATE all of these example tables).
16. Copy these four files from C:\chiapas\sample into the C:\chiapas\input directory:



seed_270.txt	8/10/2012 9:22 AM	Text Document	3 KB
seed_271.txt	8/31/2012 3:32 PM	Text Document	4 KB
seed_276.txt	9/1/2012 10:40 AM	Text Document	3 KB
seed_277.txt	9/1/2012 12:56 PM	Text Document	3 KB
seed_278_req.txt	9/1/2012 10:21 PM	Text Document	1 KB
seed_278_resp.txt	9/2/2012 12:08 PM	Text Document	1 KB
seed_820.txt	9/6/2012 10:14 AM	Text Document	1 KB
seed_834.txt	8/10/2012 9:18 AM	Text Document	6 KB
seed_835.txt	9/7/2012 11:23 PM	Text Document	4 KB
seed_837d.txt	10/9/2012 1:56 PM	Text Document	4 KB
seed_837i.txt	8/17/2013 10:26 A...	Text Document	7 KB
seed_837p.txt	9/8/2012 11:49 PM	Text Document	6 KB

17. Wait 30 seconds.
18. Re-run the the queries. Now, the output from SSMS should look like this:

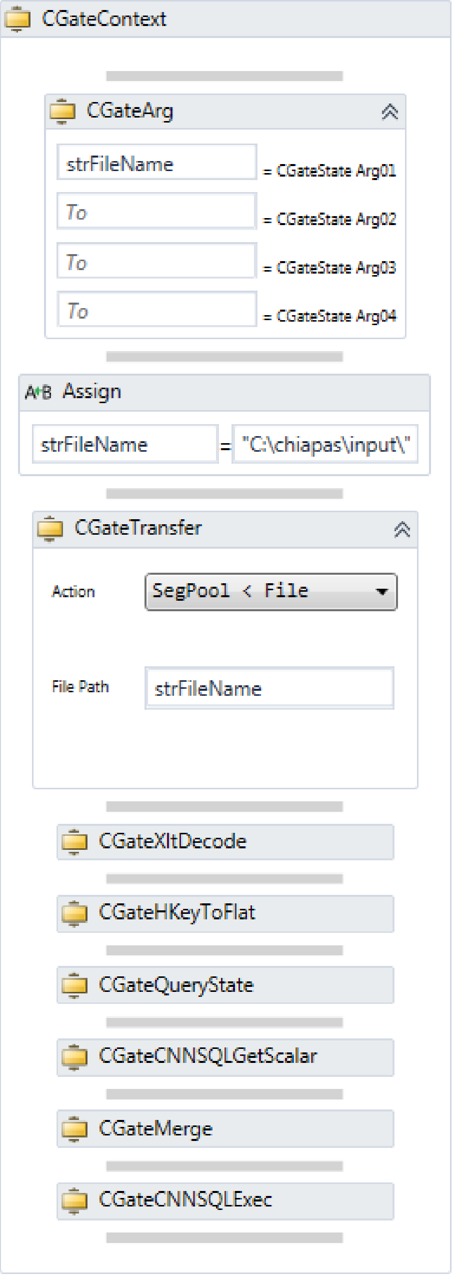


```
SELECT COUNT(*) FROM BIN_834_5
SELECT COUNT(*) FROM BIN_837D_5
SELECT COUNT(*) FROM BIN_837I_5
SELECT COUNT(*) FROM BIN_837P_5
SELECT COUNT(*) FROM BIN_LOG
```

```
   (No column na...
1  23
   (No column na...
1  19
   (No column na...
1  19
   (No column na...
1  19
   (No column na...
1  4
```

At the next poll, the automation system discovered four new files in the archive directory and passed these files to the binDecoder workflow. This workflow then worked in conjunction with two stored procedures located on the distribution database to route the data into one of four different tables. Note that because this workflow is using the 'Merge' functionality, arbitrary 837 and 834 files can be used with this process – mappings that cannot be stored in the existing tables will be logged as error messages to the MSGLOG table automatically. These missing fields can then be added to the destination tables, where subsequent decodes will detect these new fields and store information to them.

Let's go to the Workflow interface and open up the binDecoder.xaml:



strFileName = CGateState Arg01
To = CGateState Arg02
To = CGateState Arg03
To = CGateState Arg04

strFileName = "C:\chiapas\input\"

Action: SegPool < File
File Path: strFileName

CGateArg – This activity is the glue between the automation syste, and workflow. Here, the Local Archive Directory Poll will pass a single argument containing the filename (without path) that triggered this event.

Note If in the future you need the full path and filename of the triggering stimuli for a Local Archive Directory Poll, you could access the event data directly using a CGateCNNSQLGetScalar activity. Create a string variable within the CGateContext scope called TRIG_EVENT_ID. Then, set the variable via the CGateQueryState with an argument of "EVENT_ID". Then, set create a new CGateCNNSQLGetScalar activity with the "TriggerDB" connection and the following query:

```
"SELECT TSRC_DATA01 + '\' + EVENT_DATA01 FROM EVENTS INNER
JOIN TSRC ON EVENTS.TSRC_ID = TSRC.TSRC_ID WHERE EVENT_ID
= '" + TRIG_EVENT_ID + "'"
```

This will allow you to assign the full path and filename to an Activity-scoped variable.

Assign – This assignment restores the path information to the triggering filename.

CGateTransfer – This loads the passed filename into the SegPool register.

CGateXltDecode – Decodes the SegPool register to the HKey register.

CGateQueryState – This will query the Chiapas EDI engine as to the currently decoded specification and assign it to the strType string variable.

CGateCNNSQLGetScalar – This executes a SQL stored procedure on the chiapasData database. The return value passed from the stored procedure is assigned to a variable. The stored procedure is passed the specification name as an argument:

```
"USP_TEMP_TABLE '" + strType + "'"
```

When we look at the source code behind the USP_TEMP_TABLE stored procedure, we see the following:

```
-- Create a zero row temp table patterned off an existing bin table that will
receive a Chiapas Merge operation
CREATE PROCEDURE [dbo].[USP_TEMP_TABLE](@TYPE VARCHAR(20))
AS
BEGIN
      declare @random int, @upper int, @lower int, @rndtb varchar(20), @SQL
VARCHAR(1000)
      set @lower = 1
      set @upper = 999999
      select @random = ROUND(((@upper - @lower) * rand() + @lower),0)
      select @rndtb = 'temp'+substring(cast(@random as
varchar(20))+'000000',1,6)

      SET @TYPE = REPLACE(@TYPE, '.BIN','')

      SET @SQL =
      CASE WHEN @TYPE IN ('5010_834', '5010_834_A1')
             THEN 'SELECT TOP 0 * INTO ' + @rndtb + ' FROM BIN_834_5;'
             WHEN @TYPE IN ('5010_837D', '5010_837D_A1', '5010_837D_A2')
           THEN 'SELECT TOP 0 * INTO ' + @rndtb + ' FROM BIN_837D_5;'
             WHEN @TYPE IN ('5010_837I', '5010_837I_A1', '5010_837I_A2')
             THEN 'SELECT TOP 0 * INTO ' + @rndtb + ' FROM BIN_837I_5;'
             WHEN @TYPE IN ('5010_837P', '5010_837P_A1')
             THEN 'SELECT TOP 0 * INTO ' + @rndtb + ' FROM BIN_837P_5;'
      END

      SET @SQL = @SQL + 'ALTER TABLE ' + @rndtb + ' DROP COLUMN BIN_PK; ALTER
TABLE ' + @rndtb + ' ALTER COLUMN BIN_ID INT NULL;'

      EXECUTE(@SQL)
      SELECT @rndtb
END
```

This stored procedure will use the specification name to drive which of the existing BIN tables it copies and alters to a new temp table. This newly created table will be the target of the Chiapas EDI Merge activity, which will copy all mappings that the source and destination have in common, and log all mappings existing within the source but not the destination as a logged message. As all error log messages are recorded automatically at the end of every workflow execution by the automation system, no extra activities are needed for this to work.

CGateMerge - This activity will merge in the data from the loaded Flat register to the specified database table and log all fields that don't exist in the destination. In this way, Chiapas EDI can cope with files having slightly different sets of mappings and still have the bulk of the data stored successfully to a database.

CGateCNNSQLExec – This stored procedure call will transfer the contents of the temp table to one of the main BIN tables as well as log the results to the BIN_LOG table. It is executed using this syntax:

```
"USP_MOVE_TO_BIN '" + strTemp1.ToString() + "','" + strType + "','" +
strFileName + "'"
```

Looking at the SQL source code to the USP_MOVE_TO_BIN procedure, we see this:

```
- Move the newly imported file into a storage bin
CREATE PROCEDURE [dbo].[USP_MOVE_TO_BIN] (@BIN VARCHAR(100), @TYPE
VARCHAR(20), @FILENAME VARCHAR(200))
AS
BEGIN
      DECLARE @SQL VARCHAR(2000), @BIN_ID INT

      SET @TYPE = REPLACE(@TYPE, '.BIN','')

      -- First, insert a new BIN_LOG record
      BEGIN TRAN

            INSERT INTO BIN_LOG
            (                 BIN_FILENAME,
                              BIN_RECV_DATE,
                              BIN_TYPE
            )
            SELECT            @FILENAME,
                              GETDATE(),
                              @TYPE

            SET @BIN_ID = SCOPE_IDENTITY()

      COMMIT TRAN

      SET @SQL = 'UPDATE ' + @BIN + ' SET BIN_ID = ' +
CONVERT(VARCHAR,@BIN_ID) + ';'
      EXECUTE (@SQL)


      SET @SQL =
      CASE WHEN @TYPE IN ('5010_834', '5010_834_A1')
             THEN 'INSERT INTO BIN_834_5 SELECT * FROM ' + @BIN
             WHEN @TYPE IN ('5010_837D', '5010_837D_A1', '5010_837D_A2')
             THEN 'INSERT INTO BIN_837D_5 SELECT * FROM ' + @BIN
             WHEN @TYPE IN ('5010_837I', '5010_837I_A1', '5010_837I_A2')
             THEN 'INSERT INTO BIN_837I_5 SELECT * FROM ' + @BIN
             WHEN @TYPE IN ('5010_837P', '5010_837P_A1')
             THEN 'INSERT INTO BIN_837P_5 SELECT * FROM ' + @BIN
      END

      EXECUTE (@SQL)

      SET @SQL = 'DROP TABLE ' + @BIN + ';'
      EXEC(@SQL)
```

```
END
```

This code will create a new BIN_LOG entry and use that as a key to aggregate the new rows within the other BIN tables. Then, it drops the temp table to clean up.

Seed Specification Definitions

The following specifications define the output generated by the vw_Export_* views (when fed into the Chiapas EDI encoder) as well as describing the resulting seed files located within the c:\chiapas\sample directory.

Specification: 5010_270

Reference	Segment	CGIF	Mapping
Loop: ISA	GROUP HEADERS		
ISA02	Interchange Control Header	M1_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	Authorization Information - No Authorization Information Present
ISA04	Interchange Control Header	OEISA_S01_ISA04_SEC_NFO_OVL_PASSWD	Security Information - Password
ISA06	Interchange Control Header	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID	Interchange Sender ID - Mutually Defined
ISA08	Interchange Control Header	OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID	Interchange Receiver ID - Mutually Defined
ISA09	Interchange Control Header	OEISA_S01_ISA09_DT	Interchange Date
ISA10	Interchange Control Header	OEISA_S01_ISA10_TM	Interchange Time
ISA11	Interchange Control Header	OEISA_S01_ISA11_REPTN_SEPRTR	Repetition Separator
ISA12	Interchange Control Header	OEISA_S01_ISA12_VERSN_NR	Interchang Control Version Number
ISA13	Interchange Control Header	OEISA_S01_ISA13_ICN	Interchange Control Number
ISA14	Interchange Control Header	OEISA_S01_ISA14_ACK_REQ	Acknowledgment Requested
ISA15	Interchange Control Header	OEISA_S01_ISA15_USG_IND	Interchange Usage Indicator
ISA16	Interchange Control Header	OEISA_S01_ISA16_SUBELE_SEP	Component Element Separator
IEA01	Interchange Control Trailer	OEISA_S03_IEA01_GS_CT	Number of Included Functional Groups
IEA02	Interchange Control Trailer	OEISA_S03_IEA02_ICN	Interchange Control Number
Loop: GSHDR	GROUP HEADER		
GS02	Functional Group Header	GSHDR_S01_GS02_APP_SENDR_CD	Application Senders Code
GS03	Functional Group Header	GSHDR_S01_GS03_APP_RECVR_CD	Application Receivers Code
GS04	Functional Group Header	GSHDR_S01_GS04_DT	Date
GS05	Functional Group Header	GSHDR_S01_GS05_TM	Time
GS06	Functional Group Header	GSHDR_S01_GS06_GCN	Group Control Number
GS08	Functional Group Header	GSHDR_S01_GS08_VERS_ID_CD	Version Release Industry Identifier Code
GE01	Functional Group Trailer	GSHDR_S03_GE01_TS_CT	Number of Transaction Sets Included
GE02	Functional Group Trailer	GSHDR_S03_GE02_GCN	Group Control Number
Loop: STHDR	TRANSACTION SET HEADER		

ST02	Transaction Set Header	STHDR_S01_ST02_CONTRL_NR	Transaction Set Control Number
BHT01	Beginning of Hierarchical Transaction	STHDR_S02_BHT01_STRUCTR_CD	Hierarchical Structure Code
BHT02	Beginning of Hierarchical Transaction	STHDR_S02_BHT02_TS_PURPS_CD	Transaction Set Purpose Code
BHT03	Beginning of Hierarchical Transaction	STHDR_S02_BHT03_SUBMTR_TRANSCTN_ID	Submitter Transaction Identifier
BHT04	Beginning of Hierarchical Transaction	STHDR_S02_BHT04_TS_CREATN_DT	Transaction Set Creation Date
BHT05	Beginning of Hierarchical Transaction	STHDR_S02_BHT05_TS_CREATN_TIM	Transaction Set Creation Time
SE01	Transaction Set Trailer	STHDR_S04_SE01_SEG_CT	Transaction Segment Count
SE02	Transaction Set Trailer	STHDR_S04_SE02_TCN	Transaction Set Control Number
Loop: 2000A	INFORMATION SOURCE LEVEL		
HL01	Information Source Level	L2000A_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
Loop: 2100A	INFORMATION SOURCE NAME		
NM103	Information Source Name	L2100A_PR_S01_NM103_NFO_SOURC_LAST_ORG_NM_OVL_NONPRSN_ENTY	Information Source Last or Organization Name - Non-Person Entity
NM109	Information Source Name	L2100A_PR_S01_NM109_NFO_SOURC_PRIMRY_ID_OVL_FED_TAX_ID_NR	Information Source Primary Identifier - Federal Taxpayer’s Identification Number
Loop: 2000B	INFORMATION RECEIVER LEVEL		
HL01	Information Receiver Level	L2000B_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Information Receiver Level	L2000B_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
Loop: 2100B	INFORMATION RECEIVER NAME		
NM103	Information Receiver Name	L2100B_1P_S01_NM103_NFO_RECVR_LAST_ORG_NM_OVL_NONPRSN_ENTY	Information Receiver Last or Organization Name - Non-Person Entity
NM109	Information Receiver Name	L2100B_1P_S01_NM109_NFO_RECVR_ID_NR_OVL_FED_TAX_ID_NR	Information Receiver Identification Number - Federal Taxpayer’s Identification Number
Loop: 2000C	SUBSCRIBER LEVEL		
HL01	Subscriber Level	L2000C_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Subscriber Level	L2000C_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
HL04	Subscriber Level	L2000C_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
Loop: 2100C	SUBSCRIBER NAME		
NM102	Subscriber Name	L2100C_S01_NM102_ENTY_TYPE_QUAL	Entity Type Qualifier
NM103	Subscriber Name	L2100C_S01_NM103_SUB_LNAME	Subscriber Last Name
NM104	Subscriber Name	L2100C_S01_NM104_SUB_FNAME	Subscriber First Name
NM109	Subscriber Name	L2100C_S01_NM109_PRIMRY_ID_OVL_MEM_ID_NR	Subscriber Primary Identifier - Member Identification Number
DMG02	Subscriber Demographic Information	L2100C_S06_DMG02_BIRTH_DT_OVL_DT_CCYYMMDD	Subscriber Birth Date - Date Expressed in Format CCYYMMDD
DMG03	Subscriber Demographic Information	L2100C_S06_DMG03_GENDR_CD	Subscriber Gender Code
Loop: 2110C	SUBSCRIBER ELIGIBILITY OR BENEFIT INQUIRY		

EQ01	Subscriber Eligibility or Benefit Inquiry	L2110C_S01_EQ01_SVC_TYPE_CD	Service Type Code

Specification: 5010_271

Reference	Segment	CGIF	Mapping
Loop: ISA	GROUP HEADERS		
ISA02	Interchange Control Header	N1_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	Authorization Information - No Authorization Information Present
ISA04	Interchange Control Header	OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO	Security Information - No Security Information Present
ISA06	Interchange Control Header	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID	Interchange Sender ID - Mutually Defined
ISA08	Interchange Control Header	OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID	Interchange Receiver ID - Mutually Defined
ISA09	Interchange Control Header	OEISA_S01_ISA09_DT	Interchange Date
ISA10	Interchange Control Header	OEISA_S01_ISA10_TM	Interchange Time
ISA11	Interchange Control Header	OEISA_S01_ISA11_REPTN_SEPRTR	Repetition Separator
ISA12	Interchange Control Header	OEISA_S01_ISA12_VERSN_NR	Interchang Control Version Number
ISA13	Interchange Control Header	OEISA_S01_ISA13_ICN	Interchange Control Number
ISA14	Interchange Control Header	OEISA_S01_ISA14_ACK_REQ	Acknowledgment Requested
ISA15	Interchange Control Header	OEISA_S01_ISA15_USG_IND	Interchange Usage Indicator
ISA16	Interchange Control Header	OEISA_S01_ISA16_SUBELE_SEP	Component Element Separator
IEA01	Interchange Control Trailer	OEISA_S03_IEA01_GS_CT	Number of Included Functional Groups
IEA02	Interchange Control Trailer	OEISA_S03_IEA02_ICN	Interchange Control Number
Loop: GSHDR	GROUP HEADER		
GS02	Functional Group Header	GSHDR_S01_GS02_APP_SENDR_CD	Application Senders Code
GS03	Functional Group Header	GSHDR_S01_GS03_APP_RECVR_CD	Application Receivers Code
GS04	Functional Group Header	GSHDR_S01_GS04_DT	Date
GS05	Functional Group Header	GSHDR_S01_GS05_TM	Time
GS06	Functional Group Header	GSHDR_S01_GS06_GCN	Group Control Number
GS08	Functional Group Header	GSHDR_S01_GS08_VERS_ID_CD	Version Release Industry Identifier Code
GE01	Functional Group Trailer	GSHDR_S03_GE01_TS_CT	Number of Transaction Sets Included
GE02	Functional Group Trailer	GSHDR_S03_GE02_GCN	Group Control Number
Loop: STHDR	TRANSACTION SET HEADER		
ST02	Transaction Set Header	STHDR_S01_ST02_CONTRL_NR	Transaction Set Control Number
BHT01	Beginning of Hierarchical Transaction	STHDR_S02_BHT01_STRUCTR_CD	Hierarchical Structure Code
BHT02	Beginning of Hierarchical Transaction	STHDR_S02_BHT02_TS_PURPS_CD	Transaction Set Purpose Code

BHT03	Beginning of Hierarchical Transaction	STHDR_S02_BHT03_SUBMTR_TRANSCTN_ID	Submitter Transaction Identifier
BHT04	Beginning of Hierarchical Transaction	STHDR_S02_BHT04_TS_CREATN_DT	Transaction Set Creation Date
BHT05	Beginning of Hierarchical Transaction	STHDR_S02_BHT05_TS_CREATN_TIM	Transaction Set Creation Time
SE01	Transaction Set Trailer	STHDR_S04_SE01_SEG_CT	Transaction Segment Count
SE02	Transaction Set Trailer	STHDR_S04_SE02_TCN	Transaction Set Control Number
Loop: 2000A	INFORMATION SOURCE LEVEL		
HL01	Information Source Level	L2000A_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL04	Information Source Level	L2000A_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
Loop: 2100A	INFORMATION SOURCE NAME		
NM103	Information Source Name	L2100A_PR_S01_NM103_NFO_SOURC_LAST_ORG_NM_OVL_NONPRSN_ENTY	Information Source Last or Organization Name - Non-Person Entity
NM109	Information Source Name	L2100A_PR_S01_NM109_NFO_SOURC_PRIMRY_ID_OVL_FED_TAX_ID_NR	Information Source Primary Identifier - Federal Taxpayer’s Identification Number
Loop: 2000B	INFORMATION RECEIVER LEVEL		
HL01	Information Receiver Level	L2000B_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Information Receiver Level	L2000B_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
HL04	Information Receiver Level	L2000B_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
Loop: 2100B	INFORMATION RECEIVER NAME		
NM102	Information Receiver Name	L2100B_1P_S01_NM102_ENTY_TYPE_QUAL	Entity Type Qualifier
NM103	Information Receiver Name	L2100B_1P_S01_NM103_NFO_RECVR_LAST_ORG_NM	Information Receiver Last or Organization Name
NM109	Information Receiver Name	L2100B_1P_S01_NM109_NFO_RECVR_ID_NR_OVL_FED_TAX_ID_NR	Information Receiver Identification Number - Federal Taxpayer’s Identification Number
Loop: 2000C	SUBSCRIBER LEVEL		
HL01	Subscriber Level	L2000C_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Subscriber Level	L2000C_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
HL04	Subscriber Level	L2000C_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
Loop: 2100C	SUBSCRIBER NAME		
NM103	Subscriber Name	L2100C_S01_NM103_SUB_LNAME	Subscriber Last Name
NM104	Subscriber Name	L2100C_S01_NM104_SUB_FNAME	Subscriber First Name
NM109	Subscriber Name	L2100C_S01_NM109_PRIMRY_ID_OVL_MEM_ID_NR	Subscriber Primary Identifier - Member Identification Number
N301	Subscriber Address	L2100C_S03_N301_ADRS_LIN	Subscriber Address Line
N401	Subscriber City, State, ZIP Code	L2100C_S04_N401_CITY_NM	Subscriber City Name
N402	Subscriber City, State, ZIP Code	L2100C_S04_N402_STAT_CD	Subscriber State Code
N403	Subscriber City, State, ZIP Code	L2100C_S04_N403_SUB_POSTL_ZON_ZIP_CD	Subscriber Postal Zone or ZIP Code
DTP03	Subscriber Date	L2100C_S10_DTP03_DT_TM_PERD_OVL_PLAN_D8	Date Time Period - Plan (D8)
DTP03	Subscriber Date	L2100C_S10_DTP03_I02_DT_TM_PERD_OVL_ELIG_BEGN_D8	Date Time Period - Eligibility Begin (D8)

Loop: 2110C	**SUBSCRIBER ELIGIBILITY OR BENEFIT INFORMATION**		
EB01	**Subscriber Eligibility or Benefit Information**	**L2110C_S01_EB01_BENFT_NFO**	**Eligibility or Benefit Information**
EB03	**Subscriber Eligibility or Benefit Information**	**L2110C_S01_EB03_SVC_TYPE_CD**	**Service Type Code**

Specification: 5010_276

Reference	Segment	CGIF	Mapping
Loop: ISA	**GROUP HEADERS**		
ISA02	**Interchange Control Header**	**O0_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO**	**Authorization Information - No Authorization Information Present**
ISA04	**Interchange Control Header**	**OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO**	**Security Information - No Security Information Present**
ISA06	**Interchange Control Header**	**OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID**	**Interchange Sender ID - Mutually Defined**
ISA08	**Interchange Control Header**	**OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID**	**Interchange Receiver ID - Mutually Defined**
ISA09	**Interchange Control Header**	**OEISA_S01_ISA09_DT**	**Interchange Date**
ISA10	**Interchange Control Header**	**OEISA_S01_ISA10_TM**	**Interchange Time**
ISA11	**Interchange Control Header**	**OEISA_S01_ISA11_REPTN_SEPRTR**	**Repetition Separator**
ISA12	**Interchange Control Header**	**OEISA_S01_ISA12_VERSN_NR**	**Interchang Control Version Number**
ISA13	**Interchange Control Header**	**OEISA_S01_ISA13_ICN**	**Interchange Control Number**
ISA14	**Interchange Control Header**	**OEISA_S01_ISA14_ACK_REQ**	**Acknowledgment Requested**
ISA15	**Interchange Control Header**	**OEISA_S01_ISA15_USG_IND**	**Interchange Usage Indicator**
ISA16	**Interchange Control Header**	**OEISA_S01_ISA16_SUBELE_SEP**	**Component Element Separator**
IEA01	**Interchange Control Trailer**	**OEISA_S03_IEA01_GS_CT**	**Number of Included Functional Groups**
IEA02	**Interchange Control Trailer**	**OEISA_S03_IEA02_ICN**	**Interchange Control Number**
Loop: GSHDR	**GROUP HEADER**		
GS02	**Functional Group Header**	**GSHDR_S01_GS02_APP_SENDR_CD**	**Application Senders Code**
GS03	**Functional Group Header**	**GSHDR_S01_GS03_APP_RECVR_CD**	**Application Receivers Code**
GS04	**Functional Group Header**	**GSHDR_S01_GS04_DT**	**Date**
GS05	**Functional Group Header**	**GSHDR_S01_GS05_TM**	**Time**
GS06	**Functional Group Header**	**GSHDR_S01_GS06_GCN**	**Group Control Number**
GE01	**Functional Group Trailer**	**GSHDR_S03_GE01_TS_CT**	**Number of Transaction Sets Included**
GE02	**Functional Group Trailer**	**GSHDR_S03_GE02_GCN**	**Group Control Number**
Loop: STHDR	**TRANSACTION SET HEADER**		
ST02	**Transaction Set Header**	**STHDR_S01_ST02_CONTRL_NR**	**Transaction Set Control Number**
ST03	**Transaction Set Header**	**STHDR_S01_ST03_VERS_RELS_INDSTRY_ID**	**Version, Release, or Industry Identifier**

BHT01	Beginning of Hierarchical Transaction	STHDR_S02_BHT01_STRUCTR_CD	Hierarchical Structure Code
BHT03	Beginning of Hierarchical Transaction	STHDR_S02_BHT03_REF_ID	Reference Identification
BHT04	Beginning of Hierarchical Transaction	STHDR_S02_BHT04_TS_CREATN_DT	Transaction Set Creation Date
BHT05	Beginning of Hierarchical Transaction	STHDR_S02_BHT05_TS_CREATN_TIM	Transaction Set Creation Time
SE01	Transaction Set Trailer	STHDR_S04_SE01_SEG_CT	Transaction Segment Count
SE02	Transaction Set Trailer	STHDR_S04_SE02_TCN	Transaction Set Control Number
Loop: 2000A	INFORMATION SOURCE LEVEL		
HL01	Information Source Level	L2000A_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
Loop: 2100A	PAYER NAME		
NM103	Payer Name	L2100A_S01_NM103_PAYR_NM_OVL_NONPRSN_ENTY	Payer Name - Non-Person Entity
NM109	Payer Name	L2100A_S01_NM109_PAYR_ID_OVL_PAYR_ID	Payer Identifier - Payor Identification
Loop: 2000B	INFORMATION RECEIVER LEVEL		
HL01	Information Receiver Level	L2000B_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Information Receiver Level	L2000B_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
Loop: 2100B	INFORMATION RECEIVER NAME		
NM102	Information Receiver Name	L2100B_S01_NM102_ENTY_TYPE_QUAL	Entity Type Qualifier
NM103	Information Receiver Name	L2100B_S01_NM103_NFO_RECVR_LAST_ORG_NM	Information Receiver Last or Organization Name
NM109	Information Receiver Name	L2100B_S01_NM109_NFO_RECVR_ID_NR_OVL_ETN_NR	Information Receiver Identification Number - Electronic Transmitter Identification Number (ETIN)
Loop: 2000C	SERVICE PROVIDER LEVEL		
HL01	Service Provider Level	L2000C_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Service Provider Level	L2000C_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
Loop: 2100C	PROVIDER NAME		
NM102	Provider Name	L2100C_S01_NM102_ENTY_TYPE_QUAL	Entity Type Qualifier
NM103	Provider Name	L2100C_S01_NM103_PROV_LAST_ORG_NM	Provider Last or Organization Name
NM109	Provider Name	L2100C_S01_NM109_PROV_ID_OVL_FED_TAX_ID_NR	Provider Identifier - Federal Taxpayer’s Identification Number
Loop: 2000D	SUBSCRIBER LEVEL		
HL01	Subscriber Level	L2000D_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Subscriber Level	L2000D_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
HL04	Subscriber Level	L2000D_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
DMG02	Subscriber Demographic Information	L2000D_S02_DMG02_BIRTH_DT_OVL_DT_CCYYMMDD	Subscriber Birth Date - Date Expressed in Format CCYYMMDD
DMG03	Subscriber Demographic Information	L2000D_S02_DMG03_GENDR_CD	Subscriber Gender Code

Loop: 2100D	**SUBSCRIBER NAME**		
NM103	**Subscriber Name**	**L2100D_S01_NM103_SUB_LNAME_OVL_PERSN**	**Subscriber Last Name - Person**
NM104	**Subscriber Name**	**L2100D_S01_NM104_SUB_FNAME**	**Subscriber First Name**
NM109	**Subscriber Name**	**L2100D_S01_NM109_SUB_ID_OVL_MEM_ID_NR**	**Subscriber Identifier - Member Identification Number**
Loop: 2200D	**CLAIM STATUS TRACKING NUMBER**		
TRN02	**Claim Status Tracking Number**	**L2200D_S01_TRN02_CURNT_TRANSCTN_TRAC_NR**	**Current Transaction Trace Number**
REF02	**Payer Claim Control Number**	**L2200D_S02_REF02_PAYR_CLM_CONTRL_NR_OVL_PAYR_CLM_NR**	**Payer Claim Control Number - Payor's Claim Number**
DTP03	**Claim Service Date**	**L2200D_S10_DTP03_SVC_PERD_OVL_SVC_RD8**	**Claim Service Period - Service (RD8)**

Specification: 5010_277

Reference	Segment	CGIF	Mapping
Loop: ISA	**GROUP HEADERS**		
ISA02	**Interchange Control Header**	**P0_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO**	**Authorization Information - No Authorization Information Present**
ISA04	**Interchange Control Header**	**OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO**	**Security Information - No Security Information Present**
ISA06	**Interchange Control Header**	**OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID**	**Interchange Sender ID - Mutually Defined**
ISA08	**Interchange Control Header**	**OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID**	**Interchange Receiver ID - Mutually Defined**
ISA09	**Interchange Control Header**	**OEISA_S01_ISA09_DT**	**Interchange Date**
ISA10	**Interchange Control Header**	**OEISA_S01_ISA10_TM**	**Interchange Time**
ISA11	**Interchange Control Header**	**OEISA_S01_ISA11_REPTN_SEPRTR**	**Repetition Separator**
ISA12	**Interchange Control Header**	**OEISA_S01_ISA12_VERSN_NR**	**Interchang Control Version Number**
ISA13	**Interchange Control Header**	**OEISA_S01_ISA13_ICN**	**Interchange Control Number**
ISA14	**Interchange Control Header**	**OEISA_S01_ISA14_ACK_REQ**	**Acknowledgment Requested**
ISA15	**Interchange Control Header**	**OEISA_S01_ISA15_USG_IND**	**Interchange Usage Indicator**
ISA16	**Interchange Control Header**	**OEISA_S01_ISA16_SUBELE_SEP**	**Component Element Separator**
IEA01	**Interchange Control Trailer**	**OEISA_S03_IEA01_GS_CT**	**Number of Included Functional Groups**
IEA02	**Interchange Control Trailer**	**OEISA_S03_IEA02_ICN**	**Interchange Control Number**
Loop: GSHDR	**GROUP HEADER**		
GS02	**Functional Group Header**	**GSHDR_S01_GS02_APP_SENDR_CD**	**Application Senders Code**
GS03	**Functional Group Header**	**GSHDR_S01_GS03_APP_RECVR_CD**	**Application Receivers Code**
GS04	**Functional Group Header**	**GSHDR_S01_GS04_DT**	**Date**
GS05	**Functional Group Header**	**GSHDR_S01_GS05_TM**	**Time**
GS06	**Functional Group Header**	**GSHDR_S01_GS06_GCN**	**Group Control Number**
GE01	**Functional Group Trailer**	**GSHDR_S03_GE01_TS_CT**	**Number of Transaction Sets Included**
GE02	**Functional Group Trailer**	**GSHDR_S03_GE02_GCN**	**Group Control Number**

Loop: STHDR	**TRANSACTION SET HEADER**		
ST02	**Transaction Set Header**	**STHDR_S01_ST02_CONTRL_NR**	**Transaction Set Control Number**
ST03	**Transaction Set Header**	**STHDR_S01_ST03_VERS_RELS_INDSTRY_ID**	**Version, Release, or Industry Identifier**
BHT01	**Beginning of Hierarchical Transaction**	**STHDR_S02_BHT01_STRUCTR_CD**	**Hierarchical Structure Code**
BHT03	**Beginning of Hierarchical Transaction**	**STHDR_S02_BHT03_ORGNTR_APLCTN_TRANSCTN_ID**	**Originator Application Transaction Identifier**
BHT04	**Beginning of Hierarchical Transaction**	**STHDR_S02_BHT04_TS_CREATN_DT**	**Transaction Set Creation Date**
BHT05	**Beginning of Hierarchical Transaction**	**STHDR_S02_BHT05_TS_CREATN_TIM**	**Transaction Set Creation Time**
SE01	**Transaction Set Trailer**	**STHDR_S04_SE01_SEG_CT**	**Transaction Segment Count**
SE02	**Transaction Set Trailer**	**STHDR_S04_SE02_TCN**	**Transaction Set Control Number**
Loop: 2000A	**INFORMATION SOURCE LEVEL**		
HL01	**Information Source Level**	**L2000A_S01_HL01_HIERCHCL_ID_NR**	**Hierarchical ID Number**
Loop: 2100A	**PAYER NAME**		
NM103	**Payer Name**	**L2100A_S01_NM103_PAYR_NM_OVL_NONPRSN_ENTY**	**Payer Name - Non-Person Entity**
NM109	**Payer Name**	**L2100A_S01_NM109_PAYR_ID_OVL_PAYR_ID**	**Payer Identifier - Payor Identification**
Loop: 2000B	**INFORMATION RECEIVER LEVEL**		
HL01	**Information Receiver Level**	**L2000B_S01_HL01_HIERCHCL_ID_NR**	**Hierarchical ID Number**
HL02	**Information Receiver Level**	**L2000B_S01_HL02_HIERCHCL_PARNT_ID_NR**	**Hierarchical Parent ID Number**
HL04	**Information Receiver Level**	**L2000B_S01_HL04_HIERCHCL_CHILD_CD**	**Hierarchical Child Code**
Loop: 2100B	**INFORMATION RECEIVER NAME**		
NM102	**Information Receiver Name**	**L2100B_S01_NM102_ENTY_TYPE_QUAL**	**Entity Type Qualifier**
NM103	**Information Receiver Name**	**L2100B_S01_NM103_NFO_RECVR_LAST_ORG_NM**	**Information Receiver Last or Organization Name**
NM109	**Information Receiver Name**	**L2100B_S01_NM109_NFO_RECVR_ID_NR_OVL_ETN_NR**	**Information Receiver Identification Number - Electronic Transmitter Identification Number (ETIN)**
Loop: 2000C	**SERVICE PROVIDER LEVEL**		
HL01	**Service Provider Level**	**L2000C_S01_HL01_HIERCHCL_ID_NR**	**Hierarchical ID Number**
HL02	**Service Provider Level**	**L2000C_S01_HL02_HIERCHCL_PARNT_ID_NR**	**Hierarchical Parent ID Number**
HL04	**Service Provider Level**	**L2000C_S01_HL04_HIERCHCL_CHILD_CD**	**Hierarchical Child Code**
Loop: 2100C	**PROVIDER NAME**		
NM102	**Provider Name**	**L2100C_S01_NM102_ENTY_TYPE_QUAL**	**Entity Type Qualifier**
NM103	**Provider Name**	**L2100C_S01_NM103_PROV_LAST_ORG_NM**	**Provider Last or Organization Name**
NM109	**Provider Name**	**L2100C_S01_NM109_PROV_ID_OVL_FED_TAX_ID_NR**	**Provider Identifier - Federal Taxpayer’s Identification Number**
Loop: 2000D	**SUBSCRIBER LEVEL**		
HL01	**Subscriber Level**	**L2000D_S01_HL01_HIERCHCL_ID_NR**	**Hierarchical ID Number**

HL02	Subscriber Level	L2000D_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
HL04	Subscriber Level	L2000D_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
Loop: 2100D	SUBSCRIBER NAME		
NM103	Subscriber Name	L2100D_S01_NM103_SUB_LNAME_OVL_PERSN	Subscriber Last Name - Person
NM104	Subscriber Name	L2100D_S01_NM104_SUB_FNAME	Subscriber First Name
NM109	Subscriber Name	L2100D_S01_NM109_SUB_ID_OVL_MEM_ID_NR	Subscriber Identifier - Member Identification Number
Loop: 2200D	CLAIM STATUS TRACKING NUMBER		
TRN02	Claim Status Tracking Number	L2200D_S01_TRN02_REFNCD_TRANSCTN_TRAC_NR	Referenced Transaction Trace Number
TRN01-01	Claim Status Tracking Number	L2200D_S02_STC01_01_HLTHCR_CLM_STATS_CATGRY_CD	Health Care Claim Status Category Code
TRN01-02	Claim Status Tracking Number	L2200D_S02_STC01_02_STATS_CD	Status Code
STC02	Claim Level Status Information	L2200D_S02_STC02_STATS_NFO_EFF_DT	Status Information Effective Date
REF02	Payer Claim Control Number	L2200D_S03_REF02_PAYR_CLM_CONTRL_NR_OVL_PAYR_CLM_NR	Payer Claim Control Number - Payor’s Claim Number
DTP03	Claim Service Date	L2200D_S09_DTP03_SVC_PERD_OVL_SVC_RD8	Claim Service Period - Service (RD8)

Specification: 5010_277CA

Reference	Segment	CGIF	Mapping
Loop: ISA	ISA		
ISA02	Interchange Control Header	P5_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	Authorization Information - No Authorization Information Present
ISA04	Interchange Control Header	OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO	Security Information - No Security Information Present
ISA06	Interchange Control Header	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID	Interchange Sender ID - Mutually Defined
ISA08	Interchange Control Header	OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID	Interchange Receiver ID - Mutually Defined
ISA11	Interchange Control Header	OEISA_S01_ISA11_REPTN_SEPRTR	Repetition Separator
ISA12	Interchange Control Header	OEISA_S01_ISA12_VERSN_NR	Interchang Control Version Number
ISA13	Interchange Control Header	OEISA_S01_ISA13_ICN	Interchange Control Number
ISA14	Interchange Control Header	OEISA_S01_ISA14_ACK_REQ	Acknowledgment Requested
ISA15	Interchange Control Header	OEISA_S01_ISA15_USG_IND	Interchange Usage Indicator
ISA16	Interchange Control Header	OEISA_S01_ISA16_SUBELE_SEP	Component Element Separator
IEA02	Interchange Control Trailer	OEISA_S03_IEA02_ICN	Interchange Control Number
Loop: GSHDR	GROUP HEADER		
GS02	Function Group Header	GSHDR_S01_GS02_APP_SENDR_CD	Application Senders Code
GS03	Function Group Header	GSHDR_S01_GS03_APP_RECVR_CD	Application Receivers Code
GS06	Function Group Header	GSHDR_S01_GS06_GCN	Group Control Number
GS08	Function Group Header	GSHDR_S01_GS08_VERS_ID_CD	Version Release Industry Identifier Code
GE02	Functional Group Trailer	GSHDR_S03_GE02_GCN	Group Control Number
Loop: STHDR	TRANSACTION SET HEADER		

ST02	Transaction Set Header	STHDR_S01_ST02_CONTRL_NR	Transaction Set Control Number
ST03	Transaction Set Header	STHDR_S01_ST03_IMPLMNTN_CONVNTN_REFNC	Implementation Convention Reference
BHT03	Beginning of Hierarchical Transaction	STHDR_S02_BHT03_REF_ID	Reference Identification
BHT04	Beginning of Hierarchical Transaction	STHDR_S02_BHT04_TS_CREATN_DT	Transaction Set Creation Date
BHT05	Beginning of Hierarchical Transaction	STHDR_S02_BHT05_TS_CREATN_TIM	Transaction Set Creation Time
BHT06	Beginning of Hierarchical Transaction	STHDR_S02_BHT06_TRANSCTN_TYPE_CD	Transaction Type Code
SE02	Transaction Set Trailer	STHDR_S06_SE02_TCN	Transaction Set Control Number
Loop: 2000A	INFORMATION SOURCE LEVEL		
HL04	Information Source Level	L2000A_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
Loop: 2100A	INFORMATION SOURCE NAME		
NM103	Information Source Name	L2100A_PR_S01_NM103_NFO_SOURC_LAST_ORG_NM_OVL_NON_PERSN_ENTY	Information Source Last or Organization Name - Non-Person Entity
NM109	Information Source Name	L2100A_PR_S01_NM109_NFO_SOURC_PRIMRY_ID_OVL_PAYR_ID	Information Source Primary Identifier - Payor Identification
Loop: 2200A	TRANSMISSION RECEIPT CONTROL IDENTIFIER		
TRN02	Transmission Receipt Control Identifier	L2200A_S01_TRN02_NFO_SRC_APPLCTN_TRAC_ID	Information Source Application Trace Identifier
DTP03	Information Source Receipt Date	L2200A_S02_DTP03_DT_TM_PERD_OVL_RECVD_D8	Information Source Receipt Date - Received (D8)
DTP03	Information Source Process Date	L2200A_S03_DTP03_NFO_SRC_PROCSS_DT_OVL_PROCSS_D8	Information Source Process Date - Process (D8)
Loop: 2000B	INFORMATION RECEIVER LEVEL		
HL04	Information Receiver Level	L2000B_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
Loop: 2100B	INFORMATION RECEIVER NAME		
NM103	Information Receiver Name	L2100B_S01_NM103_NFO_RECVR_LAST_ORG_NM_OVL_NON_PERSN_ENTY	Information Receiver Last or Organization Name - Non-Person Entity
NM109	Information Receiver Name	L2100B_S01_NM109_NFO_SOURC_PRIMRY_ID_OVL_ETIN	Information Receiver Primary Identifier - Electronic Transmitter Identification Number (ETIN)
Loop: 2200B	INFORMATION RECEIVER APPLICATION TRACE IDENTIFIER		
TRN02	Information Receiver Application Trace Identifier	L2200B_S01_TRN02_CLM_TRANSCTN_BATCH_NR	Claim Transaction Batch Number
TRN01-01	Information Receiver Application Trace Identifier	L2200B_S02_STC01_01_HLTH_CR_CLM_STATS_CAT_CD	Health Care Claim Status Category Code
TRN01-02	Information Receiver Application Trace Identifier	L2200B_S02_STC01_02_HLTH_CR_CLM_STATS_CD	Health Care Claim Status Code
STC02	Information Receiver Status Information	L2200B_S02_STC02_STATS_NFO_EFF_DT	Status Information Effective Date
STC03	Information Receiver Status Information	L2200B_S02_STC03_ACTN_CD	Action Code
STC04	Information Receiver Status Information	L2200B_S02_STC04_TOTL_SUBMTD_CHGS_WORK	Total Submitted Charges for Work
QTY02	Total Accepted Quantity	L2200B_S03_QTY02_TOTL_ACCPTD_QTY	Total Accepted Quantity

QTY02	Total Rejected Quantity	L2200B_S04_QTY02_TOTL_REJECTD_QTY	Total Rejected Quantity

Specification: 5010_278_REQ

Reference	Segment	CGIF	Mapping
Loop: ISA	GROUP HEADERS		
ISA02	Interchange Control Header	Q0_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	Authorization Information - No Authorization Information Present
ISA04	Interchange Control Header	OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO	Security Information - No Security Information Present
ISA06	Interchange Control Header	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID	Interchange Sender ID - Mutually Defined
ISA08	Interchange Control Header	OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID	Interchange Receiver ID - Mutually Defined
ISA09	Interchange Control Header	OEISA_S01_ISA09_DT	Interchange Date
ISA10	Interchange Control Header	OEISA_S01_ISA10_TM	Interchange Time
ISA11	Interchange Control Header	OEISA_S01_ISA11_REPTN_SEPRTR	Repetition Separator
ISA12	Interchange Control Header	OEISA_S01_ISA12_VERSN_NR	Interchang Control Version Number
ISA13	Interchange Control Header	OEISA_S01_ISA13_ICN	Interchange Control Number
ISA14	Interchange Control Header	OEISA_S01_ISA14_ACK_REQ	Acknowledgment Requested
ISA15	Interchange Control Header	OEISA_S01_ISA15_USG_IND	Interchange Usage Indicator
ISA16	Interchange Control Header	OEISA_S01_ISA16_SUBELE_SEP	Component Element Separator
IEA01	Interchange Control Trailer	OEISA_S03_IEA01_GS_CT	Number of Included Functional Groups
IEA02	Interchange Control Trailer	OEISA_S03_IEA02_ICN	Interchange Control Number
Loop: GSHDR	GROUP HEADER		
GS02	Functional Group Header	GSHDR_S01_GS02_APP_SENDR_CD	Application Senders Code
GS03	Functional Group Header	GSHDR_S01_GS03_APP_RECVR_CD	Application Receivers Code
GS04	Functional Group Header	GSHDR_S01_GS04_DT	Date
GS05	Functional Group Header	GSHDR_S01_GS05_TM	Time
GS06	Functional Group Header	GSHDR_S01_GS06_GCN	Group Control Number
GE01	Functional Group Trailer	GSHDR_S03_GE01_TS_CT	Number of Transaction Sets Included
GE02	Functional Group Trailer	GSHDR_S03_GE02_GCN	Group Control Number
Loop: STHDR	TRANSACTION SET HEADER		
ST02	Transaction Set Header	STHDR_S01_ST02_CONTRL_NR	Transaction Set Control Number
ST03	Transaction Set Header	STHDR_S01_ST03_IMPLMNTN_GUID_VERSN_NM	Implementation Guide Version Name
BHT01	Beginning of Hierarchical Transaction	STHDR_S02_BHT01_STRUCTR_CD	Hierarchical Structure Code
BHT02	Beginning of Hierarchical Transaction	STHDR_S02_BHT02_TS_PURPS_CD	Transaction Set Purpose Code
BHT03	Beginning of Hierarchical Transaction	STHDR_S02_BHT03_SUBMTR_TRANSCTN_ID	Submitter Transaction Identifier

BHT04	Beginning of Hierarchical Transaction	STHDR_S02_BHT04_TS_CREATN_DT	Transaction Set Creation Date
BHT05	Beginning of Hierarchical Transaction	STHDR_S02_BHT05_TS_CREATN_TIM	Transaction Set Creation Time
SE01	Transaction Set Trailer	STHDR_S04_SE01_SEG_CT	Transaction Segment Count
SE02	Transaction Set Trailer	STHDR_S04_SE02_TCN	Transaction Set Control Number
Loop: 2000A	UTILIZATION MANAGEMENT ORGANIZATION (UMO) LEVEL		
HL01	Utilization Management Organization (UMO) Level	L2000A_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
Loop: 2010A	UTILIZATION MANAGEMENT ORGANIZATION (UMO) NAME		
NM102	Utilization Management Organization (UMO) Name	L2010A_X3_S01_NM102_ENTY_TYPE_QUAL	Entity Type Qualifier
NM103	Utilization Management Organization (UMO) Name	L2010A_X3_S01_NM103_UMO_NM_LAST_ORG_NM	Utilization Management Organization (UMO) Name Last or Organization Name
NM109	Utilization Management Organization (UMO) Name	L2010A_X3_S01_NM109_ID_CD_OVL_EMPLYR_ID_NR	Utilization Management Organization (UMO) Identification Code - Employer’s Identification Number
Loop: 2000B	REQUESTER LEVEL		
HL01	Requester Level	L2000B_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Requester Level	L2000B_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
Loop: 2010B	REQUESTER NAME		
NM102	Requester Name	L2010B_1P_S01_NM102_ENTY_TYPE_QUAL	Entity Type Qualifier
NM103	Requester Name	L2010B_1P_S01_NM103_REQSTR_LAST_ORG_NM	Requester Last or Organization Name
NM104	Requester Name	L2010B_1P_S01_NM104_REQSTR_FNAME	Requester First Name
NM109	Requester Name	L2010B_1P_S01_NM109_REQSTR_ID_OVL_ETN_NR	Requester Identifier - Electronic Transmitter Identification Number (ETIN)
Loop: 2000C	SUBSCRIBER LEVEL		
HL01	Subscriber Level	L2000C_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Subscriber Level	L2000C_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
Loop: 2010C	SUBSCRIBER NAME		
NM103	Subscriber Name	L2010C_S01_NM103_SUB_LNAME	Subscriber Last Name
NM104	Subscriber Name	L2010C_S01_NM104_SUB_FNAME	Subscriber First Name
NM109	Subscriber Name	L2010C_S01_NM109_PRIMRY_ID_OVL_MEM_ID_NR	Subscriber Primary Identifier - Member Identification Number
Loop: 2000E	PATIENT EVENT LEVEL		
HL01	Patient Event Level	L2000E_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Patient Event Level	L2000E_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
HL04	Patient Event Level	L2000E_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
UM01	Health Care Services Review Information	L2000E_S03_UM01_REQST_CATGRY_CD	Request Category Code

UM02	Health Care Services Review Information	L2000E_S03_UM02_CERT_TYPE_CD	Certification Type Code
UM04-01	Health Care Services Review Information	L2000E_S03_UM04_01_FACLTY_TYPE_CD	Facility Type Code
UM04-02	Health Care Services Review Information	L2000E_S03_UM04_02_FACLTY_CD_QUAL	Facility Code Qualifier
UM09	Health Care Services Review Information	L2000E_S03_UM09_RELS_NFO_CD	Release of Information Code
DTP03	Admission Date	L2000E_S11_DTP03_PROPSD_ACTL_ADMSN_DT_OVL_ADMSN_D8	Proposed or Actual Admission Date - Admission (D8)

Specification: 5010_278_RESP

Reference	Segment	CGIF	Mapping
Loop: ISA	GROUP HEADERS		
ISA02	Interchange Control Header	R0_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	Authorization Information - No Authorization Information Present
ISA04	Interchange Control Header	OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO	Security Information - No Security Information Present
ISA06	Interchange Control Header	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID	Interchange Sender ID - Mutually Defined
ISA08	Interchange Control Header	OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID	Interchange Receiver ID - Mutually Defined
ISA09	Interchange Control Header	OEISA_S01_ISA09_DT	Interchange Date
ISA10	Interchange Control Header	OEISA_S01_ISA10_TM	Interchange Time
ISA11	Interchange Control Header	OEISA_S01_ISA11_REPTN_SEPRTR	Repetition Separator
ISA12	Interchange Control Header	OEISA_S01_ISA12_VERSN_NR	Interchang Control Version Number
ISA13	Interchange Control Header	OEISA_S01_ISA13_ICN	Interchange Control Number
ISA14	Interchange Control Header	OEISA_S01_ISA14_ACK_REQ	Acknowledgment Requested
ISA15	Interchange Control Header	OEISA_S01_ISA15_USG_IND	Interchange Usage Indicator
ISA16	Interchange Control Header	OEISA_S01_ISA16_SUBELE_SEP	Component Element Separator
IEA01	Interchange Control Trailer	OEISA_S03_IEA01_GS_CT	Number of Included Functional Groups
IEA02	Interchange Control Trailer	OEISA_S03_IEA02_ICN	Interchange Control Number
Loop: GSHDR	GROUP HEADER		
GS02	Functional Group Header	GSHDR_S01_GS02_APP_SENDR_CD	Application Senders Code
GS03	Functional Group Header	GSHDR_S01_GS03_APP_RECVR_CD	Application Receivers Code
GS04	Functional Group Header	GSHDR_S01_GS04_DT	Date
GS05	Functional Group Header	GSHDR_S01_GS05_TM	Time
GS06	Functional Group Header	GSHDR_S01_GS06_GCN	Group Control Number
GE01	Functional Group Trailer	GSHDR_S03_GE01_TS_CT	Number of Transaction Sets Included
GE02	Functional Group Trailer	GSHDR_S03_GE02_GCN	Group Control Number
Loop: STHDR	TRANSACTION SET HEADER		
ST02	Transaction Set Header	STHDR_S01_ST02_CONTRL_NR	Transaction Set Control Number
ST03	Transaction Set Header	STHDR_S01_ST03_IMPLMNTN_GUID_VERSN_NM	Implementation Guide Version Name

BHT01	Beginning of Hierarchical Transaction	STHDR_S02_BHT01_STRUCTR_CD	Hierarchical Structure Code
BHT03	Beginning of Hierarchical Transaction	STHDR_S02_BHT03_SUBMTR_TRANSCTN_ID	Submitter Transaction Identifier
BHT04	Beginning of Hierarchical Transaction	STHDR_S02_BHT04_TS_CREATN_DT	Transaction Set Creation Date
BHT05	Beginning of Hierarchical Transaction	STHDR_S02_BHT05_TS_CREATN_TIM	Transaction Set Creation Time
BHT06	Beginning of Hierarchical Transaction	STHDR_S02_BHT06_TRANSCTN_TYPE_CD	Transaction Type Code
SE01	Transaction Set Trailer	STHDR_S04_SE01_SEG_CT	Transaction Segment Count
SE02	Transaction Set Trailer	STHDR_S04_SE02_TCN	Transaction Set Control Number
Loop: 2000A	UTILIZATION MANAGEMENT ORGANIZATION (UMO) LEVEL		
HL01	Utilization Management Organization (UMO) Level	L2000A_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL04	Utilization Management Organization (UMO) Level	L2000A_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
Loop: 2010A	UTILIZATION MANAGEMENT ORGANIZATION (UMO) NAME		
NM102	Utilization Management Organization (UMO) Name	L2010A_X3_S01_NM102_ENTY_TYPE_QUAL	Entity Type Qualifier
NM103	Utilization Management Organization (UMO) Name	L2010A_X3_S01_NM103_UMO_NM_LAST_ORG_NM	Utilization Management Organization (UMO) Name Last or Organization Name
NM109	Utilization Management Organization (UMO) Name	L2010A_X3_S01_NM109_ID_CD_OVL_EMPLYR_ID_NR	Utilization Management Organization (UMO) Identification Code - Employer’s Identification Number
Loop: 2000B	REQUESTER LEVEL		
HL01	Requester Level	L2000B_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Requester Level	L2000B_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
HL04	Requester Level	L2000B_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
Loop: 2010B	REQUESTER NAME		
NM102	Requester Name	L2010B_1P_S01_NM102_ENTY_TYPE_QUAL	Entity Type Qualifier
NM103	Requester Name	L2010B_1P_S01_NM103_REQSTR_LAST_ORG_NM	Requester Last or Organization Name
NM104	Requester Name	L2010B_1P_S01_NM104_REQSTR_FNAME	Requester First Name
NM109	Requester Name	L2010B_1P_S01_NM109_REQSTR_ID_OVL_EMPLYR_ID_NR	Requester Identifier - Employer’s Identification Number
Loop: 2000C	SUBSCRIBER LEVEL		
HL01	Subscriber Level	L2000C_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Subscriber Level	L2000C_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number

HL04	Subscriber Level	L2000C_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
Loop: 2010C	SUBSCRIBER NAME		
NM103	Subscriber Name	L2010C_S01_NM103_SUB_LNAME	Subscriber Last Name
NM104	Subscriber Name	L2010C_S01_NM104_SUB_FNAME	Subscriber First Name
NM109	Subscriber Name	L2010C_S01_NM109_PRIMRY_ID_OVL_MEM_ID_NR	Subscriber Primary Identifier - Member Identification Number
DMG02	Subscriber Demographic Information	L2010C_S06_DMG02_BIRTH_DT_OVL_DT_CCYYMMDD	Subscriber Birth Date - Date Expressed in Format CCYYMMDD
DMG03	Subscriber Demographic Information	L2010C_S06_DMG03_GENDR_CD	Subscriber Gender Code
Loop: 2000E	PATIENT EVENT LEVEL		
HL01	Patient Event Level	L2000E_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Patient Event Level	L2000E_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
HL04	Patient Event Level	L2000E_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
UM01	Health Care Services Review Information	L2000E_S04_UM01_REQST_CATGRY_CD	Request Category Code
UM02	Health Care Services Review Information	L2000E_S04_UM02_CERT_TYPE_CD	Certification Type Code
UM03	Health Care Services Review Information	L2000E_S04_UM03_TYPE_CD	Service Type Code
UM04-01	Health Care Services Review Information	L2000E_S04_UM04_01_FACLTY_TYPE_CD	Facility Type Code
UM04-02	Health Care Services Review Information	L2000E_S04_UM04_02_FACLTY_CD_QUAL	Facility Code Qualifier
HCR01	Health Care Services Review	L2000E_S05_HCR01_ACTN_CD	Action Code
HCR02	Health Care Services Review	L2000E_S05_HCR02_REVW_ID_NR	Review Identification Number

Specification: 5010_820

Reference	Segment	CGIF	Mapping
Loop: ISA	GROUP HEADERS		
ISA02	Interchange Control Header	S0_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	Authorization Information - No Authorization Information Present
ISA04	Interchange Control Header	OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO	Security Information - No Security Information Present
ISA06	Interchange Control Header	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID	Interchange Sender ID - Mutually Defined
ISA08	Interchange Control Header	OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID	Interchange Receiver ID - Mutually Defined
ISA09	Interchange Control Header	OEISA_S01_ISA09_DT	Interchange Date
ISA10	Interchange Control Header	OEISA_S01_ISA10_TM	Interchange Time
ISA11	Interchange Control Header	OEISA_S01_ISA11_REPTN_SEPRTR	Repetition Separator
ISA12	Interchange Control Header	OEISA_S01_ISA12_VERSN_NR	Interchang Control Version Number
ISA13	Interchange Control Header	OEISA_S01_ISA13_ICN	Interchange Control Number
ISA14	Interchange Control Header	OEISA_S01_ISA14_ACK_REQ	Acknowledgment Requested
ISA15	Interchange Control Header	OEISA_S01_ISA15_USG_IND	Interchange Usage Indicator
ISA16	Interchange Control Header	OEISA_S01_ISA16_SUBELE_SEP	Component Element Separator
IEA01	Interchange Control Trailer	OEISA_S03_IEA01_GS_CT	Number of Included Functional Groups
IEA02	Interchange Control Trailer	OEISA_S03_IEA02_ICN	Interchange Control Number
Loop: GSHDR	GROUP HEADER		
GS02	Functional Group Header	GSHDR_S01_GS02_APP_SENDR_CD	Application Senders Code
GS03	Functional Group Header	GSHDR_S01_GS03_APP_RECVR_CD	Application Receivers Code
GS04	Functional Group Header	GSHDR_S01_GS04_DT	Date
GS05	Functional Group Header	GSHDR_S01_GS05_TM	Time
GS06	Functional Group Header	GSHDR_S01_GS06_GCN	Group Control Number
GE01	Functional Group Trailer	GSHDR_S03_GE01_TS_CT	Number of Transaction Sets Included
GE02	Functional Group Trailer	GSHDR_S03_GE02_GCN	Group Control Number
Loop: STHDR	TRANSACTION SET HEADER		
ST02	820 Header	STHDR_S01_ST02_TS_CONTRL_NR	Transaction Set Control Number
ST03	820 Header	STHDR_S01_ST03_IMPLMNTN_CONVNTN_REFNC	Implementation Convention Reference
BPR02	Financial Information	STHDR_S02_BPR02_TOTL_PREM_PMT_AMT_OVL_REMTNC_NFO_ONLY	Total Premium Payment Amount - Remittance Information Only
BPR04	Financial Information	STHDR_S02_BPR04_PMT_METHD_CD	Payment Method Code
BPR10	Financial Information	STHDR_S02_BPR10_PAYR_ID	Payer Identifier
BPR16	Financial Information	STHDR_S02_BPR16_CHK_IS_EFT_EFF_DT	Check Issue or EFT Effective Date
TRN01	Reassociation Trace Number	STHDR_S03_TRN01_TYPE_CD	Trace Type Code
TRN02	Reassociation Trace Number	STHDR_S03_TRN02_CHK_EFT_TRAC_NR	Check or EFT Trace Number
SE01	Transaction Set Trailer	STHDR_S15_SE01_SEG_CT	Transaction Segment Count

SE02	Transaction Set Trailer	STHDR_S15_SE02_TCN	Transaction Set Control Number
Loop: 1000A	PREMIUM RECEIVERS NAME		
N102	Premium Receiver's Name	L1000A_S01_N102_LAST_ORG_NM	Premium Receiver's Last or Organization Name
N104	Premium Receiver's Name	L1000A_S01_N104_PREM_RECVR_ID_CD_OVL_FED_TAX_ID_NR	Premium Receiver's Identification Code - Federal Taxpayer's Identification Number
Loop: 1000B	PREMIUM PAYERS NAME		
N102	Premium Payer's Name	L1000B_S01_N102_PAYR_NM	Premium Payer Name
N104	Premium Payer's Name	L1000B_S01_N104_PAYR_ID_OVL_FED_TAX_ID_NR	Premium Payer Identifier - Federal Taxpayer's Identification Number
Loop: 2000B	INDIVIDUAL REMITTANCE		
ENT01	Individual Remittance	L2000B_S01_ENT01_ASGND_NR	Assigned Number
ENT04	Individual Remittance	L2000B_S01_ENT04_RECVR_INDVDL_ID_OVL_EMPLY_ID_NR	Receiver's Individual Identifier - Employee Identification Number
Loop: 2100B	INDIVIDUAL NAME		
NM103	Individual Name	L2100B_IL_S01_NM103_INDVDL_LNAME	Individual Last Name
NM104	Individual Name	L2100B_IL_S01_NM104_INDVDL_FNAME	Individual First Name
NM109	Individual Name	L2100B_IL_S01_NM109_INDVDL_ID_OVL_INSRD_UNQ_ID_NR	Individual Identifier - Insured's Unique Identification Number
Loop: 2300B	INDIVIDUAL PREMIUM REMITTANCE DETAIL		
RMR02	Individual Premium Remittance Detail	L2300B_S01_RMR02_INS_REMTNC_REF_NR_OVL_HEALTH_INS_POLCY_NR	Insurance Remittance Reference Number - Health Insurance Policy Number
RMR04	Individual Premium Remittance Detail	L2300B_S01_RMR04_DETL_PREM_PMT_AMT	Detail Premium Payment Amount

Specification: 5010_834

Reference	Segment	CGIF	Mapping
Loop: ISA	GROUP HEADERS		
ISA02	Interchange Control Header	T1_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	Authorization Information - No Authorization Information Present
ISA04	Interchange Control Header	OEISA_S01_ISA04_SEC_NFO_OVL_PASSWD	Security Information - Password
ISA06	Interchange Control Header	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID	Interchange Sender ID - Mutually Defined
ISA08	Interchange Control Header	OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID	Interchange Receiver ID - Mutually Defined
ISA09	Interchange Control Header	OEISA_S01_ISA09_DT	Interchange Date
ISA10	Interchange Control Header	OEISA_S01_ISA10_TM	Interchange Time
ISA11	Interchange Control Header	OEISA_S01_ISA11_REPTN_SEPRTR	Repetition Separator
ISA12	Interchange Control Header	OEISA_S01_ISA12_VERSN_NR	Interchang Control Version Number
ISA13	Interchange Control Header	OEISA_S01_ISA13_ICN	Interchange Control Number
ISA14	Interchange Control Header	OEISA_S01_ISA14_ACK_REQ	Acknowledgment Requested
ISA15	Interchange Control Header	OEISA_S01_ISA15_USG_IND	Interchange Usage Indicator
ISA16	Interchange Control Header	OEISA_S01_ISA16_SUBELE_SEP	Component Element Separator
IEA01	Interchange Control Trailer	OEISA_S03_IEA01_GS_CT	Number of Included Functional Groups

IEA02	Interchange Control Trailer	OEISA_S03_IEA02_ICN	Interchange Control Number
Loop: GSHDR	GROUP HEADER		
GS02	Functional Group Header	GSHDR_S01_GS02_APP_SENDR_CD	Application Senders Code
GS03	Functional Group Header	GSHDR_S01_GS03_APP_RECVR_CD	Application Receivers Code
GS04	Functional Group Header	GSHDR_S01_GS04_DT	Date
GS05	Functional Group Header	GSHDR_S01_GS05_TM	Time
GS06	Functional Group Header	GSHDR_S01_GS06_GCN	Group Control Number
GS08	Functional Group Header	GSHDR_S01_GS08_VERS_ID_CD	Version Release Industry Identifier Code
GE01	Functional Group Trailer	GSHDR_S03_GE01_TS_CT	Number of Transaction Sets Included
GE02	Functional Group Trailer	GSHDR_S03_GE02_GCN	Group Control Number
Loop: STHDR	TRANSACTION SET HEADER		
ST02	Transaction Set Header	STHDR_S01_ST02_CONTRL_NR	Transaction Set Control Number
ST03	Transaction Set Header	STHDR_S01_ST03_IMPLMNTN_CONVNTN_REFNC	Implementation Convention Reference
BGN01	Beginning Segment	STHDR_S02_BGN01_TS_PURPS_CD	Transaction Set Purpose Code
BGN02	Beginning Segment	STHDR_S02_BGN02_TS_REF_NR	Transaction Set Reference Number
BGN03	Beginning Segment	STHDR_S02_BGN03_TS_CREATN_DT	Transaction Set Creation Date
BGN04	Beginning Segment	STHDR_S02_BGN04_TS_CREATN_TIM	Transaction Set Creation Time
BGN08	Beginning Segment	STHDR_S02_BGN08_ACTN_CD	Action Code
SE01	Transaction Set Trailer	STHDR_S10_SE01_SEG_CT	Transaction Segment Count
SE02	Transaction Set Trailer	STHDR_S10_SE02_TCN	Transaction Set Control Number
Loop: 1000A	SPONSOR NAME		
N102	Sponsor Name	L1000A_S01_N102_PLAN_SPONSR_NM	Plan Sponsor Name
N104	Sponsor Name	L1000A_S01_N104_SPONSR_ID_OVL_FED_TAX_ID_NR	Sponsor Identifier - Federal Taxpayer’s Identification Number
Loop: 1000B	PAYER		
N102	Payer	L1000B_S01_N102_INSR_NM	Insurer Name
N104	Payer	L1000B_S01_N104_INSR_ID_CD_OVL_FED_TAX_ID_NR	Insurer Identification Code - Federal Taxpayer’s Identification Number
Loop: 2000	MEMBER LEVEL DETAIL		
INS01	Member Level Detail	L2000_S01_INS01_MEM_IND	Member Indicator
INS02	Member Level Detail	L2000_S01_INS02_INDVDL_REL_CD	Individual Relationship Code
INS03	Member Level Detail	L2000_S01_INS03_MAINTNC_TYPE_CD	Maintenance Type Code
INS05	Member Level Detail	L2000_S01_INS05_BENFT_STATS_CD	Benefit Status Code
INS08	Member Level Detail	L2000_S01_INS08_EMPLYMNT_STATS_CD	Employment Status Code
REF02	Subscriber Identifier	L2000_S02_REF02_SUB_ID_OVL_SUB_NR	Subscriber Identifier - Subscriber Number
REF02	Member Policy Number	L2000_S03_REF02_MEM_GRP_POLCY_NR_OVL_GRP_POLCY_NR	Member Group or Policy Number - Group or Policy Number
REF02	Member Supplemental Identifier	L2000_S04_REF02_SUPP_ID_OVL_CLIENT_REPRTNG_CATGRY	Member Supplemental Identifier - Client Reporting Category
DTP03	Member Level Dates	L2000_S05_DTP03_STATS_NFO_EFF_DT_OVL_ELIG_BEGN_D8	Status Information Effective Date - Eligibility Begin (D8)

Loop: 2100A	MEMBER NAME		
NM103	Member Name	L2100A_IL_S01_NM103_MEM_LNAME_OVL_PERSN	Member Last Name - Person
NM104	Member Name	L2100A_IL_S01_NM104_MEM_FNAME	Member First Name
PER04	Member Communications Numbers	L2100A_IL_S02_PER04_COMM_NR_OVL_HOM_PHON_NR	Communication Number - Home Phone Number
N301	Member Residence Street Address	L2100A_IL_S03_N301_ADRS_LIN	Member Address Line
N401	Member City, State, ZIP Code	L2100A_IL_S04_N401_CITY_NM	Member City Name
N402	Member City, State, ZIP Code	L2100A_IL_S04_N402_STAT_CD	Member State Code
N403	Member City, State, ZIP Code	L2100A_IL_S04_N403_MEM_POSTL_ZON_ZIP_CD	Member Postal Zone or Zip Code
DMG02	Member Demographics	L2100A_IL_S05_DMG02_BIRTH_DT_OVL_DT_CCYYMMDD	Member Birth Date - Date Expressed in Format CCYYMMDD
DMG03	Member Demographics	L2100A_IL_S05_DMG03_GENDR_CD	Gender Code

Specification: 5010_835

Reference	Segment	CGIF	Mapping
Loop: ISA	GROUP HEADERS		
ISA02	Interchange Control Header	U1_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	Authorization Information - No Authorization Information Present
ISA04	Interchange Control Header	OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO	Security Information - No Security Information Present
ISA06	Interchange Control Header	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID	Interchange Sender ID - Mutually Defined
ISA08	Interchange Control Header	OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID	Interchange Receiver ID - Mutually Defined
ISA09	Interchange Control Header	OEISA_S01_ISA09_DT	Interchange Date
ISA10	Interchange Control Header	OEISA_S01_ISA10_TM	Interchange Time
ISA11	Interchange Control Header	OEISA_S01_ISA11_REPTN_SEPRTR	Repetition Separator
ISA12	Interchange Control Header	OEISA_S01_ISA12_VERSN_NR	Interchang Control Version Number
ISA13	Interchange Control Header	OEISA_S01_ISA13_ICN	Interchange Control Number
ISA14	Interchange Control Header	OEISA_S01_ISA14_ACK_REQ	Acknowledgment Requested
ISA15	Interchange Control Header	OEISA_S01_ISA15_USG_IND	Interchange Usage Indicator
ISA16	Interchange Control Header	OEISA_S01_ISA16_SUBELE_SEP	Component Element Separator
IEA01	Interchange Control Trailer	OEISA_S03_IEA01_GS_CT	Number of Included Functional Groups
IEA02	Interchange Control Trailer	OEISA_S03_IEA02_ICN	Interchange Control Number
Loop: GSHDR	GROUP HEADER		
GS02	Functional Group Header	GSHDR_S01_GS02_APP_SENDR_CD	Application Senders Code
GS03	Functional Group Header	GSHDR_S01_GS03_APP_RECVR_CD	Application Receivers Code
GS04	Functional Group Header	GSHDR_S01_GS04_DT	Date
GS05	Functional Group Header	GSHDR_S01_GS05_TM	Time
GS06	Functional Group Header	GSHDR_S01_GS06_GCN	Group Control Number

GS08	Functional Group Header	GSHDR_S01_GS08_VERS_ID_CD	Version Release Industry Identifier Code
GE01	Functional Group Trailer	GSHDR_S03_GE01_TS_CT	Number of Transaction Sets Included
GE02	Functional Group Trailer	GSHDR_S03_GE02_GCN	Group Control Number
Loop: STHDR	TRANSACTION SET HEADER		
ST02	Transaction Set Header	STHDR_S01_ST02_CONTRL_NR	Transaction Set Control Number
BPR02	Financial Information	STHDR_S02_BPR02_TOTL_ACTL_PROV_PMT_AMT_OVL_REMTNC_NFO_ONLY	Total Actual Provider Payment Amount - Remittance Information Only
BPR03	Financial Information	STHDR_S02_BPR03_CREDT_DEBT_FLAG_CD	Credit or Debit Flag Code
BPR04	Financial Information	STHDR_S02_BPR04_PMT_METHD_CD	Payment Method Code
BPR16	Financial Information	STHDR_S02_BPR16_CHK_IS_EFT_EFF_DT	Check Issue or EFT Effective Date
TRN02	Reassociation Trace Number	STHDR_S03_TRN02_CHK_EFT_TRAC_NR	Check or EFT Trace Number
TRN03	Reassociation Trace Number	STHDR_S03_TRN03_PAYR_ID	Payer Identifier
REF02	Receiver Identification	STHDR_S05_REF02_RECVR_ID_OVL_RECVR_ID_NR	Receiver Identifier - Receiver Identification Number
DTM02	Production Date	STHDR_S07_DTM02_PRODCTN_DT_OVL_PRODCTN	Production Date - Production
SE01	Transaction Set Trailer	STHDR_S12_SE01_SEG_CT	Transaction Segment Count
SE02	Transaction Set Trailer	STHDR_S12_SE02_TCN	Transaction Set Control Number
Loop: 1000A	PAYER IDENTIFICATION		
N102	Payer Identification	L1000A_S01_N102_PAYR_NM	Payer Name
N104	Payer Identification	L1000A_S01_N104_PAYR_ID_OVL_MDCR_MDCD_SVCS_PLAND	Payer Identifier - Centers for Medicare and Medicaid Services PlanID
N301	Payer Address	L1000A_S02_N301_ADRS_LIN	Payer Address Line
N401	Payer City, State, ZIP Code	L1000A_S03_N401_CITY_NM	Payer City Name
N402	Payer City, State, ZIP Code	L1000A_S03_N402_STAT_CD	Payer State Code
N403	Payer City, State, ZIP Code	L1000A_S03_N403_PAYR_POSTL_ZON_ZIP_CD	Payer Postal Zone or ZIP Code
PER02	Payer Technical Contact Information	L1000A_S06_PER02_PAYR_TECHNCL_CONTCT_NM	Payer Technical Contact Name
PER04	Payer Technical Contact Information	L1000A_S06_PER04_PAYR_CONTCT_COMNCTN_NR_OVL_TELPHN	Payer Contact Communication Number - Telephone
Loop: 1000B	PAYEE IDENTIFICATION		
N102	Payee Identification	L1000B_S01_N102_PAY_NM	Payee Name
N104	Payee Identification	L1000B_S01_N104_ID_CD_OVL_FED_TAX_ID_NR	Payee Identification Code - Federal Taxpayer’s Identification Number
N301	Payee Address	L1000B_S02_N301_ADRS_LIN	Payee Address Line
N401	Payee City, State, ZIP Code	L1000B_S03_N401_CITY_NM	Payee City Name
N402	Payee City, State, ZIP Code	L1000B_S03_N402_STAT_CD	Payee State Code
N403	Payee City, State, ZIP Code	L1000B_S03_N403_PAY_POSTL_ZON_ZIP_CD	Payee Postal Zone or ZIP Code
Loop: 2000	HEADER NUMBER		
LX01	Header Number	L2000_S01_LX01_ASGND_NR	Assigned Number

Loop: 2100	CLAIM PAYMENT INFORMATION		
CLP01	Claim Payment Information	L2100_S01_CLP01_PATNT_CONTRL_NR	Patient Control Number
CLP02	Claim Payment Information	L2100_S01_CLP02_STATS_CD	Claim Status Code
CLP03	Claim Payment Information	L2100_S01_CLP03_TOTL_CLM_CHG_AMT	Total Claim Charge Amount
CLP04	Claim Payment Information	L2100_S01_CLP04_PMT_AMT	Claim Payment Amount
CLP06	Claim Payment Information	L2100_S01_CLP06_CLM_FILNG_IND_CD	Claim Filing Indicator Code
CLP07	Claim Payment Information	L2100_S01_CLP07_PAYR_CLM_CONTRL_NR	Payer Claim Control Number
NM103	Patient Name	L2100_S03_NM103_PATNT_LNAME	Patient Last Name
NM104	Patient Name	L2100_S03_NM104_PATNT_FNAME	Patient First Name
NM109	Patient Name	L2100_S03_NM109_PATNT_ID_OVL_MEM_ID_NR	Patient Identifier - Member Identification Number
DTM02	Statement From or To Date	L2100_S14_DTM02_CLM_DT_OVL_CLM_STATMNT_PERD_START	Claim Date - Claim Statement Period Start
DTM02	Statement From or To Date	L2100_S14_DTM02_I02_CLM_DT_OVL_CLM_STATMNT_PERD_END	Claim Date - Claim Statement Period End
AMT02	Claim Supplemental Information	L2100_S18_AMT02_CLM_SUPP_NFO_AMT_OVL_COVG_AMT	Claim Supplemental Information Amount - Coverage Amount
AMT01-02	Claim Supplemental Information	L2110_S01_SVC01_02_ADJDCTD_PROC_CD_OVL_HCPCS_CD	Adjudicated Procedure Code - Health Care Financing Administration Common Procedural Coding System (HCPCS) Codes
Loop: 2110	SERVICE PAYMENT INFORMATION		
SVC02	Service Payment Information	L2110_S01_SVC02_LIN_ITM_CHG_AMT	Line Item Charge Amount
SVC03	Service Payment Information	L2110_S01_SVC03_LIN_ITM_PROV_PMT_AMT	Line Item Provider Payment Amount
SVC05	Service Payment Information	L2110_S01_SVC05_UNTS_SVC_PD_CT	Units of Service Paid Count
DTM02	Service Date	L2110_S02_DTM02_SVC_DT_OVL_SVC	Service Date - Service
CAS01	Service Adjustment	L2110_S03_CAS01_CLM_ADJ_GRP_CD	Claim Adjustment Group Code
CAS02	Service Adjustment	L2110_S03_CAS02_ADJ_RSN_CD	Adjustment Reason Code
CAS03	Service Adjustment	L2110_S03_CAS03_ADJ_AMT	Adjustment Amount
AMT02	Service Supplemental Amount	L2110_S08_AMT02_SVC_SUPP_AMT_OVL_ALWD_ACTL	Service Supplemental Amount - Allowed - Actual
CAS01	Service Adjustment	L2110_S03_CAS01_I02_CLM_ADJ_GRP_CD	Claim Adjustment Group Code
CAS02	Service Adjustment	L2110_S03_CAS02_I02_ADJ_RSN_CD	Adjustment Reason Code
CAS03	Service Adjustment	L2110_S03_CAS03_I02_ADJ_AMT	Adjustment Amount

Specification: 5010_837D

Reference	Segment	CGIF	Mapping
Loop: ISA	GROUP HEADERS		
ISA02	Interchange Control Header	V2_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	Authorization Information - No Authorization Information Present
ISA04	Interchange Control Header	OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO	Security Information - No Security Information Present
ISA06	Interchange Control Header	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID	Interchange Sender ID - Mutually Defined
ISA08	Interchange Control Header	OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID	Interchange Receiver ID - Mutually Defined

ISA09	Interchange Control Header	OEISA_S01_ISA09_DT	Interchange Date
ISA10	Interchange Control Header	OEISA_S01_ISA10_TM	Interchange Time
ISA11	Interchange Control Header	OEISA_S01_ISA11_REPTN_SEPRTR	Repetition Separator
ISA12	Interchange Control Header	OEISA_S01_ISA12_VERSN_NR	Interchang Control Version Number
ISA13	Interchange Control Header	OEISA_S01_ISA13_ICN	Interchange Control Number
ISA14	Interchange Control Header	OEISA_S01_ISA14_ACK_REQ	Acknowledgment Requested
ISA15	Interchange Control Header	OEISA_S01_ISA15_USG_IND	Interchange Usage Indicator
ISA16	Interchange Control Header	OEISA_S01_ISA16_SUBELE_SEP	Component Element Separator
IEA01	Interchange Control Trailer	OEISA_S03_IEA01_GS_CT	Number of Included Functional Groups
IEA02	Interchange Control Trailer	OEISA_S03_IEA02_ICN	Interchange Control Number
Loop: GSHDR	GROUP HEADER		
GS02	Functional Group Header	GSHDR_S01_GS02_APP_SENDR_CD	Application Senders Code
GS03	Functional Group Header	GSHDR_S01_GS03_APP_RECVR_CD	Application Receivers Code
GS04	Functional Group Header	GSHDR_S01_GS04_DT	Date
GS05	Functional Group Header	GSHDR_S01_GS05_TM	Time
GS06	Functional Group Header	GSHDR_S01_GS06_GCN	Group Control Number
GS08	Functional Group Header	GSHDR_S01_GS08_VERS_ID_CD	Version Release Industry Identifier Code
GE01	Functional Group Trailer	GSHDR_S03_GE01_TS_CT	Number of Transaction Sets Included
GE02	Functional Group Trailer	GSHDR_S03_GE02_GCN	Group Control Number
Loop: STHDR	TRANSACTION SET HEADER		
ST02	Transaction Set Header	STHDR_S01_ST02_CONTRL_NR	Transaction Set Control Number
BHT01	Beginning of Hierarchical Transaction	STHDR_S02_BHT01_STRUCTR_CD	Hierarchical Structure Code
BHT02	Beginning of Hierarchical Transaction	STHDR_S02_BHT02_TS_PURPS_CD	Transaction Set Purpose Code
BHT03	Beginning of Hierarchical Transaction	STHDR_S02_BHT03_ORGNTR_APLCTN_TRANSCTN_ID	Originator Application Transaction Identifier
BHT04	Beginning of Hierarchical Transaction	STHDR_S02_BHT04_TS_CREATN_DT	Transaction Set Creation Date
BHT05	Beginning of Hierarchical Transaction	STHDR_S02_BHT05_TS_CREATN_TIM	Transaction Set Creation Time
BHT06	Beginning of Hierarchical Transaction	STHDR_S02_BHT06_CLM_ENCNTR_ID	Claim or Encounter Identifier
SE01	Transaction Set Trailer	STHDR_S06_SE01_SEG_CT	Transaction Segment Count
SE02	Transaction Set Trailer	STHDR_S06_SE02_TCN	Transaction Set Control Number
Loop: 1000A	SUBMITTER NAME		

NM103	Submitter Name	L1000A_S01_NM103_SUBMTR_LAST_ORG_NM_OVL_NONPRSN_ENTY	Submitter Last or Organization Name - Non-Person Entity
NM109	Submitter Name	L1000A_S01_NM109_SUBMTR_ID_OVL_ETN_NR	Submitter Identifier - Electronic Transmitter Identification Number (ETIN)
PER02	Submitter EDI Contact Information	L1000A_S02_PER02_CONTCT_NM	Submitter Contact Name
PER04	Submitter EDI Contact Information	L1000A_S02_PER04_COMM_NR_OVL_TELPHN	Communication Number - Telephone
Loop: 1000B	RECEIVER NAME		
NM103	Receiver Name	L1000B_S01_NM103_RECVR_NM_OVL_NONPRSN_ENTY	Receiver Name - Non-Person Entity
NM109	Receiver Name	L1000B_S01_NM109_PRIMRY_ID_OVL_ETN_NR	Receiver Primary Identifier - Electronic Transmitter Identification Number (ETIN)
Loop: 2000A	BILLING PROVIDER HIERARCHICAL LEVEL		
HL01	Billing Provider Hierarchical Level	L2000A_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
Loop: 2010AA	BILLING PROVIDER NAME		
NM103	Billing Provider Name	L2010AA_S01_NM103_LAST_ORGL_NM_OVL_NONPRSN_ENTY	Billing Provider Last or Organizational Name - Non-Person Entity
N301	Billing Provider Address	L2010AA_S02_N301_BILNG_PROV_ADRS_LIN	Billing Provider Address Line
N401	Billing Provider City, State, ZIP Code	L2010AA_S03_N401_BILNG_PROV_CITY_NM	Billing Provider City Name
N402	Billing Provider City, State, ZIP Code	L2010AA_S03_N402_STAT_PROVNC_CD	Billing Provider State or Province Code
N403	Billing Provider City, State, ZIP Code	L2010AA_S03_N403_POSTL_ZON_ZIP_CD	Billing Provider Postal Zone or ZIP Code
REF02	Billing Provider Tax Identification	L2010AA_S04_REF02_ID_NR_OVL_EMPLYR_ID_NR	Billing Provider Tax Identification Number - Employer's Identification Number
Loop: 2000B	SUBSCRIBER HIERARCHICAL LEVEL		
HL01	Subscriber Hierarchical Level	L2000B_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Subscriber Hierarchical Level	L2000B_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
HL04	Subscriber Hierarchical Level	L2000B_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
SBR01	Subscriber Information	L2000B_S02_SBR01_PAYR_RESP_SEQNC_NR_CD	Payer Responsibility Sequence Number Code
SBR02	Subscriber Information	L2000B_S02_SBR02_INDVDL_REL_CD	Individual Relationship Code
SBR09	Subscriber Information	L2000B_S02_SBR09_CLM_FILNG_IND_CD	Claim Filing Indicator Code
Loop: 2010BA	SUBSCRIBER NAME		
NM103	Subscriber Name	L2010BA_S01_NM103_SUB_LNAME_OVL_PERSN	Subscriber Last Name - Person
NM104	Subscriber Name	L2010BA_S01_NM104_SUB_FNAME	Subscriber First Name
NM109	Subscriber Name	L2010BA_S01_NM109_PRIMRY_ID_OVL_MEM_ID_NR	Subscriber Primary Identifier - Member Identification Number
N301	Subscriber Address	L2010BA_S02_N301_ADRS_LIN	Subscriber Address Line

N401	Subscriber City, State, ZIP Code	L2010BA_S03_N401_CITY_NM	Subscriber City Name
N402	Subscriber City, State, ZIP Code	L2010BA_S03_N402_STAT_CD	Subscriber State Code
N403	Subscriber City, State, ZIP Code	L2010BA_S03_N403_SUB_POSTL_ZON_ZIP_CD	Subscriber Postal Zone or ZIP Code
DMG02	Subscriber Demographic Information	L2010BA_S04_DMG02_BIRTH_DT_OVL_DT_CCYYMMDD	Subscriber Birth Date - Date Expressed in Format CCYYMMDD
DMG03	Subscriber Demographic Information	L2010BA_S04_DMG03_GENDR_CD	Subscriber Gender Code
Loop: 2010BB	PAYER NAME		
NM103	Payer Name	L2010BB_S01_NM103_PAYR_NM_OVL_NONPRSN_ENTY	Payer Name - Non-Person Entity
NM109	Payer Name	L2010BB_S01_NM109_PAYR_ID_OVL_PAYR_ID	Payer Identifier - Payor Identification
Loop: 2300	CLAIM INFORMATION		
CLM01	Claim Information	L2300_S01_CLM01_PATNT_CONTRL_NR	Patient Control Number
CLM02	Claim Information	L2300_S01_CLM02_TOTL_CLM_CHG_AMT	Total Claim Charge Amount
CLM05-01	Claim Information	L2300_S01_CLM05_01_PLAC_SVC_CD	Place of Service Code
CLM05-03	Claim Information	L2300_S01_CLM05_03_CLM_FREQNCY_CD_OVL_PLAC_SVC_CODS_PROF_DENTL	Claim Frequency Code - Place of Service Codes for Professional or Dental
CLM06	Claim Information	L2300_S01_CLM06_PROV_SUPLR_SIGNTR_IND	Provider or Supplier Signature Indicator
CLM07	Claim Information	L2300_S01_CLM07_ASGNMNT_PLAN_PARTCPTN_CD	Assignment or Plan Participation Code
CLM08	Claim Information	L2300_S01_CLM08_BENFTS_ASGNMNT_CERT_IND	Benefits Assignment Certification Indicator
CLM09	Claim Information	L2300_S01_CLM09_RELS_NFO_CD	Release of Information Code
DTP03	Date - Service Date	L2300_S04_DTP03_SVC_DT_OVL_SVC_RD8	Service Date - Service (RD8)
Loop: 2400	SERVICE LINE NUMBER		
LX01	Service Line Number	L2400_S01_LX01_ASGND_NR	Assigned Number
LX01-02	Service Line Number	L2400_S02_SV301_02_PROC_CD_OVL_AMRCN_DENTL_ASCTN_CD	Procedure Code - American Dental Association Codes
SV302	Dental Service	L2400_S02_SV302_LIN_ITM_CHG_AMT	Line Item Charge Amount

Specification: 5010_837I

Reference	Segment	CGIF	Mapping
Loop: ISA	GROUP HEADERS		
ISA02	Interchange Control Header	W2_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	Authorization Information - No Authorization Information Present
ISA04	Interchange Control Header	OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO	Security Information - No Security Information Present
ISA06	Interchange Control Header	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID	Interchange Sender ID - Mutually Defined
ISA08	Interchange Control Header	OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID	Interchange Receiver ID - Mutually Defined
ISA09	Interchange Control Header	OEISA_S01_ISA09_DT	Interchange Date
ISA10	Interchange Control Header	OEISA_S01_ISA10_TM	Interchange Time
ISA11	Interchange Control Header	OEISA_S01_ISA11_REPTN_SEPRTR	Repetition Separator
ISA12	Interchange Control Header	OEISA_S01_ISA12_VERSN_NR	Interchang Control Version Number
ISA13	Interchange Control Header	OEISA_S01_ISA13_ICN	Interchange Control Number

ISA14	Interchange Control Header	OEISA_S01_ISA14_ACK_REQ	Acknowledgment Requested
ISA15	Interchange Control Header	OEISA_S01_ISA15_USG_IND	Interchange Usage Indicator
ISA16	Interchange Control Header	OEISA_S01_ISA16_SUBELE_SEP	Component Element Separator
IEA01	Interchange Control Trailer	OEISA_S03_IEA01_GS_CT	Number of Included Functional Groups
IEA02	Interchange Control Trailer	OEISA_S03_IEA02_ICN	Interchange Control Number
Loop: GSHDR	GROUP HEADER		
GS02	Functional Group Header	GSHDR_S01_GS02_APP_SENDR_CD	Application Senders Code
GS03	Functional Group Header	GSHDR_S01_GS03_APP_RECVR_CD	Application Receivers Code
GS04	Functional Group Header	GSHDR_S01_GS04_DT	Date
GS05	Functional Group Header	GSHDR_S01_GS05_TM	Time
GS06	Functional Group Header	GSHDR_S01_GS06_GCN	Group Control Number
GS08	Functional Group Header	GSHDR_S01_GS08_VERS_ID_CD	Version Release Industry Identifier Code
GE01	Functional Group Trailer	GSHDR_S03_GE01_TS_CT	Number of Transaction Sets Included
GE02	Functional Group Trailer	GSHDR_S03_GE02_GCN	Group Control Number
Loop: STHDR	TRANSACTION SET HEADER		
ST02	Transaction Set Header	STHDR_S01_ST02_CONTRL_NR	Transaction Set Control Number
BHT01	Beginning of Hierarchical Transaction	STHDR_S02_BHT01_STRUCTR_CD	Hierarchical Structure Code
BHT02	Beginning of Hierarchical Transaction	STHDR_S02_BHT02_TS_PURPS_CD	Transaction Set Purpose Code
BHT03	Beginning of Hierarchical Transaction	STHDR_S02_BHT03_ORGNTR_APLCTN_TRANSCTN_ID	Originator Application Transaction Identifier
BHT04	Beginning of Hierarchical Transaction	STHDR_S02_BHT04_TS_CREATN_DT	Transaction Set Creation Date
BHT05	Beginning of Hierarchical Transaction	STHDR_S02_BHT05_TS_CREATN_TIM	Transaction Set Creation Time
BHT06	Beginning of Hierarchical Transaction	STHDR_S02_BHT06_CLM_ID	Claim Identifier
SE01	Transaction Set Trailer	STHDR_S06_SE01_SEG_CT	Transaction Segment Count
SE02	Transaction Set Trailer	STHDR_S06_SE02_TCN	Transaction Set Control Number
Loop: 1000A	SUBMITTER NAME		
NM103	Submitter Name	L1000A_S01_NM103_SUBMTR_LAST_ORG_NM_OVL_NONPRSN_ENTY	Submitter Last or Organization Name - Non-Person Entity
NM109	Submitter Name	L1000A_S01_NM109_SUBMTR_ID_OVL_ETN_NR	Submitter Identifier - Electronic Transmitter Identification Number (ETIN)
PER02	Submitter EDI Contact Information	L1000A_S02_PER02_CONTCT_NM	Submitter Contact Name
PER04	Submitter EDI Contact Information	L1000A_S02_PER04_COMM_NR_OVL_TELPHN	Communication Number - Telephone

Loop: 1000B	RECEIVER NAME		
NM103	Receiver Name	L1000B_S01_NM103_RECVR_NM_OVL_NONPRSN_ENTY	Receiver Name - Non-Person Entity
NM109	Receiver Name	L1000B_S01_NM109_PRIMRY_ID_OVL_ETN_NR	Receiver Primary Identifier - Electronic Transmitter Identification Number (ETIN)
Loop: 2000A	BILLING PROVIDER HIERARCHICAL LEVEL		
HL01	Billing Provider Hierarchical Level	L2000A_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
Loop: 2010AA	BILLING PROVIDER NAME		
NM103	Billing Provider Name	L2010AA_S01_NM103_BILNG_PROV_ORGL_NM_OVL_NONPRSN_ENTY	Billing Provider Organizational Name - Non-Person Entity
N301	Billing Provider Address	L2010AA_S02_N301_BILNG_PROV_ADRS_LIN	Billing Provider Address Line
N401	Billing Provider City, State, ZIP Code	L2010AA_S03_N401_BILNG_PROV_CITY_NM	Billing Provider City Name
N402	Billing Provider City, State, ZIP Code	L2010AA_S03_N402_STAT_PROVNC_CD	Billing Provider State or Province Code
N403	Billing Provider City, State, ZIP Code	L2010AA_S03_N403_POSTL_ZON_ZIP_CD	Billing Provider Postal Zone or ZIP Code
REF02	Billing Provider Tax Identification	L2010AA_S04_REF02_ID_NR_OVL_EMPLYR_ID_NR	Billing Provider Tax Identification Number - Employer’s Identification Number
Loop: 2000B	SUBSCRIBER HIERARCHICAL LEVEL		
HL01	Subscriber Hierarchical Level	L2000B_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Subscriber Hierarchical Level	L2000B_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number
HL04	Subscriber Hierarchical Level	L2000B_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
SBR01	Subscriber Information	L2000B_S02_SBR01_PAYR_RESP_SEQNC_NR_CD	Payer Responsibility Sequence Number Code
SBR02	Subscriber Information	L2000B_S02_SBR02_INDVDL_REL_CD	Individual Relationship Code
SBR09	Subscriber Information	L2000B_S02_SBR09_CLM_FILNG_IND_CD	Claim Filing Indicator Code
Loop: 2010BA	SUBSCRIBER NAME		
NM103	Subscriber Name	L2010BA_S01_NM103_SUB_LNAME_OVL_PERSN	Subscriber Last Name - Person
NM104	Subscriber Name	L2010BA_S01_NM104_SUB_FNAME	Subscriber First Name
NM109	Subscriber Name	L2010BA_S01_NM109_PRIMRY_ID_OVL_MEM_ID_NR	Subscriber Primary Identifier - Member Identification Number
N301	Subscriber Address	L2010BA_S02_N301_ADRS_LIN	Subscriber Address Line
N401	Subscriber City, State, ZIP Code	L2010BA_S03_N401_CITY_NM	Subscriber City Name
N402	Subscriber City, State, ZIP Code	L2010BA_S03_N402_STAT_CD	Subscriber State Code
N403	Subscriber City, State, ZIP Code	L2010BA_S03_N403_SUB_POSTL_ZON_ZIP_CD	Subscriber Postal Zone or ZIP Code
DMG02	Subscriber Demographic Information	L2010BA_S04_DMG02_BIRTH_DT_OVL_DT_CCYYMMDD	Subscriber Birth Date - Date Expressed in Format CCYYMMDD

DMG03	Subscriber Demographic Information	L2010BA_S04_DMG03_GENDR_CD	Subscriber Gender Code
Loop: 2010BB	PAYER NAME		
NM103	Payer Name	L2010BB_S01_NM103_PAYR_NM_OVL_NONPRSN_ENTY	Payer Name - Non-Person Entity
NM109	Payer Name	L2010BB_S01_NM109_PAYR_ID_OVL_PAYR_ID	Payer Identifier - Payor Identification
N301	Payer Address	L2010BB_S02_N301_ADRS_LIN	Payer Address Line
N401	Payer City, State, ZIP Code	L2010BB_S03_N401_CITY_NM	Payer City Name
N402	Payer City, State, ZIP Code	L2010BB_S03_N402_STAT_CD	Payer State Code
N403	Payer City, State, ZIP Code	L2010BB_S03_N403_PAYR_POSTL_ZON_ZIP_CD	Payer Postal Zone or ZIP Code
Loop: 2300	CLAIM INFORMATION		
CLM01	Claim Information	L2300_S01_CLM01_PATNT_CONTRL_NR	Patient Control Number
CLM02	Claim Information	L2300_S01_CLM02_TOTL_CLM_CHG_AMT	Total Claim Charge Amount
CLM05-01	Claim Information	L2300_S01_CLM05_01_FACLTY_TYPE_CD	Facility Type Code
CLM05-03	Claim Information	L2300_S01_CLM05_03_CLM_FREQNCY_CD_OVL_UBC_FORM_BIL_TYPE	Claim Frequency Code - Uniform Billing Claim Form Bill Type
CLM07	Claim Information	L2300_S01_CLM07_ASGNMNT_PLAN_PARTCPTN_CD	Assignment or Plan Participation Code
CLM08	Claim Information	L2300_S01_CLM08_BENFTS_ASGNMNT_CERT_IND	Benefits Assignment Certification Indicator
CLM09	Claim Information	L2300_S01_CLM09_RELS_NFO_CD	Release of Information Code
DTP03	Statement Dates	L2300_S03_DTP03_STATMNT_FROM_TO_DT_OVL_STATMNT_RD8	Statement From and To Date - Statement (RD8)
CL101	Institutional Claim Code	L2300_S06_CL101_ADMSN_TYPE_CD	Admission Type Code
CL102	Institutional Claim Code	L2300_S06_CL102_ADMSN_SOURC_CD	Admission Source Code
CL103	Institutional Claim Code	L2300_S06_CL103_PATNT_STATS_CD	Patient Status Code
CL101-02	Institutional Claim Code	L2300_S26_HI01_02_PRINCPL_DIAG_CD_OVL_ICD9CM_PRINCPL_DIAG	Principal Diagnosis Code - International Classification of Diseases Clinical Modification (ICD-9-CM) Principal Diagnosis
CL101-02	Institutional Claim Code	L2300_S31_HI01_02_OTHR_DIAG_OVL_ICD9CM_DIAG	Other Diagnosis - International Classification of Diseases Clinical Modification (ICD-9-CM) Diagnosis
Loop: 2310A	ATTENDING PROVIDER NAME		
NM103	Attending Provider Name	L2310A_S01_NM103_PROV_LNAME_OVL_PERSN	Attending Provider Last Name - Person
NM104	Attending Provider Name	L2310A_S01_NM104_PROV_FNAME	Attending Provider First Name
REF02	Attending Provider Secondary Identification	L2310A_S03_REF02_ATNDNG_PROV_2ND_ID_OVL_PROV_COMRCL_NR	Attending Provider Secondary Identifier - Provider Commercial Number
Loop: 2400	SERVICE LINE NUMBER		
LX01	Service Line Number	L2400_S01_LX01_ASGND_NR	Assigned Number
SV201	Institutional Service Line	L2400_S02_SV201_SVC_LIN_REVN_CD	Service Line Revenue Code
SV202-02	Institutional Service Line	L2400_S02_SV202_02_PROC_CD_OVL_HCPCS_CD	Procedure Code - Health Care Financing Administration Common Procedural Coding System (HCPCS) Codes
SV203	Institutional Service Line	L2400_S02_SV203_LIN_ITM_CHG_AMT	Line Item Charge Amount
SV205	Institutional Service Line	L2400_S02_SV205_UNT_CT_OVL_UNT	Service Unit Count - Unit

DTP03	Date - Service Date	L2400_S04_DTP03_SVC_DT_OVL_SVC_D8	Service Date - Service (D8)

Specification: 5010_837P

Reference	Segment	CGIF	Mapping
Loop: ISA	GROUP HEADERS		
ISA02	Interchange Control Header	X1_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	Authorization Information - No Authorization Information Present
ISA04	Interchange Control Header	OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO	Security Information - No Security Information Present
ISA06	Interchange Control Header	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY_DEFND_ID	Interchange Sender ID - Mutually Defined
ISA08	Interchange Control Header	OEISA_S01_ISA08_RECVR_ID_OVL_MUTLY_DEFND_ID	Interchange Receiver ID - Mutually Defined
ISA09	Interchange Control Header	OEISA_S01_ISA09_DT	Interchange Date
ISA10	Interchange Control Header	OEISA_S01_ISA10_TM	Interchange Time
ISA11	Interchange Control Header	OEISA_S01_ISA11_REPTN_SEPRTR	Repetition Separator
ISA12	Interchange Control Header	OEISA_S01_ISA12_VERSN_NR	Interchang Control Version Number
ISA13	Interchange Control Header	OEISA_S01_ISA13_ICN	Interchange Control Number
ISA14	Interchange Control Header	OEISA_S01_ISA14_ACK_REQ	Acknowledgment Requested
ISA15	Interchange Control Header	OEISA_S01_ISA15_USG_IND	Interchange Usage Indicator
ISA16	Interchange Control Header	OEISA_S01_ISA16_SUBELE_SEP	Component Element Separator
IEA01	Interchange Control Trailer	OEISA_S03_IEA01_GS_CT	Number of Included Functional Groups
IEA02	Interchange Control Trailer	OEISA_S03_IEA02_ICN	Interchange Control Number
Loop: GSHDR	GROUP HEADER		
GS02	Functional Group Header	GSHDR_S01_GS02_APP_SENDR_CD	Application Senders Code
GS03	Functional Group Header	GSHDR_S01_GS03_APP_RECVR_CD	Application Receivers Code
GS04	Functional Group Header	GSHDR_S01_GS04_DT	Date
GS05	Functional Group Header	GSHDR_S01_GS05_TM	Time
GS06	Functional Group Header	GSHDR_S01_GS06_GCN	Group Control Number
GS08	Functional Group Header	GSHDR_S01_GS08_VERS_ID_CD	Version Release Industry Identifier Code
GE01	Functional Group Trailer	GSHDR_S03_GE01_TS_CT	Number of Transaction Sets Included
GE02	Functional Group Trailer	GSHDR_S03_GE02_GCN	Group Control Number
Loop: STHDR	TRANSACTION SET HEADER		
ST02	Transaction Set Header	STHDR_S01_ST02_CONTRL_NR	Transaction Set Control Number
BHT01	Beginning of Hierarchical Transaction	STHDR_S02_BHT01_STRUCTR_CD	Hierarchical Structure Code
BHT02	Beginning of Hierarchical Transaction	STHDR_S02_BHT02_TS_PURPS_CD	Transaction Set Purpose Code

BHT03	Beginning of Hierarchical Transaction	STHDR_S02_BHT03_ORGNTR_APLCTN_TRANSCTN_ID	Originator Application Transaction Identifier
BHT04	Beginning of Hierarchical Transaction	STHDR_S02_BHT04_TS_CREATN_DT	Transaction Set Creation Date
BHT05	Beginning of Hierarchical Transaction	STHDR_S02_BHT05_TS_CREATN_TIM	Transaction Set Creation Time
BHT06	Beginning of Hierarchical Transaction	STHDR_S02_BHT06_CLM_ENCNTR_ID	Claim or Encounter Identifier
SE01	Transaction Set Trailer	STHDR_S06_SE01_SEG_CT	Transaction Segment Count
SE02	Transaction Set Trailer	STHDR_S06_SE02_TCN	Transaction Set Control Number
Loop: 1000A	SUBMITTER NAME		
NM103	Submitter Name	L1000A_S01_NM103_SUBMTR_LAST_ORG_NM_OVL_NONPRSN_ENTY	Submitter Last or Organization Name - Non-Person Entity
NM109	Submitter Name	L1000A_S01_NM109_SUBMTR_ID_OVL_ETN_NR	Submitter Identifier - Electronic Transmitter Identification Number (ETIN)
PER02	Submitter EDI Contact Information	L1000A_S02_PER02_CONTCT_NM	Submitter Contact Name
PER04	Submitter EDI Contact Information	L1000A_S02_PER04_COMM_NR_OVL_TELPHN	Communication Number - Telephone
Loop: 1000B	RECEIVER NAME		
NM103	Receiver Name	L1000B_S01_NM103_RECVR_NM_OVL_NONPRSN_ENTY	Receiver Name - Non-Person Entity
NM109	Receiver Name	L1000B_S01_NM109_PRIMRY_ID_OVL_ETN_NR	Receiver Primary Identifier - Electronic Transmitter Identification Number (ETIN)
Loop: 2000A	BILLING PROVIDER HIERARCHICAL LEVEL		
HL01	Billing Provider Hierarchical Level	L2000A_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
Loop: 2010AA	BILLING PROVIDER NAME		
NM103	Billing Provider Name	L2010AA_S01_NM103_LAST_ORGL_NM_OVL_NONPRSN_ENTY	Billing Provider Last or Organizational Name - Non-Person Entity
N301	Billing Provider Address	L2010AA_S02_N301_BILNG_PROV_ADRS_LIN	Billing Provider Address Line
N401	Billing Provider City, State, ZIP Code	L2010AA_S03_N401_BILNG_PROV_CITY_NM	Billing Provider City Name
N402	Billing Provider City, State, ZIP Code	L2010AA_S03_N402_STAT_PROVNC_CD	Billing Provider State or Province Code
N403	Billing Provider City, State, ZIP Code	L2010AA_S03_N403_POSTL_ZON_ZIP_CD	Billing Provider Postal Zone or ZIP Code
REF02	Billing Provider Tax Identification	L2010AA_S04_REF02_ID_NR_OVL_EMPLYR_ID_NR	Billing Provider Tax Identification Number - Employer’s Identification Number
Loop: 2000B	SUBSCRIBER HEIRARCHICAL LEVEL		
HL01	Subscriber Hierarchical Level	L2000B_S01_HL01_HIERCHCL_ID_NR	Hierarchical ID Number
HL02	Subscriber Hierarchical Level	L2000B_S01_HL02_HIERCHCL_PARNT_ID_NR	Hierarchical Parent ID Number

HL04	Subscriber Hierarchical Level	L2000B_S01_HL04_HIERCHCL_CHILD_CD	Hierarchical Child Code
SBR01	Subscriber Information	L2000B_S02_SBR01_PAYR_RESP_SEQNC_NR_CD	Payer Responsibility Sequence Number Code
SBR02	Subscriber Information	L2000B_S02_SBR02_INDVDL_REL_CD	Individual Relationship Code
SBR09	Subscriber Information	L2000B_S02_SBR09_CLM_FILNG_IND_CD	Claim Filing Indicator Code
Loop: 2010BA	SUBSCRIBER NAME		
NM103	Subscriber Name	L2010BA_S01_NM103_SUB_LNAME_OVL_PERSN	Subscriber Last Name - Person
NM104	Subscriber Name	L2010BA_S01_NM104_SUB_FNAME	Subscriber First Name
NM109	Subscriber Name	L2010BA_S01_NM109_PRIMRY_ID_OVL_MEM_ID_NR	Subscriber Primary Identifier - Member Identification Number
N301	Subscriber Address	L2010BA_S02_N301_ADRS_LIN	Subscriber Address Line
N401	Subscriber City, State, ZIP Code	L2010BA_S03_N401_CITY_NM	Subscriber City Name
N402	Subscriber City, State, ZIP Code	L2010BA_S03_N402_STAT_CD	Subscriber State Code
N403	Subscriber City, State, ZIP Code	L2010BA_S03_N403_SUB_POSTL_ZON_ZIP_CD	Subscriber Postal Zone or ZIP Code
DMG02	Subscriber Demographic Information	L2010BA_S04_DMG02_BIRTH_DT_OVL_DT_CCYYMMDD	Subscriber Birth Date - Date Expressed in Format CCYYMMDD
DMG03	Subscriber Demographic Information	L2010BA_S04_DMG03_GENDR_CD	Subscriber Gender Code
Loop: 2010BB	PAYER NAME		
NM103	Payer Name	L2010BB_S01_NM103_PAYR_NM_OVL_NONPRSN_ENTY	Payer Name - Non-Person Entity
NM109	Payer Name	L2010BB_S01_NM109_PAYR_ID_OVL_PAYR_ID	Payer Identifier - Payor Identification
N301	Payer Address	L2010BB_S02_N301_ADRS_LIN	Payer Address Line
N401	Payer City, State, ZIP Code	L2010BB_S03_N401_CITY_NM	Payer City Name
N402	Payer City, State, ZIP Code	L2010BB_S03_N402_PAYR_STAT_PROVNC_CD	Payer State or Province Code
N403	Payer City, State, ZIP Code	L2010BB_S03_N403_PAYR_POSTL_ZON_ZIP_CD	Payer Postal Zone or ZIP Code
Loop: 2300	CLAIM INFORMATION		
CLM01	Claim Information	L2300_S01_CLM01_PATNT_CONTRL_NR	Patient Control Number
CLM02	Claim Information	L2300_S01_CLM02_TOTL_CLM_CHG_AMT	Total Claim Charge Amount
CLM05-01	Claim Information	L2300_S01_CLM05_01_PLAC_SVC_CD	Place of Service Code
CLM05-03	Claim Information	L2300_S01_CLM05_03_CLM_FREQNCY_CD_OVL_PLAC_SVC_CODS_PROF_DENTL	Claim Frequency Code - Place of Service Codes for Professional or Dental
CLM06	Claim Information	L2300_S01_CLM06_PROV_SUPLR_SIGNTR_IND	Provider or Supplier Signature Indicator
CLM07	Claim Information	L2300_S01_CLM07_ASGNMNT_PLAN_PARTCPTN_CD	Assignment or Plan Participation Code
CLM08	Claim Information	L2300_S01_CLM08_BENFTS_ASGNMNT_CERT_IND	Benefits Assignment Certification Indicator
CLM09	Claim Information	L2300_S01_CLM09_RELS_NFO_CD	Release of Information Code
CLM01-02	Claim Information	L2300_S43_HI01_02_DIAG_CD_OVL_ICD9CM_PRINCPL_DIAG	Diagnosis Code - International Classification of Diseases Clinical Modification (ICD-9-CM) Principal Diagnosis
CLM02-02	Claim Information	L2300_S43_HI02_02_DIAG_CD_OVL_ICD9CM_DIAG	Diagnosis Code - International Classification of Diseases Clinical Modification (ICD-9-CM) Diagnosis
Loop: 2310B	RENDERING PROVIDER NAME		

NM103	**Rendering Provider Name**	**L2310B_S01_NM103_LAST_ORG_NM_OVL_NONPRSN_ENTY**	**Rendering Provider Last or Organization Name - Non-Person Entity**
REF02	**Rendering Provider Secondary Identification**	**L2310B_S03_REF02_RENDRNG_PROV_2ND_ID_OVL_PROV_COMRCL_NR**	**Rendering Provider Secondary Identifier - Provider Commercial Number**
Loop: 2400	**SERVICE LINE NUMBER**		
LX01	**Service Line Number**	**L2400_S01_LX01_ASGND_NR**	**Assigned Number**
LX01-02	**Service Line Number**	**L2400_S02_SV101_02_PROC_CD_OVL_HCPCS_CD**	**Procedure Code - Health Care Financing Administration Common Procedural Coding System (HCPCS) Codes**
SV102	**Professional Service**	**L2400_S02_SV102_LIN_ITM_CHG_AMT**	**Line Item Charge Amount**
SV104	**Professional Service**	**L2400_S02_SV104_SVC_UNT_CT_OVL_UNT**	**Service Unit Count - Unit**
SV107-01	**Professional Service**	**L2400_S02_SV107_01_DIAG_CD_POINTR**	**Diagnosis Code Pointer**
DTP03	**Date - Service Date**	**L2400_S11_DTP03_SVC_DT_OVL_SVC_D8**	**Service Date - Service (D8)**