



HIPAA Claim Master Manual

Copyright Martin Scholl Consulting, Inc.

Table of Contents

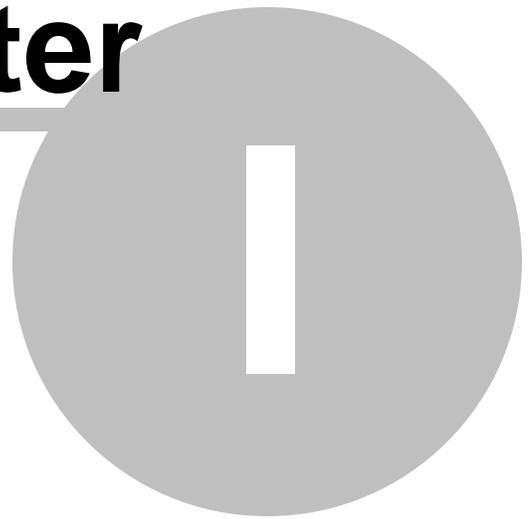
Foreword	0
Part I Introduction	2
1 About HIPAA Claim Master	2
Part II Getting Started	6
1 Starting the Application	6
2 Familiarizing with the Interface	7
3 Processing an Electronic Claim	8
4 Processing an Entire Directory	12
5 Displaying Claims	17
6 Facsimile Mode	21
7 Working with Multi-Page Claims	22
8 Printing Electronic Claims	23
9 Requesting a Trial Extension	25
10 Enabling Application Components	29
Part III Configuring the Application	32
1 Configuring Program Options	32
2 Configuring Specific Options	35
3 Configuring Claim Forms	38
4 Using the EDI Viewer and Editor	41
5 Selecting a Code Set	44
6 Accessing Daily File Log	49
7 Accessing Statistics	53
8 Selecting a Printer	54
Part IV Creating Image Files and PDFs	57
1 Creating Image Files from EDI	57
2 Doing Image Setup	60
3 Configuring Image Index	65
4 Creating Image Index Table	70
5 Viewing Image Files	75
Part V Using the Database	79
1 Exporting the Data	79
2 Setting up Database Connection	81
3 Creating Tables	88

Microsoft SQL Server Script to Create Tables	94
MySQL Script to Create Tables	131
Oracle Script to Create Tables	168
Microsoft Access Script to Create Tables	204
Other Databases Scripts to Create Tables	216
4 Selecting Fields to Export	216
5 Browsing Exported Data	225
6 Displaying the Revenue or Procedure Code Descriptions	229
7 Creating Code Look-up Tables	232
8 Troubleshooting Data Export	237
Part VI Displaying and Printing Additional Information	241
1 Understanding Types of Additional Information	241
2 Configuring Additional Info Pages	247
3 Displaying Addl Info	251
4 Displaying COB	254
5 Displaying Document Info	256
6 Printing Blank Forms	258
Part VII Using Search	261
1 Searching through a Claim File	261
Part VIII Editing and Entering Clams	264
1 Using the Claim Entry/Edit Function	264
2 Selecting a Claim Entry/Edit CodeSet	268
3 Defining Claim Entry Settings	268
4 Provider, Patient and Payer Tables	274
5 Stored Patients	277
6 Stored Payers	278
7 Stored Providers	280
8 Creating CodeSet Tables	281
9 Filling CodeSet Tables	285
10 Viewing and Editing CodeSets	288
Part IX Creating 837 EDI Files	296
1 About Creating EDI Files	296
2 Setting up Company	296
3 Creating Company Setup Table	300
4 Creating 834 EDI Files from the Attached Database	303
5 Understanding the "Create 837 Transaction Sets" Window	308

Part X Creating 277CA EDI Files	312
1 What are 277CA files	312
2 277CA in the HIPAA Claim Master	312
3 Setting up the Auto generated 277CA	314
4 Setting up trading partner for 277	315
5 Creating a 277CA file manually	316
From file	318
From Database	322
From the 'Claim in Database' screen	325
Part XI Automating	328
1 Running the Application from the Command Line	328
2 Using Command Line Arguments (CLI)	329
3 Printing and imaging from the database	333
4 Changing the Default Company	334
5 Running the Application via Scheduler	335
Part XII Using EDI Exchange Features	342
1 Getting Started	342
About EDI Exchange	342
Requesting EDI Exchange License	344
Registering EDI Exchange	346
2 Configuring EDI Exchange (Obligatory Settings)	348
1 Setting up Database Connection	349
2 Creating Database Tables	349
2b Creating 999 File Tables	354
2c Updating Database Tables	360
3 Defining Auto-Processing Options	362
4 Defining Communications Directory	364
5 Initializing EDI Exchange	366
3 Configuring EDI Exchange (Optional Settings)	372
Setting up Email Notifications	372
Setting up Incoming and Outgoing Files Options	376
Running the Application via Scheduler	379
Using the Command Line Arguments (CLI)	380
4 Working with Trading Partners	381
Setting up Trading Partners	381
Custom File Naming Conventions	395
Using Built-in FTP Client	398
Creating a Trading Partner Automatically	402
Certificate based authentication in SFTP	404
5 Using Encryption	406
About Encrytion	406
Setting up Encryption	407
Using PGP Desktop	410
Using GnuPG	411

6 Using EDI Exchange Features	413
Accessing Inbox Manager	413
Checking EDI Files	414
Downloading EDI Files From FTP Server	416
Processing EDI Files	418
Accessing EDI Receive Log	419
Accessing EDI Send Log	422
Listing SEF Files	424
 Part XIII Bug Reporting and Program Update	 427
1 Reporting an Issue	427
2 Updating the Program	430
 Index	 434

Chapter



1 Introduction

1.1 About HIPAA Claim Master

The **HIPAA Claim Master** is an advanced tool for healthcare electronic claims.

The HIPAA act of 1995 mandates that every medical claim that is transmitted electronically conforms to strict standards. These standards are developed by ANSI, the American National Standards Institute, under the umbrella of the X12 committee for Electronic Data Interchange (EDI). X12 contains standards for hundreds of electronic data interchanges. Medical claims are contained in the **837-transaction set**. X12 standards are continuously developing. The HIPAA compliant versions are **version 4010A1** and **5010**. The HIPAA Claim Master supports both versions.

There are four different flavors of the 837 transactions:

1. The institutional claim format 004010X096A1 and 005010X223A1
2. The professional claim format 004010X098A1 and 005010X222A1
3. The dental claim format 004010X097A1 and 005010X224A1
4. The reporting claim format 005010X225A2

The HIPAA Claim Master handles all four of them.

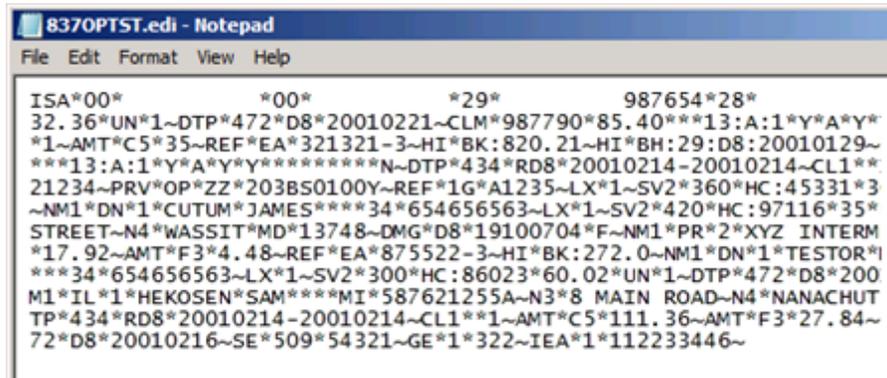
Electronic claims were introduced in the 1990's and made mandatory by 1/1/2003. During the last decade, the adaptation of this electronic format was phenomenal. More than 90% of claims are now (2012) transmitted electronically. The HIPAA Claim Master was first developed in 2001 and has undergone tremendous change over those years. In 2011, a new version 4.0 was released based on Microsoft's .Net infrastructure.

The HIPAA Claim Master reads X12 837 EDI claim files and parses the claim information with its built-in EDI translator into distinct claims. Each claim can then be rendered as an image of a UB04, CMS-1500 and ADA 2006 form or the older formats UB92, HCFA-1500 or ADA2000. These images can be viewed on the screen, printed to paper or turned into image files, PDF and TIFF formats, and added into an image management system.

In addition, the HIPAA Claim Master can export claim data to any database. Claim tables can be created within any ODBC/OleDB compliant database such as a Microsoft SQL Server, Oracle or MYSQL. The claim data is then accessible to your own applications plus you can search this data warehouse and even recreate the image.

What is an EDI file?

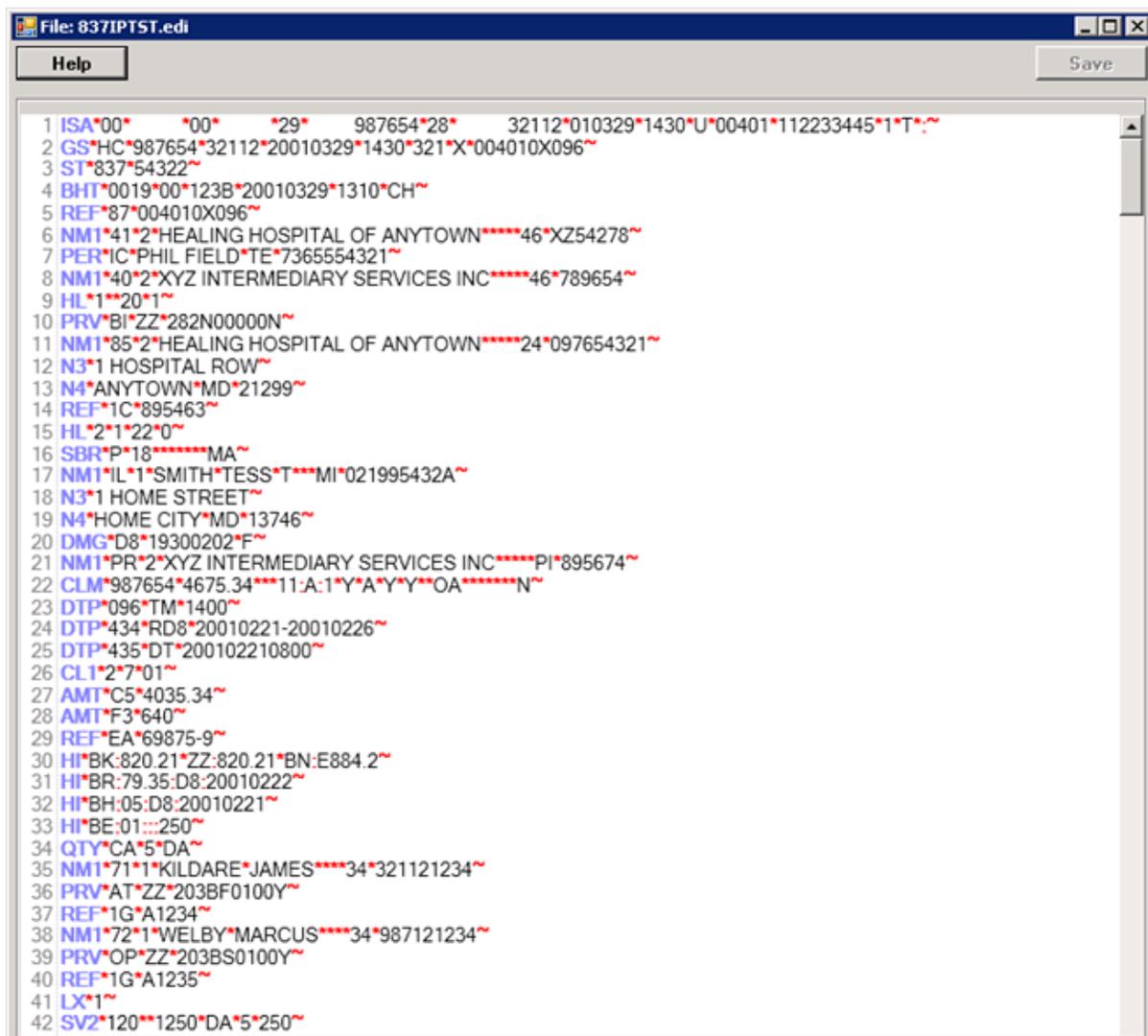
X12 is an ASCII standard. This means that all the content of a transmission consists of ASCII characters. ASCII or text files can be viewed and edited with a regular text editor but since EDI does not use line feeds and carriage returns to mark the end of a line, an EDI document is usually very hard to view.



```
837OPTST.edi - Notepad
File Edit Format View Help
ISA*00*                *00*                *29*                987654*28*
32.36*UN*1~DTP*472*D8*20010221~CLM*987790*85.40***13:A:1*Y*A*Y*
*1~AMT*C5*35~REF*EA*321321-3~HI*BK:820.21~HI*BH:29:D8:20010129~
***13:A:1*Y*A*Y*Y*****N~DTP*434*RD8*20010214-20010214~CL1**
21234~PRV*OP*ZZ*203BS0100Y~REF*1G*A1235~LX*1~SV2*360*HC:45331*3
~NM1*DN*1*CUTUM*JAMES****34*654656563~LX*1~SV2*420*HC:97116*35*
STREET~N4*WASSIT*MD*13748~DMG*D8*19100704*F~NM1*PR*2*XYZ INTERM
*17.92~AMT*F3*4.48~REF*EA*875522-3~HI*BK:272.0~NM1*DN*1*TESTOR*I
***34*654656563~LX*1~SV2*300*HC:86023*60.02*UN*1~DTP*472*D8*200
M1*IL*1*HEKOSEN*SAM****MI*587621255A~N3*8 MAIN ROAD~N4*NANACHUT
TP*434*RD8*20010214-20010214~CL1**1~AMT*C5*111.36~AMT*F3*27.84~
72*D8*20010216~SE*509*54321~GE*1*322~IEA*1*112233446~
```

A view of raw EDI data

As you can see in the previous figure, it is quite difficult to discern the individual data elements within the ASCII text file. One needs an EDI Editor to get a more meaningful representation of an EDI file. The HIPAA Claim Master has a built-in EDI Editor. Here is how this file is displayed in it:

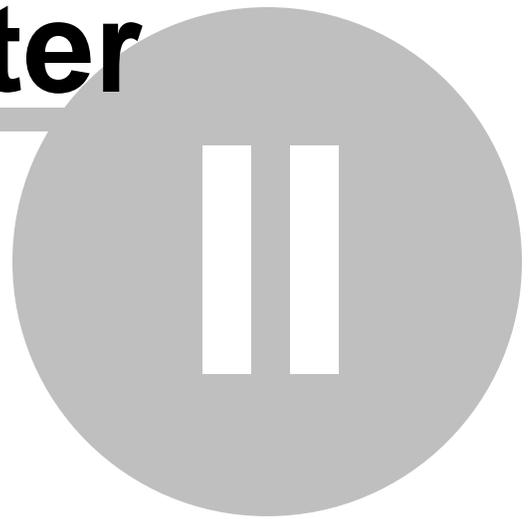


```
File: 837IPTST.edi
Help Save
1 ISA*00*      *00*      *29*      987654*28*      32112*010329*1430*U*00401*112233445*1*T::~~
2 GS*HC*987654*32112*20010329*1430*321*X*004010X096~
3 ST*837*54322~
4 BHT*0019*00*123B*20010329*1310*CH~
5 REF*87*004010X096~
6 NM1*41*2*HEALING HOSPITAL OF ANYTOWN*****46*XZ54278~
7 PER*IC*PHIL FIELD*TE*7365554321~
8 NM1*40*2*XYZ INTERMEDIARY SERVICES INC*****46*789654~
9 HL*1**20*1~
10 PRV*BI*ZZ*282N00000N~
11 NM1*85*2*HEALING HOSPITAL OF ANYTOWN*****24*097654321~
12 N3*1 HOSPITAL ROW~
13 N4*ANYTOWN*MD*21299~
14 REF*1C*895463~
15 HL*2*1*22*0~
16 SBR*P*18*****MA~
17 NM1*IL*1*SMITH*TESS*T***MI*021995432A~
18 N3*1 HOME STREET~
19 N4*HOME CITY*MD*13746~
20 DMG*D8*19300202*F~
21 NM1*PR*2*XYZ INTERMEDIARY SERVICES INC*****PI*895674~
22 CLM*987654*4675.34***11:A:1*Y*A*Y**OA*****N~
23 DTP*096*TM*1400~
24 DTP*434*RD8*20010221-20010226~
25 DTP*435*DT*200102210800~
26 CL1*2*7*01~
27 AMT*C5*4035.34~
28 AMT*F3*640~
29 REF*EA*69875-9~
30 HI*BK:820.21*ZZ:820.21*BN:E884.2~
31 HI*BR:79.35:D8:20010222~
32 HI*BH:05:D8:20010221~
33 HI*BE:01:::250~
34 QTY*CA*5*DA~
35 NM1*71*1*KILDARE*JAMES****34*321121234~
36 PRV*AT*ZZ*203BF0100Y~
37 REF*1G*A1234~
38 NM1*72*1*WELBY*MARCUS****34*987121234~
39 PRV*OP*ZZ*203BS0100Y~
40 REF*1G*A1235~
41 LX*1~
42 SV2*120**1250*DA*5*250~
```

The same data viewed with HIPAA Claim Master's built-in EDI Editor

When you view this ASCII data with a special EDI editor, the individual data segments are visible but it is still difficult to understand the content of the form. EDI tries to abbreviate complex descriptions into short codes; and without a translator only true experts know what this means. This is where the HIPAA Claim Master is indispensable. Its built-in translator parses the significant data elements from the EDI file and prints the data into the well-accepted medical forms HCFA-1500 and UB92 as well as the ADA dental forms. The program's output can be screen, paper or image file (TIFF and PDF) as well as data rows in external tables.

Chapter



2 Getting Started

2.1 Starting the Application

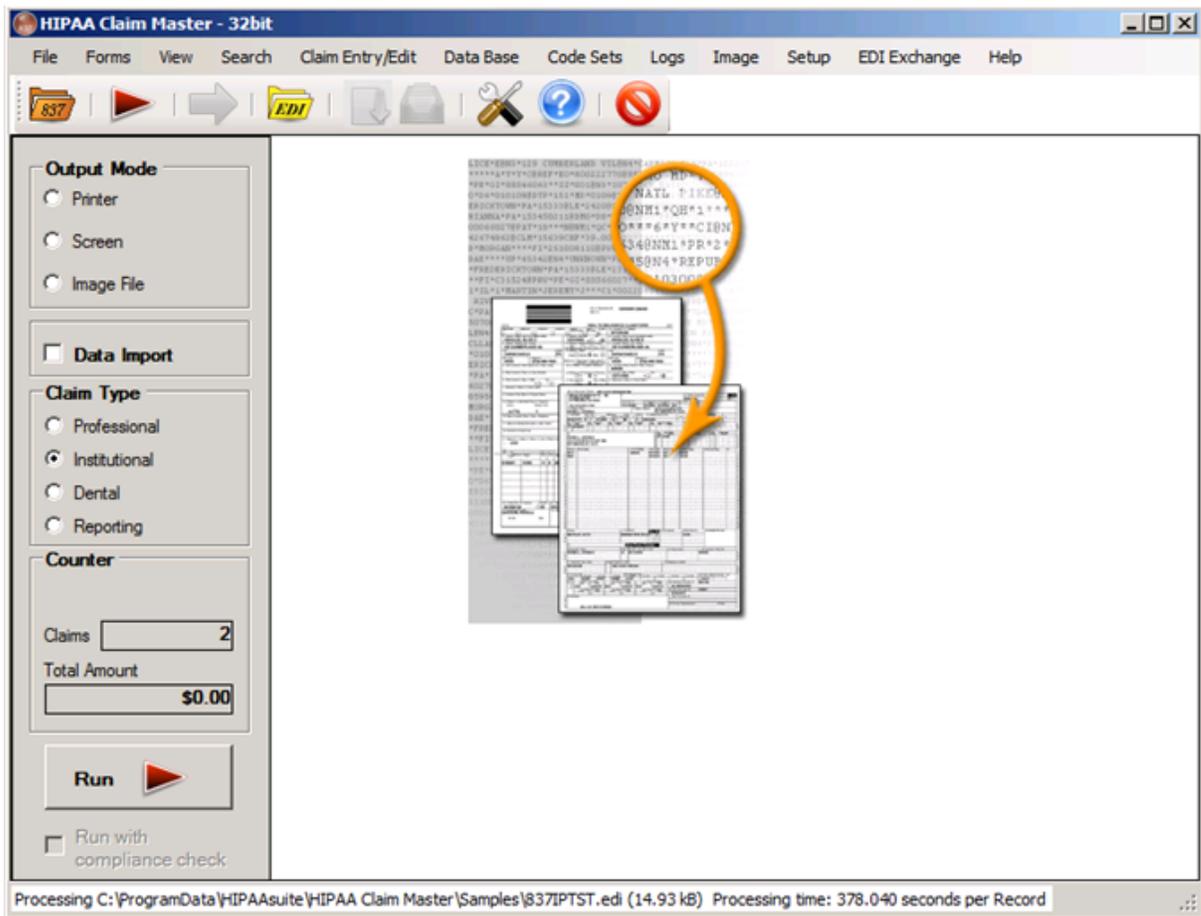
Once the Claim Master is installed, you can launch it from the "Start" menu:

Start ▶ Programs (All Programs) ▶ HIPAASuite ▶ HIPAA Claim Master.



The HIPAA Claim Master is launched from the "Start" menu

The following screen will appear:



The HIPAA Claim Master main window

Once the main program window is opened, the normal operation now would be to open an EDI file. See:

- [Processing an Electronic Claim](#)
- [Processing an Entire Directory](#)

2.2 Familiarizing with the Interface

The Top Menu Bar

The menu bar at the top shows the following menu items:

- **File** – Contains all file-related activities such as [opening a file](#) or [opening a directory](#), [printer selection](#), and exiting the program.
- **Forms** – Lets you see and print all of the 7 different claim forms that the HIPAA Claim Master renders.

- **View** – Gives you access to the built-in [EDI Editor](#), and it lets you show or hide the info bar.
- **Search** – Brings up the [EDI search tool](#).
- **Database** – Gives you access to database setup and the claim database. See [Exporting the Data](#) for details.
- **Logs** – Lets you see the logs of file processing. See [Accessing Daily File Log](#).
- **Setup** – Gives you access to the different setup screens. See [Configuring Program Options](#), [Configuring Specific Options](#), [Configuring Claim Forms](#).
- **Help** – Shows the documentation and context sensitive help topics.

Note: There may be additional menu items if you have licensed modules. Each module adds a separate menu. See [Enabling Application Components](#).

The Task Bar

The taskbar has the most important functions accessible through quick launch icons.

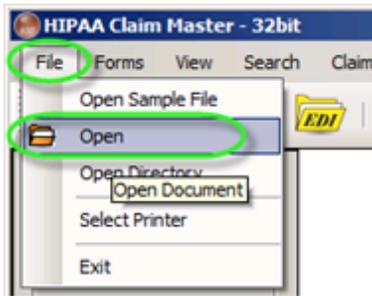
The Info Bar

The info bar is located in the left part of the interface. This panel displays the file information and has the "Run" button to start processing files. If you do not see the info bar, there is the "Info Bar" menu item that shows it (the last button located on the task bar also toggles the info bar). When you work with files, the info bar is indispensable, but when working with a database, it is optional.

2.3 Processing an Electronic Claim

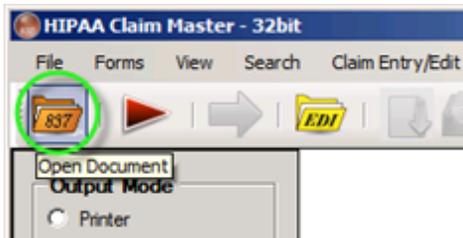
Follow the instructions below to open a single EDI file and display its details on screen in easy-to-read table format.

1. Go to *File* ► *Open* in the main menu.



The "Open" menu

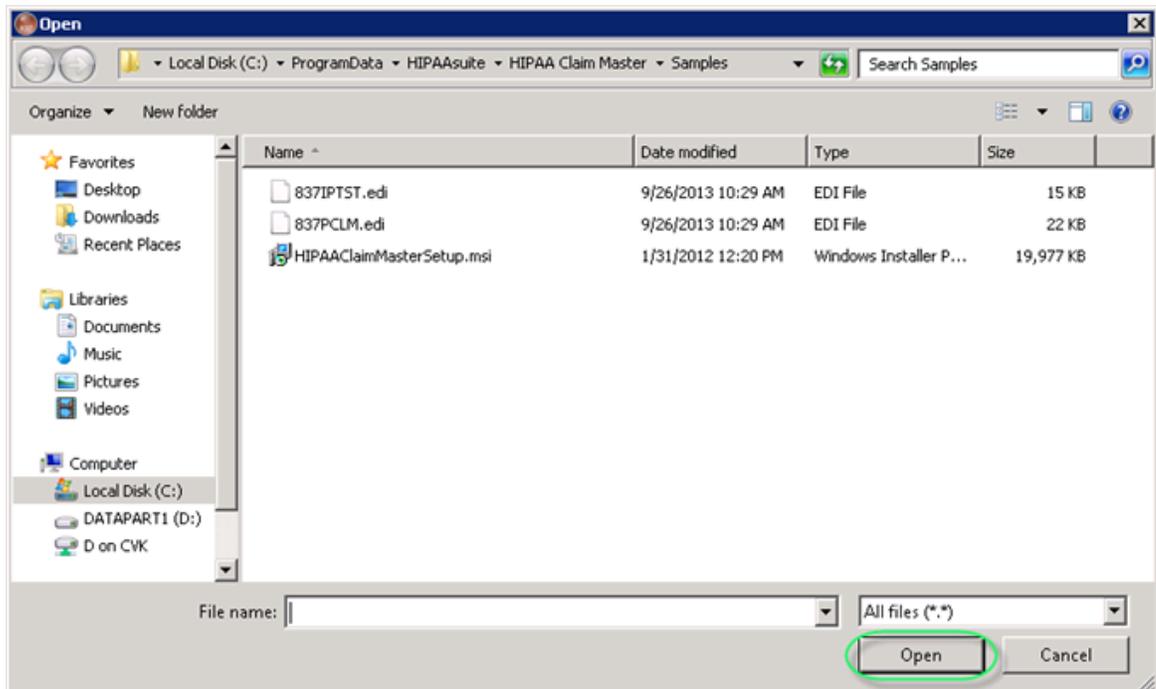
Alternatively, click on the "Open Document" icon shortcut located on the program toolbar.



The "Open Document" button

2. Browse through the folders and select a file with .edi extension. Click "Open."

Note: Only claim files in "ANSI 837" format can be opened via the Claim Master. The program will display an error message if you select a file in a different format.



The "Open" window

3. Make sure the program tool bar is displayed in the left area. To display the left program bar with options, select *View ▶ Info Bar*.
4. Select one of the following modes in the "Output Mode" box. The Output mode defines the destination where the result file will be sent to. In our example, we select the "Screen" mode.
 - **Screen** – This is a default mode. Select this option to display the EDI file on screen in a preview window.
 - **Printer** – This option allows to print the opened EDI file. See also: [Selecting a Printer](#).
 - **Image File** – This mode mode creates TIFF or PDF file. Read more in [Creating Image Files from EDI](#).



The "Output Mode" options

Note: You can choose the Output Mode, but not the Claim Type. The claim type is determined by the EDI file. If you have the EDI exchange component licensed, you will have the option to run a HIPAA compliance check before processing a file.

- Professional
- Institutional
- Dental
- Reporting

5. The file is now open. Click the "Run" button to process the file.



The "Run" buttons

6. When you click on the "Run" button, the file will be parsed and the information displayed. As in our example the output mode is "Screen," the claim is displayed in a separate window. See [Displaying Claims](#).

The screenshot shows a 'HIPAA Preview' window with a menu bar (Print, Close, Next Claim) and navigation controls (1 / 1, Claim, Add info, COB info). The main content is an 837 EDI claim form. Key fields include:

- Sender's ESI Number: 987654
- HEALING HOSPITAL OF ANYTOWN, 1 HOSPITAL ROW, ANYTOWN, MD 21299
- PAT ID: 987654
- PAT REG ID: 69875-9
- FED TAX NO: 097654321
- STATE: 02
- CITY: HOME CITY
- STATE: MD
- ZIP: 13746
- PATIENT NAME: SMITH, TESS T
- PATIENT ADDRESS: 1 HOME STREET, HOME CITY, MD 13746
- ADMISSION DATE: 02/21/01 08:00
- DISCHARGE DATE: 02/21/01 14:00
- ICD9 CODE: 05
- ICD9 DATE: 02/21/2001
- CHARGE TABLE:

42 REV CD	43 DESCRIPTION	44 RCP RATE	45 SERV DATE	46 SERV UNITS	47 TOTAL CHARGES	48 COVERED CHARGES
1	120	\$250.00		5	\$1,250.00	
2	320			5	\$1,006.00	
3	360			1	\$722.00	
4	370			1	\$310.00	
5	450			1	\$987.34	
6	710			1	\$400.00	

The display of an 837 EDI claim

If you choose an output mode other than Screen, you will see the hourglass mouse pointer until the parsing has been completed.

2.4 Processing an Entire Directory

You can process every file in an entire directory with the HIPAA Claim Master. This feature is very useful when [processing through the Windows Scheduler](#). You can then have the HIPAA Claim Master process all the files in a certain directory and move them to a processed file directory in certain intervals.

Tip: The command line option can take a directory as input. The program decides whether the command line argument is either a file or a directory and processes accordingly.

Follow the instructions below to open a batch of EDI files located in one folder.

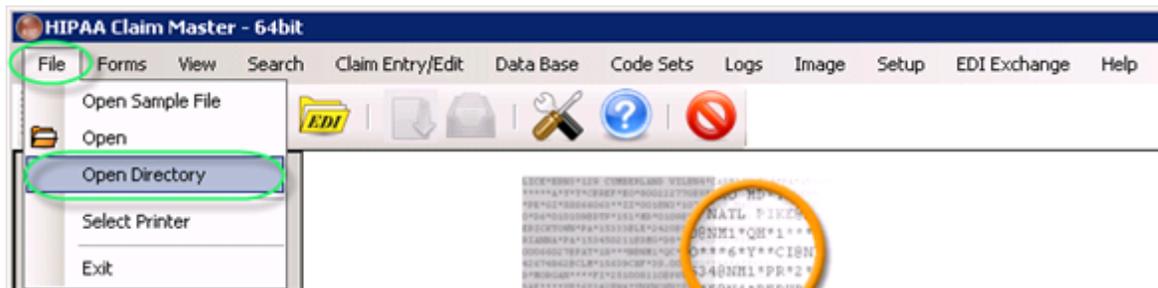
1. Make sure the "Output Mode" option is not "Screen." The Output mode defines the destination where the result file will be sent to. It does not make sense to open every claim in a directory one by one, so the directory processing is limited to "Print," "Image File" and "Data Import" option. The output modes are described in [Processing an Electronic Claim](#).

Note: You cannot display a batch of EDI files on screen in a preview window, it is possible only for one file. See [Processing an Electronic Claim](#).



The "Output Mode" options

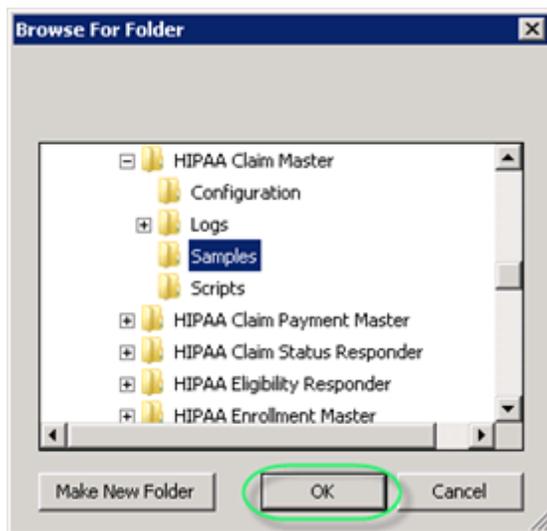
2. Go to **File** ▶ **Open Directory** in the main menu.



The "Open Directory" menu

3. Browse for the folder containing EDI files. Click on that folder and then click "OK."

Note: Only claim files in "ANSI 837" format can be opened via the HIPAA Claim Master. The program will display an error message if you select a file in a different format.



The "Browse for Folder" popup

4. Click on the "Run" button to send all files from the folder to the destination.

Note: The pre-defined settings will be applied. See also: [Configuring Program Options](#).



The "Run" buttons

You can see the current status at the bottom pane.



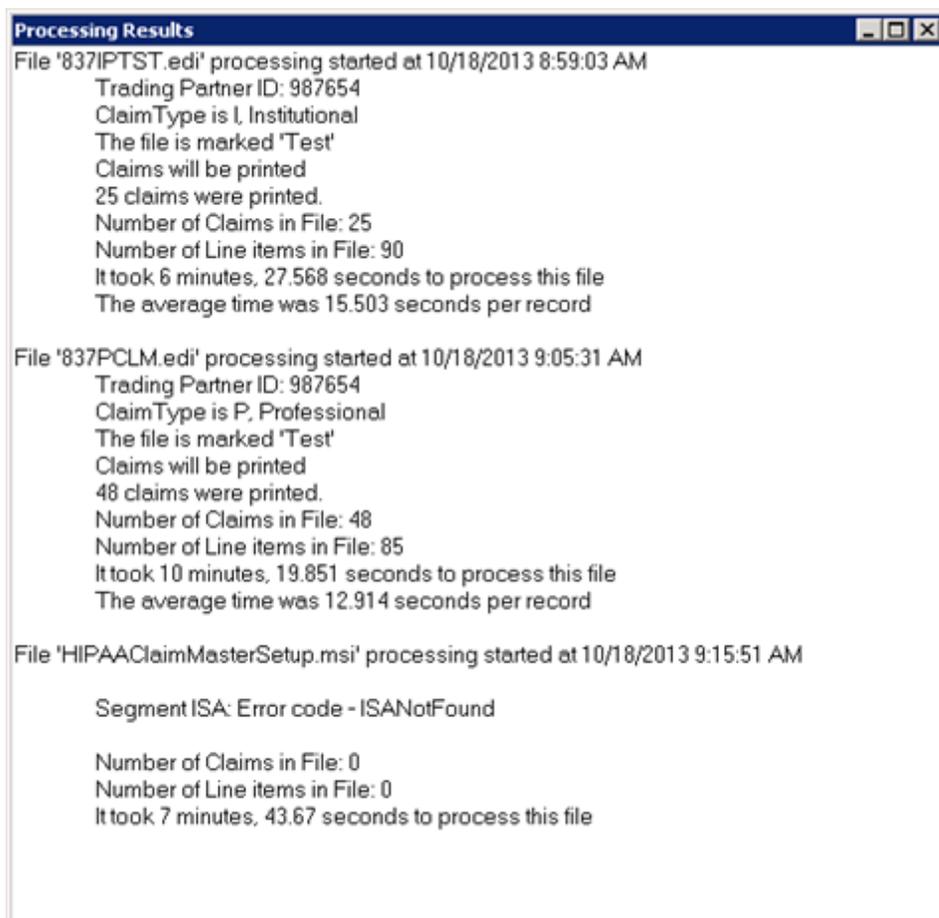
The Status pane

The file counter located at the bottom of the Info bar provides you the information about the processed files in real time.



The file counter at the bottom of the info bar

5. Once all files have been processed, the "Processing Results" popup will report the results of the operation.



The "Processing Results" window

6. The number of processed files and records is displayed in the "Counter" pane.

The "Counter" pane

2.5 Displaying Claims

After [opening a file](#) and leaving the default output mode "Screen", click on the "Run" button.

The form type is indicated in the left pane, under "Claim Type."

HEALTH INSURANCE CLAIM FORM
APPROVED BY NATIONAL UNIFORM BILLING COMMISSION USOS

1. MEDICARE (Medicare #) MEDICAID (Medicaid #) TRICARE (Sponsor's SSN) CHAMPVA (Member ID#) GROUP HEALTH PLAN (SGN or ID) FECA (SSN) OTHER (ID)

2. PATIENT'S NAME (Last Name, First Name, Middle Initial)
Nosmith, Herman

3. PATIENT'S BIRTH DATE SEX
12/25/1920 M F

5. PATIENT'S ADDRESS (No. Street)
10 9th Avenue

6. PATIENT'S RELATIONSHIP TO INSURED
Self Spouse Child Other

CITY STATE
Nanachut **MD**

ZIP CODE TELEPHONE
13747

8. PATIENT'S STATUS
Single Married Other

9. OTHER INSURED'S NAME (Last, First, Middle Initial)
a. OTHER INSURED'S POLICY OR GROUP NUMBER

10. IS PATIENT'S CONDITION RELATED TO:
a. EMPLOYMENT? (Contributor Preferred)

The Claim Type menu

Depending on the claim type (Institutional, Professional or Dental), a preview form will

come up to display the claim in the standard UB92, HCFA-1500 or ADA2000 format.

HIPAA Preview

Print Close 1 / 2 Claim Add info COB info

Create Date: 05/05/2010 Sender's EDI No: 3xxx65334

1500 Multi-Page Document Part 1

HEALTH INSURANCE CLAIM FORM
APPROVED BY NATIONAL UNIFORM BILLING COMMISSION 0805

30699 RUSSELL ROAD STE 175
WESTLAKE VILLAGE, OR 713627315

1. MEDICARE <input type="checkbox"/> (Medicare #)		2. MEDICAD <input type="checkbox"/> (Medicaid #)		3. TRICARE CHAMPUS (Sponsor's SSN)		4. CHAMPVA (Member ID)		5. GROUP HEALTH PLAN (SSN or ID)		6. FECA BUC/LUNG (SSN)		7. OTHER (ID)		18. INSURED'S I.D. NUMBER (For Program in item 1)																					
2. PATIENT'S NAME (Last Name, First Name, Middle Initial) CHARLES, SUZETTE L		3. PATIENT'S BIRTH DATE 07/05/1973		SEX M <input type="checkbox"/> F <input checked="" type="checkbox"/>		4. INSURED'S NAME (Last Name, First Name, Middle Initial) CHARLES, SUZETTE L		5. PATIENT'S ADDRESS (No. Street) 849 BAKER ROAD		6. PATIENT'S RELATIONSHIP TO INSURED Self <input checked="" type="checkbox"/> Spouse <input type="checkbox"/> Child <input type="checkbox"/> Other <input type="checkbox"/>		7. INSURED'S ADDRESS (No. Street) 849 BAKER ROAD		19. INSURED'S POLICY GROUP OR FECA NUMBER 303787711																					
CITY WANGEN		STATE ID		8. PATIENT'S STATUS Single <input type="checkbox"/> Married <input type="checkbox"/> Other <input type="checkbox"/>		CITY WANGEN		STATE ID		ZIP CODE 91239		TELEPHONE		9. INSURED'S DATE OF BIRTH 07/05/1973																					
9. OTHER INSURED'S NAME (Last, First, Middle Initial) CHARLES, SUZETTE L		10. IS PATIENT'S CONDITION RELATED TO: a. EMPLOYMENT? (Current or Previous) <input type="checkbox"/> YES <input checked="" type="checkbox"/> NO		b. AUTO ACCIDENT PLACE (SRRR) <input checked="" type="checkbox"/> YES <input type="checkbox"/> NO		c. OTHER ACCIDENT <input type="checkbox"/> YES <input checked="" type="checkbox"/> NO		11. INSURED'S POLICY GROUP OR FECA NUMBER 303787711		8. EMPLOYER'S NAME OR SCHOOL NAME		9. EMPLOYER'S NAME OR SCHOOL NAME		10. IS THERE ANOTHER HEALTH BENEFIT PLAN? <input checked="" type="checkbox"/> YES <input type="checkbox"/> NO																					
8. OTHER INSURED'S POLICY OR GROUP NUMBER YRxxxxx55698		9. OTHER INSURED'S DATE OF BIRTH 07/05/1973		SEX M <input type="checkbox"/> F <input checked="" type="checkbox"/>		c. EMPLOYER'S NAME OR SCHOOL NAME		10. INSURANCE PLAN NAME OR PROGRAM NAME ANTHEM BCBS VA		11. INSURED'S OR AUTHORIZED PERSON'S SIGNATURE Source Release Cert Date on File Y		12. INSURED'S OR AUTHORIZED PERSON'S SIGNATURE Payment/Assignment Cert. Indicator Y		13. HOSPITALIZATION DATES RELATED TO CURRENT SERVICES FROM 02/26/2010 TO																					
14. DATE OF CURRENT ILLNESS (First Day) OR INJURY (Accident or Pregnancy (LMP)) 08/15/2009		15. IF PATIENT HAS HAD SAME OR SIMILAR ILLNESS GIVE FIRST DATE		16. DATES PATIENT UNABLE TO WORK IN CURRENT OCCUPATION FROM TO		17. NAME OF REFERRING PROVIDER OR OTHER SOURCE SRADERS, MARISS		18. HOSPITALIZATION DATES RELATED TO CURRENT SERVICES FROM 02/26/2010 TO		19. RESERVED FOR LOCAL USE		20. OUTSIDE LAST \$ CHARGES <input type="checkbox"/> YES <input checked="" type="checkbox"/> NO		21. MEDICAD RESUBMISSION CODE																					
21. DIAGNOSIS OR NATURE OF ILLNESS OR INJURY (Relate items 1,2,3 or 4 to item 24a by line) 1. 845.09 2. 719.47 3. 728.87 4.		22. MEDICAD RESUBMISSION CODE		23. PRIOR AUTHORIZATION NUMBER		A. DATE(S) OF SERVICE FROM TO		B. PLACE		C. ICD-9-CM PROCEDURE CODE		D. PROCEDURES, SERVICES OR SUPPLY CPT/HCPCS MODIFIER		E. DIAGNOSIS POINTER		F. \$ CHARGES		G. DAYS UNITS		H. EPS		I. ID. QUAL		J. RENDERING PROVIDER ID #											
1		04/02/2010		11		97110		1,2,3		\$55.00		1		NR		1487646196		2		04/02/2010		11		97112		1,2,3		\$100.00		2		NR		1487646196	
3		04/02/2010		11		97140		1,2,3		\$60.00		1		NR		1487646196		4		04/02/2010		11		97530		1,2,3		\$55.00		1		NR		1487646196	
5		04/05/2010		11		97110		1,2,3		\$110.00		2		NR		1487646196		6		04/05/2010		11		97112		1,2,3		\$100.00		2		NR		1487646196	
25. FEDERAL TAX I.D. NUMBER 631133454		26. PATIENT'S ACCOUNT NO. 400713216CLAIM		27. ACCEPT ASSIGNMENT? <input checked="" type="checkbox"/> YES <input type="checkbox"/> NO		28. TOTAL CHARGE		29. AMOUNT PAID		30. BALANCE DUE		31. RENDERING PROVIDER BISHOP, JULIE MARIE Specialty 225100000X On File Date		32. SERVICE FACILITY LOCATION INFORMATION POST ROAD-IN 1260 NORTH POST ROAD SUITE B INDIANAPOLIS, IN 46219		33. BILLING PROVIDER'S INFO & PH # PHYSICAL THERAPY PARTNERS IN 3544 SOLUTIONS CENTER MENDOZA, IL 606223005																			

A professional claim form CMS-1500, the first page of two

Sender's EDI Number: 987654

1 HEALING HOSPITAL OF ANYTOWN
1 HOSPITAL ROW
ANYTOWN MD 21299

2 PATIENT NAME: SMITH TESS T
3 PATIENT ADDRESS: HOME CITY, MD 13746

4 BIRTH DATE: 02021930
5 SEX: F
6 DATE OF ADMISSION: 022101
7 ADMISSION TYPE: 08
8 ICD-9-CM PROCEDURE CODE: 2
9 ICD-9-CM DIAGNOSIS CODE: 7
10 ICD-9-CM DRUG CODE: 14
11 ICD-9-CM STATUS: 01

25 CODE	26 VALUE CODES AMOUNT	40 CODE	41 VALUE CODES AMOUNT	42 REV./CD.	43 DESCRIPTION	44 NO. PROC./RATE/HIPPS CODE	45 SERV. DATE	46 SERV. UNITS	47 TOTAL CHARGES	48 NO. UN-COVED CHARGES	49
01	250.00			120		\$250.00		5	1250.00		
				320				5	1006.00		
				360				1	722.00		
				370				1	310.00		
				450				1	987.34		
				710				1	400.00		

View of an Institutional claim in Facsimile Mode.

The image above shows a UB-04 in facsimile mode.

2.7 Working with Multi-Page Claims

Professional claims with more than 6 lines and institutional claims with more than 23 lines do not fit on one form. The HIPAA Claim Master will display such claims on multiple forms. The header of each form will indicate that this is a multi-part document as well as which segment the present page is. The screen form has navigation buttons, so that you can page forwards and backwards through the claim.

← 1 / 2 → | Claim | AddInfo | COB info

Sender's EDI No: 987654
XYZ Carrier Services Inc

Navigating through multi-page claim. The page number is indicated

Only the line information will differ on each form. The claim header information will be identical. HIPAAsuite does not show page totals but will always display the claim total.

Multi-page claims will be printed on several sheets of paper.

The above figure also shows that this claim has additional info that does not fit on the UB04 and there is extra COB info.

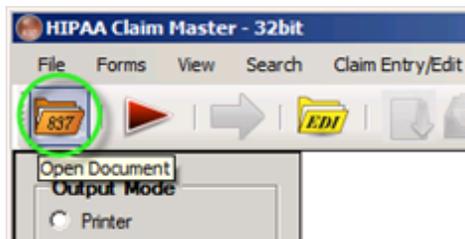
2.8 Printing Electronic Claims

If you want to print to a printer other than the computer's defined default printer, select the desired printer. See [Selecting a Printer](#).

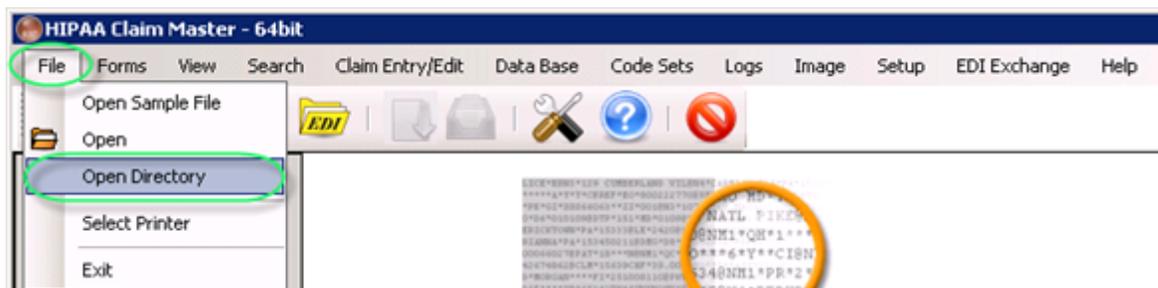
Note: The HIPAA Claim Master Version 4.0 onwards does not need anymore third party software for the creation of image files. Older versions of the HIPAA Claim Master needed to have the [Peernet Tiff Image Printer](#) or [PDF Image Printer](#) driver installed.

Follow the instructions below to print the Electronic Claim.

1. Open an EDI file or directory containing the EDI files in the HIPAA Claim Master. Read more in [Processing an Electronic Claim](#), [Processing an Entire Directory](#).



The "Open Document" button



The "Open Directory" menu

2. Select the "Printer" option in the "Output Mode" block.



The "Printer" mode

3. Click on the "Run" button.



The "Run" buttons

4. The "Processing Results" window displays the report.



The "Processing Results" window

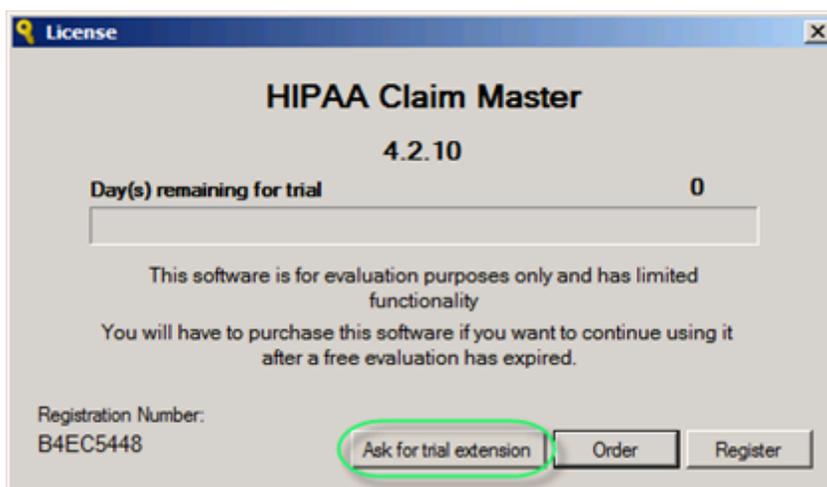
2.9 Requesting a Trial Extension

The HIPAA Claim Master comes with a free trial of 15 days. Once your trial period has expired, the application will stop working. There are limitations to the trial version; for example, you can only export the first 50 records of EDI files, and only 5 files per directory processing. We are sorry to have to limit the experience, but we had instances where organizations used the software for production purposes without purchasing it.

We do not mind extending your trial if you need extra time to evaluate the software. You will only need to send us the registration number and we will gladly give you another 15 day trial. If your trial time has expired and you wish to continue your testing of the software, please send an email to info@HIPAAsuite.com with the Registration number and we will give you a trial extension.

If you need to continue your trial of the product, follow the instructions below.

1. Launch the application.
2. Click on the "Ask for Trial Extension" button on the displayed window.

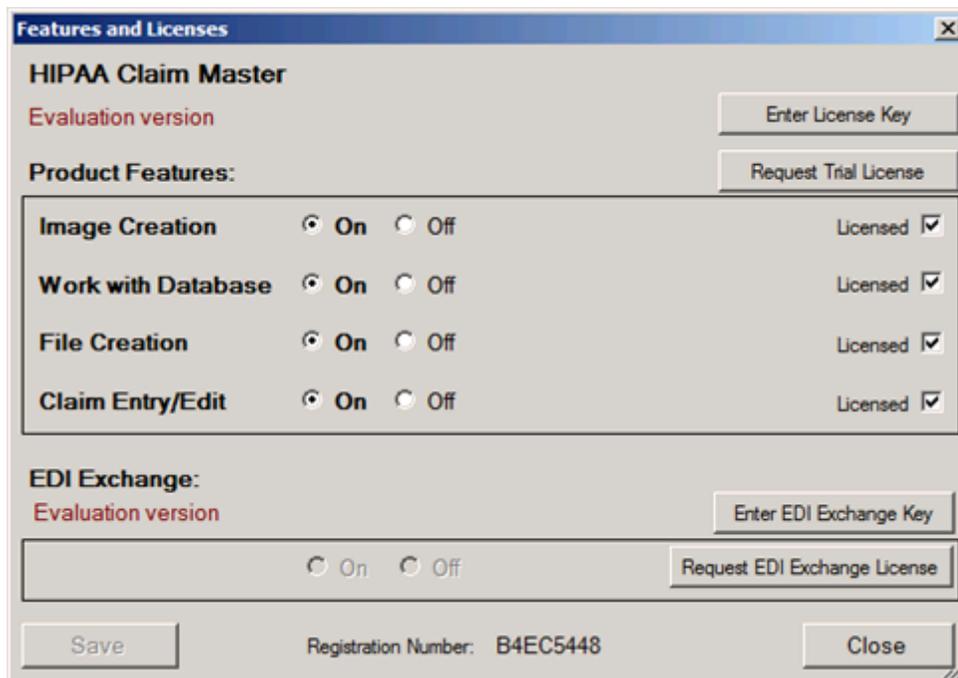


The "Ask for Trial Extension" button

Tip: You can see the registration number in the lower left hand corner. This number is needed for the registration as well as trial extensions. It is unique to your computer and hardware. You can hover over the registration number and copy it to the clipboard, so that you can easily paste it into the email. This avoids human error. (The letter "O" is never used, only zeros.) Hovering over the number will give you the following floating menu:



3. Clicking on the "Ask for Trial Extension" button will bring up the feature control window.

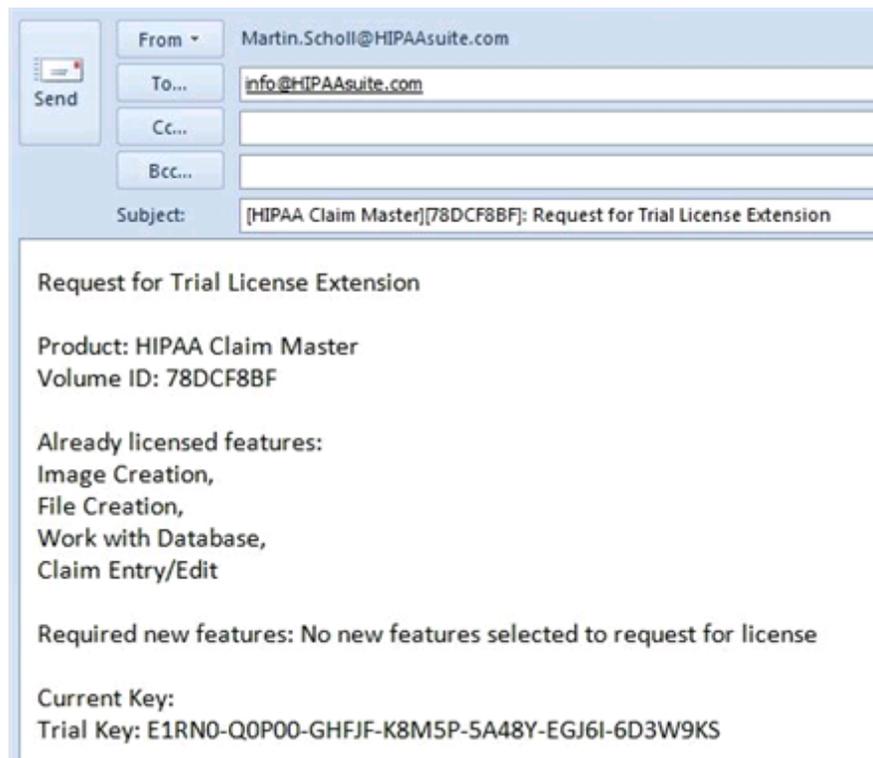


The "Features and Licenses" window

4. Select features you need and click on "Request Trial license."

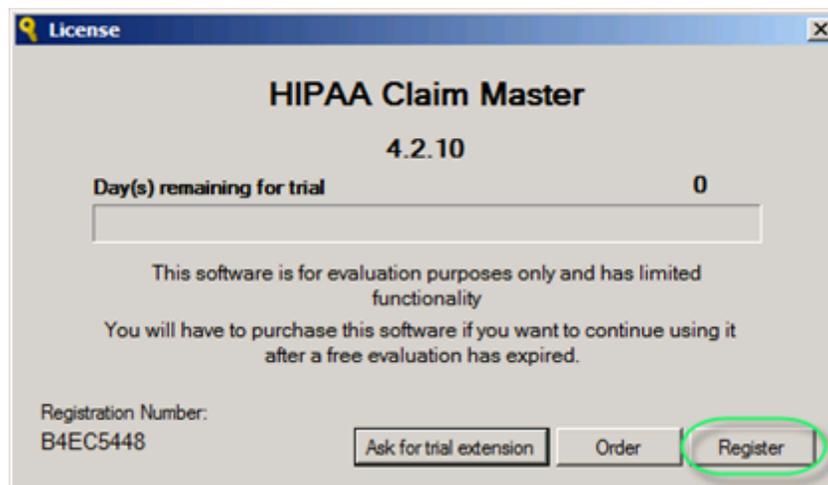
5. An email to HIPAAsuite will be created. The message will contain the registration number so we can then create the extension key. For example, this will open Outlook with the following data inserted:

- To: info@HIPAAsuite.com
- Subject: [HIPAA Claim Master][123456789]: Request for Trial License Extension
- Message: Volume ID: 123456789



The trial extension request email

6. The response from the support team should include an activation key.
7. Once you have the activation/unlock key, click "Register" on the program welcome window.



The "Register" button

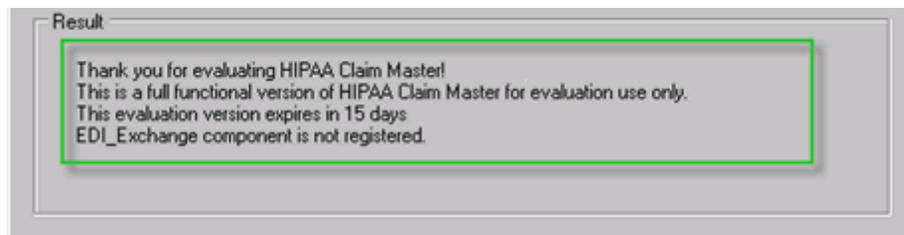
8. Paste the activation key into the field on the Registration Form and then click "Register."

Tip: The field is case-sensitive. The best thing to do is to copy and paste the License Key from our email.



The "Registration Form" window with entered license key and highlighted "Register" button

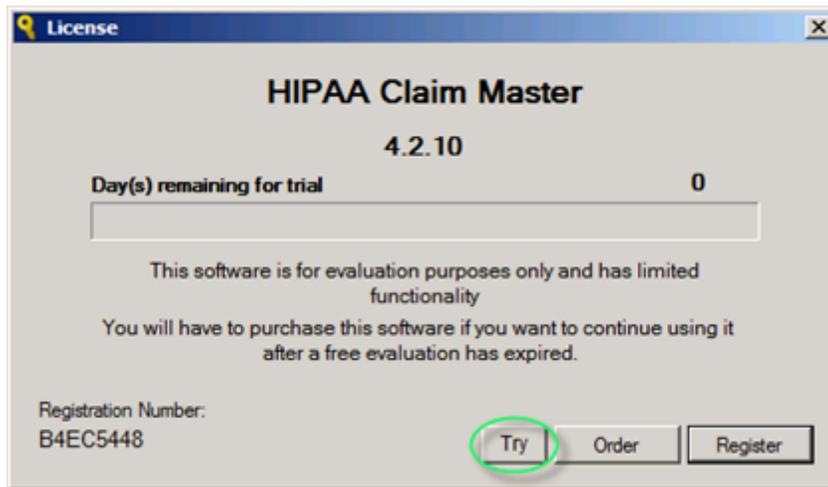
9. The success message will appear in the "Result" border. You will see what components have been registered.



The license activation result

Click "Close."

10. Once the license key has been entered, click "Try" on the welcome window to continue using the program for another 15 days.



The "Try" button

Once the product is registered with a permanent unlock code, future upgrades will find this key and install without user action.

2.10 Enabling Application Components

With version 4.1 we split up the HIPAA Claim Master into components to make the product more affordable and to allow the customers to license only the features that they need. The following components can be licensed individually:

- A base version that displays claims on the screen and prints them to paper.
- **Image file creation** – See [Creating Image Files from EDI](#).
- **Database integration** – See [Exporting the Data](#).
- **EDI creation** – See [Creating EDI Files](#).
- **Claim edit/entry** – See [Using the Claim Entry/Edit Function](#).

There is also the **EDI Exchange** module that works across all HIPAAsuite products and has a separate license so that it can be employed by multiple HIPAAsuite products without requiring to be licensed for each one.

Features and Licenses [X]

HIPAA Claim Master

Evaluation version [Enter License Key]

Product Features: [Request Trial License]

Image Creation	<input checked="" type="radio"/> On <input type="radio"/> Off	Licensed <input checked="" type="checkbox"/>
Work with Database	<input checked="" type="radio"/> On <input type="radio"/> Off	Licensed <input checked="" type="checkbox"/>
File Creation	<input checked="" type="radio"/> On <input type="radio"/> Off	Licensed <input checked="" type="checkbox"/>
Claim Entry/Edit	<input checked="" type="radio"/> On <input type="radio"/> Off	Licensed <input checked="" type="checkbox"/>

EDI Exchange:

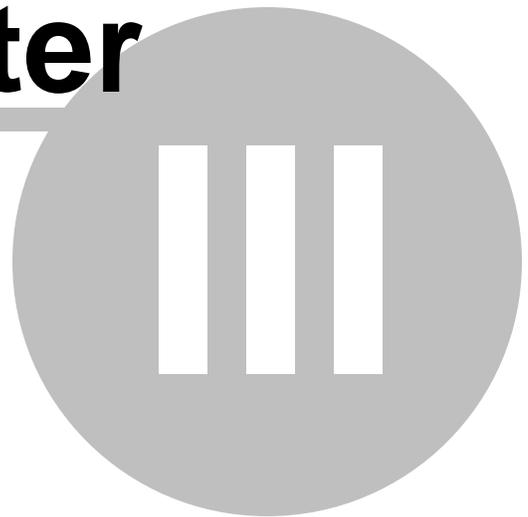
Evaluation version [Enter EDI Exchange Key]

On Off [Request EDI Exchange License]

[Save] Registration Number: B4EC5448 [Close]

The "Features and Licenses" window

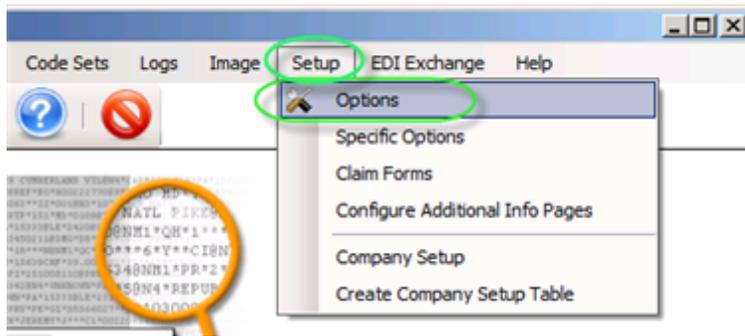
Chapter



3 Configuring the Application

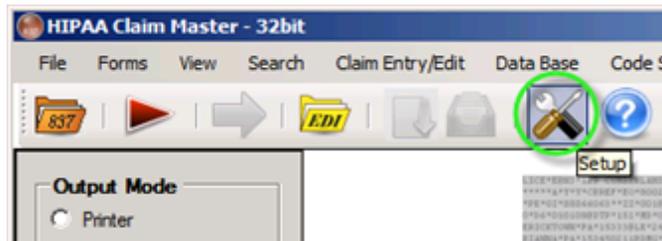
3.1 Configuring Program Options

1. To configure the program options, select *Setup* ▶ *Options* in the main menu.



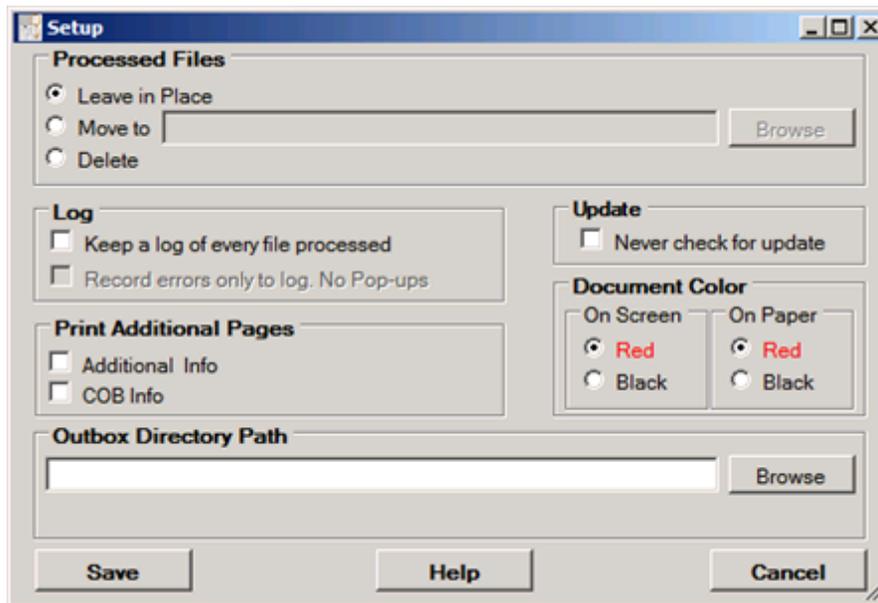
The "Options" menu

Alternatively, you can click on the "Setup" button located on the main toolbar.



The "Setup" button

2. The following screen will appear.



The "Setup" window

You can configure the options:

- **Processed Files**
- **Log**
- **Update**
- **Print Additional Pages**
- **Document Color**
- **Outbox Directory Path**

Detailed descriptions to follow.

3. Once you have finished editing the options, click "Save."

Processed Files

In order to prevent the duplicate processing of files, the HIPAA Claim Master can move or delete files after they have been processed. Since the HIPAA Claim Master program can be launched by the Window's scheduler (see [Running the Application via Scheduler](#)), it is important to either move or delete processed files, so that they will not be picked up again. You have the choice to either do nothing, move the processed files to a specific folder or delete the files:

- **Leave in Place** – Select this option to do nothing with processed files.

Tip: This option is not recommended.

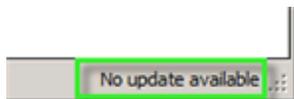
- **Move** – Select this option to move the processed files to a specified folder.
- **Delete** – Select this option to delete the processed files.

Log

- **Keep a log of every file processed** – Determines whether the log entry should be created for each file processed. By default, the log is only active in command line use (without human interaction.) If you want a log entry created for each manual processing, except viewing on screen, place a check mark here. See also: [Accessing Daily File Log](#).

Update

- **Never Check for Update** – To check the HIPAA Claim Master version update, uncheck the "Never Check for Update" check box. If the check box is unchecked, the application will automatically look for the new version available and a pop up window displayed. On selecting the new version, the application will get updated to the current version available. The lower right corner of the application window displays information including alerts on version updates.



The message on the bottom bar

Print Additional Pages

There may be additional pages with data that have no representation on the standard forms.

This option determines whether the additional pages with either [COB information](#) or segments with [additional information](#) that have no representation on the standard forms should be printed on either paper or image files. This option helps you save paper. If you only need the main claim form, there is no need to print 2 extra sheets.

If you only need the main claim form, there is no need to print these two (2) extra sheets, in this case, leave the following options unchecked to save your paper.

- **Additional Info** – There may be additional pages with segments containing additional information that have no representation on the standard forms. If this option is checked, then these additional pages will be printed on paper and image files.
- **COB Info** – There may be additional pages containing COB information that have no representation on the standard forms. If this option is checked, then the additional pages will be printed on paper and image files.

Document Color

This section of the setup window is concerned with the output colors. You have the choice to render the form in red or in black. The form background can be set to red to make the form visually more appealing and to allow the use of a color filter for OCR software. The color can be selected for the following media:

Note: The data is always black.

- **On Screen** – Choose a color for the form when displaying claims on screen.
- **On Paper** – Choose a color for the form when printing claims.

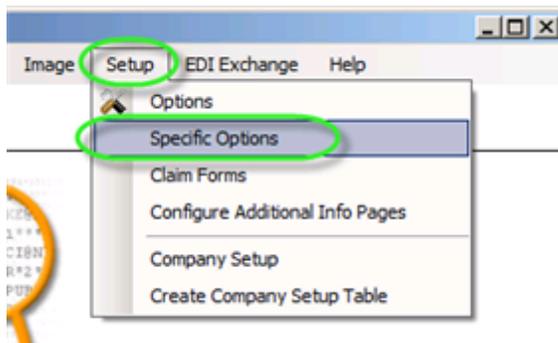
Outbox Directory Path

Define the destination folder where the result files will be stored.

3.2 Configuring Specific Options

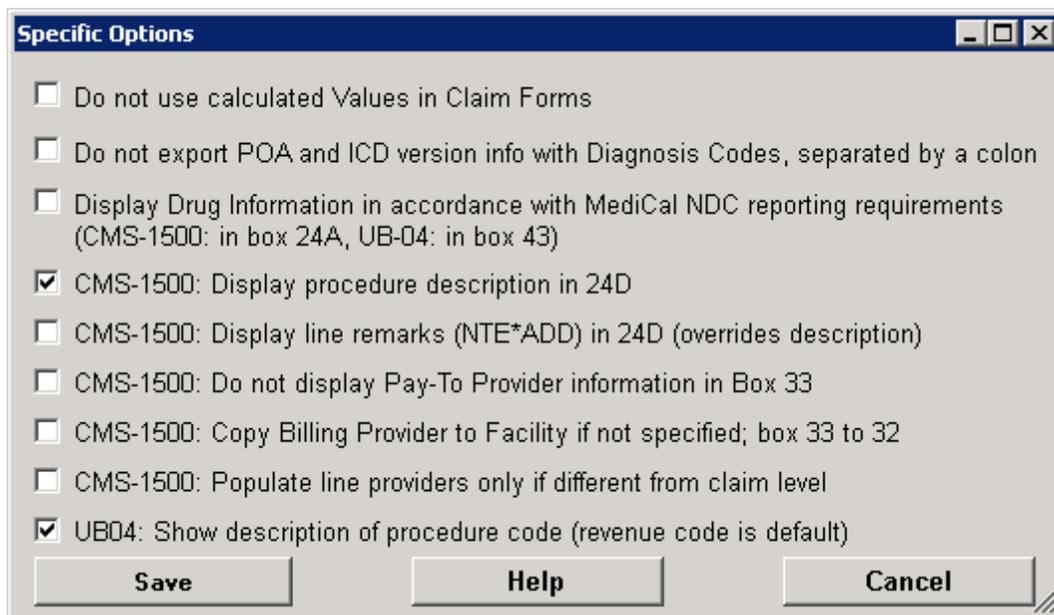
You can set regional and other user-specific options that do not apply to all users. Follow the instructions below.

1. To configure the program options, select *Setup* ▶ *Specific Options* in the main menu.



The "Specific Options" menu

2. The following screen will appear.



The "Specific Options" window

3. You can enable/disable the following options:

- **Do not use calculated Values in Claim Forms** – The HIPAA Claim Master calculates the Open Amount from charges minus previous payments and patient paid amounts and some other values. If you prefer not to do this, check this option.
- **Export POA and ICD Version information with Diagnosis Codes, separated by a colon** – You can opt out of writing POA and ICD version information to the database. The default is now to write this information to the database. This option changed. The default behavior is required to create valid claims.

Present on Admission (POA) information is now often required by insurers to exclude "hospital infections" from coverage. These are diseases that are acquired by the patient after admission to the hospital and often cause high costs and even death. In

the SQL export the POA codes are colon separated from the diagnosis codes. If this causes problems, you can deselect this here. With the introduction of ICD-10 it is important to save the version of ICD codes that are in the claim to the database. Since it is intrinsically linked to the Diagnosis code we decided against creating another set of fields. It would mean creating another 80 or so fields for POA and ICD version. Instead we opted to concatenate the ICD version qualifier and the POA indicator with Diagnosis Code qualifier separated by a colon (:)

- Principal Diagnosis Code has POA and ICD version. Example:

Z77.22:Y:ABK with the diagnosis code in first position, followed by the POA indicator 'Y' and the ICD-10 qualifier

- Other Diagnosis Code has POA and ICD version. same as above

Example: *J151::ABF* Here the POA code is committed

- E-Codes also have both POA and ICD version, same as above

Example *E8600:Y:ABN*

- Admit Diagnosis Code has no POA info. By definition is the diagnosis code under which a patient was admitted present at admission.

Example *L01.03:ABJ* Only the ICD version qualifier is added to the diagnosis code

- Reason for Visit codes have no POA info. Same as Admit diagnosis

This way of connecting diagnosis information with POA and ICD codes is bi-directional, meaning if you create 837 files from database, then the above example is the way to do it. Internally HIPAA Claim Master will parse out the information and assemble the respective HI segment within all applicable rules.

- **Display Drug Information with accordance with Medical NDC reporting requirements (CMS-1500: in box 24A, UB-04: in box 43) –** The Medi-Cal NDC reporting requirement is concerned with the transmission of drug data that belongs to a specific procedure. In the 837, this information is in the LIN and CPT segments of the 2400 loop. Here the display of NDC in the CMS-1500:

	A. DATE(S) OF SERVICE FROM	TO	B. PLACE	C. EMG	D. PROCEDURES, SERVICES OR SUPPLY CPT/HCPCS	MODIFIER	E. DIAGNOSIS POINTER	F. \$ CHARGES	G. DAYS UNITS	H. EPS	I. ID. QUAL	J. RENDERING PROVIDER ID. #
1	N400075800120	05/01/2009	05/01/2009	11	ML000020000	J9170	1,2	\$1,650.00	3			1234567890
2	N400641244045	05/01/2009	05/01/2009	11	UN001000000	J1644	1,2	\$5.00	1			1234567890
3	N4	05/01/2009	05/01/2009	11	ML000250000	J7050	1,2	\$30.00	3			1234567890
4	N400088120632	05/01/2009	05/01/2009	11	GR000010000	J1260	3	\$200.00	10			1234567890

The NDC code is in the upper half of field 24 A. The quantity in 24 D.

The format for the quantity is a full 10-digit number. The 10 digits consist of seven digits for the whole number, followed by the three-digit decimal portion of the number. In the example above line 1 indicates 20 milliliter, line 2 1000 units, line 3 250 milliliter and line 4 10 grams.

	42 REV.CD.	43 DESCRIPTION	44 HCPCS/RATE/NIPPS CODE	45 SERV. DATE	46 SERV. UNITS	47 TOTAL CHARGES	48 NO. COVERED CHARGES	49
1	0250	N400409128331UN000001000	J1170	04/07/09	1	\$25.00		
2	0250	N400338802072ML000011000	J3490	04/07/09	11	\$445.50		
3	0250	N400008092351UN000001000	S0164	04/07/09	1	\$40.50		
4	0250	N400173044202UN000001000	X7050	04/07/09	1	\$150.70		

The NDC information in the UB04 form

In the UB04 the NDC information goes into the "Description" field 43 as one string with quantities in the same format as in the CMS-1500.

- **CMS-1500: Display procedure description in 24D** – See [Displaying the Revenue or Procedure Code Descriptions](#).
- **CMS-1500: Display line remarks (NTE*ADD) in 24D (overrides description)**
- **CMS-1500: Do not display Pay-To Provider information in Box 33** - When a Pay-To provider is specified in loop 2010AB, the address information is usually displayed in box 33.
- **CMS-1500: Copy Billing Provider to Facility if not specified; box 33 to 32**
- **CMS-1500: Populate line providers only if different from claim level** - Display cleaned up in EDI files where the rendering provider is repeated on every line.
- **UB 04: Show description of Procedure Code (Revenue code is default)**

Field locator 43 will be populated with the procedure code description in case such code is in SV2 segment

- **Use Facsimile Mode in UB 04 and CMS-1500** - For the purists, this mode strives to make the claim look like it has been filled with a Dot-Matrix printer. More [here](#).

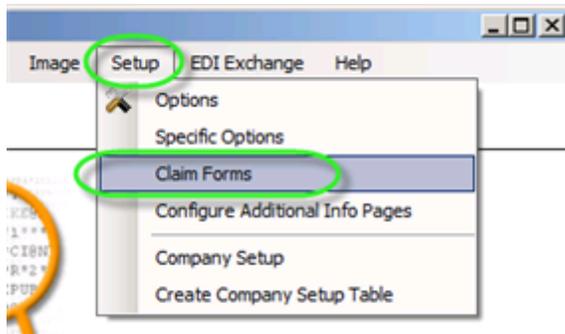
4. Once you have finished editing the options, click "Save."

3.3 Configuring Claim Forms

The introduction of the National Provider ID (NPI) brought new claim forms. All three types of claims now have a form that contains the NPI. The HCFA-1500 became the CMS-1500, the UB92 was replaced with the UB04 and the ADA 2006 replaced the earlier versions.

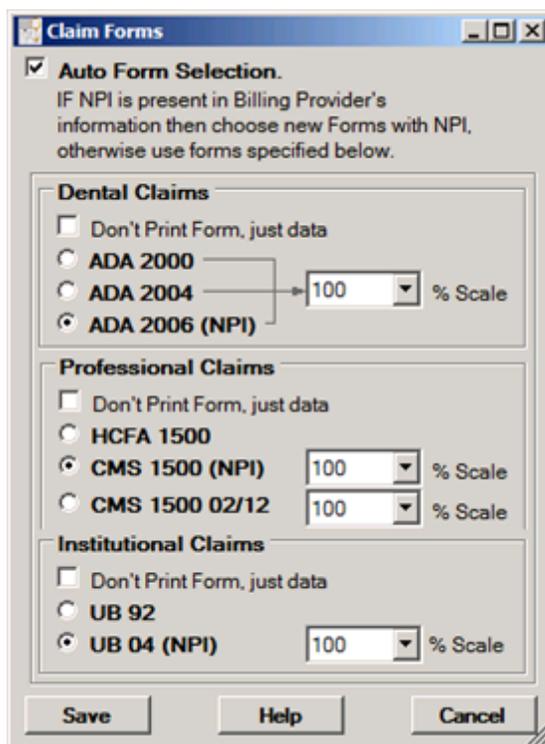
You can choose which forms to use and it will give you the choice to either print the form and the data or to suppress the form, so that only the data appears. This is important if you want to print on pre-printed forms or use OCR to read the data.

1. To configure the program options, select *Setup* ▶ *Claim Forms* in the main menu.



The "Claim Forms" menu

2. The following screen will appear.



The "Claim Forms" window

3. You can configure the following options:

- Auto Form Selection

- **Dental Claims**
- **Professional Claims**
- **Institutional Claims**

They are described further below.

4. Once you have finished editing the options, click "Save."

Auto Form Selection

Check this option to automatically select the forms. The system selects the forms based on the following logics. If NPI is present in Billing Provider's information, then choose new Forms with NPI, otherwise use forms pecified below.

Tip: We recommend to activate this setting. Older Pre-NPI claim files will then be displayed in the old forms while newer files with NPI will be rendered in the new forms.

Dental, Professional, Institutional Claims

Select the forms for the following claims:

- **Dental Claims**
- **Professional Claims**
- **Institutional Claims**

Specify the following settings:

- **Scale** – The UB04 form is bigger than the other claim forms. It is 12" x 9" instead of the customary 8 1/2" x 11" and it usually does not print in its entirety due to the print margins. For that reason it is possible to scale the form down to a printable size.
- **Don't print form, just data** – Additionally, you have the choice to print just the data and suppress the form.
Tip: This feature comes in handy when you want to print to pre-printed forms or if you have an OCR system that does not need the forms.

3.4 Using the EDI Viewer and Editor

EDI files are often hard to read, especially if they have no carriage returns and line feeds to put each segment on a line of its own.

The EDI Editor available in the HIPAA Claim Master replaces all element separators with a star "*", all sub element separators with a colon ":" and all segment separators with a tilde "~" even if the original file uses different delimiters.

Tip: Changing EDI files can be a tricky undertaking. You should be experienced in the format of the 837 and understand that an 837 claim file could be a legal document that should not be altered without the consent of the originator.

To Open EDI Editor

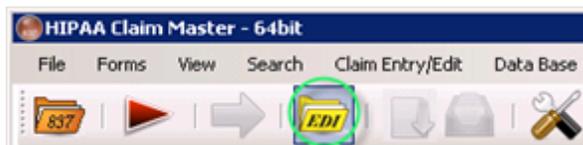
To access the EDI Editor, select the "EDI Editor" option under "View" menu item.



The "EDI Editor" menu

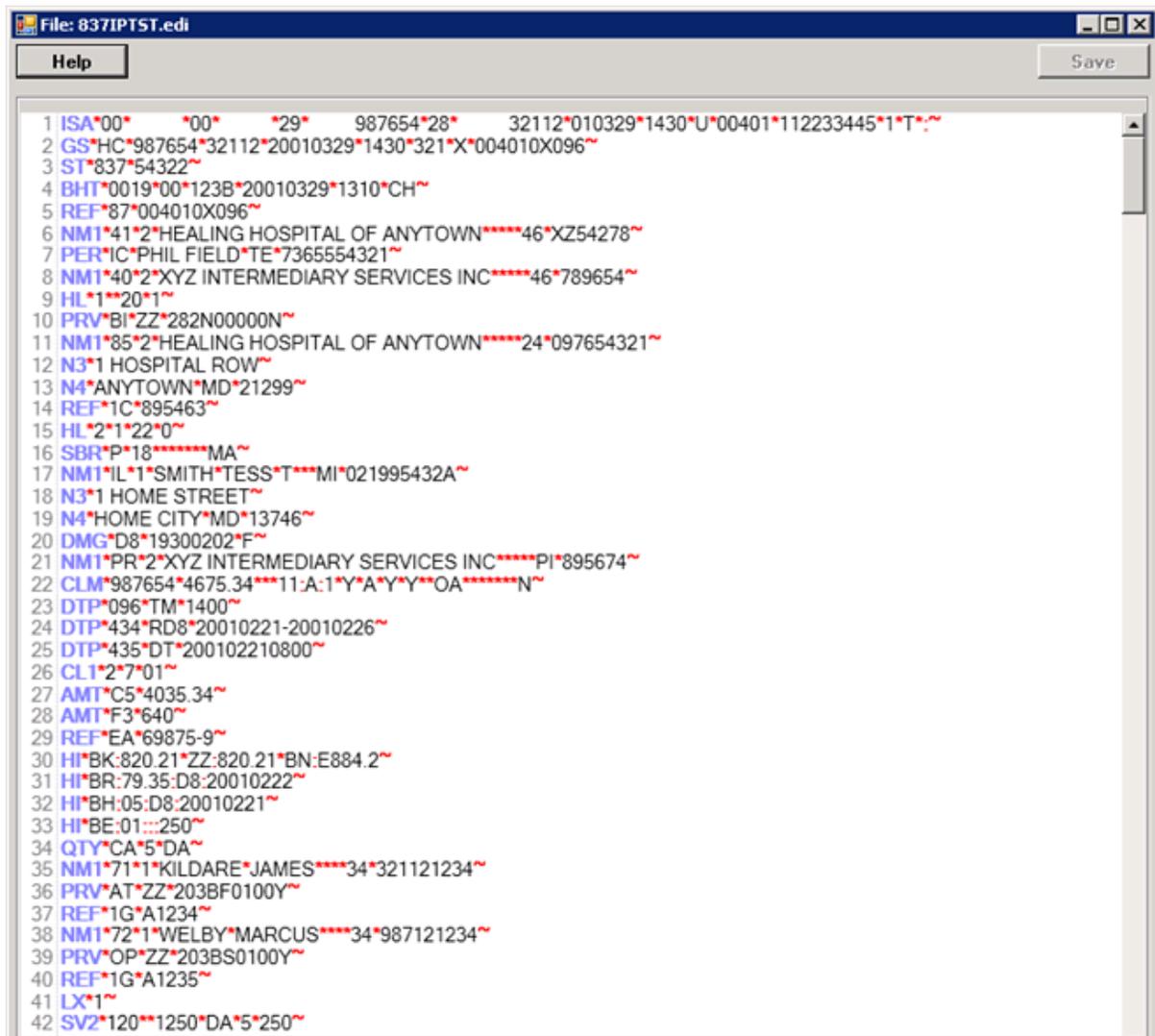
Alternatively, you can click on the "EDI" icon on the toolbar to invoke the EDI Editor.

This option is only enabled when you have [opened an EDI file](#).



The "EDI Editor" button

The following screen will appear:

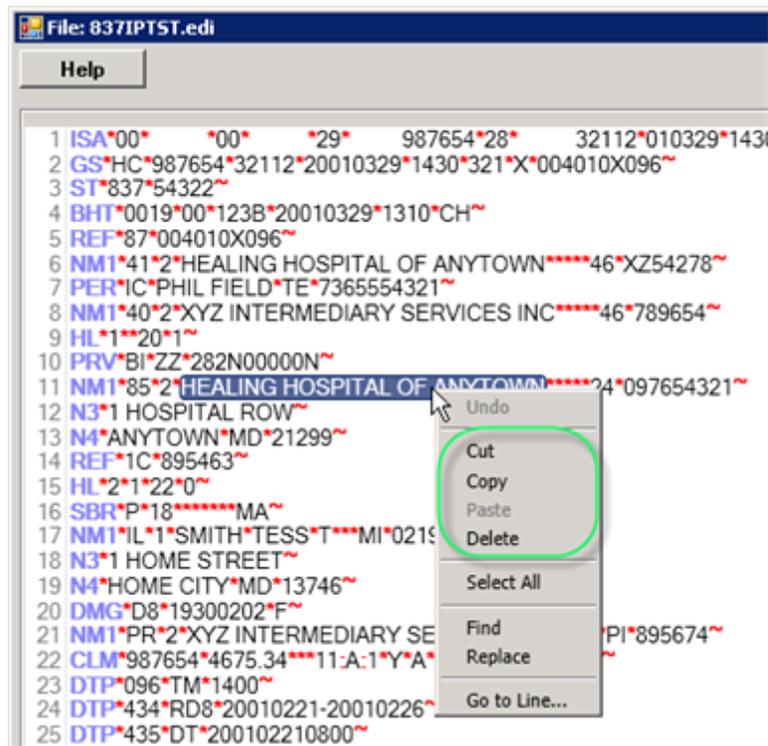
The image shows a screenshot of a software window titled "File: 837IPTST.edi". The window has a "Help" button on the left and a "Save" button on the right. The main area contains a list of 42 EDI segments, each on a new line and numbered from 1 to 42. The segments are formatted as EDI syntax, including headers like ISA, GS, ST, BHT, REF, NMT, PER, NM1, HL, PRV, N3, N4, DMG, DTP, CL1, AMT, REF, HI, QTY, and SV2, followed by their respective data values and control characters like tilde (~) and asterisks (*).

```
1 ISA*00*      *00*      *29*      987654*28*      32112*010329*1430*U*00401*112233445*1*T::~~
2 GS*HC*987654*32112*20010329*1430*321*X*004010X096~
3 ST*837*54322~
4 BHT*0019*00*123B*20010329*1310*CH~
5 REF*87*004010X096~
6 NMT*41*2*HEALING HOSPITAL OF ANYTOWN*****46*XZ54278~
7 PER*IC*PHIL FIELD*TE*7365554321~
8 NM1*40*2*XYZ INTERMEDIARY SERVICES INC*****46*789654~
9 HL*1**20*1~
10 PRV*BI*ZZ*282N00000N~
11 NM1*85*2*HEALING HOSPITAL OF ANYTOWN*****24*097654321~
12 N3*1 HOSPITAL ROW~
13 N4*ANYTOWN*MD*21299~
14 REF*1C*895463~
15 HL*2*1*22*0~
16 SBR*P*18*****MA~
17 NM1*IL*1*SMITH*TESS*T***MI*021995432A~
18 N3*1 HOME STREET~
19 N4*HOME CITY*MD*13746~
20 DMG*D8*19300202*F~
21 NMT*PR*2*XYZ INTERMEDIARY SERVICES INC*****PI*895674~
22 CLM*987654*4675.34***11:A:1*Y*A*Y**OA*****N~
23 DTP*096*TM*1400~
24 DTP*434*RD8*20010221-20010226~
25 DTP*435*DT*200102210800~
26 CL1*2*7*01~
27 AMT*C5*4035.34~
28 AMT*F3*640~
29 REF*EA*69875-9~
30 HI*BK:820.21*ZZ:820.21*BN:E884.2~
31 HI*BR:79.35:D8:20010222~
32 HI*BH:05:D8:20010221~
33 HI*BE:01:::250~
34 QTY*CA*5*DA~
35 NMT*71*1*KILDARE*JAMES****34*321121234~
36 PRV*AT*ZZ*203BF0100Y~
37 REF*1G*A1234~
38 NM1*72*1*WELBY*MARCUS****34*987121234~
39 PRV*OP*ZZ*203BS0100Y~
40 REF*1G*A1235~
41 LX*1~
42 SV2*120**1250*DA*5*250~
```

The EDI Editor window

To Cut, Copy, Paste

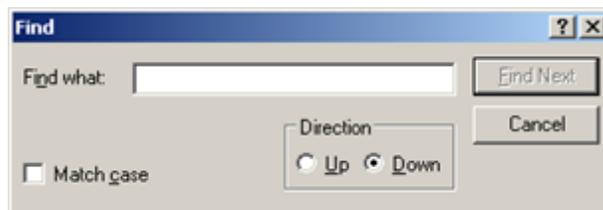
When you right-click anywhere in the text, a context menu will appear with typical text edit options such as cut, copy and paste.



Context menu

To Find and Replace

The "Find" and "Replace" utilities work as in a common text editor. To use them, right-click somewhere in the document and then from the displayed context menu select the "Find" or "Replace" command. Alternatively, you can press `ctrl F` simultaneously on your keyboard to bring the text search utility.



The "Find" pop-up

Pressing `ctrl H` simultaneously will bring up the Replace utility.



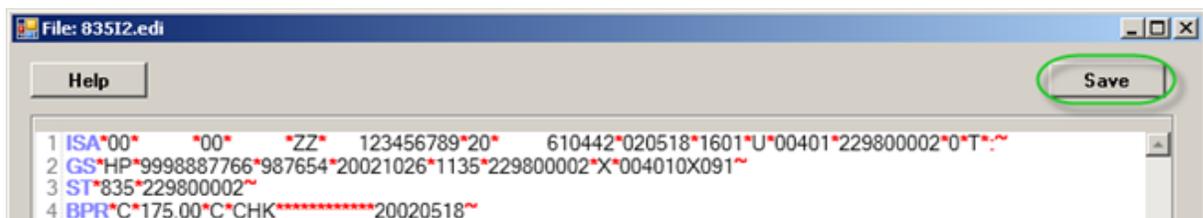
The "Replace" popup

To Save the Changes

As long as you don't save, there will be no changes to the file.

If you make changes to the file, the "Save" button becomes enabled and you can save any changes. The HIPAA Claim Master will further work with those changed files, so you don't have to open the file again.

Notice: Changing EDI files can be a tricky undertaking. You should be experienced in the format of the 835 and understand that an 835 claim file could be a legal document that should not be altered without the consent of the originator.



The "Save" button

3.5 Selecting a Code Set

EDI relies on transaction code sets. These codes represent longer explanations and descriptions.

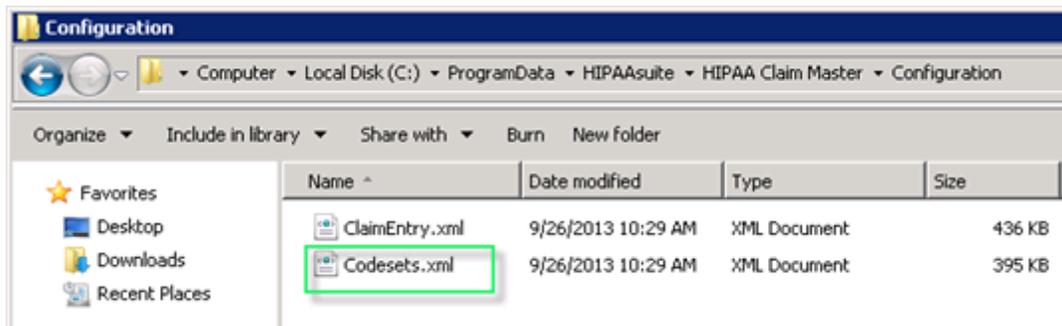
At the onset of EDI one of the main design ideas was to make the EDI files as short as

possible and to allow different languages to use the same descriptions by reducing long verbose explanations to 2-3 byte long codes. Computer storage was incredibly expensive and making any file larger than it absolutely had to was considered wasteful.

The code sets are stored in the file `codesets.xml` in the `Configuration` subdirectory:

Windows XP: `C:\Documents and Settings\All Users\Application Data\HIPAAsuite\HIPAA Claim Master\Configuration\`

Windows 7, Vista: `C:\ProgramData\HIPAAsuite\HIPAA Claim Master\Configuration\`

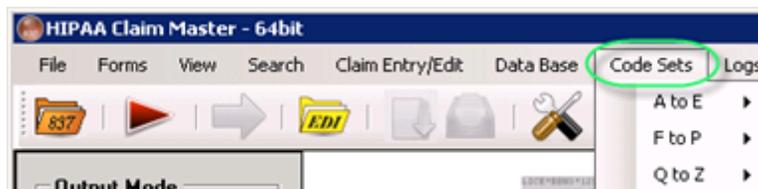


The Configuration folder in Windows 7

Tip: The folder is hidden.

To view/edit transaction code sets in HIPAA Claim Master, follow the instructions below.

1. Go to the "Code Sets" item in the main menu.



The "Code Sets" menu

2. From the sub-menu, select a code set name. The following menu items are available via the "Code Sets" menu:

A to E

- Adjustment Reason
- Adjustment Reason Group
- Admission Source

- Admission Type
- Amount Qualifier Codes
- Attachment Report Type
- Certification Type
- Communication Codes
- DTP Codes
- Delay Reason Codes

F to P

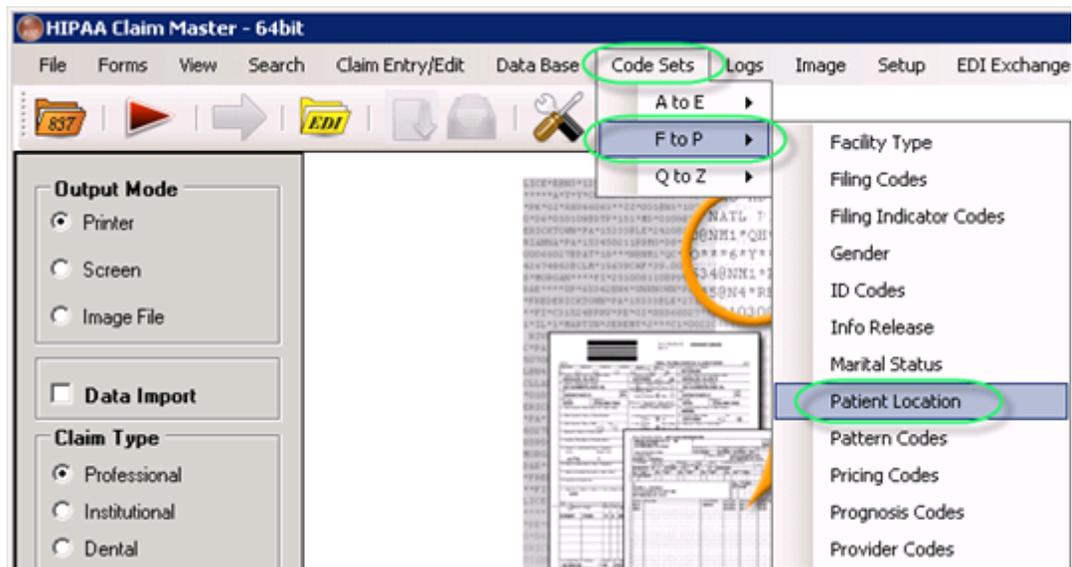
- Facility Type
- Filing Codes
- Filing Indicator Codes
- Gender
- ID Codes
- Info Release
- Marital Status
- Patient Location
- Pattern Codes
- Pricing Codes
- Prognosis Codes
- Provider Codes

Q to Z

- REF Codes
- Relationship
- REFCODE
- Remark Codes
- Signature Source

- Subluxation Codes
- Tooth Numbering
- Unit Codes
- Yes/No Codes

The following screen-shot illustrates how to select the "Patient Location" code set for display.



The "Patient Location" sub-menu

3. The Code Set configuration window will be displayed.

You can add, delete or edit codes. Read further how.

Below you can see this window for the Patient Location.

The window provides the following information:

- **Number of records in codeset** – The total number of codes. For example, for Payment Location it is equal to 14.
- **Codes and their corresponding descriptions.** The codes are indicated in the source EDI files. The descriptions are displayed instead of codes in the result easy-to-read files. For example, for Payment Location, there are the following codes and corresponding descriptions:

- A – Acute Care Facility
- B – Boarding Home
- C – Hospice
- D – Intermediate Care Facility
- etc.

Code	Description
A	Acute Care Facility
B	Boarding Home
C	Hospice
D	Intermediate Care Facility
E	Long_term
F	Not Specified
G	Nursing Home
H	Sub-acute care facility
L	Other Location
M	ARehabilitation Facility
O	OutPatient Facility
R	Residential Treatment Facility
S	Skilled Nursing Home
T	Rest Home
*	

The "Patient Location" code set configuration window

4. Click on the "Save" button to save the changes.

Adding a New Row

To add, click on the last row. Now you can see the indicator in the first grey field.

O	OutPatient Facility
R	Residential Treatment Facility
S	Skilled Nursing Home
T	Rest Home
▶*	

The indicator

Click the field again to enter to the Edit mode. Now you can enter the text.

Note: Only after you leave this row, the changes will take place.

Deleting a Row

To delete, click the left margin. This action will highlight the whole row.

O	OutPatient Facility
R	Residential Treatment Facility
S	Skilled Nursing Home
▶	T Rest Home
*	

Highlighting a row by clicking the left margin

Now press the <DELETE> key on your keyboard.

Editing the Row

To edit, click two times the cell you want to change. Make your changes. Clicking or moving outside this cell overwrites the cell contents with the new value. To make these changes permanent, click the 'Save' button.

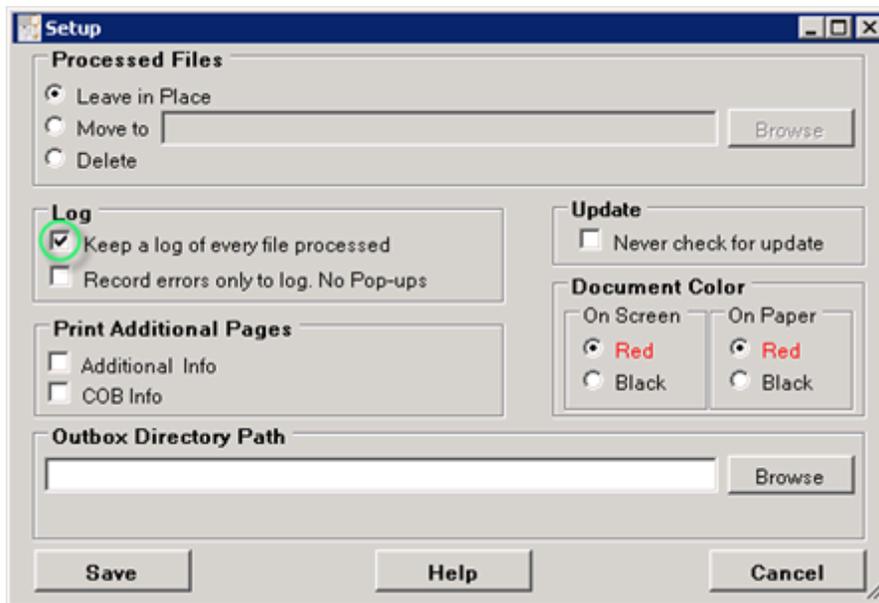
Notice: Don't forget to save the changes by pressing the 'Save' button.

3.6 Accessing Daily File Log

The HIPAA Claim Master can keep a log entry for each file that gets processed. This feature is especially helpful when the HIPAA Claim Master is run through automated processes. The log is essential in tracking errors and documentation.

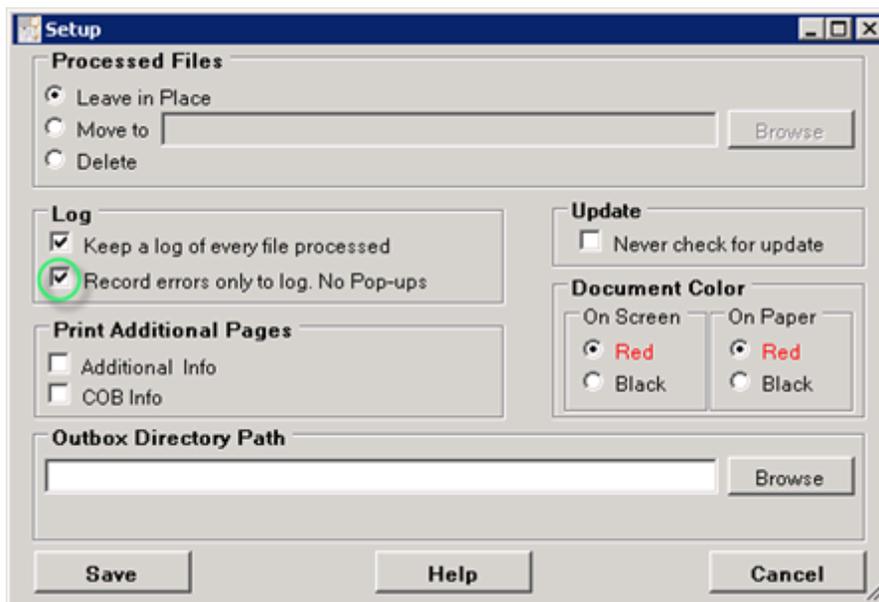
Enabling Logging

To enable logs, check the "Keep a log of every file processed" checkbox under *Setup* ▶ *Options* ▶ *Log*.



The "Keep a log of every file processed" check-box

To record the errors only in the log, check the "Record errors only in log" box displayed in the "Log" section.

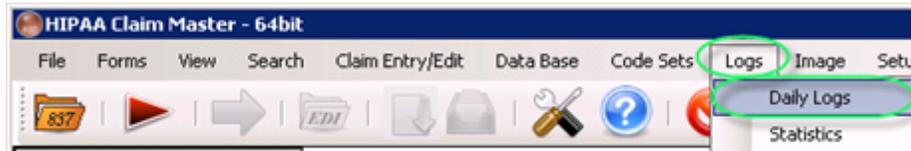


The "Record errors only in log" check-box

Accessing Logs

Follow the instructions below to access the log files for your HIPAA Claim Master.

Click *Logs* ▶ *Daily Logs* in the main menu.



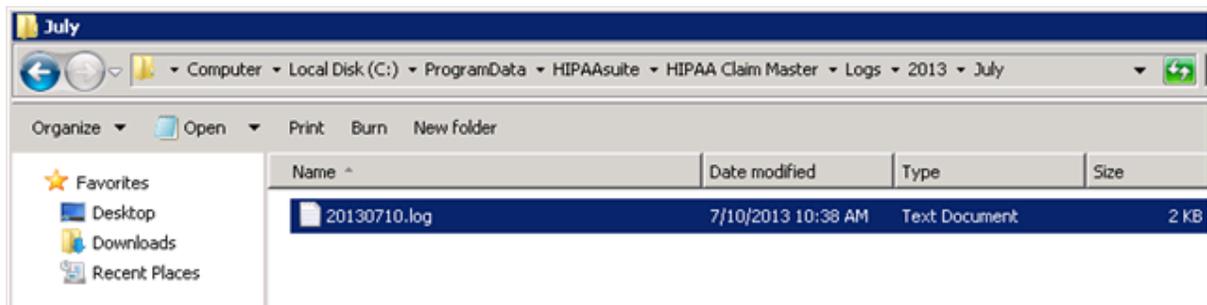
The "Log" menu

If there is a log file for the current day, it will be displayed. If not, the page will be blank. You can navigate to each date with the calendar control on top of the page.

Log files are simple text files, one for each day that are stored in date hashed folders. Alternatively, access the `Logs` folder in the following directory:

Windows XP: `C:\Documents and Settings\All Users\Application Data\HIPAAsuite\HIPAA Claim Master\Logs`

Windows 7, Vista: `C:\ProgramData\HIPAAsuite\HIPAA Claim Master\Logs`



The directory structure for the daily logs

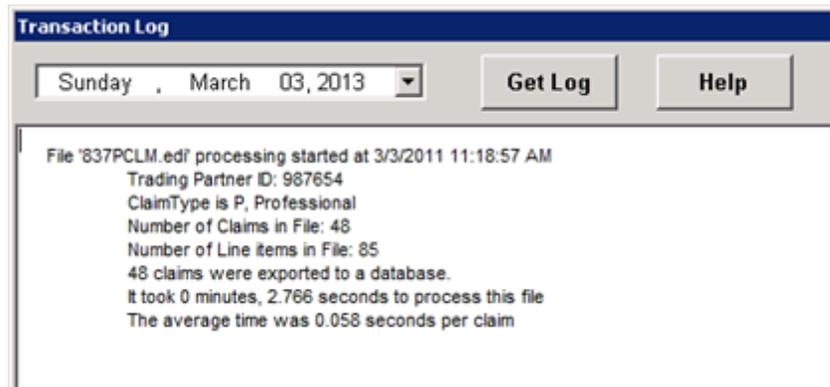
Tip: The folder is hidden by default. But if you type in the address in the top bar of the Windows explorer, you will find that it exists. Or you can change the explorer settings to show hidden files and directories.

Tip: The `ProgramData` folder contains all your application data, like program settings, user data, etc, for all of the installed programs on the computer. The `ProgramData` folder holds the same files that were contained in the `\Documents and Settings\All Users` and `\Documents and Settings\All Users\Application Data` folders in XP.

For each day that the HIPAA Claim Master is used (If logs are enabled), there will be a log

file which carries the name in the filename.

Within the directory, find a folder for the year you are looking log for. Then find a sub-folder for the necessary month with files for each day. Double-click a file which has the name equal to the necessary date. The log will look like this:

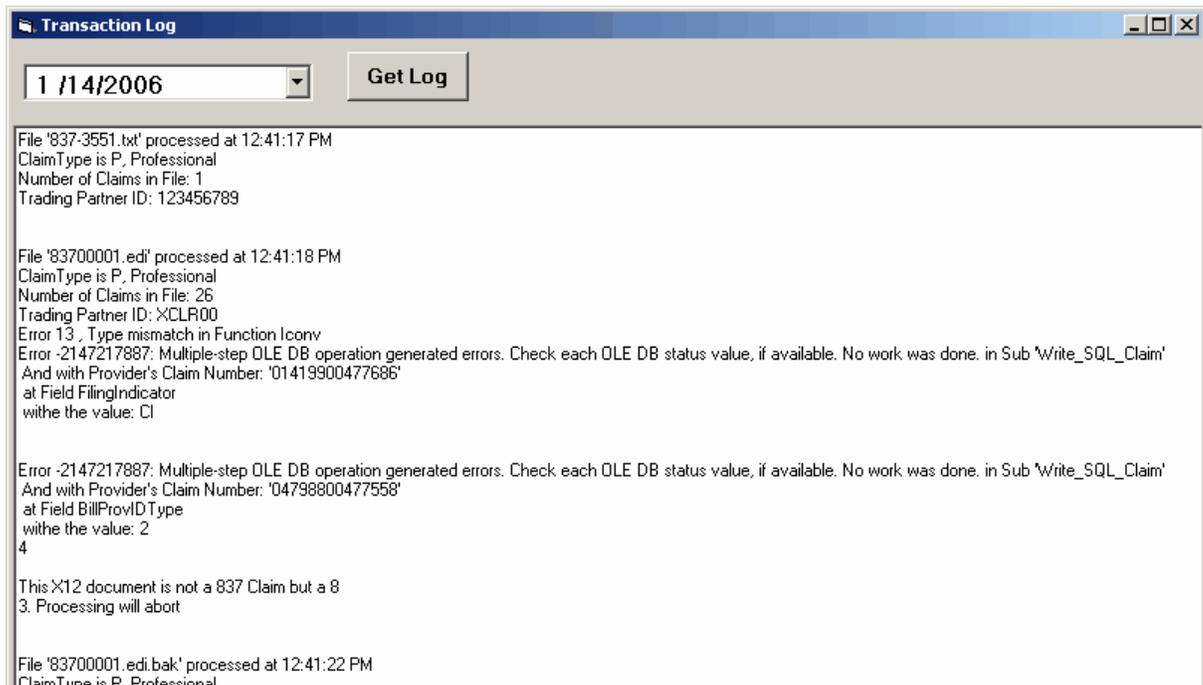


The daily file log

This entry is for a problem free file. Out of this entry you will get:

- Process start time
- Filename
- Trading partner ID
- Claim type
- Number of claims
- Number of lines
- Number of exports/image files/ prints
- Processing time
- Average time per claim

An entry with errors might look like the following:



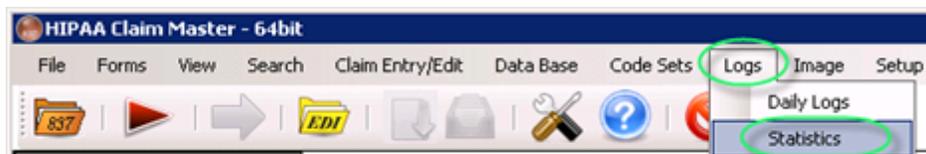
The second entry shows a database export problem. The message contains details that allow tracking of the problem. In this case a carriage return was in the file between the 2 and the 4.

3.7 Accessing Statistics

The HIPAA Claim Master stores the amount of claims processed per month. These usage statistics are needed for a per-claim licensing model.

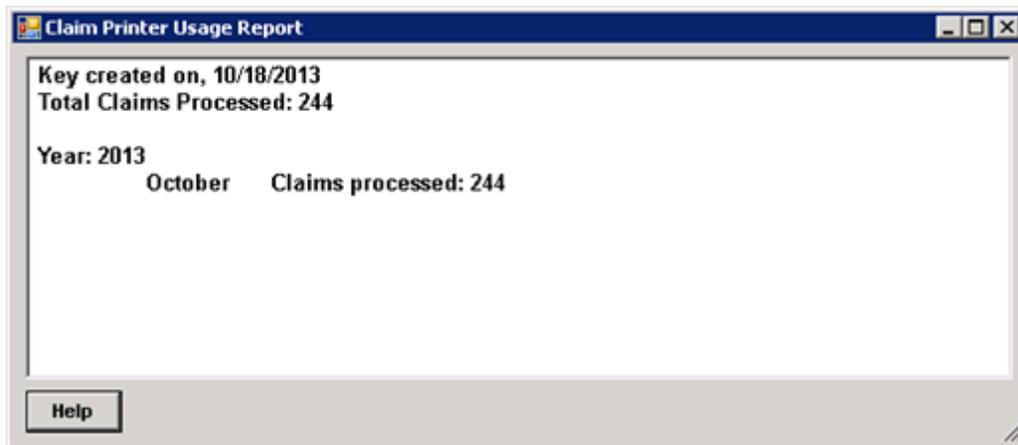
Follow the instructions below to access the statistics in your HIPAA Claim Master.

Click *Logs* ▶ *Statistics* in the main menu.



The "Statistics" menu

The following screen will display.



The "Claim Printer Usage Report" window

The screen shows the number of claims processed in total and by year and month. The data is stored as an encrypted key in the registry.

Clicking the "Send Usage Report for Billing Purposes" button creates an email that contains this report and sends it to HIPAAsuite. The per-claim licensing model depends on this report for billing purposes. If you have the unlimited Licensing Scheme, you do not need to worry about this function.

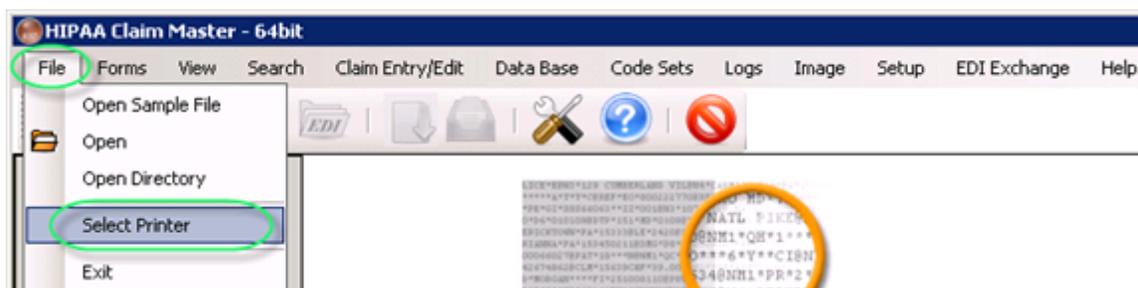
3.8 Selecting a Printer

If you want to print to a printer other than the computer's defined default printer, you will need to select the desired printer.

Note: The HIPAA Claim Master Version 4.0 onwards does not need anymore third party software for the creation of image files. Older versions of the HIPAA Claim Master needed to have the [Peernet Tiff Image Printer](#) or [PDF Image Printer](#) driver installed.

Follow the instructions below.

Go to *File* ▶ *Select Printer* in the main menu.



The "Printer" window will appear. Select the necessary printer.



Selecting the Printer for output on paper

Though you do not have any documents in your print queue, click on "Print" to select this printer for future output.

Chapter

IV

4 Creating Image Files and PDFs

4.1 Creating Image Files from EDI

The greatest disadvantage of electronic claims is that they are near impossible to understand in raw form. Storing an image associated with a claim makes a lot of sense for the most HIPAA Claim Master's users. If you have an investment in a claim imaging system, you can now create images from EDI files and treat them in your imaging system just like scanned images from paper claims. If your staff can understand a claim on a UB04 in a second, why not make it available in this form? Having an image file makes it possible for case workers to call up an image of the claim in seconds.

The HIPAA Claim Master can also be used to create image files and PDFs of the EDI claims. Image file creation is a component that can be licensed separately.

The HIPAA Claim Master displays the EDI document in a page that makes it easy to read the 837 transactions. The following formats are supported:

- TIFF
- PDF – The PDF files have searchable meta information in them.

Note: Earlier versions of the HIPAA Claim Master used a third party image printer driver. This is no longer necessary. The HIPAA Claim Master now has PDF and TIFF capabilities built-in.

Certain parameters of the formats are adjustable.

This is the same information that is in the [image index files](#). This small subset of the claim data plus the image path and filename serves most image libraries well.

Image filename is also exported to the [claim export](#). Here, we can also associate an image filename with the more complete claim data.

You can create image files containing the data that you see in Screen mode. You can create image files manually or automatically using the command line arguments and Windows Scheduler. Read more in:

- [Using Command Line Arguments \(CLI\)](#)
- [Running the Application via Scheduler](#)

Before creating image files, make sure all settings are defined correctly. Read more in:

- [Doing Image Setup](#)

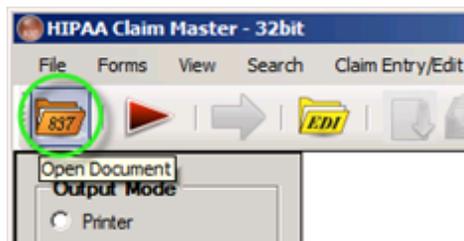
- [Configuring Image Index](#)
- [Creating Image Index Table](#)

How to Create Image Files

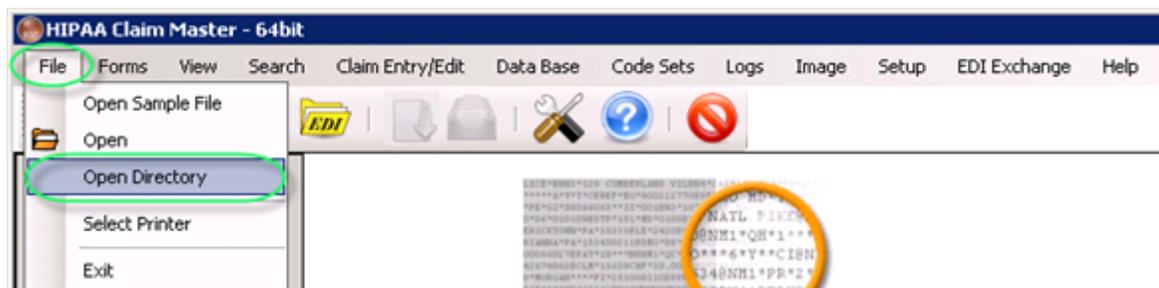
Follow the instructions below to create an image file from EDI file.

Notice: The image file will be created based on the pre-defined settings. Read more in [Doing Image Setup](#), [Configuring Image Index](#).

1. Open an EDI file or directory containing the EDI files in the HIPAA Claim Master. Read more in [Processing an Electronic Claim](#), [Processing an Entire Directory](#).

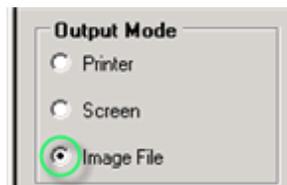


The "Open Document" button



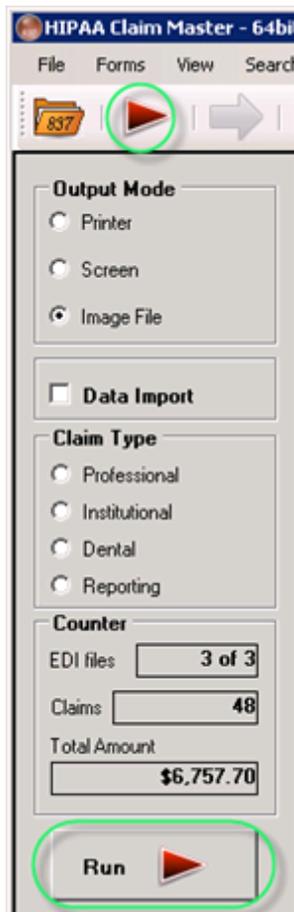
The "Open Directory" menu

2. Select the "Image File" option in the "Output Mode" block.



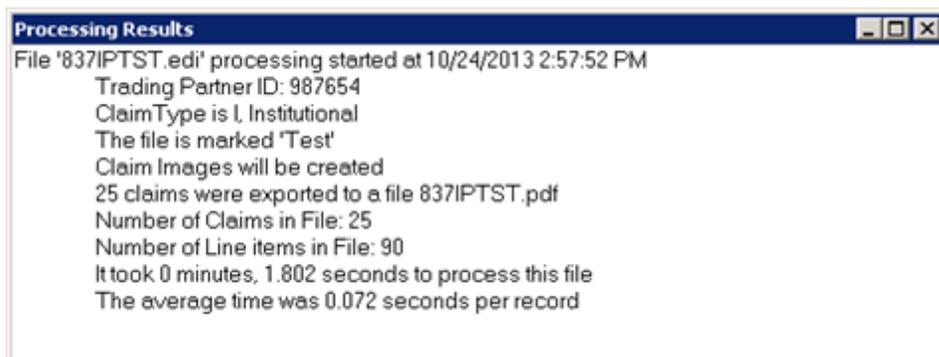
The "Image File" mode

3. Click on the "Run" button.



The "Run" buttons

4. The "Processing Results" window displays the report. You can find the resulting TIFF or PDF files in the specified folder.

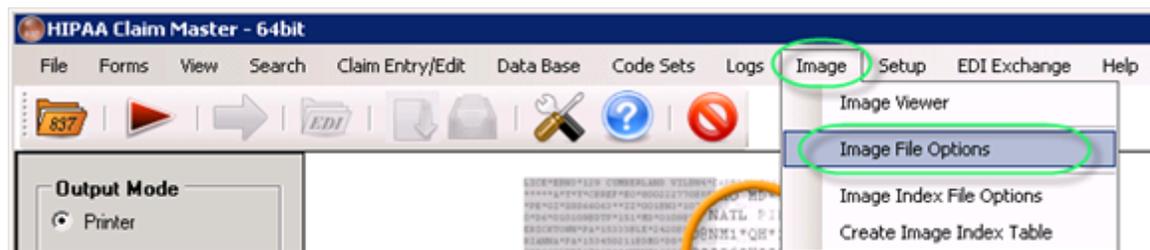


The "Processing Results" window

4.2 Doing Image Setup

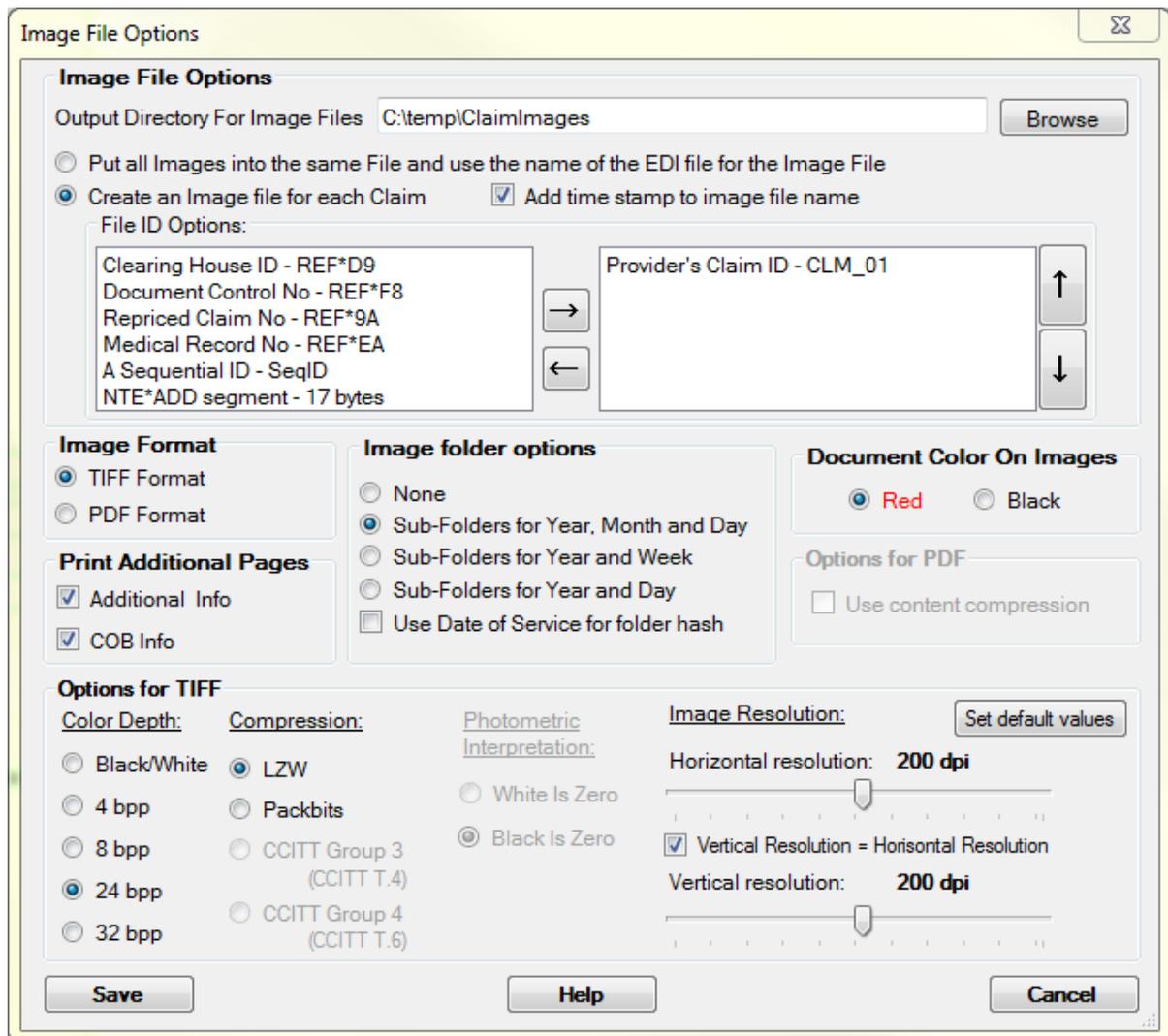
Before proceeding with image export, do the image setup as described further.

Select *Image* ► *Image File Options* in the main menu to access the "Image File Options" window.



The "Image File Options" menu

The following window will appear:



The "Image File Options" window

You can set the parameters for image file creation; they are described below.

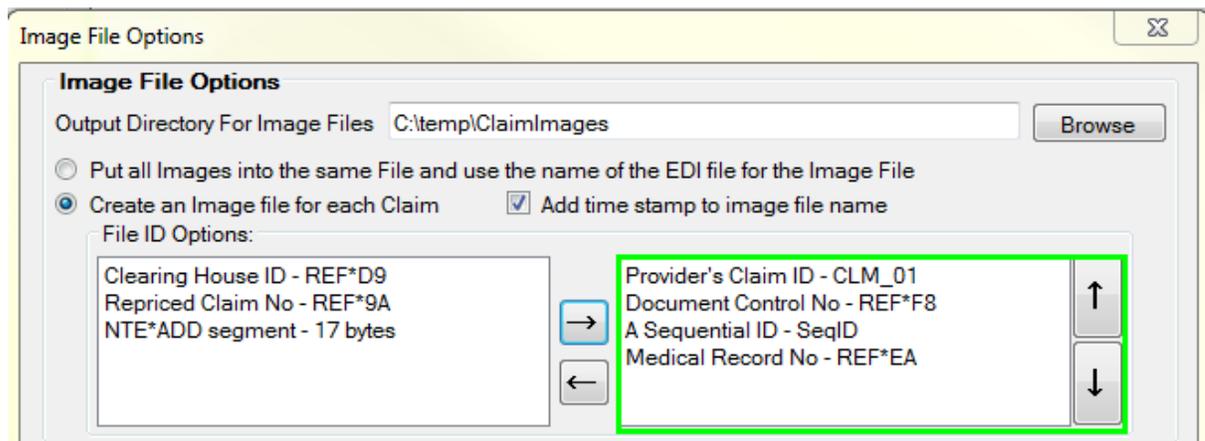
Once the options have been set, click "Save."

Image File Options

- **Output Directory for Image Files** – Here you can set the path to the folder where the image files will be saved to.
- **Put All Images into the Same File and Use the Name of the EDI File for the Image File** – This option allows to create a multi-page file with all the claim images in it using the EDI filename with a `.tif` or `.pdf` extension.

- **Create an Image File for Each Claim** – This option creates a separate image file for each individual claim. You can specify how the filename will be generated (what information will be included to the filename):
 - **Provider's Claim ID** – The Provider's Claim ID from the CML_01 segment.
 - **Clearing House ID** – The Clearinghouse ID from the REF*D9 value.
 - **Document Control No** – The Medical Record Number (REF*F8).
 - **Repriced Claim No** – The Repricer's Claim ID (REF*9A).
 - **Medical Record No** – For example, REF*EA.
 - **A Sequential ID** – A sequential number starting at one (SeqID).
 - **NTE*ADD Segment** – 17 bytes.

You can create a hierarchy that determines which ID will be used for the filename. In the picture below, you see that the document control number (REF*F8) is the preferred filename. If such a segment is not present then the second option is the Clearing house ID, third is the Medical record number. Since none of these segments are mandatory, you might encounter claims that do not produce an image name. We strongly recommend to also use the Provider's Claim ID (CLM_01) as a fall back option in the naming of the image files. Your choice will then be stored in the Windows registry.



The image naming hierarchy

To prevent the overwriting of existing image files, the HIPAA Claim Master checks if a file with that name already exists and starts to append enumerations to the filename. For example, `WD34222_3.pdf` would be the 4th instance of a file with `WD34222` as the Claim ID.

Image Format

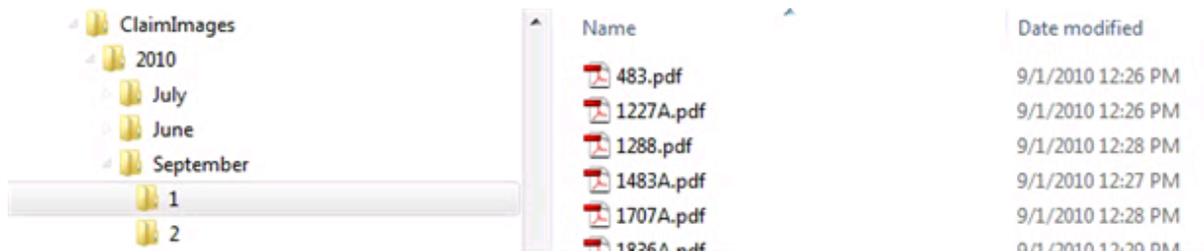
Choose either PDF or TIFF. These are the two most widely used image formats. TIFF (Tagged Image File Format), historically derived from Telefax technology, has not been updated since 1992. PDF (Portable Document Format) has been the open standard since 2008 and has become very popular, due to the ease of the format.

Print Additional Pages

This option determines whether the additional pages with either [COB information](#) or segments with [additional information](#) that have no representation on the standard forms, should be printed on either paper or image files. This option helps you save paper. If you only need the main claim form, there is no need to create a multi-page image file with 2 extra pages.

Image Folder Options

When you create image files of claims, you could soon end up with thousands of images in one folder. Windows has real problems with folders containing more than 1,000 files. For this reason, the HIPAA Claim Master can hash the images over many folders to make sure that file quantities per folder remain low.



Folder Options Year, Month and Day

You can see that there is a folder with the year 2010 under the Output directory, then subdirectories with the Month which are, finally, further subdivided into days of the month. The 2010/September/1 folder contains all the images created that day.

Document Color on Images

This section of the setup window is concerned with the output colors. The form background can be set to red to make the form more visually appealing and to allow the use of a color filter for OCR software.

Options for PDF

- **Use Content Compression** – Checking this option does result in considerably smaller files. You can choose the option to compress the content, thereby reducing the file size by roughly 50%.

Options for TIFF

Here you can adjust the settings of the Image files.

- **Color Depth** – You can change color depth of the image (black / white, 4 bpp / 8bpp / 24bpp / 32bpp).

Note: TIFF in Window's GDI library does not work with 16 bpp (bits per pixel). Color depths set lower than 16 bpp results in some losses in image quality.

Reference: The black-and-white (bi-level) image means that for each pixel (picture element) in the image we do not need three bytes as in the true color image (over 16.7 million colours), but only one bit that is 1/24 part of the pixel size from the true colour image. The one-bit pixel can express only black or white colors, nothing more.

- **Compression** – You can use compression with TIFF files: LZW and Packbits are used for all Colors, CCITT4 for black-and-white.
 - **LZW** – You can compress any type of TIFF image of any bit depth using LZW, a lossless method. The application can store the compressed image in a TIFF 5.0 file or keep it in memory. An average 2:1 compression ratio is achieved with LZW compression on images.
 - **Packbits** – You can compress and decompress gray scale, palette, and bitonal images using Packbits, a lossless method. Packbits are fast, widely-supported, and provide good compression of sparse images, such as scanned documents. The application can store the compressed image in a TIFF 5.0 file or keep it in memory.
- **Photometric Interpretation** – For the black-and-white and grayscale images (when "Black/White" or "4 bpp" options are selected as color depth), you can specify the intended interpretation of the image pixel data. Select one of the following options:
 - **White is Zero** – The minimum sample value is displayed as white.
 - **Black is Zero** – Default value. The minimum sample value is displayed as black.
- **Image Resolution** – You can change horizontal and vertical resolution of images

from 96 to 300 dpi (dots per inch). Default resolution is 200 dpi. The same rule applies here: the larger the Resolution, the more computing time and power is consumed and the larger the image file sizes.

Note: Theoretically we could use even higher resolutions, but it will slow down processing significantly result in approximately the same result in quality.

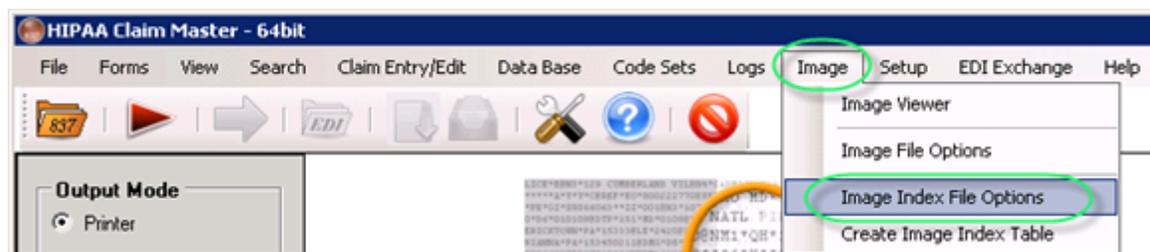
4.3 Configuring Image Index

Many of HIPAAsuite's clients have Document Management Systems into which they scan paper claims. Often the whole claim department is built around this imaging system. With the prevalence of electronic claims, the need arises to create images of the EDI claims as well. This is the whole reason behind the HIPAA Claim Master. These images had to be imported with associated indexes into the document management system. The image index capability was added for two different clients:

- The first client used an ASCII flat file with the basic information of claims and images to import into his image management system.
- The second used a database with basic information about the claim.

Both of these solutions are now part of the HIPAA Claim Master and most of our other products.

Select *Image* ► *Image Index File Options* in the main menu to access the "Image Index Configuration" window.



The "Image Index File Options" menu

You can define which fields to include and configure your database settings. Here are all options:

Create Image Index

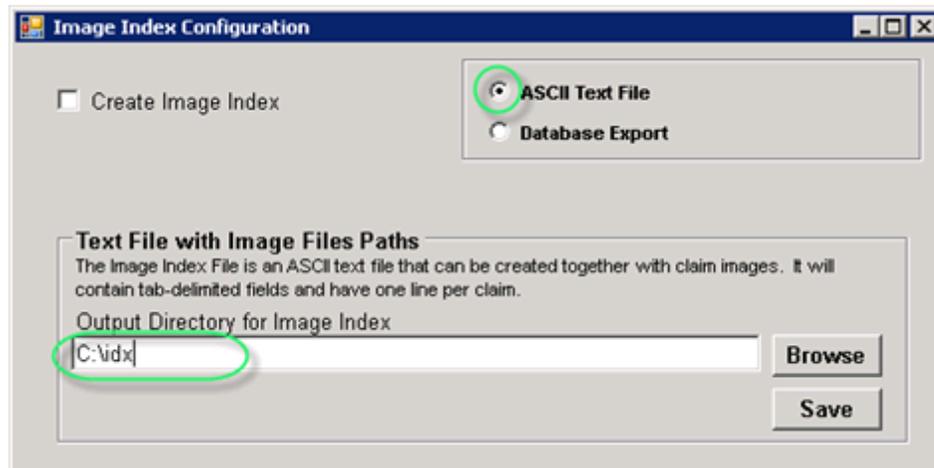
Check this option to create the image index. Select one of the following options to specify the format:

- **ASCII Text File** – The image Index File is an ASCII text file that can be created

together with claim images. It will contain tab-delimited fields and have one line per claim.

In the "Text File with Image Files Path" field specify the output directory for image index. Click "Browse" to select the destination place.

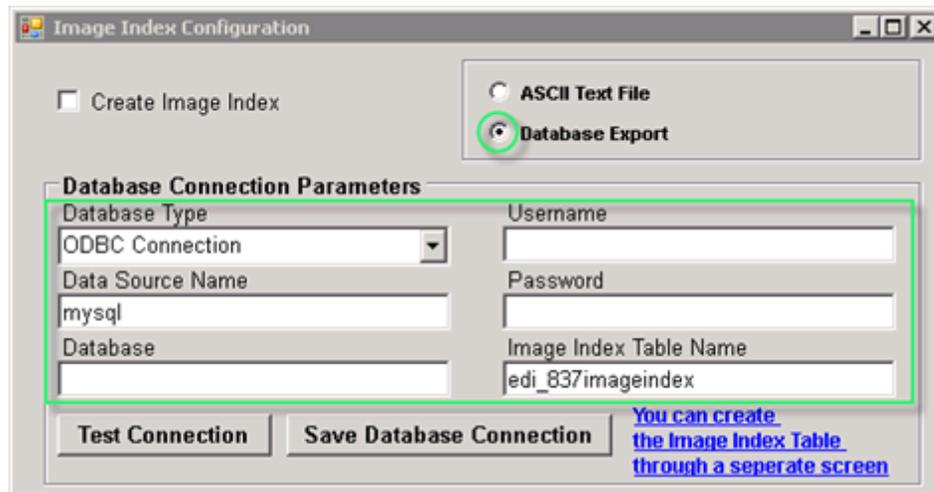
Note: The image index ASCII file currently works only when you save individual images for each claim.



The "ASCII Text File" option

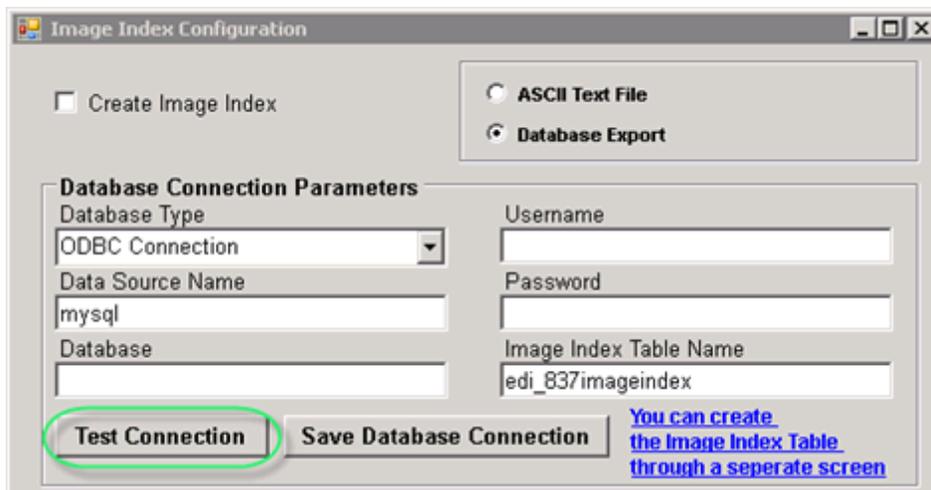
Then click "Save."

- **Database Export** – Select this option to save the image index through ODBC into any database, independent of the database used for file downloads. The database setup is similar to that of the [claim data export](#). Specify the following database connection parameters:
 - **Database Type**
 - **Database Server Name or DSN**
 - **Database**
 - **Username**
 - **Password**
 - **Image Index Table Name**



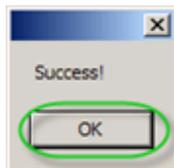
The "Database Export" option

Once all database connection parameters have been specified, click "Test Connection" to verify the connection.



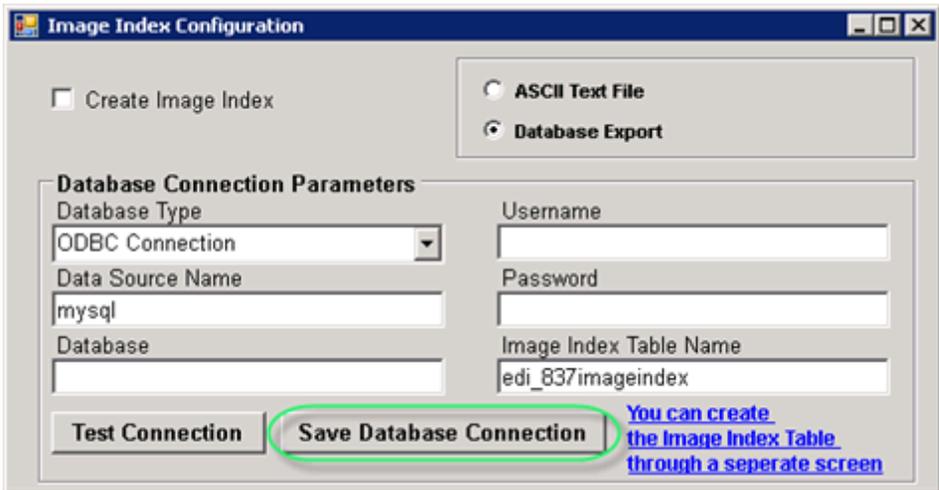
The "Test Connection" button

The configuration is correct if you receive a message like this. Click "OK."



The success message

Then click "Save Database Connection."

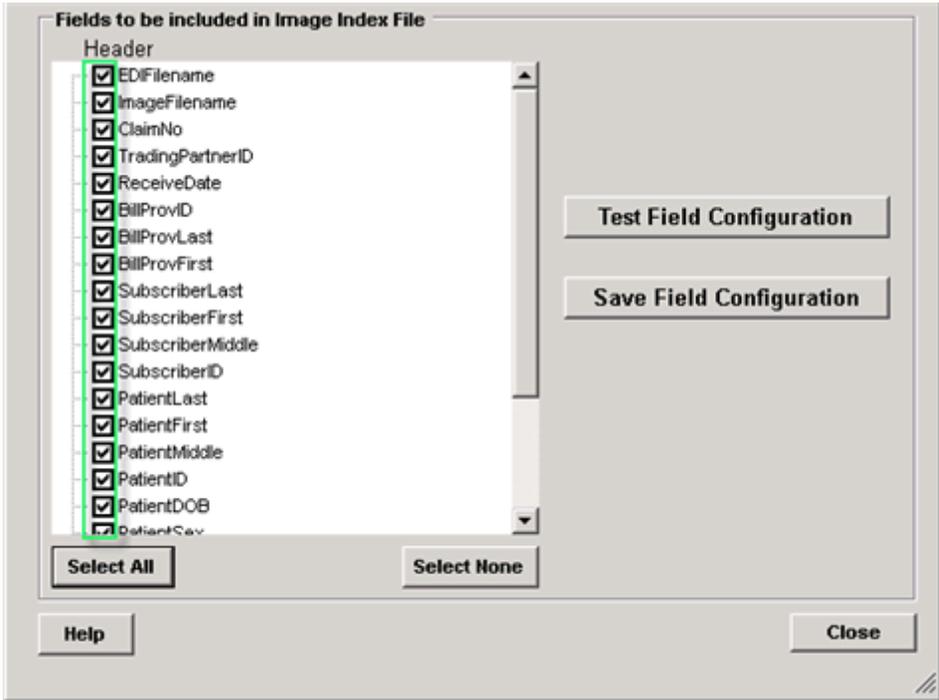


The "Save Database Connection" button

To create the image index table, click on the "You can create the Image Index Table through a separate screen." See [Creating Image Index Table](#).

Selecting Fields to Be Included to the Image Index File

You can select the fields that get imported into the image index by selecting the checkboxes in front of the field names.



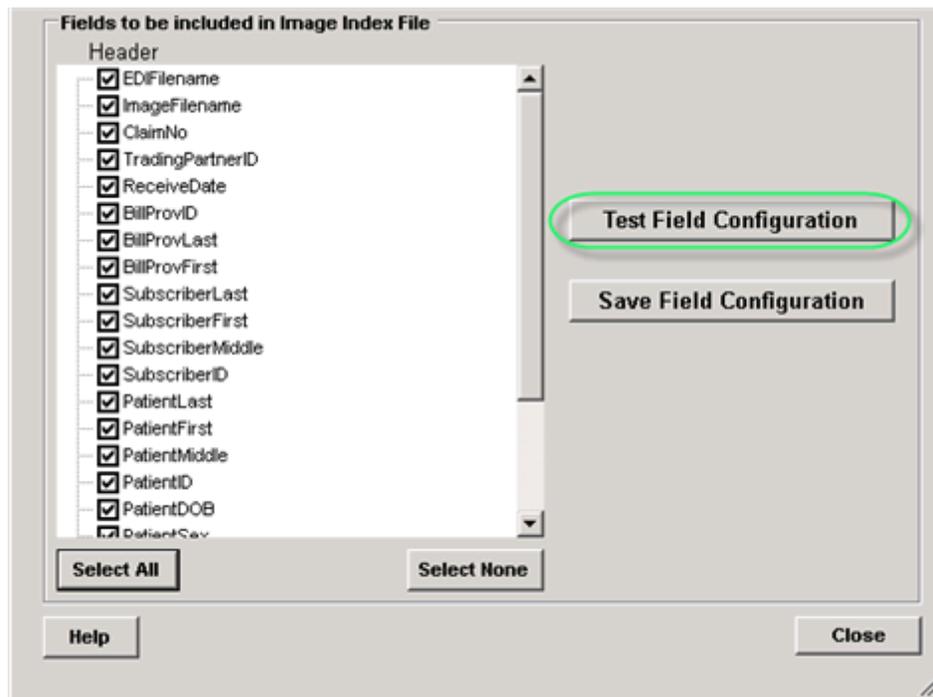
The fields to be included in Image Index File

Tip: As with the other field selectors, you can change the field names by clicking twice on them. This is helpful if you have an existing database that you want to access with different names.



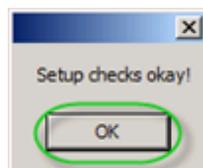
Changing the field names

For Database export, you can test the fields configuration by clicking on the "Test Field Configuration" button.



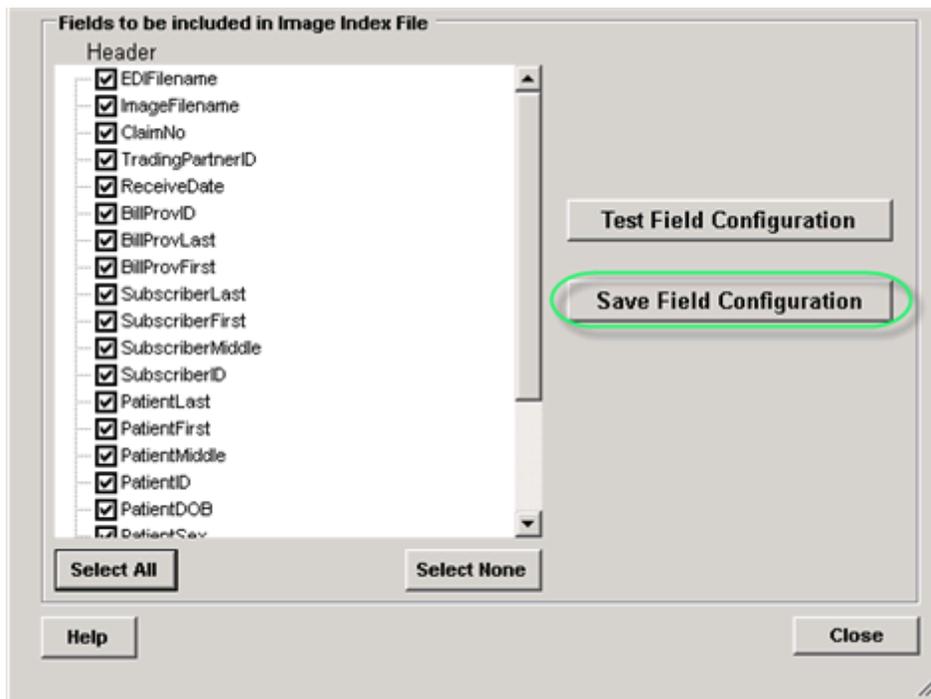
The "Test Field Configuration" button

You will receive a message like this. Click "OK."



The success message

When ready, click on the "Save Field Configuration" button.



The "Save Field Configuration" button

Once all image index options have been defined and saved, click "Close."

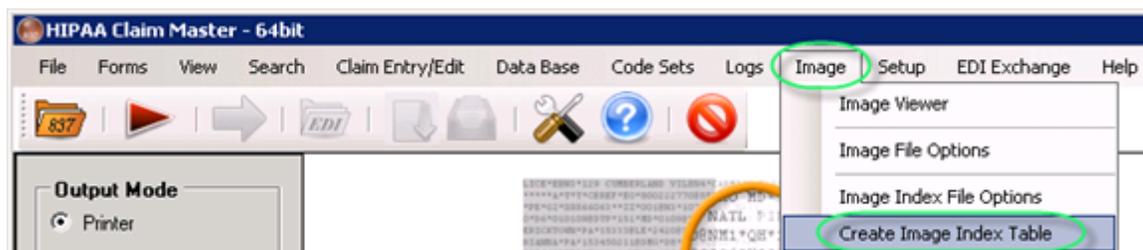


The "Close" button

4.4 Creating Image Index Table

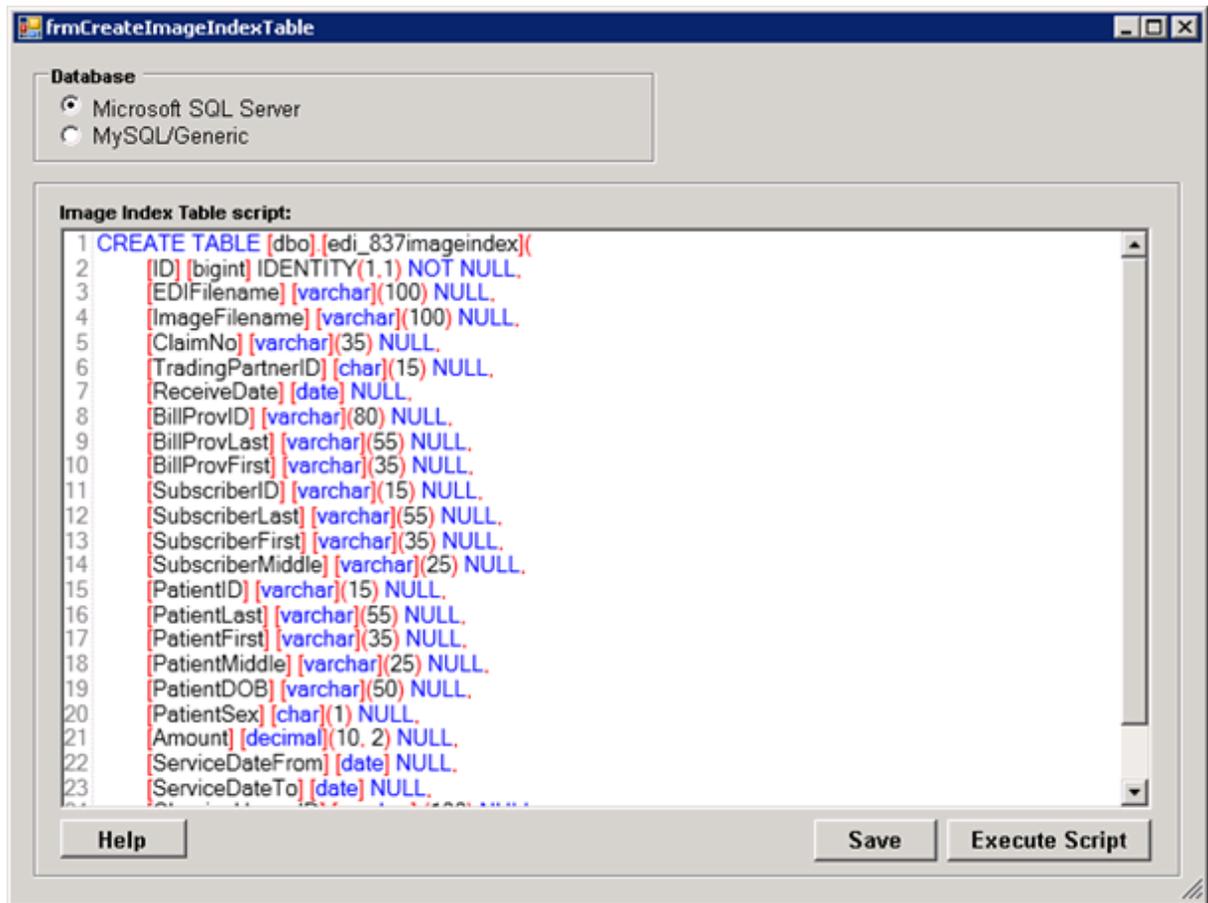
To create the table for the image index, follow the instructions below.

1. Select *Image* ► *Create Image Index Table* in the main menu.



The "Create Image Index Table" menu

2. The following window will appear.



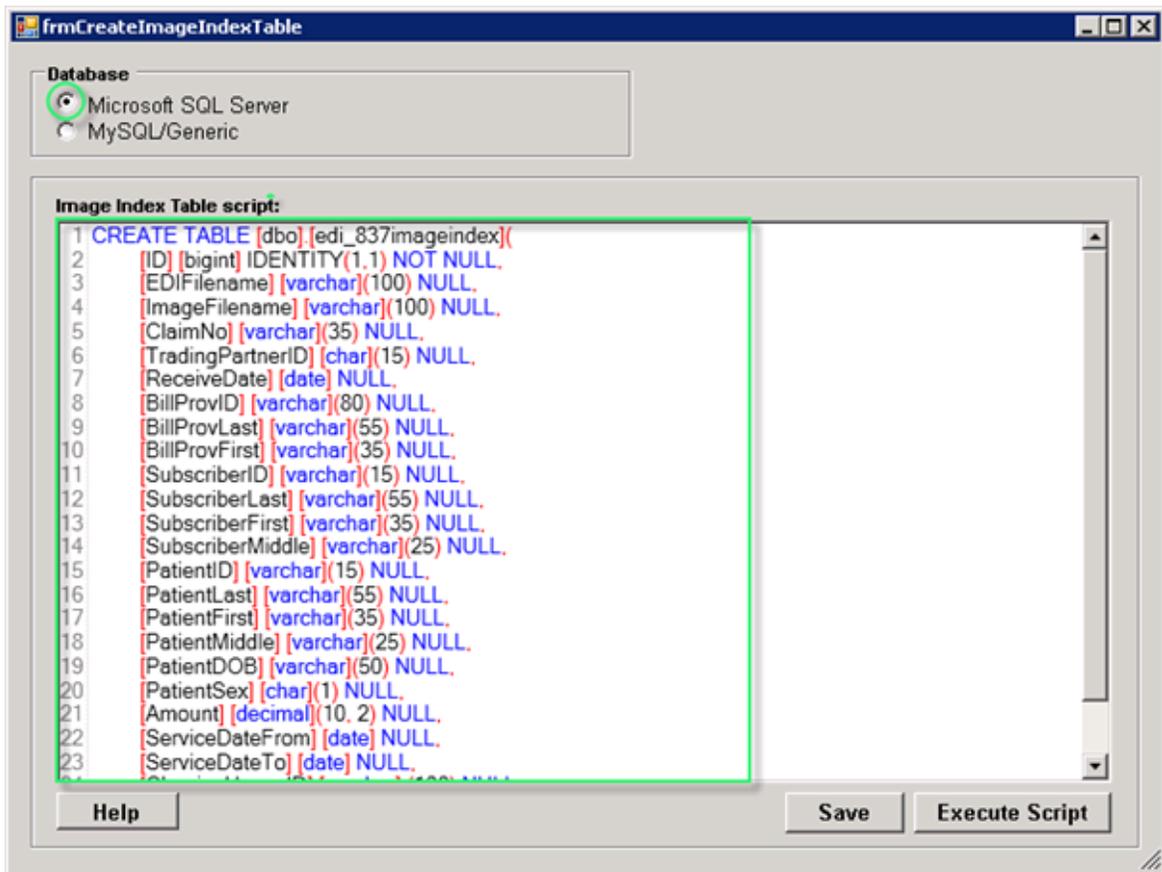
The "Create Image Index Table" window

3. Select one of the necessary options:

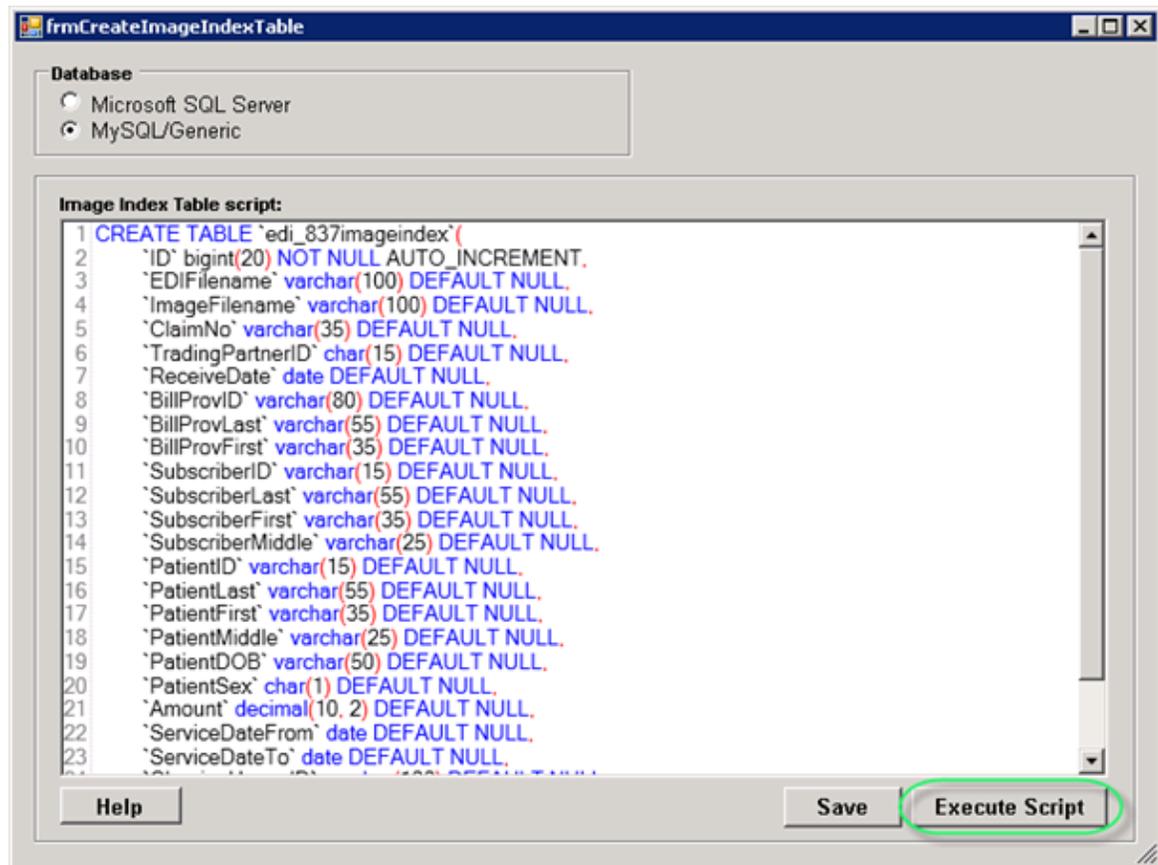
- Microsoft SQL Server
- MySQL/Genetic

Note: We do not have table scripts for all databases. Most likely, only Microsoft SQL servers and MySQL are supplied. And even here, different versions of the database behave differently. If you have database expertise, please, modify the scripts accordingly. We invite you to submit improvements and requests.

4. Then make sure the image index table script is correct.

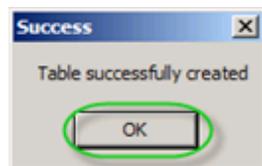


The "Create Image Index Table" window with the "Microsoft SQL Server" option selected



The "Execute Script" button

Then you will receive a notification like this. Click "OK."



The success message

6. Then click "Save."

Example

You can use the following MySQL/SQL script:

```
CREATE TABLE `edi_837imageindex` (  
  `ID` bigint(20) NOT NULL AUTO_INCREMENT,  
  `EDIFilename` varchar(100) DEFAULT NULL,  
  `ImageFilename` varchar(100) DEFAULT NULL,  
  `ClaimNo` varchar(35) DEFAULT NULL,
```

```
`TradingPartnerID` char(15) DEFAULT NULL,  
`ReceiveDate` date DEFAULT NULL,  
`BillProvID` varchar(80) DEFAULT NULL,  
`BillProvLast` varchar(55) DEFAULT NULL,  
`BillProvFirst` varchar(35) DEFAULT NULL,  
`SubscriberID` varchar(15) DEFAULT NULL,  
`SubscriberLast` varchar(55) DEFAULT NULL,  
`SubscriberFirst` varchar(35) DEFAULT NULL,  
`SubscriberMiddle` varchar(25) DEFAULT NULL,  
`PatientID` varchar(15) DEFAULT NULL,  
`PatientLast` varchar(55) DEFAULT NULL,  
`PatientFirst` varchar(35) DEFAULT NULL,  
`PatientMiddle` varchar(25) DEFAULT NULL,  
`PatientDOB` varchar(50) DEFAULT NULL,  
`PatientSex` char(1) DEFAULT NULL,  
`Amount` decimal(10, 2) DEFAULT NULL,  
`ServiceDateFrom` date DEFAULT NULL,  
`ServiceDateTo` date DEFAULT NULL,  
`ClearingHouseID` varchar(100) DEFAULT NULL,  
`OrigRefNo` varchar(50) DEFAULT NULL,  
UNIQUE KEY `ID` (`ID`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=latin1;
```

Tip: Make sure you use a correct symbol ` and not ' in the script.

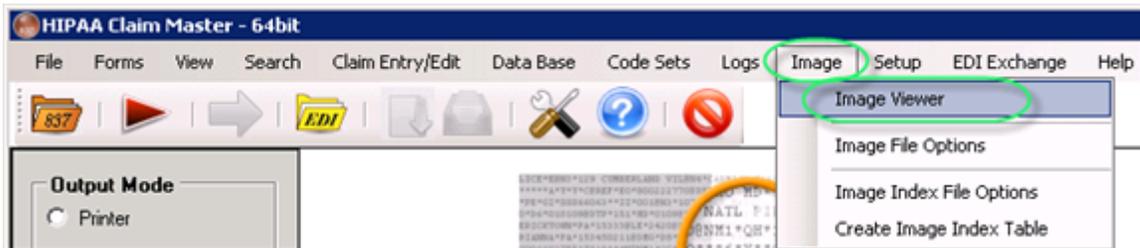
Tip: If you encounter errors running the script on MySQL or other database systems, then modify it until it works or consult with your database administrator to make the modifications. SQL Scripts share over all platforms a lot of syntax, but still each flavor demands its own adjustments.

Tip: If you encounter persistent errors with the script, we recommend working with the scripts within the database environment. SQL Server Enterprise Manager or SQLyog, for example, have an SQL query analyzer that helps you tweak the script until it runs.

4.5 Viewing Image Files

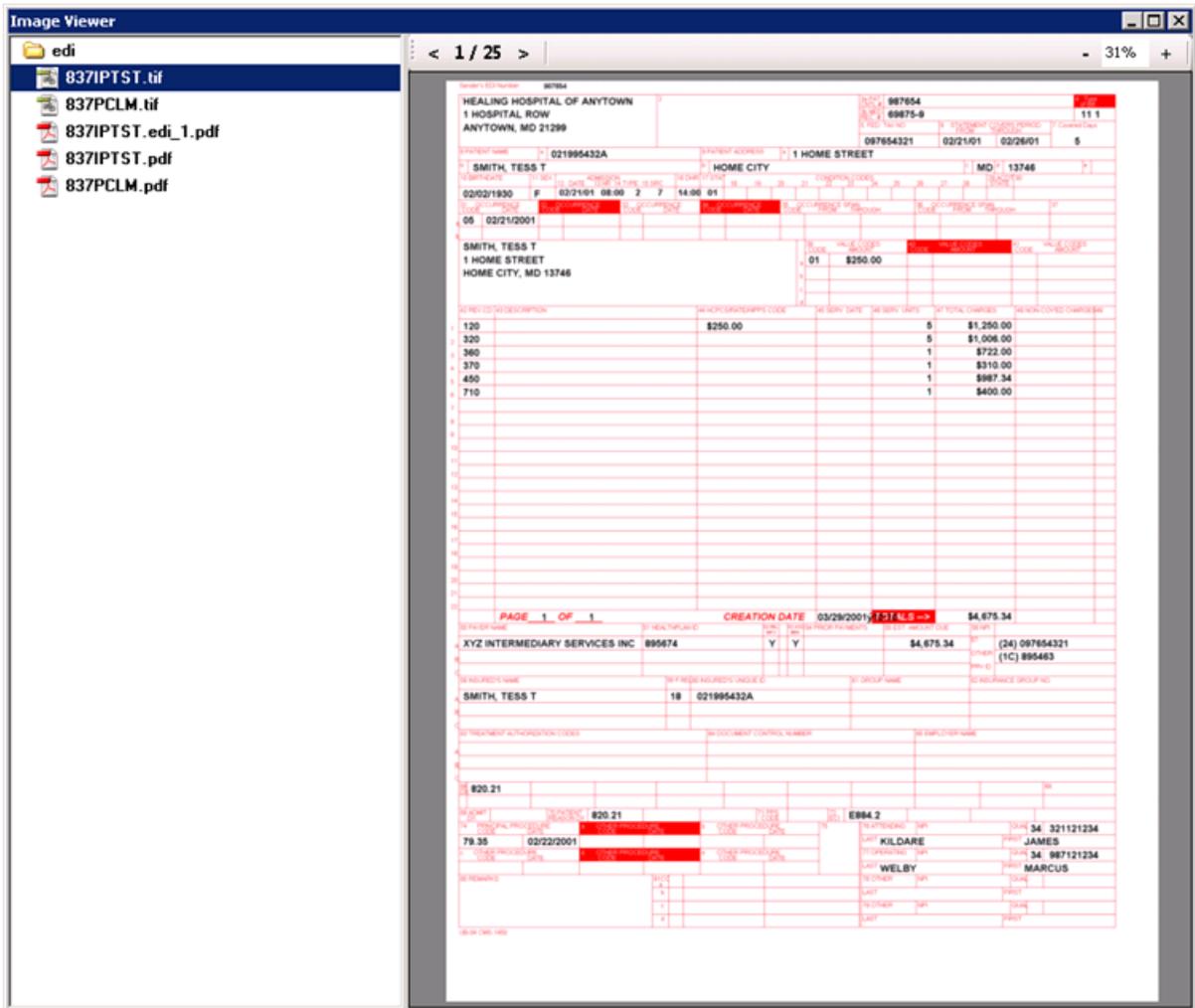
The HIPAA Claim Master has a built-in image viewer that uses light weight dll's to display TIFF and PDF images. It also allows you to navigate through the folder structure that has been set up.

Select *Image* ► *Image Viewer* in the main menu to access the "Image Viewer" window.



The "Image Viewer" menu

The following window will appear:



The "Image Viewer" window

The Viewer lets you navigate through multi-page documents with the navigation buttons in the upper left corner of the image. You can adjust the display size according to your monitor size or preferences in the upper right corner.

The left panel lists all generated TIFF and PDF files. Click a file name to display its content.

TIFF: Microsoft Windows has a Windows Photo Viewer that renders TIFF images. Unfortunately, it is not that practical. It omits lines in lower magnification and makes the image look sloppy, but is sufficient for most. There are much better TIFF viewers out there but we will not make any recommendations.

PDF: PDF files can be opened with [Acrobat Reader](#) by Adobe, which is available for free. There are also other free PDF viewers and editors. We will not make any recommendations.

Chapter



5 Using the Database

5.1 Exporting the Data

The HIPAA Claim Master does not come with a database. But it can link to any ODBC- or OleDB-compliant database. Database integration is a component of the HIPAA Claim Master and can be licensed as add-in.

Most businesses that do HIPAA transactions have a database engine on their premises. Microsoft SQL Server is probably the most widely used business database; Oracle or IBM database also have broad acceptance. The open source and free database MySQL finds more and more acceptance in the business world and is great for developers.

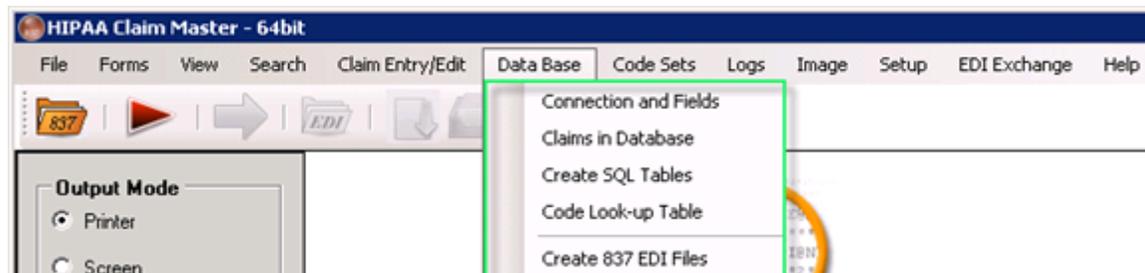
The HIPAA Claim Master uses two database tables for the claim information:

- a claim header table
- a claim detail table with the line item information.

Scripts are provided to create these tables for some database types. You might have to modify the scripts to suit your actual taste of the SQL language.

Once these tables are created, claims from EDI files can be loaded into this database. Your application can take the claim information from there and you can run your business logic on these records. The HIPAA Claim Master can in turn read the database record and re-create the claim image from the database. In the next implementation you will even be able to create an 837 EDI file from those database records.

The HIPAA Claim Master lets you set up two connections, to a test and to a production database. The deciding factor on which database is used is the ISA_15 data element of the 837 file. This indicator is either a "P" for production or "T" for test. If you need only one database, set up both connections identically.



The "Data Base" menu

The sequence of actions to **setup** the data export is the following:

- [Setting up Database Connection](#)
- [Creating Tables](#)
- [Selecting Fields to Export](#)

How to Save Data to the Database

The HIPAA Claim Master provides the "Data Import" option. You can import the claim data to a database while printing or imaging claims (see [Processing an Electronic Claim, Printing Electronic Claims](#)). In the "Screen" mode, the data import will not work since you can interrupt the viewing of EDI claims at any time and this might lead to incomplete data import.

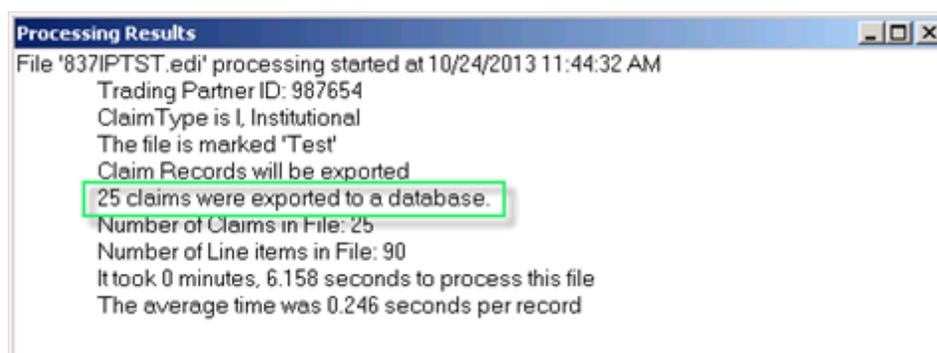
Notice: The image file will be created based on the pre-defined settings. Read more in



The "Data Import" check-box

The data is exported once the user clicks on the "Run" button. The data is written to the database once the "Run" button has been clicked and only if the "Data Import" checkbox is activated.

The "Processing Results" window will display notification about data export.



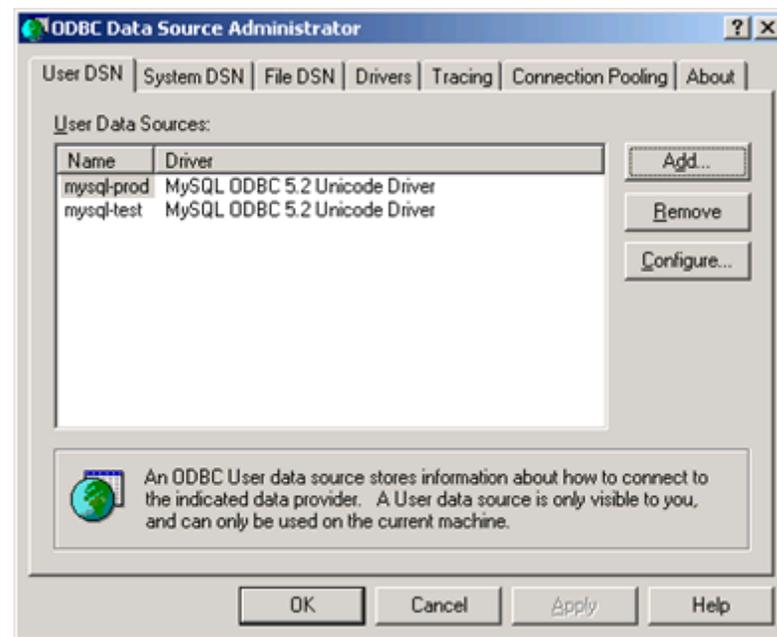
The "Processing Results" window

Now you can [browse exported data](#).

5.2 Setting up Database Connection

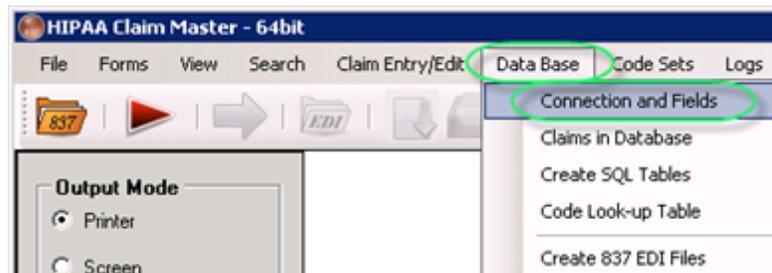
The HIPAA Claim Master relies on ODBC; or in the case of Microsoft SQL server on OleDB to connect to a database.

Tip: If you use ODBC for the connection, you will need to set up the ODBC connection before using HIPAA Claim Master with the database. Typically, ODBC connections are established in the *Windows Control Panel Administrative Tools Data Sources (ODBC)* setup screen. Setting up the ODBC connection varies from database to database.



The "ODBC Data Source Administrator" screen in Windows

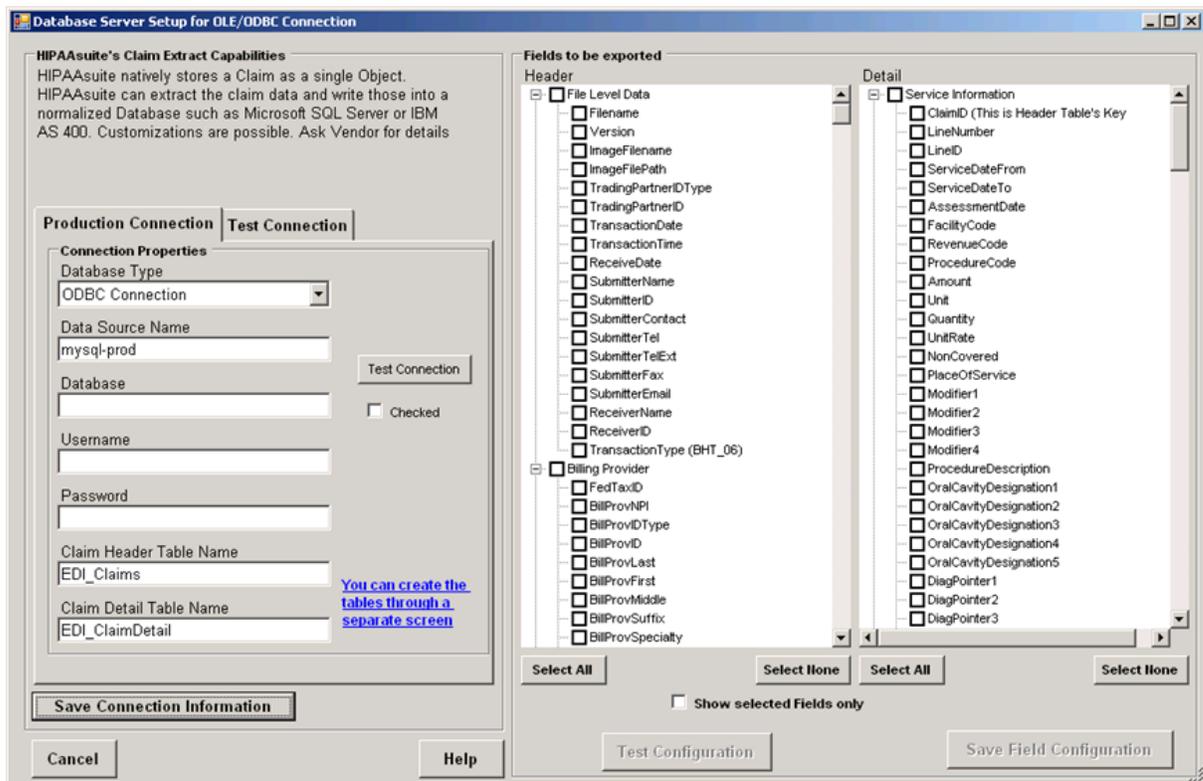
Select *Data Base Connection and Fields* in the main menu.



The "Connection and Fields" menu

2. You will be confronted with the following screen where you can set up the database

connection and enter the names of the database tables.



The "Database Server Setup for OLE/ODBC Connection

On the left side of the screen, define the database connection properties.

There are two tabs:

- **Production Connection**
- **Test Connection**

Each tab has the following fields:

- **Database Type** - Select the type of the database from the drop-down menu: is either ODBC or Microsoft SQL Server either with Windows Authentication or with explicit credentials.

Note: If you need other types, please contact us for customizations.

- **MS SQL Server**
- **MS SQL Server Integrated Security**
- **ODBC Connection**
- **ODBC Connection for Oracle**

- **Database Server Name** – Specify the database server name or DSN. In case of SQL Server, it is the IP address or the name of the database server; for ODBC it is the Data Source Name (DSN) that is defined through the ODBC setup in the Control Panel of Windows.
- **Database** - Enter the database name under the above connection. Typically, this field will be empty for ODBC setup.
- **Username** - Specify the user name that has privileges to the database. Typically, this field will be empty for ODBC setup.
- **Password** - Specify user's password. Typically, this field will be empty for ODBC setup.
- **Claim Header Table Name** - Enter the name that you give to your Claim header table. The default name is suggested but can be changed.
- **Claim Detail Table Name** - Enter the name of the table that contains the line information. A default name is suggested but can be changed.

We will configure the Test Connection and show the process on the screen-shots.

Database Server Setup for OLE/ODBC Connection

HIPAAsuite's Claim Extract Capabilities
HIPAAsuite natively stores a Claim as a single Object. HIPAAsuite can extract the claim data and write those into a normalized Database such as Microsoft SQL Server or IBM AS 400. Customizations are possible. Ask Vendor for details

Production Connection **Test Connection**

Connection Properties

Database Type
ODBC Connection

Data Source Name
mysql-test

Database

Username

Password

Claim Header Table Name
EDI_Claims

Claim Detail Table Name
EDI_ClaimDetail

Test Connection

Checked

[You can create the tables through a separate screen](#)

Save Connection Information

Cancel Help

The connection properties fields

To create tables in your database, click on the "You can create the tables through a separate screen" link. For instructions, refer to [Creating Tables](#).

Database Server Setup for OLE/ODBC Connection

HIPAA Suite's Claim Extract Capabilities
HIPAA Suite natively stores a Claim as a single Object. HIPAA Suite can extract the claim data and write those into a normalized Database such as Microsoft SQL Server or IBM AS 400. Customizations are possible. Ask Vendor for details

Production Connection **Test Connection**

Connection Properties

Database Type
ODBC Connection

Data Source Name
mysql-test

Database

Username

Password

Claim Header Table Name
EDI_Claims

Claim Detail Table Name
EDI_ClaimDetail

Test Connection

Checked

[You can create the tables through a separate screen](#)

Save Connection Information

Cancel Help

A link to create tables

Once the connection parameters have been entered, click "Save Connection Information."

Database Server Setup for OLE/ODBC Connection

HIPAA Suite's Claim Extract Capabilities
HIPAA Suite natively stores a Claim as a single Object. HIPAA Suite can extract the claim data and write those into a normalized Database such as Microsoft SQL Server or IBM AS 400. Customizations are possible. Ask Vendor for details

Production Connection **Test Connection**

Connection Properties

Database Type
ODBC Connection

Data Source Name
mysql-test

Database

Username

Password

Claim Header Table Name
EDI_Claims

Claim Detail Table Name
EDI_ClaimDetail

Test Connection

Checked

[You can create the tables through a separate screen](#)

Save Connection Information

Cancel Help

The "Save Connection Information" button

To verify if the entered parameters are correct, click on the "Test Connection" button. You cannot export claims before this connection tests successfully.

Database Server Setup for OLE/ODBC Connection

HIPAAsuite's Claim Extract Capabilities
HIPAAsuite natively stores a Claim as a single Object.
HIPAAsuite can extract the claim data and write those into a normalized Database such as Microsoft SQL Server or IBM AS 400. Customizations are possible. Ask Vendor for details

Production Connection **Test Connection**

Connection Properties

Database Type
ODBC Connection

Data Source Name
mysql-test

Database

Username

Password

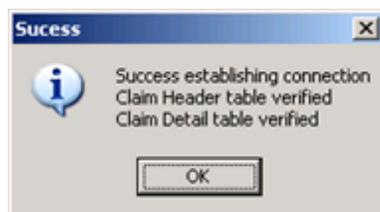
Claim Header Table Name
EDI_Claims

Claim Detail Table Name
EDI_ClaimDetail

[You can create the tables through a separate screen](#)

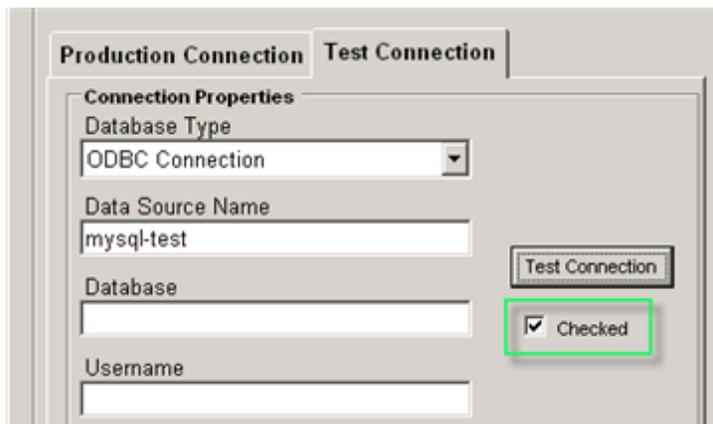
Checked

If the test has passed successfully, you will see the following notification:



The success message

Once the test has passed successfully, the "Checked" box becomes enabled.



The "Checked" checkbox

Once you have the database connection defined and tested, you can enter the table names. See [Selecting Fields to Export](#).

If you need to create the tables first, then see [Creating Tables](#).

5.3 Creating Tables

Before Creating Tables...

The scripts that are supplied with the HIPAA Claim Master will generate every field that is available for import. These are more than 260 fields in the claim header. You will not need all these fields. You can edit the scripts to only use the fields that your adjudication system requires. System performance will be better if you only select the needed fields.

Not all databases are supported. Most likely only Microsoft SQL server and MYSQL scripts are supplied. And even here different versions of the database behave differently. If you have database expertise, please modify the scripts accordingly. We invite you to submit improvements and requests.

If you encounter persistent errors with the script, we recommend you to work with the scripts within the database environment. SQL Server Enterprise Manager or SQLyog for example have an SQL query analyzer that helps you tweak the scripts until they run.

Also, feel free to contact us if you have any problems.

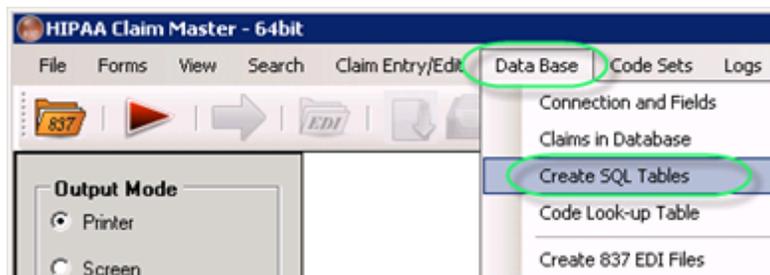
How to Create Tables

In order to facilitate the development, HIPAA Claim Master allows you to create the staging tables for your database where HIPAA Claim Master will export the claim data.

Before [data export](#), you need to create the tables through SQL table statements. You can do it using the built-in function of the HIPAA Claim Master. Creating these tables is a quite intimidating task. Once you have your [connection defined and tested](#), follow the instructions below.

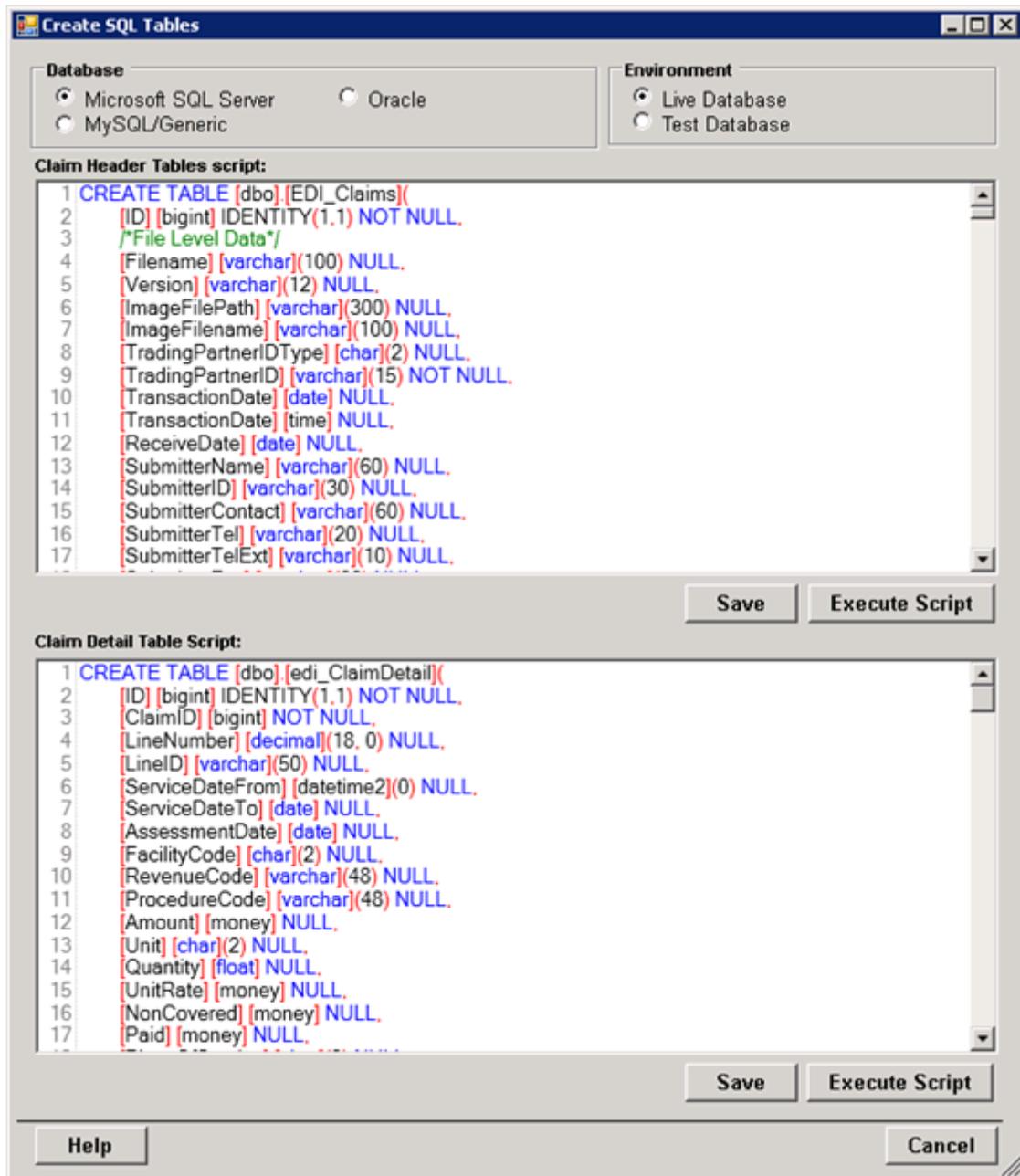
Note: The tables are created via the "Drop Table" and "Create Table" commands. The scripts come from two files: `EDI_Claims.sql` and `EDI_ClaimDetail.sql` that are distributed with the application. Please, contact HIPAAsuite if you need help with the database setup.

1. Select the *Data Base* ▶ *Create SQL Table* in the main menu.



The "Create SQL Table" menu

2. The following window will be displayed.



The "Create SQL Tables" window

3. Define the following options:

Database

- Microsoft SQL Server
- MySQL/Genetic
- Oracle

Environment

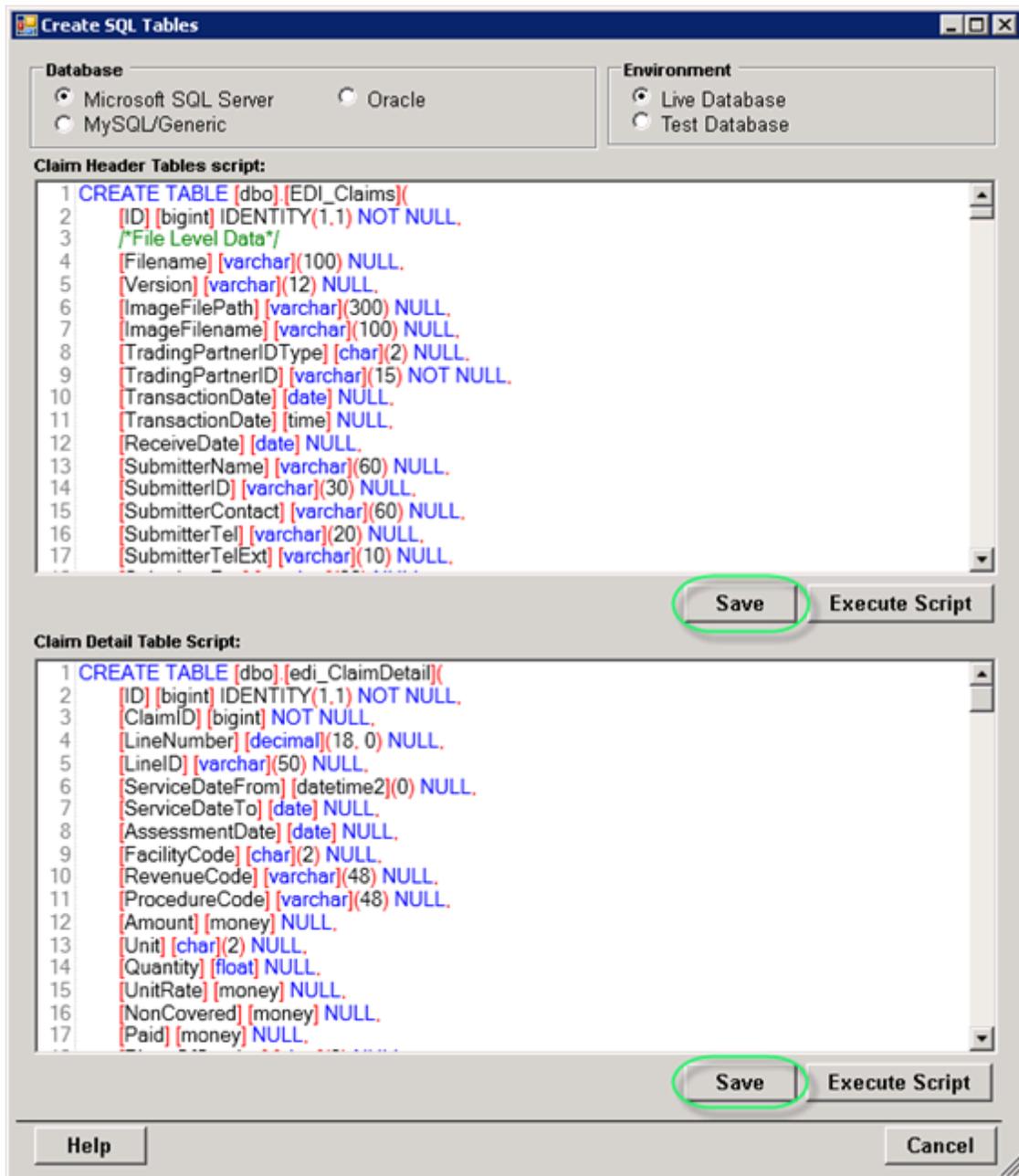
- Live Database
- Test Database

Scripts

On the following tabs there are SQL scripts used to create the appropriate tables:

- Claim Header Tables Script
- Claim Detail Table Script

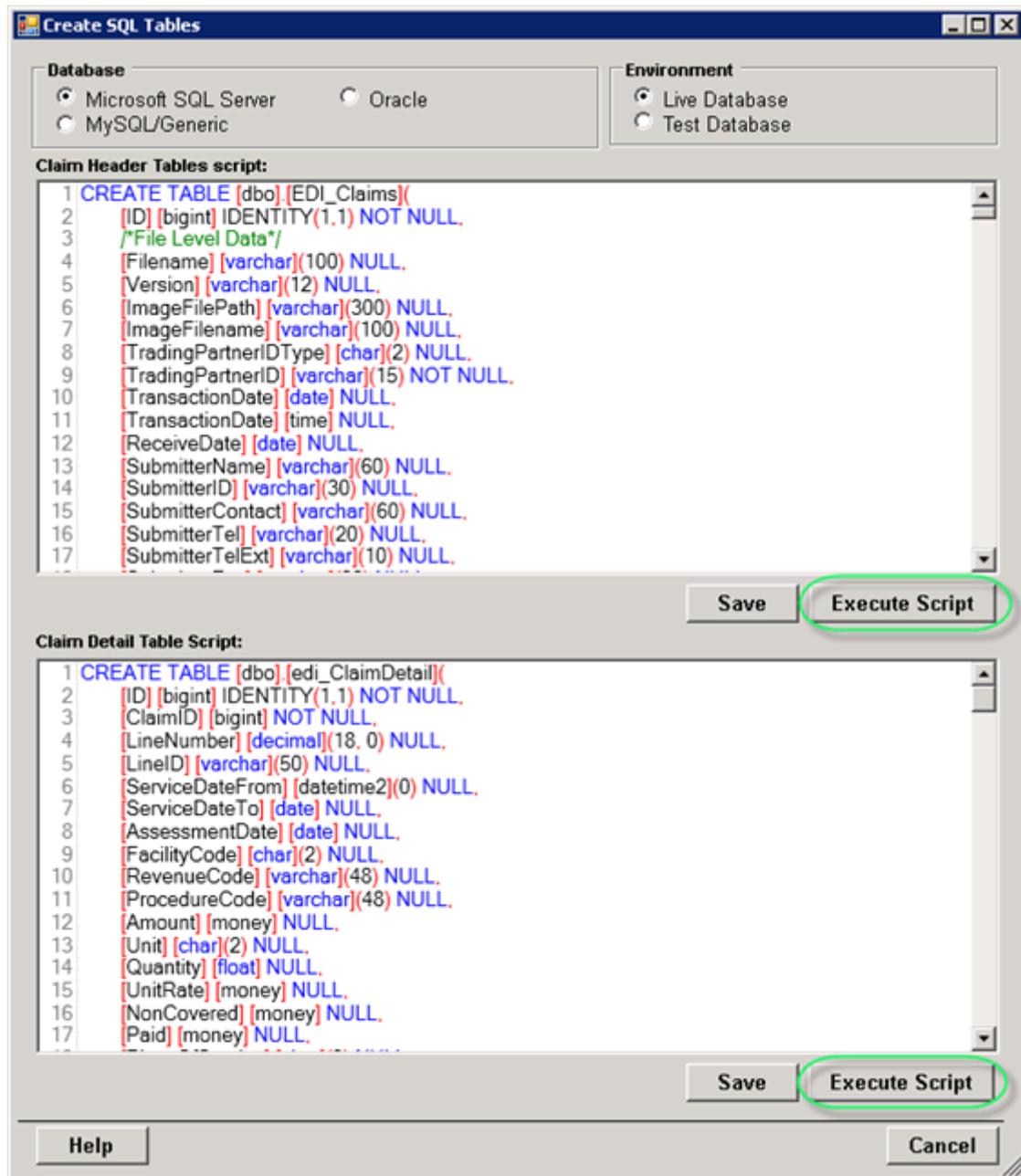
Tip: You can modify the scripts so that they run on your specific database and to conform to your database's specific SQL syntax. For each script, click "Save."



The "Save" button

For each script, click "Execute Script" to create the corresponding table in the database.

Warning: Executing the scripts will wipe out any previous tables that you created and you will lose all data.



The "Execute Script" button

Once the table has been created successfully, you will see the following notification:



The success message

5.3.1 Microsoft SQL Server Script to Create Tables

Header

```
CREATE TABLE [dbo].[EDI_Claims](
    [ID] [bigint] IDENTITY(1,1) NOT NULL,
    /*File Level Data*/
    [Filename] [varchar](100) NULL,
    [Version] [varchar](12) NULL,
    [ImageFilePath] [varchar](300) NULL,
    [ImageFilename] [varchar](100) NULL,
    [TradingPartnerIDType] [char](2) NULL,
    [TradingPartnerID] [varchar](15) NOT NULL,
    [TransactionDate] [date] NULL,
    [ReceiveDate] [date] NULL,
    [SubmitterName] [varchar](60) NULL,
    [SubmitterID] [varchar](30) NULL,
    [SubmitterContact] [varchar](60) NULL,
    [SubmitterTel] [varchar](20) NULL,
    [SubmitterTelExt] [varchar](10) NULL,
    [SubmitterFax] [varchar](20) NULL,
    [SubmitterEmail] [varchar](80) NULL,
    [ReceiverName] [varchar](60) NULL,
    [ReceiverID] [varchar](30) NULL,
    [TransactionType] [char](2) NULL,
    /*Billing Provider*/
    [FedTaxID] [varchar](15) NULL,
```

[BillProvIDType] [char](2) NULL,
[BillProvID] [varchar](80) NULL,
[BillProvNPI] [varchar](12) NULL,
[BillProvLast] [varchar](60) NULL,
[BillProvFirst] [varchar](35) NULL,
[BillProvMiddle] [varchar](25) NULL,
[BillProvSuffix] [varchar](10) NULL,
[BillProvSpecialty] [varchar](10) NULL,
[BillProvAddress] [varchar](55) NULL,
[BillProvAddress2] [varchar](55) NULL,
[BillProvCity] [varchar](30) NULL,
[BillProvState] [char](2) NULL,
[BillProvZip] [varchar](15) NULL,
[BillProvCountry] [varchar](3) NULL,
[BillProvSubdivision] [varchar](3) NULL,
[BillProvContact] [varchar](60) NULL,
[BillProvTel] [varchar](80) NULL,
[BillProvTelExt] [varchar](80) NULL,
[BillProvFax] [varchar](80) NULL,
[BillProvEmail] [varchar](80) NULL,
[BillProvOtherIDQual1] [char](2) NULL,
[BillProvOtherID1] [varchar](30) NULL,
[BillProvOtherIDQual2] [char](2) NULL,
[BillProvOtherID2] [varchar](30) NULL,
[BillProvOtherIDQual3] [char](2) NULL,
[BillProvOtherID3] [varchar](30) NULL,

```
[BillProvOtherIDQual4] [char](2) NULL,  
[BillProvOtherID4] [varchar](30) NULL,  
[BillProvOtherIDQual5] [char](2) NULL,  
[BillProvOtherID5] [varchar](30) NULL,  
  
/*Pay-to Provider*/  
  
[PayToProvIDType] [char](2) NULL,  
[PayToProvID] [varchar](30) NULL,  
[PayToProvTaxID] [varchar](12) NULL,  
[PayToProvNPI] [varchar](12) NULL,  
[PayToProvLast] [varchar](60) NULL,  
[PayToProvFirst] [varchar](35) NULL,  
[PayToProvMiddle] [varchar](25) NULL,  
[PayToProvSuffix] [varchar](10) NULL,  
[PayToProvSpecialty] [varchar](10) NULL,  
[PayToProvAddress] [varchar](55) NULL,  
[PayToProvAddress2] [varchar](55) NULL,  
[PayToProvCity] [varchar](30) NULL,  
[PayToProvState] [char](2) NULL,  
[PayToProvZip] [varchar](15) NULL,  
  
/*Pay-to Payer*/  
  
[PayToPayerIDType] [char](2) NULL,  
[PayToPayerID] [varchar](80) NULL,  
[PayToPayerTaxID] [varchar](12) NULL,  
[PayToPayerLast] [varchar](60) NULL,  
[PayToPayerFirst] [varchar](35) NULL,  
[PayToPayerMiddle] [varchar](25) NULL,
```

```
[PayToPayerSuffix] [varchar](10) NULL,  
[PayToPayerAddress] [varchar](55) NULL,  
[PayToPayerAddress2] [varchar](55) NULL,  
[PayToPayerCity] [varchar](30) NULL,  
[PayToPayerState] [char](2) NULL,  
[PayToPayerZip] [varchar](15) NULL,  
[PayToPayerOtherIDQual] [char](2) NULL,  
[PayToPayerOtherID] [varchar](30) NULL,  
/*Subscriber*/  
[SubscriberLast] [varchar](60) NULL,  
[SubscriberFirst] [varchar](35) NULL,  
[SubscriberMiddle] [varchar](25) NULL,  
[SubscriberSuffix] [varchar](10) NULL,  
[SubscriberIDType] [char](2) NULL,  
[SubscriberID] [varchar](30) NULL,  
[SubscriberAddress] [varchar](55) NULL,  
[SubscriberAddress2] [varchar](55) NULL,  
[SubscriberCity] [varchar](30) NULL,  
[SubscriberState] [char](2) NULL,  
[SubscriberZip] [varchar](15) NULL,  
[SubscriberCountry] [varchar](3) NULL,  
[SubscriberLocation] [varchar](30) NULL,  
[SubscriberSubdivision] [varchar](3) NULL,  
[SubscriberDOB] [varchar](10) NULL,  
[SubscriberSex] [char](1) NULL,  
[SubscriberEthnicity] [varchar](75) NULL,
```

```
[SubscriberMaritalStatus] [char](1) NULL,  
[SubscriberCollectionMethod] [varchar](30) NULL,  
[SubscriberSSN] [varchar](12) NULL,  
[SubscriberAgencyClaimNo] [varchar](50) NULL,  
[SubscriberMemberID] [varchar](50) NULL,  
[SubscriberPersonalID] [varchar](50) NULL,  
[SubscriberContact] [varchar](60) NULL,  
[SubscriberTel] [varchar](15) NULL,  
[SubscriberTelExt] [varchar](10) NULL,  
/*Destination Payer*/  
[PayerName] [varchar](60) NULL,  
[PayerIDType] [char](2) NULL,  
[PayerID] [varchar](30) NULL,  
[PayerAddress] [varchar](55) NULL,  
[PayerAddress2] [varchar](55) NULL,  
[PayerCity] [varchar](30) NULL,  
[PayerState] [char](2) NULL,  
[PayerZip] [varchar](15) NULL,  
[PayerResponsibility] [char](1) NULL,  
[PayerOtherIDQual1] [char](2) NULL,  
[PayerOtherID1] [varchar](30) NULL,  
[PayerOtherIDQual2] [char](2) NULL,  
[PayerOtherID2] [varchar](30) NULL,  
[PayerOtherIDQual3] [char](2) NULL,  
[PayerOtherID3] [varchar](30) NULL,  
[GroupNo] [varchar](50) NULL,
```

```
[GroupName] [varchar](60) NULL,  
[InsuranceType] [char](2) NULL,  
[FilingIndicator] [char](2) NULL,  
[COBIndicator] [char](1) NULL,  
  
/*Responsible*/  
[ResponsibleLast] [varchar](60) NULL,  
[ResponsibleFirst] [varchar](35) NULL,  
[ResponsibleMiddle] [varchar](25) NULL,  
[ResponsibleSuffix] [varchar](10) NULL,  
[ResponsibleAddress] [varchar](55) NULL,  
[ResponsibleAddress2] [varchar](55) NULL,  
[ResponsibleCity] [varchar](30) NULL,  
[ResponsibleState] [char](2) NULL,  
[ResponsibleZip] [varchar](15) NULL,  
[ResponsibleIDType] [char](2) NULL,  
[ResponsibleID] [varchar](80) NULL,  
  
/*Patient*/  
[PatientLast] [varchar](60) NULL,  
[PatientFirst] [varchar](35) NULL,  
[PatientMiddle] [varchar](25) NULL,  
[PatientSuffix] [varchar](10) NULL,  
[PatientAddress] [varchar](55) NULL,  
[PatientAddress2] [varchar](55) NULL,  
[PatientCity] [varchar](30) NULL,  
[PatientState] [char](2) NULL,  
[PatientZip] [varchar](15) NULL,
```

```
[PatientCountry] [varchar](3) NULL,  
[PatientLocation] [varchar](30) NULL,  
[PatientSubdivision] [varchar](3) NULL,  
[PatientIDType] [char](2) NULL,  
[PatientID] [varchar](30) NULL,  
[PatientSSN] [varchar](12) NULL,  
[PatientMemberID] [varchar](50) NULL,  
[PatientPersonalID] [varchar](50) NULL,  
[PatientInsurancePolicy] [varchar](50) NULL,  
[PatientDOB] [varchar](50) NULL,  
[PatientSex] [char](1) NULL,  
[PatientEthnicity] [varchar](75) NULL,  
[PatientMaritalStatus] [char](1) NULL,  
[PatientCollectionMethod] [varchar](30) NULL,  
[PatientRelationship] [char](2) NULL,  
[PatientPaid] [varchar](10) NULL,  
[PregnancyIndicator] [char](1) NULL,  
/*Rendering Provider*/  
[RendProvIDType] [char](2) NULL,  
[RendProvID] [varchar](30) NULL,  
[RendProvNPI] [varchar](12) NULL,  
[RendProvTaxID] [varchar](12) NULL,  
[RendProvLast] [varchar](60) NULL,  
[RendProvFirst] [varchar](35) NULL,  
[RendProvMiddle] [varchar](25) NULL,  
[RendProvSuffix] [varchar](10) NULL,
```

```
[RendProvSpecialty] [varchar](10) NULL,  
[RendProvOtherIDQual1] [char](2) NULL,  
[RendProvOtherID1] [varchar](30) NULL,  
[RendProvOtherIDQual2] [char](2) NULL,  
[RendProvOtherID2] [varchar](30) NULL,  
[RendProvOtherIDQual3] [char](2) NULL,  
[RendProvOtherID3] [varchar](30) NULL,  
/*Attending Provider*/  
[AttendProvLast] [varchar](60) NULL,  
[AttendProvFirst] [varchar](35) NULL,  
[AttendProvMiddle] [varchar](25) NULL,  
[AttendProvSuffix] [varchar](10) NULL,  
[AttendProvIDType] [char](2) NULL,  
[AttendProvID] [varchar](30) NULL,  
[AttendProvTaxID] [varchar](12) NULL,  
[AttendProvNPI] [varchar](12) NULL,  
[AttendProvOtherIDQual1] [char](2) NULL,  
[AttendProvOtherID1] [varchar](30) NULL,  
[AttendProvSpecialty] [varchar](10) NULL,  
/*Operating Provider*/  
[OperatingProvLast] [varchar](60) NULL,  
[OperatingProvFirst] [varchar](35) NULL,  
[OperatingProvMiddle] [varchar](25) NULL,  
[OperatingProvSuffix] [varchar](10) NULL,  
[OperatingProvIDType] [char](2) NULL,  
[OperatingProvID] [varchar](30) NULL,
```

```
[OperatingProvNPI] [varchar](12) NULL,  
[OperatingProvTaxID] [varchar](12) NULL,  
[OperatingProvOtherIDQual1] [char](2) NULL,  
[OperatingProvOtherID1] [varchar](30) NULL,  
[OperatingProvSpecialty] [varchar](10) NULL,  
/*Other Provider*/  
[OtherProvLast] [varchar](60) NULL,  
[OtherProvFirst] [varchar](35) NULL,  
[OtherProvMiddle] [varchar](25) NULL,  
[OtherProvSuffix] [varchar](10) NULL,  
[OtherProvIDType] [char](2) NULL,  
[OtherProvID] [varchar](30) NULL,  
[OtherProvTaxID] [varchar](12) NULL,  
[OtherProvNPI] [varchar](12) NULL,  
[OtherProvOtherIDQual1] [char](2) NULL,  
[OtherProvOtherID1] [varchar](30) NULL,  
[OtherProvSpecialty] [varchar](50) NULL,  
/*Facility*/  
[FacilityType] [char](2) NULL,  
[FacilityIDType] [char](2) NULL,  
[FacilityID] [varchar](80) NULL,  
[FacilityNPI] [varchar](12) NULL,  
[FacilityTaxID] [varchar](12) NULL,  
[FacilityOtherIDQual1] [char](2) NULL,  
[FacilityOtherID1] [varchar](30) NULL,  
[FacilityName] [varchar](80) NULL,
```

```
[FacilityAddress] [varchar](55) NULL,  
[FacilityAddress2] [varchar](55) NULL,  
[FacilityCity] [varchar](30) NULL,  
[FacilitySpecialty] [varchar](50) NULL,  
[FacilityState] [char](2) NULL,  
[FacilityZip] [varchar](15) NULL,  
  
/*Referring Provider*/  
  
[RefProvLast] [varchar](60) NULL,  
[RefProvFirst] [varchar](35) NULL,  
[RefProvMiddle] [varchar](25) NULL,  
[RefProvSuffix] [varchar](10) NULL,  
[RefProvIDType] [char](2) NULL,  
[RefProvID] [varchar](80) NULL,  
[RefProvTaxID] [varchar](12) NULL,  
[RefProvNPI] [varchar](12) NULL,  
[RefProvOtherIDQual1] [char](2) NULL,  
[RefProvOtherID1] [varchar](300) NULL,  
[RefProvSpecialty] [varchar](20) NULL,  
  
/*Supervising Provider*/  
  
[SupervProvLast] [varchar](60) NULL,  
[SupervProvFirst] [varchar](35) NULL,  
[SupervProvMiddle] [varchar](25) NULL,  
[SupervProvSuffix] [varchar](10) NULL,  
[SupervProvIDType] [char](2) NULL,  
[SupervProvID] [varchar](30) NULL,  
[SupervProvTaxID] [varchar](12) NULL,
```

```
[SupervProvNPI] [varchar](12) NULL,  
[SupervProvOtherIDQual1] [char](2) NULL,  
[SupervProvOtherID1] [varchar](50) NULL,  
[SupervProvSpecialty] [varchar](20) NULL,  
/*Assistant Surgeon*/  
[AssistSurgLast] [varchar](60) NULL,  
[AssistSurgFirst] [varchar](35) NULL,  
[AssistSurgMiddle] [varchar](25) NULL,  
[AssistSurgSuffix] [varchar](10) NULL,  
[AssistSurgIDType] [char](2) NULL,  
[AssistSurgID] [varchar](30) NULL,  
[AssistSurgTaxID] [varchar](12) NULL,  
[AssistSurgNPI] [varchar](12) NULL,  
[AssistSurgOtherIDQual1] [varchar](3) NULL,  
[AssistSurgOtherID1] [varchar](50) NULL,  
[AssistSurgSpecialty] [varchar](10) NULL,  
/*Ambulance Pick-up Location*/  
[PickUpAddress] [varchar](55) NULL,  
[PickUpAddress2] [varchar](55) NULL,  
[PickUpCity] [varchar](30) NULL,  
[PickUpState] [char](2) NULL,  
[PickUpZip] [varchar](15) NULL,  
/*Ambulance Drop-off Location*/  
[DropOffName] [varchar](80) NULL,  
[DropOffAddress] [varchar](55) NULL,  
[DropOffAddress2] [varchar](55) NULL,
```

```
[DropOffCity] [varchar](30) NULL,  
[DropOffState] [char](2) NULL,  
[DropOffZip] [varchar](15) NULL,  
/*COB Payer 1*/  
[OtherInsuredLast] [varchar](60) NULL,  
[OtherInsuredFirst] [varchar](35) NULL,  
[OtherInsuredMiddle] [varchar](25) NULL,  
[OtherInsuredSuffix] [varchar](10) NULL,  
[OtherInsuredIDQual] [char](2) NULL,  
[OtherInsuredID] [varchar](30) NULL,  
[OtherInsuredAddress] [varchar](55) NULL,  
[OtherInsuredAddress2] [varchar](55) NULL,  
[OtherInsuredCity] [varchar](30) NULL,  
[OtherInsuredState] [char](2) NULL,  
[OtherInsuredZip] [varchar](15) NULL,  
[OtherInsuredCountry] [varchar](3) NULL,  
[OtherInsuredSubdivision] [varchar](3) NULL,  
[OtherInsuredPolicy] [varchar](50) NULL,  
[OtherInsuredDOB] [varchar](10) NULL,  
[OtherInsuredSex] [char](1) NULL,  
[OtherInsuredRelationship] [char](2) NULL,  
[OtherInsuredOtherIDQual1] [char](3) NULL,  
[OtherInsuredOtherID1] [varchar](50) NULL,  
[OtherInsuredOtherIDQual2] [char](3) NULL,  
[OtherInsuredOtherID2] [varchar](50) NULL,  
[OtherInsuredOtherIDQual3] [char](3) NULL,
```

```
[OtherInsuredOtherID3] [varchar](50) NULL,  
[OtherPayerResponsibility] [char](1) NULL,  
[OtherPayerName] [varchar](35) NULL,  
[OtherPayerIDQual] [char](2) NULL,  
[OtherPayerID] [varchar](30) NULL,  
[OtherPayerAddress1] [varchar](55) NULL,  
[OtherPayerAddress2] [varchar](55) NULL,  
[OtherPayerCity] [varchar](30) NULL,  
[OtherPayerState] [char](2) NULL,  
[OtherPayerZip] [varchar](15) NULL,  
[OtherPayerPaid] [varchar](10) NULL,  
[OtherPayerGroupName] [varchar](60) NULL,  
[OtherPayerGroupNo] [varchar](50) NULL,  
[OtherPayerInsuranceType] [char](2) NULL,  
[OtherPayerFilingIndicator] [char](2) NULL,  
[OtherPayerAdjustmentReasonGroup] [varchar](20) NULL,  
[OtherPayerAdjustmentReason] [varchar](50) NULL,  
[OtherPayerAdjustmentAmount] [varchar](120) NULL,  
[OtherPayerPaidDate] [date] NULL,  
[OtherPayerInfoRelease] [char](1) NULL,  
[OtherPayerBenefitAssign] [char](1) NULL,  
[OtherPayerDocControlNo] [varchar](50) NULL,  
[OtherPayerPriorAuth] [varchar](50) NULL,  
[OtherPayerOtherIDQual1] [char](3) NULL,  
[OtherPayerOtherID1] [varchar](50) NULL,  
[OtherPayerOtherIDQual2] [char](3) NULL,
```

```
[OtherPayerOtherID2] [varchar](50) NULL,  
  
/*COB Payer 2*/  
  
[OtherInsured2Last] [varchar](60) NULL,  
[OtherInsured2First] [varchar](35) NULL,  
[OtherInsured2Middle] [varchar](25) NULL,  
[OtherInsured2Suffix] [varchar](10) NULL,  
[OtherInsured2IDQual] [char](2) NULL,  
[OtherInsured2ID] [varchar](30) NULL,  
[OtherInsured2Address] [varchar](55) NULL,  
[OtherInsured2Address2] [varchar](55) NULL,  
[OtherInsured2City] [varchar](30) NULL,  
[OtherInsured2State] [char](2) NULL,  
[OtherInsured2Zip] [varchar](15) NULL,  
[OtherInsured2Country] [varchar](3) NULL,  
[OtherInsured2Subdivision] [varchar](3) NULL,  
[OtherInsured2Policy] [varchar](50) NULL,  
[OtherInsured2DOB] [varchar](10) NULL,  
[OtherInsured2Sex] [char](1) NULL,  
[OtherInsured2Relationship] [char](2) NULL,  
[OtherInsured2OtherIDQual1] [char](3) NULL,  
[OtherInsured2OtherID1] [varchar](50) NULL,  
[OtherInsured2OtherIDQual2] [char](3) NULL,  
[OtherInsured2OtherID2] [varchar](50) NULL,  
[OtherInsured2OtherIDQual3] [char](3) NULL,  
[OtherInsured2OtherID3] [varchar](50) NULL,  
[OtherPayer2Responsibility] [char](1) NULL,
```

```
[OtherPayer2Name] [varchar](35) NULL,  
[OtherPayer2IDQual] [char](2) NULL,  
[OtherPayer2ID] [varchar](80) NULL,  
[OtherPayer2Address1] [varchar](55) NULL,  
[OtherPayer2Address2] [varchar](55) NULL,  
[OtherPayer2City] [varchar](30) NULL,  
[OtherPayer2State] [char](2) NULL,  
[OtherPayer2Zip] [varchar](15) NULL,  
[OtherPayer2Paid] [varchar](10) NULL,  
[OtherPayer2GroupName] [varchar](60) NULL,  
[OtherPayer2GroupNo] [varchar](50) NULL,  
[OtherPayer2InsuranceType] [char](2) NULL,  
[OtherPayer2FilingIndicator] [char](2) NULL,  
[OtherPayer2AdjustmentReasonGroup] [char](20) NULL,  
[OtherPayer2AdjustmentReason] [varchar](50) NULL,  
[OtherPayer2AdjustmentAmount] [varchar](120) NULL,  
[OtherPayer2PaidDate] [date] NULL,  
[OtherPayer2InfoRelease] [char](1) NULL,  
[OtherPayer2BenefitAssign] [char](1) NULL,  
[OtherPayer2DocControlNo] [varchar](50) NULL,  
[OtherPayer2PriorAuth] [varchar](50) NULL,  
[OtherPayer2OtherIDQual1] [char](3) NULL,  
[OtherPayer2OtherID1] [varchar](50) NULL,  
[OtherPayer2OtherIDQual2] [char](3) NULL,  
[OtherPayer2OtherID2] [varchar](50) NULL,  
  
/*Claim Data*/
```

[ClaimNo] [varchar](50) NOT NULL,
[Amount] [money] NULL,
[EstimatedAmountDue] [money] NULL,
[PatientEstimatedAmountDue] [money] NULL,
[PlaceOfService] [char](2) NULL,
[ClaimFrequency] [char](1) NULL,
[SubmitReason] [char](1) NULL,
[ProviderSignature] [char](1) NULL,
[ProviderAcceptsAssignment] [char](1) NULL,
[BenefitAssignment] [char](1) NULL,
[InfoReleaseCode] [char](1) NULL,
[PatientSignatureCode] [char](1) NULL,
[RelatedCauses] [char](2) NULL,
[RelatedCauses2] [char](2) NULL,
[RelatedCausesState] [char](2) NULL,
[SpecialProgramCode] [char](3) NULL,
[ProviderParticipation] [char](1) NULL,
[EOBIndicator] [char](1) NULL,
[DelayReasonCode] [char](2) NULL,
[ServiceDateFrom] [date] NULL,
[ServiceDateTo] [date] NULL,
[OnsetDate] [date] NULL,
[SimilarSymptomsDate] [date] NULL,
[DisabilityBegin] [date] NULL,
[DisabilityEnd] [date] NULL,
[HospitalizationBegin] [date] NULL,

```
[HospitalizationEnd] [date] NULL,  
[AccidentDate] [date] NULL,  
[LastMenstrualPeriod] [date] NULL,  
[RepricerReceivedDate] [date] NULL,  
[AdmissionDate] [date] NULL,  
[AdmissionHour] [time](7) NULL,  
[AdmissionType] [char](1) NULL,  
[AdmissionSource] [char](1) NULL,  
[DischargeHour] [time](7) NULL,  
[PatientStatus] [char](2) NULL,  
[CoveredDays] [varchar](4) NULL,  
[NonCoveredDays] [varchar](4) NULL,  
[COBDays] [varchar](4) NULL,  
[LifeTimeReserveDays] [varchar](4) NULL,  
[PriorAuthorization] [varchar](50) NULL,  
[ClearingHouseID] [varchar](50) NULL,  
[MedicalRecordNumber] [varchar](50) NULL,  
[MothersMedicalRecordNumber] [varchar](50) NULL,  
[ServiceAuthorizationException] [varchar](50) NULL,  
[ReferralNumber] [varchar](50) NULL,  
[PayerClaimControlNumber] [varchar](50) NULL,  
[AdjustedRepricedClaimNumber] [varchar](50) NULL,  
[AutoAccidentState] [varchar](50) NULL,  
[OrigRefNo] [varchar](50) NULL,  
[ClaimType] [char](1) NULL,  
[TypeOfBill] [varchar](20) NULL,
```

```
[Remark1] [varchar](90) NULL,  
[Remark2] [varchar](90) NULL,  
[Remark3] [varchar](90) NULL,  
[Remark4] [varchar](90) NULL,  
[K3_1] [varchar](80) NULL,  
[K3_2] [varchar](80) NULL,  
[K3_3] [varchar](80) NULL,  
[K3_4] [varchar](80) NULL,  
[K3_5] [varchar](80) NULL,  
[K3_6] [varchar](80) NULL,  
[K3_7] [varchar](80) NULL,  
[K3_8] [varchar](80) NULL,  
[K3_9] [varchar](80) NULL,  
[K3_10] [varchar](80) NULL,  
[OutsideLab] [char](1) NULL,  
[LabCharge] [varchar](10) NULL,  
[Test_Prod] [char](1) NULL,  
[ReportTypeCode1] [char](2) NULL,  
[ReportTransmissionCode1] [varchar](2) NULL,  
[AttachmentControlNumber1] [varchar](80) NULL,  
/*Dental Claims*/  
[Predetermination] [char](2) NULL,  
[OrthodonticTotal] [varchar](5) NULL,  
[OrthodonticRemaining] [varchar](5) NULL,  
[OrthodonticYesNo] [char](1) NULL,  
[ToothStatus] [varchar](100) NULL,
```

```
[AppliancePlacementDate] [date] NULL,  
  
/*Diagnosis Information*/  
  
[AdmitDiagnosis] [varchar](10) NULL,  
  
[ECode] [varchar](30) NULL,  
  
[ECode2] [varchar](30) NULL,  
  
[ECode3] [varchar](30) NULL,  
  
[ECode4] [varchar](30) NULL,  
  
[ECode5] [varchar](30) NULL,  
  
[ECode6] [varchar](30) NULL,  
  
[ECode7] [varchar](30) NULL,  
  
[ECode8] [varchar](30) NULL,  
  
--[ECode9] [varchar](30) NULL,  
  
--[ECode10] [varchar](30) NULL,  
  
--[ECode11] [varchar](30) NULL,  
  
--[ECode12] [varchar](30) NULL,  
  
[ReasonForVisit] [varchar](30) NULL,  
  
[ReasonForVisit2] [varchar](30) NULL,  
  
[ReasonForVisit3] [varchar](30) NULL,  
  
[PrincipalDiagnosis] [varchar](10) NULL,  
  
[Diag2] [varchar](10) NULL,  
  
[Diag3] [varchar](10) NULL,  
  
[Diag4] [varchar](10) NULL,  
  
[Diag5] [varchar](10) NULL,  
  
[Diag6] [varchar](10) NULL,  
  
[Diag7] [varchar](10) NULL,  
  
[Diag8] [varchar](10) NULL,
```

```
[Diag9] [varchar](10) NULL,  
[Diag10] [varchar](10) NULL,  
[Diag11] [varchar](10) NULL,  
[Diag12] [varchar](10) NULL,  
[Diag13] [varchar](10) NULL,  
[Diag14] [varchar](10) NULL,  
[Diag15] [varchar](10) NULL,  
[Diag16] [varchar](10) NULL,  
[Diag17] [varchar](10) NULL,  
[Diag18] [varchar](10) NULL,  
[Diag19] [varchar](10) NULL,  
[Diag20] [varchar](10) NULL,  
[Diag21] [varchar](10) NULL,  
[Diag22] [varchar](10) NULL,  
[Diag23] [varchar](10) NULL,  
[Diag24] [varchar](10) NULL,  
[Diag25] [varchar](10) NULL,  
[DRG] [varchar](15) NULL,  
  
/*Procedure Information*/  
[PrincipalProcedure] [varchar](10) NULL,  
[PrincipalProcedureDate] [datetime2](0) NULL,  
[Proc2] [varchar](50) NULL,  
[Proc2Date] [datetime2](0) NULL,  
[Proc3] [varchar](10) NULL,  
[Proc3Date] [datetime2](0) NULL,  
[Proc4] [varchar](10) NULL,
```

```
[Proc4Date] [datetime2](0) NULL,  
[Proc5] [varchar](10) NULL,  
[Proc5Date] [datetime2](0) NULL,  
[Proc6] [varchar](10) NULL,  
[Proc6Date] [datetime2](0) NULL,  
[Proc7] [varchar](10) NULL,  
[Proc7Date] [datetime2](0) NULL,  
[Proc8] [varchar](10) NULL,  
[Proc8Date] [datetime2](0) NULL,  
[Proc9] [varchar](10) NULL,  
[Proc9Date] [datetime2](0) NULL,  
[Proc10] [varchar](10) NULL,  
[Proc10Date] [datetime2](0) NULL,  
[Proc11] [varchar](10) NULL,  
[Proc11Date] [datetime2](0) NULL,  
[Proc12] [varchar](10) NULL,  
[Proc12Date] [datetime2](0) NULL,  
[Proc13] [varchar](10) NULL,  
[Proc13Date] [datetime2](0) NULL,  
[Proc14] [varchar](10) NULL,  
[Proc14Date] [datetime2](0) NULL,  
[Proc15] [varchar](10) NULL,  
[Proc15Date] [datetime2](0) NULL,  
[Proc16] [varchar](10) NULL,  
[Proc16Date] [datetime2](0) NULL,  
[Proc17] [varchar](10) NULL,
```

```
[Proc17Date] [datetime2](0) NULL,  
[Proc18] [varchar](10) NULL,  
[Proc18Date] [datetime2](0) NULL,  
[Proc19] [varchar](10) NULL,  
[Proc19Date] [datetime2](0) NULL,  
[Proc20] [varchar](10) NULL,  
[Proc20Date] [datetime2](0) NULL,  
[Proc21] [varchar](10) NULL,  
[Proc21Date] [datetime2](0) NULL,  
[Proc22] [varchar](10) NULL,  
[Proc22Date] [datetime2](0) NULL,  
[Proc23] [varchar](10) NULL,  
[Proc23Date] [datetime2](0) NULL,  
[Proc24] [varchar](10) NULL,  
[Proc24Date] [datetime2](0) NULL,  
[Proc25] [varchar](10) NULL,  
[Proc25Date] [datetime2](0) NULL,  
  
/*Value Codes*/  
[ValueCode1] [char](3) NULL,  
[ValueAmount1] [varchar](10) NULL,  
[ValueCode2] [char](3) NULL,  
[ValueAmount2] [varchar](10) NULL,  
[ValueCode3] [char](3) NULL,  
[ValueAmount3] [varchar](10) NULL,  
[ValueCode4] [char](3) NULL,  
[ValueAmount4] [varchar](10) NULL,
```

```
[ValueCode5] [char](3) NULL,  
[ValueAmount5] [varchar](10) NULL,  
[ValueCode6] [char](3) NULL,  
[ValueAmount6] [varchar](10) NULL,  
[ValueCode7] [varchar](10) NULL,  
[ValueAmount7] [varchar](10) NULL,  
[ValueCode8] [char](3) NULL,  
[ValueAmount8] [varchar](10) NULL,  
[ValueCode9] [char](3) NULL,  
[ValueAmount9] [varchar](10) NULL,  
[ValueCode10] [char](3) NULL,  
[ValueAmount10] [varchar](10) NULL,  
[ValueCode11] [char](3) NULL,  
[ValueAmount11] [varchar](10) NULL,  
[ValueCode12] [char](3) NULL,  
[ValueAmount12] [varchar](10) NULL,  
--[ValueCode13] [char](3) NULL,  
--[ValueAmount13] [varchar](10) NULL,  
--[ValueCode14] [char](3) NULL,  
--[ValueAmount14] [varchar](10) NULL,  
--[ValueCode15] [char](3) NULL,  
--[ValueAmount15] [varchar](10) NULL,  
--[ValueCode16] [char](3) NULL,  
--[ValueAmount16] [varchar](10) NULL,  
--[ValueCode17] [char](3) NULL,  
--[ValueAmount17] [varchar](10) NULL,
```

```
--[ValueCode18] [char](3) NULL,  
--[ValueAmount18] [varchar](10) NULL,  
--[ValueCode19] [char](3) NULL,  
--[ValueAmount19] [varchar](10) NULL,  
--[ValueCode20] [char](3) NULL,  
--[ValueAmount20] [varchar](10) NULL,  
--[ValueCode21] [char](3) NULL,  
--[ValueAmount21] [varchar](10) NULL,  
--[ValueCode22] [char](3) NULL,  
--[ValueAmount22] [varchar](10) NULL,  
--[ValueCode23] [char](3) NULL,  
--[ValueAmount23] [varchar](10) NULL,  
--[ValueCode24] [char](3) NULL,  
--[ValueAmount24] [varchar](10) NULL,  
  
/*Condition Codes*/  
[ConditionCode1] [varchar](30) NULL,  
[ConditionCode2] [varchar](30) NULL,  
[ConditionCode3] [varchar](30) NULL,  
[ConditionCode4] [varchar](30) NULL,  
[ConditionCode5] [varchar](30) NULL,  
[ConditionCode6] [varchar](30) NULL,  
[ConditionCode7] [varchar](30) NULL,  
[ConditionCode8] [varchar](30) NULL,  
[ConditionCode9] [varchar](30) NULL,  
[ConditionCode10] [varchar](30) NULL,  
--[ConditionCode11] [varchar](30) NULL,
```

```
--[ConditionCode12] [varchar](30) NULL,  
--[ConditionCode13] [varchar](30) NULL,  
--[ConditionCode14] [varchar](30) NULL,  
--[ConditionCode15] [varchar](30) NULL,  
--[ConditionCode16] [varchar](30) NULL,  
--[ConditionCode17] [varchar](30) NULL,  
--[ConditionCode18] [varchar](30) NULL,  
--[ConditionCode19] [varchar](30) NULL,  
--[ConditionCode20] [varchar](30) NULL,  
--[ConditionCode21] [varchar](30) NULL,  
--[ConditionCode22] [varchar](30) NULL,  
--[ConditionCode23] [varchar](30) NULL,  
--[ConditionCode24] [varchar](30) NULL,  
  
/*Occurrence Codes*/  
  
[OccurrenceCode1] [char](3) NULL,  
[OccurrenceDate1] [date] NULL,  
[OccurrenceCode2] [char](3) NULL,  
[OccurrenceDate2] [date] NULL,  
[OccurrenceCode3] [char](3) NULL,  
[OccurrenceDate3] [date] NULL,  
[OccurrenceCode4] [char](3) NULL,  
[OccurrenceDate4] [date] NULL,  
[OccurrenceCode5] [char](3) NULL,  
[OccurrenceDate5] [date] NULL,  
[OccurrenceCode6] [char](3) NULL,  
[OccurrenceDate6] [date] NULL,
```

```
[OccurrenceCode7] [char](3) NULL,  
[OccurrenceDate7] [date] NULL,  
[OccurrenceCode8] [char](3) NULL,  
[OccurrenceDate8] [date] NULL,  
--[OccurrenceCode9] [char](3) NULL,  
--[OccurrenceDate9] [date] NULL,  
--[OccurrenceCode10] [char](3) NULL,  
--[OccurrenceDate10] [date] NULL,  
--[OccurrenceCode11] [char](3) NULL,  
--[OccurrenceDate11] [date] NULL,  
--[OccurrenceCode12] [char](3) NULL,  
--[OccurrenceDate12] [date] NULL,  
--[OccurrenceCode13] [char](3) NULL,  
--[OccurrenceDate13] [date] NULL,  
--[OccurrenceCode14] [char](3) NULL,  
--[OccurrenceDate14] [date] NULL,  
--[OccurrenceCode15] [char](3) NULL,  
--[OccurrenceDate15] [date] NULL,  
--[OccurrenceCode16] [char](3) NULL,  
--[OccurrenceDate16] [date] NULL,  
--[OccurrenceCode17] [char](3) NULL,  
--[OccurrenceDate17] [date] NULL,  
--[OccurrenceCode18] [char](3) NULL,  
--[OccurrenceDate18] [date] NULL,  
--[OccurrenceCode19] [char](3) NULL,  
--[OccurrenceDate19] [date] NULL,
```

```
--[OccuranceCode20] [char](3) NULL,  
--[OccuranceDate20] [date] NULL,  
--[OccuranceCode21] [char](3) NULL,  
--[OccuranceDate21] [date] NULL,  
--[OccuranceCode22] [char](3) NULL,  
--[OccuranceDate22] [date] NULL,  
--[OccuranceCode23] [char](3) NULL,  
--[OccuranceDate23] [date] NULL,  
--[OccuranceCode24] [char](3) NULL,  
--[OccuranceDate24] [date] NULL,  
[OccuranceSpanCode1] [char](3) NULL,  
[OccuranceSpanFrom1] [date] NULL,  
[OccuranceSpanTo1] [date] NULL,  
[OccuranceSpanCode2] [char](3) NULL,  
[OccuranceSpanFrom2] [date] NULL,  
[OccuranceSpanTo2] [date] NULL,  
[OccuranceSpanCode3] [char](3) NULL,  
[OccuranceSpanFrom3] [date] NULL,  
[OccuranceSpanTo3] [date] NULL,  
[OccuranceSpanCode4] [char](3) NULL,  
[OccuranceSpanFrom4] [date] NULL,  
[OccuranceSpanTo4] [date] NULL,  
  
/*CR1*/  
[PatientWeight] [varchar](10) NULL,  
[AmbulanceTransportCode] [char](1) NULL,  
[AmbulanceTransportReasonCode] [char](1) NULL,
```

```
[TransportDistance] [varchar](15) NULL,  
[RoundTripPurposeDescription] [varchar](80) NULL,  
[StretcherPurposeDescription] [varchar](80) NULL,  
/* CRC - Ambulance Certification */  
[AmbulanceConditionIndicator] [char](1) NULL,  
[AmbulanceConditionCode1] [char](3) NULL,  
[AmbulanceConditionCode2] [char](3) NULL,  
[AmbulanceConditionCode3] [char](3) NULL,  
[AmbulanceConditionCode4] [char](3) NULL,  
[AmbulanceConditionCode5] [char](3) NULL,  
/*Repricing*/  
[RepricedClaimNumber] [varchar](50) NULL,  
[RepricingMethodology] [varchar](3) NULL,  
[RepricedAmount] [money] NULL,  
[SavingsAmount] [money] NULL,  
[RepricerID] [varchar](30) NULL,  
[RepricingRate] [varchar](9) NULL,  
[APG_Code] [varchar](30) NULL,  
[APG_Amount] [money] NULL,  
[ApprovedRevenueCode] [varchar] (48) NULL,  
[ApprovedProcedureCode] [varchar] (48) NULL,  
[ApprovedUnitCode] [char](2) NULL,  
[ApprovedUnits] [bigint] NULL,  
[RejectReason] [varchar](3) NULL,  
[ComplianceCode] [varchar](2) NULL,  
[ExceptionCode] [varchar](2) NULL
```

```
/*SQL To EDI*/  
  
--,[ClaimAction] [char](1) NULL,  
  
--[ErrorMessage] [varchar](1000) NULL  
  
) ON [PRIMARY]
```

Detail

```
CREATE TABLE [dbo].[edi_ClaimDetail](  
  
    [ID] [bigint] IDENTITY(1,1) NOT NULL,  
  
    [ClaimID] [bigint] NOT NULL,  
  
    [LineNumber] [decimal](18, 0) NULL,  
  
    [LineID] [varchar](50) NULL,  
  
    [ServiceDateFrom] [datetime2](0) NULL,  
  
    [ServiceDateTo] [date] NULL,  
  
    [AssessmentDate] [date] NULL,  
  
    [FacilityCode] [char](2) NULL,  
  
    [RevenueCode] [varchar](48) NULL,  
  
    [ProcedureCode] [varchar](48) NULL,  
  
    [Amount] [money] NULL,  
  
    [Unit] [char](2) NULL,  
  
    [Quantity] [float] NULL,  
  
    [UnitRate] [money] NULL,  
  
    [NonCovered] [money] NULL,  
  
    [Paid] [money] NULL,  
  
    [PlaceOfService] [char](3) NULL,  
  
    [Modifier1] [char](3) NULL,  
  
    [Modifier2] [char](3) NULL,
```

```
[Modifier3] [char](3) NULL,  
[Modifier4] [char](3) NULL,  
[ProcedureDescription] [varchar](80) NULL,  
[OralCavityDesignation1] [char](3) NULL,  
[OralCavityDesignation2] [char](3) NULL,  
[OralCavityDesignation3] [char](3) NULL,  
[OralCavityDesignation4] [char](3) NULL,  
[OralCavityDesignation5] [char](3) NULL,  
[DiagPointer1] [char](1) NULL,  
[DiagPointer2] [char](1) NULL,  
[DiagPointer3] [char](1) NULL,  
[DiagPointer4] [char](1) NULL,  
[ToothNumber] [char](2) NULL,  
[Surface] [varchar](10) NULL,  
[EmergencyIndicator] [char](1) NULL,  
[ServiceTax] [money] NULL,  
[FacilityTax] [money] NULL,  
[SalesTax] [money] NULL,  
[ApprovedAmount] [money] NULL,  
[LineK3_01] [varchar](80) NULL,  
[LineK3_02] [varchar](80) NULL,  
[LineK3_03] [varchar](80) NULL,  
[LineK3_04] [varchar](80) NULL,  
[LineK3_05] [varchar](80) NULL,  
[LineK3_06] [varchar](80) NULL,  
[LineK3_07] [varchar](80) NULL,
```

```
[LineK3_08] [varchar](80) NULL,  
[LineK3_09] [varchar](80) NULL,  
[LineK3_10] [varchar](80) NULL,  
[Remark] [varchar](100) NULL,  
[AmbulancePatientCount] [bigint] NULL,  
  
/****PWK****/  
  
[ReportType] [char](2) NULL,  
[ReportTransmission] [char](2) NULL,  
[AttachmentControlNumber] [varchar](80) NULL,  
  
/*HCP*/  
  
[RepricingMethodology] [varchar](3) NULL,  
[RepricedAmount] [money] NULL,  
[SavingsAmount] [money] NULL,  
[RepricerID] [varchar](30) NULL,  
[RepricingRate] [varchar](9) NULL,  
[APG_Code] [varchar](30) NULL,  
[APG_Amount] [money] NULL,  
[ApprovedRevenueCode] [varchar] (48) NULL,  
[ApprovedProcedureCode] [varchar] (48) NULL,  
[ApprovedUnitCode] [char](2) NULL,  
[ApprovedUnits] [bigint] NULL,  
[RejectReason] [varchar](3) NULL,  
[ComplianceCode] [varchar](2) NULL,  
[ExceptionCode] [varchar](2) NULL,  
  
/* Drugs */  
  
[DrugCode] [varchar](48) NULL,
```

```
[DrugUnitPrice] [money] NULL,  
[DrugUnitCode] [char](2) NULL,  
[DrugUnits] [float] NULL,  
[PrescriptionNumber] [varchar](30) NULL,  
/* CR1 */  
[PatientWeight] [varchar](10) NULL,  
[AmbulanceTransportCode] [char](1) NULL,  
[AmbulanceTransportReasonCode] [char](1) NULL,  
[TransportDistance] [varchar](15) NULL,  
[RoundTripPurposeDescription] [varchar](80) NULL,  
[StretcherPurposeDescription] [varchar](80) NULL,  
/* CRC - Ambulance Certification */  
[AmbulanceConditionIndicator] [char](1) NULL,  
[AmbulanceConditionCode1] [char](3) NULL,  
[AmbulanceConditionCode2] [char](3) NULL,  
[AmbulanceConditionCode3] [char](3) NULL,  
[AmbulanceConditionCode4] [char](3) NULL,  
[AmbulanceConditionCode5] [char](3) NULL,  
/* Attending Provider */  
[AttendingProviderLast] [varchar](35) NULL,  
[AttendingProviderFirst] [varchar](25) NULL,  
[AttendingProviderMiddle] [varchar](25) NULL,  
[AttendingProviderSuffix] [varchar](10) NULL,  
[AttendingProviderIDQual] [char](2) NULL,  
[AttendingProviderID] [varchar](35) NULL,  
[AttendingProviderOtherIDQual] [char](2) NULL,
```

```
[AttendingProviderOtherID] [varchar](30) NULL,  
  
/* Operating Provider */  
  
[OperatingProviderLast] [varchar](35) NULL,  
  
[OperatingProviderFirst] [varchar](25) NULL,  
  
[OperatingProviderMiddle] [varchar](25) NULL,  
  
[OperatingProviderSuffix] [varchar](10) NULL,  
  
[OperatingProviderIDQual] [char](2) NULL,  
  
[OperatingProviderID] [varchar](35) NULL,  
  
[OperatingProviderOtherIDQual] [char](2) NULL,  
  
[OperatingProviderOtherID] [varchar](30) NULL,  
  
/* Other Provider */  
  
[OtherProviderLast] [varchar](35) NULL,  
  
[OtherProviderFirst] [varchar](25) NULL,  
  
[OtherProviderMiddle] [varchar](25) NULL,  
  
[OtherProviderSuffix] [varchar](10) NULL,  
  
[OtherProviderIDQual] [char](2) NULL,  
  
[OtherProviderID] [varchar](35) NULL,  
  
[OtherProviderOtherIDQual] [char](2) NULL,  
  
[OtherProviderOtherID] [varchar](30) NULL,  
  
/* Rendering Provider */  
  
[RenderingProviderLast] [varchar](35) NULL,  
  
[RenderingProviderFirst] [varchar](25) NULL,  
  
[RenderingProviderMiddle] [varchar](25) NULL,  
  
[RenderingProviderSuffix] [varchar](10) NULL,  
  
[RenderingProviderIDQual] [char](2) NULL,  
  
[RenderingProviderID] [varchar](35) NULL,
```

```
[RenderingProviderOtherIDQual] [char](2) NULL,  
[RenderingProviderOtherID] [varchar](30) NULL,  
  
/* PurchasedService Provider */  
[PurchasedServiceProviderLast] [varchar](35) NULL,  
[PurchasedServiceProviderFirst] [varchar](25) NULL,  
[PurchasedServiceProviderMiddle] [varchar](25) NULL,  
[PurchasedServiceProviderSuffix] [varchar](10) NULL,  
[PurchasedServiceProviderIDQual] [char](2) NULL,  
[PurchasedServiceProviderID] [varchar](35) NULL,  
[PurchasedServiceProviderOtherIDQual] [char](2) NULL,  
[PurchasedServiceProviderOtherID] [varchar](30) NULL,  
  
/* Facility Provider */  
[FacilityName] [varchar](35) NULL,  
[FacilityIDQual] [char](2) NULL,  
[FacilityID] [varchar](35) NULL,  
[FacilityAddress1] [varchar](25) NULL,  
[FacilityAddress2] [varchar](25) NULL,  
[FacilityCity] [varchar](30) NULL,  
[FacilityState] [char](2) NULL,  
[FacilityZip] [varchar](15) NULL,  
[FacilityOtherIDQual] [char](2) NULL,  
[FacilityOtherID] [varchar](30) NULL,  
  
/* Supervising Provider */  
[SupervisingProviderLast] [varchar](35) NULL,  
[SupervisingProviderFirst] [varchar](25) NULL,  
[SupervisingProviderMiddle] [varchar](25) NULL,
```

```
[SupervisingProviderSuffix] [varchar](10) NULL,  
[SupervisingProviderIDQual] [char](2) NULL,  
[SupervisingProviderID] [varchar](35) NULL,  
[SupervisingProviderOtherIDQual] [char](2) NULL,  
[SupervisingProviderOtherID] [varchar](30) NULL,  
  
/* Ordering Provider */  
  
[OrderingProviderLast] [varchar](35) NULL,  
[OrderingProviderFirst] [varchar](25) NULL,  
[OrderingProviderMiddle] [varchar](25) NULL,  
[OrderingProviderSuffix] [varchar](10) NULL,  
[OrderingProviderIDQual] [char](2) NULL,  
[OrderingProviderID] [varchar](35) NULL,  
[OrderingProviderOtherIDQual] [char](2) NULL,  
[OrderingProviderOtherID] [varchar](30) NULL,  
  
/* Referring Provider */  
  
[ReferringProviderLast] [varchar](35) NULL,  
[ReferringProviderFirst] [varchar](25) NULL,  
[ReferringProviderMiddle] [varchar](25) NULL,  
[ReferringProviderSuffix] [varchar](10) NULL,  
[ReferringProviderIDQual] [char](2) NULL,  
[ReferringProviderID] [varchar](35) NULL,  
[ReferringProviderOtherIDQual] [char](2) NULL,  
[ReferringProviderOtherID] [varchar](30) NULL,  
  
/*COB 1 */  
  
[OtherPayer1ID] [varchar](30) NULL,  
  
[OtherPayer1Paid] [money] NULL,
```

```
[OtherPayer1PaidProcedure] [varchar](30) NULL,  
[OtherPayer1PaidRevenueCode] [varchar](30) NULL,  
[OtherPayer1PaidQuantity] [bigint] NULL,  
[OtherPayer1BundledLine] [int] NULL,  
[OtherPayer1AdjustmentReasonGroup1] [char](2) NULL,  
[OtherPayer1AdjustmentReason1] [varchar](5) NULL,  
[OtherPayer1AdjustmentAmount1] [money] NULL,  
[OtherPayer1AdjustmentQuantity1] [bigint] NULL,  
[OtherPayer1AdjustmentReasonGroup2] [char](2) NULL,  
[OtherPayer1AdjustmentReason2] [varchar](5) NULL,  
[OtherPayer1AdjustmentAmount2] [money] NULL,  
[OtherPayer1AdjustmentQuantity2] [bigint] NULL,  
[OtherPayer1AdjustmentReasonGroup3] [char](2) NULL,  
[OtherPayer1AdjustmentReason3] [varchar](5) NULL,  
[OtherPayer1AdjustmentAmount3] [money] NULL,  
[OtherPayer1AdjustmentQuantity3] [bigint] NULL,  
[OtherPayer1AdjustmentReasonGroup4] [char](2) NULL,  
[OtherPayer1AdjustmentReason4] [varchar](5) NULL,  
[OtherPayer1AdjustmentAmount4] [money] NULL,  
[OtherPayer1AdjustmentQuantity4] [bigint] NULL,  
--[OtherPayer1AdjustmentReasonGroup5Plus] [varchar](77) NULL,  
--[OtherPayer1AdjustmentReason5Plus] [varchar](155) NULL,  
--[OtherPayer1AdjustmentAmount5Plus] [varchar](493) NULL,  
--[OtherPayer1AdjustmentQuantity5Plus] [varchar](415) NULL,  
[OtherPayer1PaidDate] [date] NULL,  
  
/*COB 2 */
```

```
[OtherPayer2ID] [varchar](30) NULL,  
[OtherPayer2Paid] [money] NULL,  
[OtherPayer2PaidProcedure] [varchar](30) NULL,  
[OtherPayer2PaidRevenueCode] [varchar](30) NULL,  
[OtherPayer2PaidQuantity] [bigint] NULL,  
[OtherPayer2BundledLine] [int] NULL,  
[OtherPayer2AdjustmentReasonGroup1] [char](2) NULL,  
[OtherPayer2AdjustmentReason1] [varchar](5) NULL,  
[OtherPayer2AdjustmentAmount1] [money] NULL,  
[OtherPayer2AdjustmentQuantity1] [bigint] NULL,  
[OtherPayer2AdjustmentReasonGroup2] [char](2) NULL,  
[OtherPayer2AdjustmentReason2] [varchar](5) NULL,  
[OtherPayer2AdjustmentAmount2] [money] NULL,  
[OtherPayer2AdjustmentQuantity2] [bigint] NULL,  
[OtherPayer2AdjustmentReasonGroup3] [char](2) NULL,  
[OtherPayer2AdjustmentReason3] [varchar](5) NULL,  
[OtherPayer2AdjustmentAmount3] [money] NULL,  
[OtherPayer2AdjustmentQuantity3] [bigint] NULL,  
[OtherPayer2AdjustmentReasonGroup4] [char](2) NULL,  
[OtherPayer2AdjustmentReason4] [varchar](5) NULL,  
[OtherPayer2AdjustmentAmount4] [money] NULL,  
[OtherPayer2AdjustmentQuantity4] [bigint] NULL,  
--[OtherPayer2AdjustmentReasonGroup5Plus] [varchar](77) NULL,  
--[OtherPayer2AdjustmentReason5Plus] [varchar](155) NULL,  
--[OtherPayer2AdjustmentAmount5Plus] [varchar](493) NULL,  
--[OtherPayer2AdjustmentQuantity5Plus] [varchar](415) NULL,
```

```
[OtherPayer2PaidDate] [date] NULL  
) ON [PRIMARY]
```

5.3.2 MySQL Script to Create Tables

This is a free, industrial strength Open Source database available at www.MySQL.com

Header

```
CREATE TABLE `EDI_Claims`(  
    `ID` int(11) NOT NULL auto_increment,  
    /*File Level Data*/  
    `Filename` varchar(100) default NULL,  
    `ImagePath` varchar(300) default NULL,  
    `ImageFilename` varchar(100) default NULL,  
    `Version` varchar(12) default NULL,  
    `TradingPartnerIDType` char(2) default NULL,  
    `TradingPartnerID` varchar(15) NOT NULL,  
    `TransactionDate` varchar(15) default NULL,  
    `ReceiveDate` varchar(15) default NULL,  
    `SubmitterName` varchar(60) default NULL,  
    `SubmitterID` varchar(30) default NULL,  
    `SubmitterContact` varchar(60) default NULL,  
    `SubmitterTel` varchar(20) default NULL,  
    `SubmitterTelExt` varchar(10) default NULL,  
    `SubmitterFax` varchar(20) default NULL,  
    `SubmitterEmail` varchar(80) default NULL,  
    `ReceiverName` varchar(60) default NULL,  
    `ReceiverID` varchar(30) default NULL,
```

```
`TransactionType` char(2) default NULL,  
  
/*Billing Provider*/  
  
`FedTaxID` varchar(15) default NULL,  
  
`BillProvIDType` char(2) default NULL,  
  
`BillProvID` varchar(80) default NULL,  
  
`BillProvNPI` varchar(12) default NULL,  
  
`BillProvLast` varchar(60) default NULL,  
  
`BillProvFirst` varchar(35) default NULL,  
  
`BillProvMiddle` varchar(25) default NULL,  
  
`BillProvSuffix` varchar(10) default NULL,  
  
`BillProvSpecialty` varchar(10) default NULL,  
  
`BillProvAddress` varchar(55) default NULL,  
  
`BillProvAddress2` varchar(55) default NULL,  
  
`BillProvCity` varchar(30) default NULL,  
  
`BillProvState` char(2) default NULL,  
  
`BillProvZip` varchar(15) default NULL,  
  
`BillProvCountry` varchar(3) default NULL,  
  
`BillProvSubdivision` varchar(3) default NULL,  
  
`BillProvContact` varchar(60) default NULL,  
  
`BillProvTel` varchar(80) default NULL,  
  
`BillProvTelExt` varchar(80) default NULL,  
  
`BillProvFax` varchar(80) default NULL,  
  
`BillProvEmail` varchar(80) default NULL,  
  
`BillProvOtherIDQual1` varchar(100) default NULL,  
  
`BillProvOtherID1` varchar(100) default NULL,  
  
`BillProvOtherIDQual2` varchar(100) default NULL,
```

```
`BillProvOtherID2` varchar(100) default NULL,  
`BillProvOtherIDQual3` varchar(100) default NULL,  
`BillProvOtherID3` varchar(100) default NULL,  
`BillProvOtherIDQual4` varchar(100) default NULL,  
`BillProvOtherID4` varchar(100) default NULL,  
`BillProvOtherIDQual5` varchar(100) default NULL,  
`BillProvOtherID5` varchar(100) default NULL,  
  
/*Pay-to Provider*/  
  
`PayToProvIDType` char(2) default NULL,  
`PayToProvID` varchar(80) default NULL,  
`PayToProvTaxID` varchar(12) default NULL,  
`PayToProvNPI` varchar(12) default NULL,  
`PayToProvLast` varchar(60) default NULL,  
`PayToProvFirst` varchar(35) default NULL,  
`PayToProvMiddle` varchar(25) default NULL,  
`PayToProvSuffix` varchar(10) default NULL,  
`PayToProvSpecialty` varchar(10) default NULL,  
`PayToProvAddress` varchar(55) default NULL,  
`PayToProvAddress2` varchar(55) default NULL,  
`PayToProvCity` varchar(30) default NULL,  
`PayToProvState` char(2) default NULL,  
`PayToProvZip` varchar(15) default NULL,  
  
/*Pay-to Payer*/  
  
`PayToPayerIDType` char(2) default NULL,  
`PayToPayerID` varchar(80) default NULL,  
`PayToPayerTaxID` varchar(12) default NULL,
```

```
`PayToPayerLast` varchar(60) default NULL,  
`PayToPayerFirst` varchar(35) default NULL,  
`PayToPayerMiddle` varchar(25) default NULL,  
`PayToPayerSuffix` varchar(10) default NULL,  
`PayToPayerAddress` varchar(55) default NULL,  
`PayToPayerAddress2` varchar(55) default NULL,  
`PayToPayerCity` varchar(30) default NULL,  
`PayToPayerState` char(2) default NULL,  
`PayToPayerZip` varchar(15) default NULL,  
`PayToPayerOtherIDQual` char(2) default NULL,  
`PayToPayerOtherID` varchar(30) default NULL,  
/*Subscriber*/  
`SubscriberLast` varchar(60) default NULL,  
`SubscriberFirst` varchar(35) default NULL,  
`SubscriberMiddle` varchar(25) default NULL,  
`SubscriberSuffix` varchar(10) default NULL,  
`SubscriberIDType` char(2) default NULL,  
`SubscriberID` varchar(80) default NULL,  
`SubscriberAddress` varchar(55) default NULL,  
`SubscriberAddress2` varchar(55) default NULL,  
`SubscriberCity` varchar(30) default NULL,  
`SubscriberState` char(2) default NULL,  
`SubscriberZip` varchar(15) default NULL,  
`SubscriberCountry` varchar(3) default NULL,  
`SubscriberLocation` varchar(30) default NULL,  
`SubscriberSubdivision` varchar(3) default NULL,
```

```
`SubscriberDOB` varchar(10) default NULL,  
`SubscriberSex` char(1) default NULL,  
`SubscriberEthnicity` varchar(75) default NULL,  
`SubscriberMaritalStatus` char(1) default NULL,  
`SubscriberCollectionMethod` varchar(30) default NULL,  
`SubscriberSSN` varchar(12) default NULL,  
`SubscriberAgencyClaimNo` varchar(50) default NULL,  
`SubscriberMemberID` varchar(50) default NULL,  
`SubscriberPersonalID` varchar(50) default NULL,  
`SubscriberContact` varchar(60) default NULL,  
`SubscriberTel` varchar(15) default NULL,  
`SubscriberTelExt` varchar(10) default NULL,  
  
/*Destination Payer*/  
`PayerName` varchar(60) default NULL,  
`PayerIDType` char(2) default NULL,  
`PayerID` varchar(30) default NULL,  
`PayerAddress` varchar(55) default NULL,  
`PayerAddress2` varchar(55) default NULL,  
`PayerCity` varchar(30) default NULL,  
`PayerState` char(2) default NULL,  
`PayerZip` varchar(15) default NULL,  
`PayerResponsibility` char(1) default NULL,  
`PayerOtherIDQual1` varchar(3) default NULL,  
`PayerOtherID1` varchar(50) default NULL,  
`PayerOtherIDQual2` varchar(3) default NULL,  
`PayerOtherID2` varchar(50) default NULL,
```

```
`PayerOtherIDQual3` varchar(3) default NULL,  
`PayerOtherID3` varchar(50) default NULL,  
`GroupNo` varchar(50) default NULL,  
`GroupName` varchar(60) default NULL,  
`InsuranceType` char(2) default NULL,  
`FilingIndicator` char(2) default NULL,  
`COBIndicator` char(1) default NULL,  
  
/*Responsible*/  
`ResponsibleLast` varchar(60) default NULL,  
`ResponsibleFirst` varchar(35) default NULL,  
`ResponsibleMiddle` varchar(25) default NULL,  
`ResponsibleSuffix` varchar(10) default NULL,  
`ResponsibleAddress` varchar(55) default NULL,  
`ResponsibleAddress2` varchar(55) default NULL,  
`ResponsibleCity` varchar(30) default NULL,  
`ResponsibleState` char(2) default NULL,  
`ResponsibleZip` varchar(15) default NULL,  
`ResponsibleIDType` char(2) default NULL,  
`ResponsibleID` varchar(80) default NULL,  
  
/*Patient*/  
`PatientLast` varchar(60) default NULL,  
`PatientFirst` varchar(35) default NULL,  
`PatientMiddle` varchar(25) default NULL,  
`PatientSuffix` varchar(10) default NULL,  
`PatientAddress` varchar(55) default NULL,  
`PatientAddress2` varchar(55) default NULL,
```

```
`PatientCity` varchar(30) default NULL,  
`PatientState` char(2) default NULL,  
`PatientZip` varchar(15) default NULL,  
`PatientCountry` varchar(3) default NULL,  
`PatientLocation` varchar(30) default NULL,  
`PatientSubdivision` varchar(3) default NULL,  
`PatientIDType` char(2) default NULL,  
`PatientID` varchar(80) default NULL,  
`PatientSSN` varchar(12) default NULL,  
`PatientMemberID` varchar(50) default NULL,  
`PatientPersonalID` varchar(50) default NULL,  
`PatientInsurancePolicy` varchar(50) default NULL,  
`PatientDOB` varchar(50) default NULL,  
`PatientSex` char(1) default NULL,  
`PatientEthnicity` varchar(75) default NULL,  
`PatientMaritalStatus` char(1) default NULL,  
`PatientCollectionMethod` varchar(30) default NULL,  
`PatientRelationship` char(2) default NULL,  
`PatientPaid` varchar(10) default NULL,  
`PregnancyIndicator` char(1) default NULL,  
  
/*Rendering Provider*/  
  
`RendProvIDType` char(2) default NULL,  
`RendProvID` varchar(80) default NULL,  
`RendProvNPI` varchar(12) default NULL,  
`RendProvTaxID` varchar(12) default NULL,  
`RendProvLast` varchar(60) default NULL,
```

```
`RendProvFirst` varchar(35) default NULL,  
`RendProvMiddle` varchar(25) default NULL,  
`RendProvSuffix` varchar(10) default NULL,  
`RendProvSpecialty` varchar(10) default NULL,  
`RendProvOtherIDQual1` varchar(3) default NULL,  
`RendProvOtherID1` varchar(50) default NULL,  
`RendProvOtherIDQual2` varchar(3) default NULL,  
`RendProvOtherID2` varchar(50) default NULL,  
`RendProvOtherIDQual3` varchar(3) default NULL,  
`RendProvOtherID3` varchar(50) default NULL,  
/*Attending Provider*/  
`AttendProvLast` varchar(60) default NULL,  
`AttendProvFirst` varchar(35) default NULL,  
`AttendProvMiddle` varchar(25) default NULL,  
`AttendProvSuffix` varchar(10) default NULL,  
`AttendProvIDType` char(2) default NULL,  
`AttendProvID` varchar(80) default NULL,  
`AttendProvTaxID` varchar(12) default NULL,  
`AttendProvNPI` varchar(12) default NULL,  
`AttendProvOtherIDQual1` varchar(100) default NULL,  
`AttendProvOtherID1` varchar(100) default NULL,  
`AttendProvSpecialty` varchar(10) default NULL,  
/*Operating Provider*/  
`OperatingProvLast` varchar(60) default NULL,  
`OperatingProvFirst` varchar(35) default NULL,  
`OperatingProvMiddle` varchar(25) default NULL,
```

```
`OperatingProvSuffix` varchar(10) default NULL,  
`OperatingProvIDType` char(2) default NULL,  
`OperatingProvID` varchar(80) default NULL,  
`OperatingProvNPI` varchar(12) default NULL,  
`OperatingProvTaxID` varchar(12) default NULL,  
`OperatingProvOtherIDQual1` varchar(100) default NULL,  
`OperatingProvOtherID1` varchar(100) default NULL,  
`OperatingProvSpecialty` varchar(10) default NULL,  
/*Other Provider*/  
`OtherProvLast` varchar(60) default NULL,  
`OtherProvFirst` varchar(35) default NULL,  
`OtherProvMiddle` varchar(25) default NULL,  
`OtherProvSuffix` varchar(10) default NULL,  
`OtherProvIDType` varchar(100) default NULL,  
`OtherProvID` varchar(100) default NULL,  
`OtherProvTaxID` varchar(12) default NULL,  
`OtherProvNPI` varchar(12) default NULL,  
`OtherProvOtherIDQual1` varchar(100) default NULL,  
`OtherProvOtherID1` varchar(100) default NULL,  
`OtherProvSpecialty` varchar(100) default NULL,  
/*Facility*/  
`FacilityType` char(2) default NULL,  
`FacilityIDType` char(2) default NULL,  
`FacilityID` varchar(80) default NULL,  
`FacilityNPI` varchar(12) default NULL,  
`FacilityTaxID` varchar(12) default NULL,
```

```
`FacilityOtherIDQual1` varchar(100) default NULL,  
`FacilityOtherID1` varchar(100) default NULL,  
`FacilityName` varchar(80) default NULL,  
`FacilityAddress` varchar(55) default NULL,  
`FacilityAddress2` varchar(55) default NULL,  
`FacilityCity` varchar(30) default NULL,  
`FacilitySpecialty` varchar(100) default NULL,  
`FacilityState` char(2) default NULL,  
`FacilityZip` varchar(15) default NULL,  
/*Referring Provider*/  
`RefProvLast` varchar(60) default NULL,  
`RefProvFirst` varchar(35) default NULL,  
`RefProvMiddle` varchar(25) default NULL,  
`RefProvSuffix` varchar(10) default NULL,  
`RefProvIDType` char(2) default NULL,  
`RefProvID` varchar(80) default NULL,  
`RefProvTaxID` varchar(12) default NULL,  
`RefProvNPI` varchar(12) default NULL,  
`RefProvOtherIDQual1` varchar(100) default NULL,  
`RefProvOtherID1` varchar(100) default NULL,  
`RefProvSpecialty` varchar(10) default NULL,  
/*Supervising Provider*/  
`SupervProvLast` varchar(60) default NULL,  
`SupervProvFirst` varchar(35) default NULL,  
`SupervProvMiddle` varchar(25) default NULL,  
`SupervProvSuffix` varchar(10) default NULL,
```

```
`SupervProvIDType` char(2) default NULL,  
`SupervProvID` varchar(80) default NULL,  
`SupervProvTaxID` varchar(12) default NULL,  
`SupervProvNPI` varchar(12) default NULL,  
`SupervProvOtherIDQual1` varchar(3) default NULL,  
`SupervProvOtherID1` varchar(50) default NULL,  
`SupervProvSpecialty` varchar(10) default NULL,  
/*Assistant Surgeon*/  
`AssistSurgLast` varchar(60) default NULL,  
`AssistSurgFirst` varchar(35) default NULL,  
`AssistSurgMiddle` varchar(25) default NULL,  
`AssistSurgSuffix` varchar(10) default NULL,  
`AssistSurgIDType` char(2) default NULL,  
`AssistSurgID` varchar(80) default NULL,  
`AssistSurgTaxID` varchar(12) default NULL,  
`AssistSurgNPI` varchar(12) default NULL,  
`AssistSurgOtherIDQual1` varchar(3) default NULL,  
`AssistSurgOtherID1` varchar(50) default NULL,  
`AssistSurgSpecialty` varchar(10) default NULL,  
/*Ambulance Pick-up Location*/  
`PickUpAddress` varchar(55) default NULL,  
`PickUpAddress2` varchar(55) default NULL,  
`PickUpCity` varchar(30) default NULL,  
`PickUpState` char(2) default NULL,  
`PickUpZip` varchar(15) default NULL,  
/*Ambulance Drop-off Location*/
```

```
`DropOffName` varchar(80) default NULL,  
`DropOffAddress` varchar(55) default NULL,  
`DropOffAddress2` varchar(55) default NULL,  
`DropOffCity` varchar(30) default NULL,  
`DropOffState` char(2) default NULL,  
`DropOffZip` varchar(15) default NULL,  
  
/*COB Payer 1*/  
  
`OtherInsuredLast` varchar(60) default NULL,  
`OtherInsuredFirst` varchar(35) default NULL,  
`OtherInsuredMiddle` varchar(25) default NULL,  
`OtherInsuredSuffix` varchar(10) default NULL,  
`OtherInsuredIDQual` char(2) default NULL,  
`OtherInsuredID` varchar(10) default NULL,  
`OtherInsuredAddress` varchar(55) default NULL,  
`OtherInsuredAddress2` varchar(55) default NULL,  
`OtherInsuredCity` varchar(30) default NULL,  
`OtherInsuredState` char(2) default NULL,  
`OtherInsuredZip` varchar(15) default NULL,  
`OtherInsuredCountry` varchar(3) default NULL,  
`OtherInsuredSubdivision` varchar(3) default NULL,  
`OtherInsuredPolicy` varchar(50) default NULL,  
`OtherInsuredDOB` varchar(10) default NULL,  
`OtherInsuredSex` char(1) default NULL,  
`OtherInsuredRelationship` char(2) default NULL,  
`OtherInsuredOtherIDQual1` char(3) default NULL,  
`OtherInsuredOtherID1` varchar(50) default NULL,
```

```
`OtherInsuredOtherIDQual2` char(3) default NULL,  
`OtherInsuredOtherID2` varchar(50) default NULL,  
`OtherInsuredOtherIDQual3` char(3) default NULL,  
`OtherInsuredOtherID3` varchar(50) default NULL,  
`OtherPayerResponsibility` varchar(100) default NULL,  
`OtherPayerName` varchar(35) default NULL,  
`OtherPayerIDQual` char(2) default NULL,  
`OtherPayerID` varchar(80) default NULL,  
`OtherPayerAddress1` varchar(55) default NULL,  
`OtherPayerAddress2` varchar(55) default NULL,  
`OtherPayerCity` varchar(30) default NULL,  
`OtherPayerState` char(2) default NULL,  
`OtherPayerZip` varchar(15) default NULL,  
`OtherPayerPaid` varchar(10) default NULL,  
`OtherPayerGroupName` varchar(100) default NULL,  
`OtherPayerGroupNo` varchar(100) default NULL,  
`OtherPayerInsuranceType` char(2) default NULL,  
`OtherPayerFilingIndicator` char(2) default NULL,  
`OtherPayerAdjustmentReasonGroup` varchar(100) default NULL,  
`OtherPayerAdjustmentReason` varchar(100) default NULL,  
`OtherPayerAdjustmentAmount` varchar(100) default NULL,  
`OtherPayerPaidDate` varchar(15) default NULL,  
`OtherPayerInfoRelease` varchar(100) default NULL,  
`OtherPayerBenefitAssign` varchar(100) default NULL,  
`OtherPayerDocControlNo` varchar(100) default NULL,  
`OtherPayerPriorAuth` varchar(100) default NULL,
```

```
`OtherPayerOtherIDQual1` char(3) default NULL,  
`OtherPayerOtherID1` varchar(50) default NULL,  
`OtherPayerOtherIDQual2` char(3) default NULL,  
`OtherPayerOtherID2` varchar(50) default NULL,  
/*COB Payer 2*/  
`OtherInsured2Last` varchar(60) default NULL,  
`OtherInsured2First` varchar(35) default NULL,  
`OtherInsured2Middle` varchar(25) default NULL,  
`OtherInsured2Suffix` varchar(10) default NULL,  
`OtherInsured2IDQual` char(2) default NULL,  
`OtherInsured2ID` varchar(10) default NULL,  
`OtherInsured2Address` varchar(55) default NULL,  
`OtherInsured2Address2` varchar(55) default NULL,  
`OtherInsured2City` varchar(30) default NULL,  
`OtherInsured2State` char(2) default NULL,  
`OtherInsured2Zip` varchar(15) default NULL,  
`OtherInsured2Country` varchar(3) default NULL,  
`OtherInsured2Subdivision` varchar(3) default NULL,  
`OtherInsured2Policy` varchar(50) default NULL,  
`OtherInsured2DOB` varchar(10) default NULL,  
`OtherInsured2Sex` char(1) default NULL,  
`OtherInsured2Relationship` char(2) default NULL,  
`OtherInsured2OtherIDQual1` char(3) default NULL,  
`OtherInsured2OtherID1` varchar(50) default NULL,  
`OtherInsured2OtherIDQual2` char(3) default NULL,  
`OtherInsured2OtherID2` varchar(50) default NULL,
```

```
`OtherInsured2OtherIDQual3` char(3) default NULL,  
`OtherInsured2OtherID3` varchar(50) default NULL,  
`OtherPayer2Responsibility` varchar(100) default NULL,  
`OtherPayer2Name` varchar(35) default NULL,  
`OtherPayer2IDQual` char(2) default NULL,  
`OtherPayer2ID` varchar(80) default NULL,  
`OtherPayer2Address1` varchar(55) default NULL,  
`OtherPayer2Address2` varchar(55) default NULL,  
`OtherPayer2City` varchar(30) default NULL,  
`OtherPayer2State` char(2) default NULL,  
`OtherPayer2Zip` varchar(15) default NULL,  
`OtherPayer2Paid` varchar(10) default NULL,  
`OtherPayer2GroupName` varchar(100) default NULL,  
`OtherPayer2GroupNo` varchar(100) default NULL,  
`OtherPayer2InsuranceType` char(2) default NULL,  
`OtherPayer2FilingIndicator` char(2) default NULL,  
`OtherPayer2AdjustmentReasonGroup` varchar(100) default NULL,  
`OtherPayer2AdjustmentReason` varchar(100) default NULL,  
`OtherPayer2AdjustmentAmount` varchar(100) default NULL,  
`OtherPayer2PaidDate` varchar(15) default NULL,  
`OtherPayer2InfoRelease` varchar(100) default NULL,  
`OtherPayer2BenefitAssign` varchar(100) default NULL,  
`OtherPayer2DocControlNo` varchar(100) default NULL,  
`OtherPayer2PriorAuth` varchar(100) default NULL,  
`OtherPayer2OtherIDQual1` char(3) default NULL,  
`OtherPayer2OtherID1` varchar(50) default NULL,
```

```
`OtherPayer2OtherIDQual2` char(3) default NULL,  
`OtherPayer2OtherID2` varchar(50) default NULL,  
  
/*Claim Data*/  
  
`ClaimNo` varchar(50) NOT NULL,  
  
`Amount` decimal(10,2) default NULL,  
  
`EstimatedAmountDue` decimal(10,2) default NULL,  
  
`PatientEstimatedAmountDue` decimal(10,2) default NULL,  
  
`PlaceOfService` char(2) default NULL,  
  
`ClaimFrequency` char(1) default NULL,  
  
`SubmitReason` char(1) default NULL,  
  
`ProviderSignature` char(1) default NULL,  
  
`ProviderAcceptsAssignment` char(1) default NULL,  
  
`BenefitAssignment` char(1) default NULL,  
  
`InfoReleaseCode` char(1) default NULL,  
  
`PatientSignatureCode` char(1) default NULL,  
  
`RelatedCauses` char(2) default NULL,  
  
`RelatedCauses2` char(2) default NULL,  
  
`RelatedCausesState` char(2) default NULL,  
  
`SpecialProgramCode` char(3) default NULL,  
  
`ProviderParticipation` char(1) default NULL,  
  
`EOBIndicator` char(1) default NULL,  
  
`DelayReasonCode` char(2) default NULL,  
  
`ServiceDateFrom` varchar(15) default NULL,  
  
`ServiceDateTo` varchar(15) default NULL,  
  
`OnsetDate` varchar(15) default NULL,  
  
`SimilarSymptomsDate` varchar(10) default NULL,
```

```
`DisabilityBegin` varchar(10) default NULL,  
`DisabilityEnd` varchar(10) default NULL,  
`HospitalizationBegin` varchar(20) default NULL,  
`HospitalizationEnd` varchar(10) default NULL,  
`AccidentDate` varchar(20) default NULL,  
`LastMenstrualPeriod` varchar(15) default NULL,  
`RepricerReceivedDate` varchar(15) default NULL,  
`AdmissionDate` varchar(15) default NULL,  
`AdmissionHour` varchar(10) default NULL,  
`AdmissionType` char(1) default NULL,  
`AdmissionSource` char(1) default NULL,  
`DischargeHour` varchar(10) default NULL,  
`PatientStatus` char(2) default NULL,  
`CoveredDays` varchar(4) default NULL,  
`NonCoveredDays` varchar(4) default NULL,  
`COBDays` varchar(4) default NULL,  
`LifeTimeReserveDays` varchar(4) default NULL,  
`PriorAuthorization` varchar(50) default NULL,  
`ClearingHouseID` varchar(100) default NULL,  
`MedicalRecordNumber` varchar(50) default NULL,  
`MothersMedicalRecordNumber` varchar(50) default NULL,  
`ServiceAuthorizationException` varchar(50) default NULL,  
`ReferralNumber` varchar(50) default NULL,  
`PayerClaimControlNumber` varchar(50) default NULL,  
`RepricedClaimNumber` varchar(50) default NULL,  
`AdjustedRepricedClaimNumber` varchar(50) default NULL,
```

```
`AutoAccidentState` varchar(50) default NULL,  
`OrigRefNo` varchar(50) default NULL,  
`ClaimType` char(1) default NULL,  
`TypeOfBill` varchar(20) default NULL,  
`Remark1` varchar(90) default NULL,  
`Remark2` varchar(90) default NULL,  
`Remark3` varchar(90) default NULL,  
`Remark4` varchar(90) default NULL,  
`K3_1` varchar(80) default NULL,  
`K3_2` varchar(80) default NULL,  
`K3_3` varchar(80) default NULL,  
`K3_4` varchar(80) default NULL,  
`K3_5` varchar(80) default NULL,  
`K3_6` varchar(80) default NULL,  
`K3_7` varchar(80) default NULL,  
`K3_8` varchar(80) default NULL,  
`K3_9` varchar(80) default NULL,  
`K3_10` varchar(80) default NULL,  
`OutsideLab` char(1) default NULL,  
`LabCharge` varchar(10) default NULL,  
`Test_Prod` char(1) default NULL,  
`ReportTypeCode1` varchar(100) default NULL,  
`ReportTransmissionCode1` varchar(100) default NULL,  
`AttachmentControlNumber1` varchar(100) default NULL,  
/*Dental Claims*/  
`Predetermination` varchar(100) default NULL,
```

```
`OrthodonticTotal` varchar(100) default NULL,  
`OrthodonticRemaining` varchar(100) default NULL,  
`OrthodonticYesNo` varchar(100) default NULL,  
`ToothStatus` varchar(100) default NULL,  
`AppliancePlacementDate` varchar(100) default NULL,  
  
/*Diagnosis Information*/  
`AdmitDiagnosis` varchar(10) default NULL,  
`ECode` varchar(100) default NULL,  
`ECode2` varchar(100) default NULL,  
`ECode3` varchar(100) default NULL,  
`ECode4` varchar(100) default NULL,  
`ECode5` varchar(100) default NULL,  
`ECode6` varchar(100) default NULL,  
`ECode7` varchar(100) default NULL,  
`ECode8` varchar(100) default NULL,  
#`ECode9` varchar(100) default NULL,  
#`ECode10` varchar(100) default NULL,  
#`ECode11` varchar(100) default NULL,  
#`ECode12` varchar(100) default NULL,  
  
`ReasonForVisit` varchar(100) default NULL,  
`ReasonForVisit2` varchar(100) default NULL,  
`ReasonForVisit3` varchar(100) default NULL,  
  
`PrincipalDiagnosis` varchar(10) default NULL,  
  
`Diag2` varchar(10) default NULL,  
`Diag3` varchar(10) default NULL,  
`Diag4` varchar(10) default NULL,
```

```
`Diag5` varchar(10) default NULL,  
`Diag6` varchar(10) default NULL,  
`Diag7` varchar(10) default NULL,  
`Diag8` varchar(10) default NULL,  
`Diag9` varchar(10) default NULL,  
`Diag10` varchar(10) default NULL,  
`Diag11` varchar(10) default NULL,  
`Diag12` varchar(10) default NULL,  
`Diag13` varchar(10) default NULL,  
`Diag14` varchar(10) default NULL,  
`Diag15` varchar(10) default NULL,  
`Diag16` varchar(10) default NULL,  
`Diag17` varchar(10) default NULL,  
`Diag18` varchar(10) default NULL,  
`Diag19` varchar(10) default NULL,  
`Diag20` varchar(10) default NULL,  
`Diag21` varchar(10) default NULL,  
`Diag22` varchar(10) default NULL,  
`Diag23` varchar(10) default NULL,  
`Diag24` varchar(10) default NULL,  
`Diag25` varchar(10) default NULL,  
`DRG` varchar(15) default NULL,  
  
/*Procedure Information*/  
  
`PrincipalProcedure` varchar(10) default NULL,  
`PrincipalProcedureDate` datetime default NULL,  
  
`Proc2` varchar(50) default NULL,
```

```
`Proc2Date` datetime default NULL,  
`Proc3` varchar(10) default NULL,  
`Proc3Date` datetime default NULL,  
`Proc4` varchar(10) default NULL,  
`Proc4Date` datetime default NULL,  
`Proc5` varchar(10) default NULL,  
`Proc5Date` datetime default NULL,  
`Proc6` varchar(10) default NULL,  
`Proc6Date` datetime default NULL,  
`Proc7` varchar(10) default NULL,  
`Proc7Date` datetime default NULL,  
`Proc8` varchar(10) default NULL,  
`Proc8Date` datetime default NULL,  
`Proc9` varchar(10) default NULL,  
`Proc9Date` datetime default NULL,  
`Proc10` varchar(10) default NULL,  
`Proc10Date` datetime default NULL,  
`Proc11` varchar(10) default NULL,  
`Proc11Date` datetime default NULL,  
`Proc12` varchar(10) default NULL,  
`Proc12Date` datetime default NULL,  
`Proc13` varchar(10) default NULL,  
`Proc13Date` datetime default NULL,  
`Proc14` varchar(10) default NULL,  
`Proc14Date` datetime default NULL,  
`Proc15` varchar(10) default NULL,
```

```
`Proc15Date` datetime default NULL,  
`Proc16` varchar(10) default NULL,  
`Proc16Date` datetime default NULL,  
`Proc17` varchar(10) default NULL,  
`Proc17Date` datetime default NULL,  
`Proc18` varchar(10) default NULL,  
`Proc18Date` datetime default NULL,  
`Proc19` varchar(10) default NULL,  
`Proc19Date` datetime default NULL,  
`Proc20` varchar(10) default NULL,  
`Proc20Date` datetime default NULL,  
`Proc21` varchar(10) default NULL,  
`Proc21Date` datetime default NULL,  
`Proc22` varchar(10) default NULL,  
`Proc22Date` datetime default NULL,  
`Proc23` varchar(10) default NULL,  
`Proc23Date` datetime default NULL,  
`Proc24` varchar(10) default NULL,  
`Proc24Date` datetime default NULL,  
`Proc25` varchar(10) default NULL,  
`Proc25Date` datetime default NULL,  
  
/*Value Codes*/  
`ValueCode1` char(3) default NULL,  
`ValueAmount1` varchar(10) default NULL,  
`ValueCode2` char(3) default NULL,  
`ValueAmount2` varchar(10) default NULL,
```

```
`ValueCode3` char(3) default NULL,  
`ValueAmount3` varchar(10) default NULL,  
`ValueCode4` char(3) default NULL,  
`ValueAmount4` varchar(10) default NULL,  
`ValueCode5` char(3) default NULL,  
`ValueAmount5` varchar(10) default NULL,  
`ValueCode6` char(3) default NULL,  
`ValueAmount6` varchar(10) default NULL,  
`ValueCode7` varchar(10) default NULL,  
`ValueAmount7` varchar(10) default NULL,  
`ValueCode8` char(3) default NULL,  
`ValueAmount8` varchar(10) default NULL,  
`ValueCode9` char(3) default NULL,  
`ValueAmount9` varchar(10) default NULL,  
`ValueCode10` char(3) default NULL,  
`ValueAmount10` varchar(10) default NULL,  
`ValueCode11` char(3) default NULL,  
`ValueAmount11` varchar(10) default NULL,  
`ValueCode12` char(3) default NULL,  
`ValueAmount12` varchar(10) default NULL,  
#`ValueCode13` char(3) default NULL,  
#`ValueAmount13` varchar(10) default NULL,  
#`ValueCode14` char(3) default NULL,  
#`ValueAmount14` varchar(10) default NULL,  
#`ValueCode15` char(3) default NULL,  
#`ValueAmount15` varchar(10) default NULL,
```

```
#`ValueCode16` char(3) default NULL,  
#`ValueAmount16` varchar(10) default NULL,  
#`ValueCode17` varchar(10) default NULL,  
#`ValueAmount17` varchar(10) default NULL,  
#`ValueCode18` char(3) default NULL,  
#`ValueAmount18` varchar(10) default NULL,  
#`ValueCode19` char(3) default NULL,  
#`ValueAmount19` varchar(10) default NULL,  
#`ValueCode20` char(3) default NULL,  
#`ValueAmount20` varchar(10) default NULL,  
#`ValueCode21` char(3) default NULL,  
#`ValueAmount21` varchar(10) default NULL,  
#`ValueCode22` char(3) default NULL,  
#`ValueAmount22` varchar(10) default NULL,  
#`ValueCode23` char(3) default NULL,  
#`ValueAmount23` varchar(10) default NULL,  
#`ValueCode24` char(3) default NULL,  
#`ValueAmount24` varchar(10) default NULL,  
  
/*Condition Codes*/  
`ConditionCode1` varchar(30) default NULL,  
`ConditionCode2` varchar(30) default NULL,  
`ConditionCode3` varchar(30) default NULL,  
`ConditionCode4` varchar(30) default NULL,  
`ConditionCode5` varchar(30) default NULL,  
`ConditionCode6` varchar(30) default NULL,  
`ConditionCode7` varchar(30) default NULL,
```

```
`ConditionCode8` varchar(30) default NULL,  
`ConditionCode9` varchar(30) default NULL,  
`ConditionCode10` varchar(30) default NULL,  
#`ConditionCode11` varchar(30) default NULL,  
#`ConditionCode12` varchar(30) default NULL,  
#`ConditionCode13` varchar(30) default NULL,  
#`ConditionCode14` varchar(30) default NULL,  
#`ConditionCode15` varchar(30) default NULL,  
#`ConditionCode16` varchar(30) default NULL,  
#`ConditionCode17` varchar(30) default NULL,  
#`ConditionCode18` varchar(30) default NULL,  
#`ConditionCode19` varchar(30) default NULL,  
#`ConditionCode20` varchar(30) default NULL,  
#`ConditionCode21` varchar(30) default NULL,  
#`ConditionCode22` varchar(30) default NULL,  
#`ConditionCode23` varchar(30) default NULL,  
#`ConditionCode24` varchar(30) default NULL,  
  
/*Occurrence Codes*/  
  
`OccurrenceCode1` char(3) default NULL,  
`OccurrenceDate1` varchar(10) default NULL,  
`OccurrenceCode2` char(3) default NULL,  
`OccurrenceDate2` varchar(10) default NULL,  
`OccurrenceCode3` char(3) default NULL,  
`OccurrenceDate3` varchar(10) default NULL,  
`OccurrenceCode4` char(3) default NULL,  
`OccurrenceDate4` varchar(10) default NULL,
```

```
`OccurrenceCode5` char(3) default NULL,  
`OccurrenceDate5` varchar(10) default NULL,  
`OccurrenceCode6` char(3) default NULL,  
`OccurrenceDate6` varchar(10) default NULL,  
`OccurrenceCode7` char(3) default NULL,  
`OccurrenceDate7` varchar(10) default NULL,  
`OccurrenceCode8` char(3) default NULL,  
`OccurrenceDate8` varchar(10) default NULL,  
#`OccurrenceCode9` char(3) default NULL,  
#`OccurrenceDate9` varchar(10) default NULL,  
#`OccurrenceCode10` char(3) default NULL,  
#`OccurrenceDate10` varchar(10) default NULL,  
#`OccurrenceCode11` char(3) default NULL,  
#`OccurrenceDate11` varchar(10) default NULL,  
#`OccurrenceCode12` char(3) default NULL,  
#`OccurrenceDate12` varchar(10) default NULL,  
#`OccurrenceCode13` char(3) default NULL,  
#`OccurrenceDate13` varchar(10) default NULL,  
#`OccurrenceCode14` char(3) default NULL,  
#`OccurrenceDate14` varchar(10) default NULL,  
#`OccurrenceCode15` char(3) default NULL,  
#`OccurrenceDate15` varchar(10) default NULL,  
#`OccurrenceCode16` char(3) default NULL,  
#`OccurrenceDate16` varchar(10) default NULL,  
#`OccurrenceCode17` char(3) default NULL,  
#`OccurrenceDate17` varchar(10) default NULL,
```

```
#`OccurrenceCode18` char(3) default NULL,  
#`OccurrenceDate18` varchar(10) default NULL,  
#`OccurrenceCode19` char(3) default NULL,  
#`OccurrenceDate19` varchar(10) default NULL,  
#`OccurrenceCode20` char(3) default NULL,  
#`OccurrenceDate20` varchar(10) default NULL,  
#`OccurrenceCode21` char(3) default NULL,  
#`OccurrenceDate21` varchar(10) default NULL,  
#`OccurrenceCode22` char(3) default NULL,  
#`OccurrenceDate22` varchar(10) default NULL,  
#`OccurrenceCode23` char(3) default NULL,  
#`OccurrenceDate23` varchar(10) default NULL,  
#`OccurrenceCode24` char(3) default NULL,  
#`OccurrenceDate24` varchar(10) default NULL,  
`OccurrenceSpanCode1` char(3) default NULL,  
`OccurrenceSpanFrom1` varchar(10) default NULL,  
`OccurrenceSpanTo1` varchar(10) default NULL,  
`OccurrenceSpanCode2` char(3) default NULL,  
`OccurrenceSpanFrom2` varchar(10) default NULL,  
`OccurrenceSpanTo2` varchar(10) default NULL,  
`OccurrenceSpanCode3` char(3) default NULL,  
`OccurrenceSpanFrom3` varchar(10) default NULL,  
`OccurrenceSpanTo3` varchar(10) default NULL,  
`OccurrenceSpanCode4` char(3) default NULL,  
`OccurrenceSpanFrom4` varchar(10) default NULL,  
`OccurrenceSpanTo4` varchar(10) default NULL,
```

```
/* CR1 */

`PatientWeight` varchar(10) NULL,

`AmbulanceTransportCode` char(1) NULL,

`AmbulanceTransportReasonCode` char(1) NULL,

`TransportDistance` varchar(15) NULL,

`RoundTripPurposeDescription` varchar(80) NULL,

`StretcherPurposeDescription` varchar(80) NULL,

/* CRC - Ambulance Certification */

`AmbulanceConditionIndicator` char(1) NULL,

`AmbulanceConditionCode1` char(3) NULL,

`AmbulanceConditionCode2` char(3) NULL,

`AmbulanceConditionCode3` char(3) NULL,

`AmbulanceConditionCode4` char(3) NULL,

`AmbulanceConditionCode5` char(3) NULL,

/*Repricing*/

`RepricingMethodology` varchar(3) NULL,

`RepricedAmount` decimal(10,2) NULL,

`SavingsAmount` decimal(10,2) NULL,

`RepricerID` varchar(30) NULL,

`RepricingRate` varchar(9) NULL,

`APG_Code` varchar(30) NULL,

`APG_Amount` decimal(10,2) NULL,

`ApprovedRevenueCode` varchar (48) NULL,

`ApprovedProcedureCode` varchar (48) NULL,

`ApprovedUnitCode` char(2) NULL,

`ApprovedUnits` bigint NULL,
```

```
    `RejectReason` varchar(3) NULL,  
    `ComplianceCode` varchar(2) NULL,  
    `ExceptionCode` varchar(2) NULL,  
    /*SQL To EDI*/  
    #`ClaimAction` char(1) NULL,  
    #`ErrorMessage` varchar(1000) NULL,  
    PRIMARY KEY (`ID`)  
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Detail

```
CREATE TABLE `edi_ClaimDetail`(  
    `ID` int(11) NOT NULL auto_increment,  
    `ClaimID` int(11) NOT NULL,  
    `LineNumber` decimal(18, 0) NULL,  
    `LineID` varchar(50) NULL,  
    `ServiceDateFrom` datetime NULL,  
    `ServiceDateTo` date NULL,  
    `AssessmentDate` date NULL,  
    `FacilityCode` char(2) NULL,  
    `RevenueCode` varchar(48) NULL,  
    `ProcedureCode` varchar(48) NULL,  
    `Amount` decimal(10,2) NULL,  
    `Unit` varchar(50) NULL,  
    `Quantity` float NULL,  
    `UnitRate` decimal(10,2) NULL,  
    `NonCovered` decimal(10,2) NULL,
```

```
`Paid` decimal(10,2) NULL,  
`PlaceOfService` char(3) NULL,  
`Modifier1` char(3) NULL,  
`Modifier2` char(3) NULL,  
`Modifier3` char(3) NULL,  
`Modifier4` char(3) NULL,  
`ProcedureDescription` varchar(80) NULL,  
`OralCavityDesignation1` char(3) NULL,  
`OralCavityDesignation2` char(3) NULL,  
`OralCavityDesignation3` char(3) NULL,  
`OralCavityDesignation4` char(3) NULL,  
`OralCavityDesignation5` char(3) NULL,  
`DiagPointer1` char(1) NULL,  
`DiagPointer2` char(1) NULL,  
`DiagPointer3` char(1) NULL,  
`DiagPointer4` char(1) NULL,  
`ToothNumber` char(2) NULL,  
`Surface` varchar(10) NULL,  
`EmergencyIndicator` char(1) NULL,  
`ServiceTax` decimal(10,2) NULL,  
`FacilityTax` decimal(10,2) NULL,  
`SalesTax` decimal(10,2) NULL,  
`ApprovedAmount` decimal(10,2) NULL,  
`LineK3_01` varchar(80) NULL,  
`LineK3_02` varchar(80) NULL,  
`LineK3_03` varchar(80) NULL,
```

```
`LineK3_04` varchar(80) NULL,  
`LineK3_05` varchar(80) NULL,  
`LineK3_06` varchar(80) NULL,  
`LineK3_07` varchar(80) NULL,  
`LineK3_08` varchar(80) NULL,  
`LineK3_09` varchar(80) NULL,  
`LineK3_10` varchar(80) NULL,  
`Remark` varchar(100) NULL,  
`AmbulancePatientCount` bigint NULL,  
  
/****PWK****/  
  
`ReportType` char(2) NULL,  
`ReportTransmission` char(2) NULL,  
`AttachmentControlNumber` varchar(80) NULL,  
  
/*HCP*/  
  
`RepricingMethodology` varchar(3) NULL,  
`RepricedAmount` decimal(10,2) NULL,  
`SavingsAmount` decimal(10,2) NULL,  
`RepricerID` varchar(30) NULL,  
`RepricingRate` varchar(9) NULL,  
`APG_Code` varchar(30) NULL,  
`APG_Amount` decimal(10,2) NULL,  
`ApprovedRevenueCode` varchar (48) NULL,  
`ApprovedProcedureCode` varchar (48) NULL,  
`ApprovedUnitCode` char(2) NULL,  
`ApprovedUnits` bigint NULL,  
`RejectReason` varchar(3) NULL,
```

```
`ComplianceCode` varchar(2) NULL,  
`ExceptionCode` varchar(2) NULL,  
  
/* Drugs */  
`DrugCode` varchar(48) NULL,  
`DrugUnitPrice` decimal(10,2) NULL,  
`DrugUnitCode` char(2) NULL,  
`DrugUnits` double NULL,  
`PrescriptionNumber` varchar(30) NULL,  
  
/* CR1 */  
`PatientWeight` varchar(10) NULL,  
`AmbulanceTransportCode` char(1) NULL,  
`AmbulanceTransportReasonCode` char(1) NULL,  
`TransportDistance` varchar(15) NULL,  
`RoundTripPurposeDescription` varchar(80) NULL,  
`StretcherPurposeDescription` varchar(80) NULL,  
  
/* CRC - Ambulance Certification */  
`AmbulanceConditionIndicator` char(1) NULL,  
`AmbulanceConditionCode1` char(3) NULL,  
`AmbulanceConditionCode2` char(3) NULL,  
`AmbulanceConditionCode3` char(3) NULL,  
`AmbulanceConditionCode4` char(3) NULL,  
`AmbulanceConditionCode5` char(3) NULL,  
  
/* Attending Provider */  
`AttendingProviderLast` varchar(35) NULL,  
`AttendingProviderFirst` varchar(25) NULL,  
`AttendingProviderMiddle` varchar(25) NULL,
```

```
`AttendingProviderSuffix` varchar(10) NULL,  
`AttendingProviderIDQual` char(2) NULL,  
`AttendingProviderID` varchar(35) NULL,  
`AttendingProviderOtherIDQual` char(2) NULL,  
`AttendingProviderOtherID` varchar(30) NULL,  
  
/* Operating Provider */  
  
`OperatingProviderLast` varchar(35) NULL,  
`OperatingProviderFirst` varchar(25) NULL,  
`OperatingProviderMiddle` varchar(25) NULL,  
`OperatingProviderSuffix` varchar(10) NULL,  
`OperatingProviderIDQual` char(2) NULL,  
`OperatingProviderID` varchar(35) NULL,  
`OperatingProviderOtherIDQual` char(2) NULL,  
`OperatingProviderOtherID` varchar(30) NULL,  
  
/* Other Provider */  
  
`OtherProviderLast` varchar(35) NULL,  
`OtherProviderFirst` varchar(25) NULL,  
`OtherProviderMiddle` varchar(25) NULL,  
`OtherProviderSuffix` varchar(10) NULL,  
`OtherProviderIDQual` char(2) NULL,  
`OtherProviderID` varchar(35) NULL,  
`OtherProviderOtherIDQual` char(2) NULL,  
`OtherProviderOtherID` varchar(30) NULL,  
  
/* Rendering Provider */  
  
`RenderingProviderLast` varchar(35) NULL,  
`RenderingProviderFirst` varchar(25) NULL,
```

```
`RenderingProviderMiddle` varchar(25) NULL,  
`RenderingProviderSuffix` varchar(10) NULL,  
`RenderingProviderIDQual` char(2) NULL,  
`RenderingProviderID` varchar(35) NULL,  
`RenderingProviderOtherIDQual` char(2) NULL,  
`RenderingProviderOtherID` varchar(30) NULL,  
  
/* PurchasedService Provider */  
`PurchasedServiceProviderLast` varchar(35) NULL,  
`PurchasedServiceProviderFirst` varchar(25) NULL,  
`PurchasedServiceProviderMiddle` varchar(25) NULL,  
`PurchasedServiceProviderSuffix` varchar(10) NULL,  
`PurchasedServiceProviderIDQual` char(2) NULL,  
`PurchasedServiceProviderID` varchar(35) NULL,  
`PurchasedServiceProviderOtherIDQual` char(2) NULL,  
`PurchasedServiceProviderOtherID` varchar(30) NULL,  
  
/* Facility Provider */  
`FacilityName` varchar(35) NULL,  
`FacilityIDQual` char(2) NULL,  
`FacilityID` varchar(35) NULL,  
`FacilityAddress1` varchar(25) NULL,  
`FacilityAddress2` varchar(25) NULL,  
`FacilityCity` varchar(30) NULL,  
`FacilityState` char(2) NULL,  
`FacilityZip` varchar(15) NULL,  
`FacilityOtherIDQual` char(2) NULL,  
`FacilityOtherID` varchar(30) NULL,
```

```
/* Supervising Provider */
`SupervisingProviderLast` varchar(35) NULL,
`SupervisingProviderFirst` varchar(25) NULL,
`SupervisingProviderMiddle` varchar(25) NULL,
`SupervisingProviderSuffix` varchar(10) NULL,
`SupervisingProviderIDQual` char(2) NULL,
`SupervisingProviderID` varchar(35) NULL,
`SupervisingProviderOtherIDQual` char(2) NULL,
`SupervisingProviderOtherID` varchar(30) NULL,
/* Ordering Provider */
`OrderingProviderLast` varchar(35) NULL,
`OrderingProviderFirst` varchar(25) NULL,
`OrderingProviderMiddle` varchar(25) NULL,
`OrderingProviderSuffix` varchar(10) NULL,
`OrderingProviderIDQual` char(2) NULL,
`OrderingProviderID` varchar(35) NULL,
`OrderingProviderOtherIDQual` char(2) NULL,
`OrderingProviderOtherID` varchar(30) NULL,
/* Referring Provider */
`ReferringProviderLast` varchar(35) NULL,
`ReferringProviderFirst` varchar(25) NULL,
`ReferringProviderMiddle` varchar(25) NULL,
`ReferringProviderSuffix` varchar(10) NULL,
`ReferringProviderIDQual` char(2) NULL,
`ReferringProviderID` varchar(35) NULL,
`ReferringProviderOtherIDQual` char(2) NULL,
```

```
`ReferringProviderOtherID` varchar(30) NULL,  
  
/*COB 1 */  
  
`OtherPayer1ID` varchar(30) NULL,  
  
`OtherPayer1Paid` decimal(10,2) NULL,  
  
`OtherPayer1PaidProcedure` varchar(30) NULL,  
  
`OtherPayer1PaidRevenueCode` varchar(30) NULL,  
  
`OtherPayer1PaidQuantity` bigint NULL,  
  
`OtherPayer1BundledLine` int NULL,  
  
`OtherPayer1AdjustmentReasonGroup1` char(2) NULL,  
  
`OtherPayer1AdjustmentReason1` varchar(5) NULL,  
  
`OtherPayer1AdjustmentAmount1` decimal(10,2) NULL,  
  
`OtherPayer1AdjustmentQuantity1` bigint NULL,  
  
`OtherPayer1AdjustmentReasonGroup2` char(2) NULL,  
  
`OtherPayer1AdjustmentReason2` varchar(5) NULL,  
  
`OtherPayer1AdjustmentAmount2` decimal(10,2) NULL,  
  
`OtherPayer1AdjustmentQuantity2` bigint NULL,  
  
`OtherPayer1AdjustmentReasonGroup3` char(2) NULL,  
  
`OtherPayer1AdjustmentReason3` varchar(5) NULL,  
  
`OtherPayer1AdjustmentAmount3` decimal(10,2) NULL,  
  
`OtherPayer1AdjustmentQuantity3` bigint NULL,  
  
`OtherPayer1AdjustmentReasonGroup4` char(2) NULL,  
  
`OtherPayer1AdjustmentReason4` varchar(5) NULL,  
  
`OtherPayer1AdjustmentAmount4` decimal(10,2) NULL,  
  
`OtherPayer1AdjustmentQuantity4` bigint NULL,  
  
#`OtherPayer1AdjustmentReasonGroup5Plus` varchar(77) NULL,  
  
#`OtherPayer1AdjustmentReason5Plus` varchar(155) NULL,
```

```
#`OtherPayer1AdjustmentAmount5Plus` varchar(493) NULL,  
#`OtherPayer1AdjustmentQuantity5Plus` varchar(415) NULL,  
`OtherPayer1PaidDate` varchar(15) default NULL,  
/*COB 2 */  
`OtherPayer2ID` varchar(30) NULL,  
`OtherPayer2Paid` decimal(10,2) NULL,  
`OtherPayer2PaidProcedure` varchar(30) NULL,  
`OtherPayer2PaidRevenueCode` varchar(30) NULL,  
`OtherPayer2PaidQuantity` bigint NULL,  
`OtherPayer2BundledLine` int NULL,  
`OtherPayer2AdjustmentReasonGroup1` char(2) NULL,  
`OtherPayer2AdjustmentReason1` varchar(5) NULL,  
`OtherPayer2AdjustmentAmount1` decimal(10,2) NULL,  
`OtherPayer2AdjustmentQuantity1` bigint NULL,  
`OtherPayer2AdjustmentReasonGroup2` char(2) NULL,  
`OtherPayer2AdjustmentReason2` varchar(5) NULL,  
`OtherPayer2AdjustmentAmount2` decimal(10,2) NULL,  
`OtherPayer2AdjustmentQuantity2` bigint NULL,  
`OtherPayer2AdjustmentReasonGroup3` char(2) NULL,  
`OtherPayer2AdjustmentReason3` varchar(5) NULL,  
`OtherPayer2AdjustmentAmount3` decimal(10,2) NULL,  
`OtherPayer2AdjustmentQuantity3` bigint NULL,  
`OtherPayer2AdjustmentReasonGroup4` char(2) NULL,  
`OtherPayer2AdjustmentReason4` varchar(5) NULL,  
`OtherPayer2AdjustmentAmount4` decimal(10,2) NULL,  
`OtherPayer2AdjustmentQuantity4` bigint NULL,
```

```
#`OtherPayer2AdjustmentReasonGroup5Plus` varchar(77) NULL,  
#`OtherPayer2AdjustmentReason5Plus` varchar(155) NULL,  
#`OtherPayer2AdjustmentAmount5Plus` varchar(493) NULL,  
#`OtherPayer2AdjustmentQuantity5Plus` varchar(415) NULL,  
`OtherPayer2PaidDate` date NULL,  
  
PRIMARY KEY (`ID`)  
  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

5.3.3 Oracle Script to Create Tables

Header

```
CREATE TABLE EDI_Claims(  
  
    ID number(19) NOT NULL,  
  
    /*File Level Data*/  
  
    Filename varchar2(100) NULL,  
  
    Version varchar2(12) NULL,  
  
    ImageFilePath varchar2(300) NULL,  
  
    ImageFilename varchar2(100) NULL,  
  
    TradingPartnerIDType char(2) NULL,  
  
    TradingPartnerID varchar2(15) NOT NULL,  
  
    TransactionDate date NULL,  
  
    ReceiveDate date NULL,  
  
    SubmitterName varchar2(60) NULL,  
  
    SubmitterID varchar2(30) NULL,  
  
    SubmitterContact varchar2(60) NULL,  
  
    SubmitterTel varchar2(20) NULL,  
  
    SubmitterTelExt varchar2(10) NULL,
```

```
SubmitterFax varchar2(20) NULL,  
SubmitterEmail varchar2(80) NULL,  
ReceiverName varchar2(60) NULL,  
ReceiverID varchar2(30) NULL,  
TransactionType char(2) NULL,  
/*Billing Provider*/  
FedTaxID varchar2(15) NULL,  
BillProvIDType char(2) NULL,  
BillProvID varchar2(80) NULL,  
BillProvNPI varchar2(12) NULL,  
BillProvLast varchar2(60) NULL,  
BillProvFirst varchar2(35) NULL,  
BillProvMiddle varchar2(25) NULL,  
BillProvSuffix varchar2(10) NULL,  
BillProvSpecialty varchar2(10) NULL,  
BillProvAddress varchar2(55) NULL,  
BillProvAddress2 varchar2(55) NULL,  
BillProvCity varchar2(30) NULL,  
BillProvState char(2) NULL,  
BillProvZip varchar2(15) NULL,  
BillProvCountry varchar2(3) NULL,  
BillProvSubdivision varchar2(3) NULL,  
BillProvContact varchar2(60) NULL,  
BillProvTel varchar2(80) NULL,  
BillProvTelExt varchar2(80) NULL,  
BillProvFax varchar2(80) NULL,
```

```
BillProvEmail varchar2(80) NULL,  
BillProvOtherIDQual1 char(2) NULL,  
BillProvOtherID1 varchar2(30) NULL,  
BillProvOtherIDQual2 char(2) NULL,  
BillProvOtherID2 varchar2(30) NULL,  
BillProvOtherIDQual3 char(2) NULL,  
BillProvOtherID3 varchar2(30) NULL,  
BillProvOtherIDQual4 char(2) NULL,  
BillProvOtherID4 varchar2(30) NULL,  
BillProvOtherIDQual5 char(2) NULL,  
BillProvOtherID5 varchar2(30) NULL,  
  
/*Pay-to Provider*/  
  
PayToProvIDType char(2) NULL,  
PaytoProvID varchar2(30) NULL,  
PaytoProvTaxID varchar2(12) NULL,  
PaytoProvNPI varchar2(12) NULL,  
PaytoProvLast varchar2(60) NULL,  
PaytoProvFirst varchar2(35) NULL,  
PayToProvMiddle varchar2(25) NULL,  
PayToProvSuffix varchar2(10) NULL,  
PayToProvSpecialty varchar2(10) NULL,  
PaytoProvAddress varchar2(55) NULL,  
PayToProvAddress2 varchar2(55) NULL,  
PaytoProvCity varchar2(30) NULL,  
PaytoProvState char(2) NULL,  
PaytoProvZip varchar2(15) NULL,
```

```
/*Pay-to Payer*/  
  
PayToPayerIDType char(2) NULL,  
  
PaytoPayerID varchar2(80) NULL,  
  
PaytoPayerTaxID varchar2(12) NULL,  
  
PaytoPayerNPI varchar2(12) NULL,  
  
PaytoPayerLast varchar2(60) NULL,  
  
PaytoPayerFirst varchar2(35) NULL,  
  
PayToPayerMiddle varchar2(25) NULL,  
  
PayToPayerSuffix varchar2(10) NULL,  
  
PaytoPayerAddress varchar2(55) NULL,  
  
PayToPayerAddress2 varchar2(55) NULL,  
  
PaytoPayerCity varchar2(30) NULL,  
  
PaytoPayerState char(2) NULL,  
  
PaytoPayerZip varchar2(15) NULL,  
  
/*Rendering Provider*/  
  
RendProvIDType char(2) NULL,  
  
RendProvID varchar2(30) NULL,  
  
RendProvNPI varchar2(12) NULL,  
  
RendProvTaxID varchar2(12) NULL,  
  
RendProvLast varchar2(60) NULL,  
  
RendProvFirst varchar2(35) NULL,  
  
RendProvMiddle varchar2(25) NULL,  
  
RendProvSuffix varchar2(10) NULL,  
  
RendProvSpecialty varchar2(10) NULL,  
  
RendProvOtherIDQual1 char(2) NULL,  
  
RendProvOtherID1 varchar2(30) NULL,
```

```
RendProvOtherIDQual2 char(2) NULL,  
RendProvOtherID2 varchar2(30) NULL,  
RendProvOtherIDQual3 char(2) NULL,  
RendProvOtherID3 varchar2(30) NULL,  
  
/*Attending Provider*/  
  
AttendProvLast varchar2(60) NULL,  
AttendProvFirst varchar2(35) NULL,  
AttendProvMiddle varchar2(25) NULL,  
AttendProvSuffix varchar2(10) NULL,  
AttendProvIDType char(2) NULL,  
AttendProvID varchar2(30) NULL,  
AttendProvTaxID varchar2(12) NULL,  
AttendProvNPI varchar2(12) NULL,  
AttendProvOtherIDQual1 char(2) NULL,  
AttendProvOtherID1 varchar2(30) NULL,  
AttendProvSpecialty varchar2(10) NULL,  
  
/*Operating Provider*/  
  
OperatingProvLast varchar2(60) NULL,  
OperatingProvFirst varchar2(35) NULL,  
OperatingProvMiddle varchar2(25) NULL,  
OperatingProvSuffix varchar2(10) NULL,  
OperatingProvIDType char(2) NULL,  
OperatingProvID varchar2(30) NULL,  
OperatingProvNPI varchar2(12) NULL,  
OperatingProvTaxID varchar2(12) NULL,  
OperatingProvOtherIDQual1 char(2) NULL,
```

```
OperatingProvOtherID1 varchar2(30) NULL,  
OperatingProvSpecialty varchar2(10) NULL,  
  
/*Other Provider*/  
  
OtherProvLast varchar2(60) NULL,  
OtherProvFirst varchar2(35) NULL,  
OtherProvMiddle varchar2(25) NULL,  
OtherProvSuffix varchar2(10) NULL,  
OtherProvIDType char(2) NULL,  
OtherProvID varchar2(30) NULL,  
OtherProvTaxID varchar2(12) NULL,  
OtherProvNPI varchar2(12) NULL,  
OtherProvOtherIDQual1 char(2) NULL,  
OtherProvOtherID1 varchar2(30) NULL,  
OtherProvSpecialty varchar2(50) NULL,  
  
/*Facility*/  
  
FacilityType char(2) NULL,  
FacilityIDType char(2) NULL,  
FacilityID varchar2(80) NULL,  
FacilityNPI varchar2(12) NULL,  
FacilityTaxID varchar2(12) NULL,  
FacilityOtherIDQual1 char(2) NULL,  
FacilityOtherID1 varchar2(30) NULL,  
FacilityName varchar2(80) NULL,  
FacilityAddress varchar2(55) NULL,  
FacilityAddress2 varchar2(55) NULL,  
FacilityCity varchar2(30) NULL,
```

```
FacilitySpecialty varchar2(50) NULL,  
FacilityState char(2) NULL,  
FacilityZip varchar2(15) NULL,  
/*Referring Provider*/  
RefProvLast varchar2(60) NULL,  
RefProvFirst varchar2(35) NULL,  
RefProvMiddle varchar2(25) NULL,  
RefProvSuffix varchar2(10) NULL,  
RefProvIDType char(2) NULL,  
RefProvID varchar2(80) NULL,  
RefProvTaxID varchar2(12) NULL,  
RefProvNPI varchar2(12) NULL,  
RefProvOtherIDQual1 char(2) NULL,  
RefProvOtherID1 varchar2(300) NULL,  
RefProvSpecialty varchar2(20) NULL,  
/*Supervising Provider*/  
SupervProvLast varchar2(60) NULL,  
SupervProvFirst varchar2(35) NULL,  
SupervProvMiddle varchar2(25) NULL,  
SupervProvSuffix varchar2(10) NULL,  
SupervProvIDType char(2) NULL,  
SupervProvID varchar2(30) NULL,  
SupervProvTaxID varchar2(12) NULL,  
SupervProvNPI varchar2(12) NULL,  
SupervProvOtherIDQual1 char(2) NULL,  
SupervProvOtherID1 varchar2(50) NULL,
```

```
SupervProvSpecialty varchar2(20) NULL,  
  
/*Assistant Surgeon*/  
  
AssistSurgLast varchar2(60) NULL,  
  
AssistSurgFirst varchar2(35) NULL,  
  
AssistSurgMiddle varchar2(25) NULL,  
  
AssistSurgSuffix varchar2(10) NULL,  
  
AssistSurgIDType char(2) NULL,  
  
AssistSurgID varchar2(30) NULL,  
  
AssistSurgTaxID varchar2(12) NULL,  
  
AssistSurgNPI varchar2(12) NULL,  
  
AssistSurgOtherIDQual1 varchar2(3) NULL,  
  
AssistSurgOtherID1 varchar2(50) NULL,  
  
AssistSurgSpecialty varchar2(10) NULL,  
  
/*Ambulance Pick-up Location*/  
  
PickUpAddress varchar2(55) NULL,  
  
PickUpAddress2 varchar2(55) NULL,  
  
PickUpCity varchar2(30) NULL,  
  
PickUpState char(2) NULL,  
  
PickUpZip varchar2(15) NULL,  
  
/*Ambulance Drop-off Location*/  
  
DropOffName varchar2(80) NULL,  
  
DropOffAddress varchar2(55) NULL,  
  
DropOffAddress2 varchar2(55) NULL,  
  
DropOffCity varchar2(30) NULL,  
  
DropOffState char(2) NULL,  
  
DropOffZip varchar2(15) NULL,
```

```
/*Subscriber*/
SubscriberLast varchar2(60) NULL,
SubscriberFirst varchar2(35) NULL,
SubscriberMiddle varchar2(25) NULL,
SubscriberSuffix varchar2(10) NULL,
SubscriberAddress varchar2(55) NULL,
SubscriberAddress2 varchar2(55) NULL,
SubscriberCity varchar2(30) NULL,
SubscriberState char(2) NULL,
SubscriberZip varchar2(15) NULL,
SubscriberCountry varchar2(3) NULL,
SubscriberSubdivision varchar2(3) NULL,
SubscriberIDType char(2) NULL,
SubscriberID varchar2(30) NULL,
SubscriberSSN varchar2(12) NULL,
SubscriberDOB varchar2(10) NULL,
SubscriberSex char(1) NULL,
SubscriberEthnicity varchar2(55) NULL,
/*Responsible*/
ResponsibleLast varchar2(60) NULL,
ResponsibleFirst varchar2(35) NULL,
ResponsibleMiddle varchar2(25) NULL,
ResponsibleSuffix varchar2(10) NULL,
ResponsibleAddress varchar2(55) NULL,
ResponsibleAddress2 varchar2(55) NULL,
ResponsibleCity varchar2(30) NULL,
```

```
ResponsibleState char(2) NULL,  
ResponsibleZip varchar2(15) NULL,  
ResponsibleIDType char(2) NULL,  
ResponsibleID varchar2(80) NULL,  
/*Destination Payer*/  
PayerResponsibility char(1) NULL,  
PayerName varchar2(60) NULL,  
PayerIDType char(2) NULL,  
PayerID varchar2(30) NULL,  
PayerAddress varchar2(55) NULL,  
PayerAddress2 varchar2(55) NULL,  
PayerCity varchar2(30) NULL,  
PayerState char(2) NULL,  
PayerZip varchar2(15) NULL,  
GroupNo varchar2(50) NULL,  
GroupName varchar2(60) NULL,  
InsuranceType char(2) NULL,  
FilingIndicator char(2) NULL,  
COBIndicator char(1) NULL,  
/*Patient*/  
PatientLast varchar2(60) NULL,  
PatientFirst varchar2(35) NULL,  
PatientMiddle varchar2(25) NULL,  
PatientSuffix varchar2(10) NULL,  
PatientAddress varchar2(55) NULL,  
PatientAddress2 varchar2(55) NULL,
```

```
PatientCity varchar2(30) NULL,  
PatientState char(2) NULL,  
PatientZip varchar2(15) NULL,  
PatientCountry varchar2(3) NULL,  
PatientSubdivision varchar2(3) NULL,  
PatientIDType char(2) NULL,  
PatientID varchar2(30) NULL,  
PatientSSN varchar2(12) NULL,  
PatientDOB varchar2(50) NULL,  
PatientSex char(1) NULL,  
PatientEthnicity varchar2(35) NULL,  
PatientRelationship char(2) NULL,  
PatientPaid varchar2(10) NULL,  
PregnancyIndicator char(1) NULL,  
/*COB Payer 1*/  
OtherInsuredLast varchar2(60) NULL,  
OtherInsuredFirst varchar2(35) NULL,  
OtherInsuredMiddle varchar2(25) NULL,  
OtherInsuredSuffix varchar2(10) NULL,  
OtherInsuredIDQual char(2) NULL,  
OtherInsuredID varchar2(30) NULL,  
OtherInsuredAddress varchar2(55) NULL,  
OtherInsuredAddress2 varchar2(55) NULL,  
OtherInsuredCity varchar2(30) NULL,  
OtherInsuredState char(2) NULL,  
OtherInsuredZip varchar2(15) NULL,
```

```
OtherInsuredCountry varchar2(3) NULL,  
OtherInsuredSubdivision varchar2(3) NULL,  
OtherInsuredPolicy varchar2(50) NULL,  
OtherInsuredDOB varchar2(10) NULL,  
OtherInsuredSex char(1) NULL,  
OtherInsuredRelationship char(2) NULL,  
OtherPayerResponsibility char(1) NULL,  
OtherPayerName varchar2(35) NULL,  
OtherPayerIDQual char(2) NULL,  
OtherPayerID varchar2(30) NULL,  
OtherPayerPaid varchar2(10) NULL,  
OtherPayerGroupName varchar2(60) NULL,  
OtherPayerGroupNo varchar2(50) NULL,  
OtherPayerInsuranceType char(2) NULL,  
OtherPayerFilingIndicator char(2) NULL,  
OtherPayerAdjReasonGroup varchar2(20) NULL,  
OtherPayerAdjReason varchar2(50) NULL,  
OtherPayerAdjAmount varchar2(120) NULL,  
OtherPayerPaidDate date NULL,  
OtherPayerInfoRelease char(1) NULL,  
OtherPayerBenefitAssign char(1) NULL,  
OtherPayerDocControlNo varchar2(50) NULL,  
OtherPayerPriorAuth varchar2(50) NULL,  
  
/*COB Payer 2*/  
  
OtherInsured2Last varchar2(60) NULL,  
OtherInsured2First varchar2(35) NULL,
```

OtherInsured2Middle varchar2(25) NULL,
OtherInsured2Suffix varchar2(10) NULL,
OtherInsured2IDQual char(2) NULL,
OtherInsured2ID varchar2(30) NULL,
OtherInsured2Address varchar2(55) NULL,
OtherInsured2Address2 varchar2(55) NULL,
OtherInsured2City varchar2(30) NULL,
OtherInsured2State char(2) NULL,
OtherInsured2Zip varchar2(15) NULL,
OtherInsured2Country varchar2(3) NULL,
OtherInsured2Subdivision varchar2(3) NULL,
OtherInsured2Policy varchar2(50) NULL,
OtherInsured2DOB varchar2(10) NULL,
OtherInsured2Sex char(1) NULL,
OtherInsured2Relationship char(2) NULL,
OtherPayer2Responsibility char(1) NULL,
OtherPayer2Name varchar2(35) NULL,
OtherPayer2IDQual char(2) NULL,
OtherPayer2ID varchar2(80) NULL,
OtherPayer2Paid varchar2(10) NULL,
OtherPayer2GroupName varchar2(60) NULL,
OtherPayer2GroupNo varchar2(50) NULL,
OtherPayer2InsuranceType char(2) NULL,
OtherPayer2FilingIndicator char(2) NULL,
OtherPayer2AdjReasonGroup char(20) NULL,
OtherPayer2AdjReason varchar2(50) NULL,

```
OtherPayer2AdjAmount varchar2(120) NULL,  
OtherPayer2PaidDate date NULL,  
OtherPayer2InfoRelease char(1) NULL,  
OtherPayer2BenefitAssign char(1) NULL,  
OtherPayer2DocControlNo varchar2(50) NULL,  
OtherPayer2PriorAuth varchar2(50) NULL,  
/*Claim Data*/  
ClaimNo varchar2(50) NOT NULL,  
Amount NUMBER(19,4) NULL,  
EstimatedAmountDue varchar2(10) NULL,  
PlaceOfService char(2) NULL,  
SubmitReason char(1) NULL,  
ProviderSignature char(1) NULL,  
ProviderAcceptsAssignment char(1) NULL,  
BenefitAssignment char(1) NULL,  
InfoReleaseCode char(1) NULL,  
PatientSignatureCode char(1) NULL,  
RelatedCauses char(2) NULL,  
RelatedCauses2 char(2) NULL,  
RelatedCausesState char(2) NULL,  
SpecialProgramCode char(3) NULL,  
ProviderParticipation char(1) NULL,  
EOBIndicator char(1) NULL,  
DelayReasonCode char(2) NULL,  
ServiceDateFrom date NULL,  
ServiceDateTo date NULL,
```

OnsetDate date NULL,
SimilarSymptomsDate date NULL,
DisabilityBegin date NULL,
DisabilityEnd date NULL,
HospitalizationBegin date NULL,
HospitalizationEnd date NULL,
AccidentDate date NULL,
LastMenstrualPeriod date NULL,
RepricerReceivedDate date NULL,
AdmissionDate date NULL,
AdmissionHour VARCHAR2(10) NULL,
AdmissionType char(1) NULL,
AdmissionSource char(1) NULL,
DischargeHour VARCHAR2(10) NULL,
PatientStatus char(2) NULL,
CoveredDays varchar2(4) NULL,
NonCoveredDays varchar2(4) NULL,
COBDays varchar2(4) NULL,
LifeTimeReserveDays varchar2(4) NULL,
PriorAuthorization varchar2(50) NULL,
ClearingHouseID varchar2(50) NULL,
MedicalRecordNumber varchar2(50) NULL,
ServiceAuthorizationException varchar2(50) NULL,
ReferralNumber varchar2(50) NULL,
PayerClaimControlNumber varchar2(50) NULL,
AdjustedRepricedClaimNumber varchar2(50) NULL,

```
AutoAccidentState varchar2(50) NULL,  
OrigRefNo varchar2(50) NULL,  
ClaimType char(1) NULL,  
TypeOfBill varchar2(20) NULL,  
Remark1 varchar2(90) NULL,  
Remark2 varchar2(90) NULL,  
Remark3 varchar2(90) NULL,  
Remark4 varchar2(90) NULL,  
K3_1 varchar2(80) NULL,  
K3_2 varchar2(80) NULL,  
K3_3 varchar2(80) NULL,  
K3_4 varchar2(80) NULL,  
K3_5 varchar2(80) NULL,  
K3_6 varchar2(80) NULL,  
K3_7 varchar2(80) NULL,  
K3_8 varchar2(80) NULL,  
K3_9 varchar2(80) NULL,  
K3_10 varchar2(80) NULL,  
OutsideLab char(1) NULL,  
LabCharge varchar2(10) NULL,  
Test_Prod char(1) NULL,  
ReportTypeCode1 char(2) NULL,  
ReportTransmissionCode1 varchar2(2) NULL,  
AttachmentControlNumber1 varchar2(80) NULL,  
/*Dental Claims*/  
Predetermination char(2) NULL,
```

```
OrthodonticTotal varchar2(5) NULL,  
OrthodonticRemaining varchar2(5) NULL,  
OrthodonticYesNo char(1) NULL,  
ToothStatus varchar2(100) NULL,  
AppliancePlacementDate date NULL,  
/*Diagnosis Information*/  
AdmitDiagnosis varchar2(10) NULL,  
ECode varchar2(30) NULL,  
ECode2 varchar2(30) NULL,  
ECode3 varchar2(30) NULL,  
ECode4 varchar2(30) NULL,  
ECode5 varchar2(30) NULL,  
ECode6 varchar2(30) NULL,  
ECode7 varchar2(30) NULL,  
ECode8 varchar2(30) NULL,  
--ECode9 varchar2(30) NULL,  
--ECode10 varchar2(30) NULL,  
--ECode11 varchar2(30) NULL,  
--ECode12 varchar2(30) NULL,  
ReasonForVisit varchar2(30) NULL,  
ReasonForVisit2 varchar2(30) NULL,  
ReasonForVisit3 varchar2(30) NULL,  
PrincipalDiagnosis varchar2(10) NULL,  
Diag2 varchar2(10) NULL,  
Diag3 varchar2(10) NULL,  
Diag4 varchar2(10) NULL,
```

```
Diag5 varchar2(10) NULL,  
Diag6 varchar2(10) NULL,  
Diag7 varchar2(10) NULL,  
Diag8 varchar2(10) NULL,  
Diag9 varchar2(10) NULL,  
Diag10 varchar2(10) NULL,  
Diag11 varchar2(10) NULL,  
Diag12 varchar2(10) NULL,  
Diag13 varchar2(10) NULL,  
Diag14 varchar2(10) NULL,  
Diag15 varchar2(10) NULL,  
Diag16 varchar2(10) NULL,  
Diag17 varchar2(10) NULL,  
Diag18 varchar2(10) NULL,  
Diag19 varchar2(10) NULL,  
Diag20 varchar2(10) NULL,  
Diag21 varchar2(10) NULL,  
Diag22 varchar2(10) NULL,  
Diag23 varchar2(10) NULL,  
Diag24 varchar2(10) NULL,  
Diag25 varchar2(10) NULL,  
DRG varchar2(15) NULL,  
  
/*Procedure Information*/  
  
PrincipalProcedure varchar2(10) NULL,  
PrincipalProcedureDate date NULL,  
Proc2 varchar2(50) NULL,
```

```
Proc2Date date NULL,  
  
Proc3 varchar2(10) NULL,  
  
Proc3Date date NULL,  
  
Proc4 varchar2(10) NULL,  
  
Proc4Date date NULL,  
  
Proc5 varchar2(10) NULL,  
  
Proc5Date date NULL,  
  
Proc6 varchar2(10) NULL,  
  
Proc6Date date NULL,  
  
Proc7 varchar2(10) NULL,  
  
Proc7Date date NULL,  
  
Proc8 varchar2(10) NULL,  
  
Proc8Date date NULL,  
  
Proc9 varchar2(10) NULL,  
  
Proc9Date date NULL,  
  
Proc10 varchar2(10) NULL,  
  
Proc10Date date NULL,  
  
Proc11 varchar2(10) NULL,  
  
Proc11Date date NULL,  
  
Proc12 varchar2(10) NULL,  
  
Proc12Date date NULL,  
  
Proc13 varchar2(10) NULL,  
  
Proc13Date date NULL,  
  
Proc14 varchar2(10) NULL,  
  
Proc14Date date NULL,  
  
Proc15 varchar2(10) NULL,
```

```
Proc15Date date NULL,  
  
Proc16 varchar2(10) NULL,  
  
Proc16Date date NULL,  
  
Proc17 varchar2(10) NULL,  
  
Proc17Date date NULL,  
  
Proc18 varchar2(10) NULL,  
  
Proc18Date date NULL,  
  
Proc19 varchar2(10) NULL,  
  
Proc19Date date NULL,  
  
Proc20 varchar2(10) NULL,  
  
Proc20Date date NULL,  
  
Proc21 varchar2(10) NULL,  
  
Proc21Date date NULL,  
  
Proc22 varchar2(10) NULL,  
  
Proc22Date date NULL,  
  
Proc23 varchar2(10) NULL,  
  
Proc23Date date NULL,  
  
Proc24 varchar2(10) NULL,  
  
Proc24Date date NULL,  
  
Proc25 varchar2(10) NULL,  
  
Proc25Date date NULL,  
  
/*Value Codes*/  
  
ValueCode1 char(3) NULL,  
  
ValueAmount1 varchar2(10) NULL,  
  
ValueCode2 char(3) NULL,  
  
ValueAmount2 varchar2(10) NULL,
```

```
ValueCode3 char(3) NULL,  
ValueAmount3 varchar2(10) NULL,  
ValueCode4 char(3) NULL,  
ValueAmount4 varchar2(10) NULL,  
ValueCode5 char(3) NULL,  
ValueAmount5 varchar2(10) NULL,  
ValueCode6 char(3) NULL,  
ValueAmount6 varchar2(10) NULL,  
ValueCode7 varchar2(10) NULL,  
ValueAmount7 varchar2(10) NULL,  
ValueCode8 char(3) NULL,  
ValueAmount8 varchar2(10) NULL,  
ValueCode9 char(3) NULL,  
ValueAmount9 varchar2(10) NULL,  
ValueCode10 char(3) NULL,  
ValueAmount10 varchar2(10) NULL,  
ValueCode11 char(3) NULL,  
ValueAmount11 varchar2(10) NULL,  
ValueCode12 char(3) NULL,  
ValueAmount12 varchar2(10) NULL,  
--ValueCode13 char(3) NULL,  
--ValueAmount13 varchar2(10) NULL,  
--ValueCode14 char(3) NULL,  
--ValueAmount14 varchar2(10) NULL,  
--ValueCode15 char(3) NULL,  
--ValueAmount15 varchar2(10) NULL,
```

```
--ValueCode16 char(3) NULL,  
--ValueAmount16 varchar2(10) NULL,  
--ValueCode17 char(3) NULL,  
--ValueAmount17 varchar2(10) NULL,  
--ValueCode18 char(3) NULL,  
--ValueAmount18 varchar2(10) NULL,  
--ValueCode19 char(3) NULL,  
--ValueAmount19 varchar2(10) NULL,  
--ValueCode20 char(3) NULL,  
--ValueAmount20 varchar2(10) NULL,  
--ValueCode21 char(3) NULL,  
--ValueAmount21 varchar2(10) NULL,  
--ValueCode22 char(3) NULL,  
--ValueAmount22 varchar2(10) NULL,  
--ValueCode23 char(3) NULL,  
--ValueAmount23 varchar2(10) NULL,  
--ValueCode24 char(3) NULL,  
--ValueAmount24 varchar2(10) NULL,  
  
/*Condition Codes*/  
ConditionCode1 varchar2(30) NULL,  
ConditionCode2 varchar2(30) NULL,  
ConditionCode3 varchar2(30) NULL,  
ConditionCode4 varchar2(30) NULL,  
ConditionCode5 varchar2(30) NULL,  
ConditionCode6 varchar2(30) NULL,  
ConditionCode7 varchar2(30) NULL,
```

```
ConditionCode8 varchar2(30) NULL,  
ConditionCode9 varchar2(30) NULL,  
ConditionCode10 varchar2(30) NULL,  
--ConditionCode11 varchar2(30) NULL,  
--ConditionCode12 varchar2(30) NULL,  
--ConditionCode13 varchar2(30) NULL,  
--ConditionCode14 varchar2(30) NULL,  
--ConditionCode15 varchar2(30) NULL,  
--ConditionCode16 varchar2(30) NULL,  
--ConditionCode17 varchar2(30) NULL,  
--ConditionCode18 varchar2(30) NULL,  
--ConditionCode19 varchar2(30) NULL,  
--ConditionCode20 varchar2(30) NULL,  
--ConditionCode21 varchar2(30) NULL,  
--ConditionCode22 varchar2(30) NULL,  
--ConditionCode23 varchar2(30) NULL,  
--ConditionCode24 varchar2(30) NULL,  
  
/*Occurrence Codes*/  
  
OccurrenceCode1 char(3) NULL,  
  
OccurrenceDate1 date NULL,  
  
OccurrenceCode2 char(3) NULL,  
  
OccurrenceDate2 date NULL,  
  
OccurrenceCode3 char(3) NULL,  
  
OccurrenceDate3 date NULL,  
  
OccurrenceCode4 char(3) NULL,  
  
OccurrenceDate4 date NULL,
```

```
OccurrenceCode5 char(3) NULL,  
OccurrenceDate5 date NULL,  
OccurrenceCode6 char(3) NULL,  
OccurrenceDate6 date NULL,  
OccurrenceCode7 char(3) NULL,  
OccurrenceDate7 date NULL,  
OccurrenceCode8 char(3) NULL,  
OccurrenceDate8 date NULL,  
--OccurrenceCode9 char(3) NULL,  
--OccurrenceDate9 date NULL,  
--OccurrenceCode10 char(3) NULL,  
--OccurrenceDate10 date NULL,  
--OccurrenceCode11 char(3) NULL,  
--OccurrenceDate11 date NULL,  
--OccurrenceCode12 char(3) NULL,  
--OccurrenceDate12 date NULL,  
--OccurrenceCode13 char(3) NULL,  
--OccurrenceDate13 date NULL,  
--OccurrenceCode14 char(3) NULL,  
--OccurrenceDate14 date NULL,  
--OccurrenceCode15 char(3) NULL,  
--OccurrenceDate15 date NULL,  
--OccurrenceCode16 char(3) NULL,  
--OccurrenceDate16 date NULL,  
--OccurrenceCode17 char(3) NULL,  
--OccurrenceDate17 date NULL,
```

```
--OccurrenceCode18 char(3) NULL,  
--OccurrenceDate18 date NULL,  
--OccurrenceCode19 char(3) NULL,  
--OccurrenceDate19 date NULL,  
--OccurrenceCode20 char(3) NULL,  
--OccurrenceDate20 date NULL,  
--OccurrenceCode21 char(3) NULL,  
--OccurrenceDate21 date NULL,  
--OccurrenceCode22 char(3) NULL,  
--OccurrenceDate22 date NULL,  
--OccurrenceCode23 char(3) NULL,  
--OccurrenceDate23 date NULL,  
--OccurrenceCode24 char(3) NULL,  
--OccurrenceDate24 date NULL,  
OccurrenceSpanCode1 char(3) NULL,  
OccurrenceSpanFrom1 date NULL,  
OccurrenceSpanTo1 date NULL,  
OccurrenceSpanCode2 char(3) NULL,  
OccurrenceSpanFrom2 date NULL,  
OccurrenceSpanTo2 date NULL,  
OccurrenceSpanCode3 char(3) NULL,  
OccurrenceSpanFrom3 date NULL,  
OccurrenceSpanTo3 date NULL,  
OccurrenceSpanCode4 char(3) NULL,  
OccurrenceSpanFrom4 date NULL,  
OccurrenceSpanTo4 date NULL,
```

```
/*CR1*/

PatientWeight varchar2(10) NULL,

AmbulanceTransportCode char(1) NULL,

AmbulanceTransportReasonCode char(1) NULL,

TransportDistance varchar2(15) NULL,

RoundTripPurposeDescription varchar2(80) NULL,

StretcherPurposeDescription varchar2(80) NULL,

/* CRC - Ambulance Certification */

AmbulanceConditionIndicator char(1) NULL,

AmbulanceConditionCode1 char(3) NULL,

AmbulanceConditionCode2 char(3) NULL,

AmbulanceConditionCode3 char(3) NULL,

AmbulanceConditionCode4 char(3) NULL,

AmbulanceConditionCode5 char(3) NULL,

/*Repricing*/

RepricedClaimNumber varchar2(50) NULL,

RepricingMethodology varchar2(3) NULL,

RepricedAmount number(19,4) NULL,

SavingsAmount number(19,4) NULL,

RepricerID varchar2(30) NULL,

RepricingRate varchar2(9) NULL,

APG_Code varchar2(30) NULL,

APG_Amount number(19,4) NULL,

ApprovedRevenueCode varchar2 (48) NULL,

ApprovedProcedureCode varchar2 (48) NULL,

ApprovedUnitCode char(2) NULL,
```

```
    ApprovedUnits number(19) NULL,
    RejectReason varchar2(3) NULL,
    ComplianceCode varchar2(2) NULL,
    ExceptionCode varchar2(2) NULL,
    /*SQL To EDI*/
   --[ClaimAction] char(1) NULL,
   --[ErrorMessage] varchar2(1000) NULL
)NOLOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;

create sequence edi_claims_seq start with 1 increment by 1;

create or replace trigger edi_claims_trigger
before insert on EDI_Claims
REFERENCING NEW AS newrow
for each row
begin
    select edi_claims_seq.nextval into :newrow.ID from dual;
end;
```

Detail

```
CREATE TABLE EDI_ClaimDetail(
    ID number(19) NOT NULL,
    ClaimID number(19) NOT NULL,
```

LineNumber number(18, 0) NULL,
LineID varchar2(50) NULL,
ServiceDateFrom varchar2(50) NULL,
ServiceDateTo varchar2(50) NULL,
AssessmentDate varchar2(50) NULL,
FacilityCode varchar2(50) NULL,
RevenueCode varchar2(50) NULL,
ProcedureCode varchar2(50) NULL,
Amount varchar2(50) NULL,
Unit varchar2(50) NULL,
Quantity varchar2(50) NULL,
UnitRate varchar2(50) NULL,
NonCovered varchar2(50) NULL,
Paid varchar2(50) NULL,
PlaceOfService char(3) NULL,
Modifier1 char(3) NULL,
Modifier2 char(3) NULL,
Modifier3 char(3) NULL,
Modifier4 char(3) NULL,
ProcedureDescription varchar2(80) NULL,
OralCavityDesignation1 char(3) NULL,
OralCavityDesignation2 char(3) NULL,
OralCavityDesignation3 char(3) NULL,
OralCavityDesignation4 char(3) NULL,
OralCavityDesignation5 char(3) NULL,
DiagPointer1 char(1) NULL,

```
DiagPointer2 char(1) NULL,  
DiagPointer3 char(1) NULL,  
DiagPointer4 char(1) NULL,  
ToothNumber char(2) NULL,  
Surface varchar2(10) NULL,  
EmergencyIndicator char(1) NULL,  
ServiceTax number(19,4) NULL,  
FacilityTax number(19,4) NULL,  
SalesTax number(19,4) NULL,  
ApprovedAmount number(19,4) NULL,  
LineK3_01 varchar2(80) NULL,  
LineK3_02 varchar2(80) NULL,  
LineK3_03 varchar2(80) NULL,  
LineK3_04 varchar2(80) NULL,  
LineK3_05 varchar2(80) NULL,  
LineK3_06 varchar2(80) NULL,  
LineK3_07 varchar2(80) NULL,  
LineK3_08 varchar2(80) NULL,  
LineK3_09 varchar2(80) NULL,  
LineK3_10 varchar2(80) NULL,  
Remark varchar2(100) NULL,  
AmbulancePatientCount bigint NULL,  
  
/****PWK****/  
  
ReportType char(2) NULL,  
ReportTransmission char(2) NULL,  
  
AttachmentControlNumber varchar2(80) NULL,
```

```
/*HCP*/

RepricingMethodology varchar2(3) NULL,

RepricedAmount number(19,4) NULL,

SavingsAmount number(19,4) NULL,

RepricerID varchar2(30) NULL,

RepricingRate varchar2(9) NULL,

APG_Code varchar2(30) NULL,

APG_Amount number(19,4) NULL,

ApprovedRevenueCode varchar2 (48) NULL,

ApprovedProcedureCode varchar2 (48) NULL,

ApprovedUnitCode char(2) NULL,

ApprovedUnits number(19) NULL,

RejectReason varchar2(3) NULL,

ComplianceCode varchar2(2) NULL,

ExceptionCode varchar2(2) NULL,

/* Drugs */

DrugCode varchar2(48) NULL,

DrugUnitPrice number(19,4) NULL,

DrugUnitCode char(2) NULL,

DrugUnits float NULL,

PrescriptionNumber varchar2(30) NULL,

/* CR1 */

PatientWeight varchar2(10) NULL,

AmbulanceTransportCode char(1) NULL,

AmbulanceTransportReasonCode char(1) NULL,

TransportDistance varchar2(15) NULL,
```

```
RoundTripPurposeDescription varchar2(80) NULL,  
StretcherPurposeDescription varchar2(80) NULL,  
  
/* CRC - Ambulance Certification */  
  
AmbulanceConditionIndicator char(1) NULL,  
AmbulanceConditionCode1 char(3) NULL,  
AmbulanceConditionCode2 char(3) NULL,  
AmbulanceConditionCode3 char(3) NULL,  
AmbulanceConditionCode4 char(3) NULL,  
AmbulanceConditionCode5 char(3) NULL,  
  
/* Attending Provider */  
  
AttendingProviderLast varchar2(35) NULL,  
AttendingProviderFirst varchar2(25) NULL,  
AttendingProviderMiddle varchar2(25) NULL,  
AttendingProviderSuffix varchar2(10) NULL,  
AttendingProviderIDQual char(2) NULL,  
AttendingProviderID varchar2(35) NULL,  
AttendingProviderOtherIDQual char(2) NULL,  
AttendingProviderOtherID varchar2(30) NULL,  
  
/* Operating Provider */  
  
OperatingProviderLast varchar2(35) NULL,  
OperatingProviderFirst varchar2(25) NULL,  
OperatingProviderMiddle varchar2(25) NULL,  
OperatingProviderSuffix varchar2(10) NULL,  
OperatingProviderIDQual char(2) NULL,  
OperatingProviderID varchar2(35) NULL,  
OperatingProviderOtherIDQual char(2) NULL,
```

```
OperatingProviderOtherID varchar2(30) NULL,  
  
/* Other Provider */  
  
OtherProviderLast varchar2(35) NULL,  
  
OtherProviderFirst varchar2(25) NULL,  
  
OtherProviderMiddle varchar2(25) NULL,  
  
OtherProviderSuffix varchar2(10) NULL,  
  
OtherProviderIDQual char(2) NULL,  
  
OtherProviderID varchar2(35) NULL,  
  
OtherProviderOtherIDQual char(2) NULL,  
  
OtherProviderOtherID varchar2(30) NULL,  
  
/* Rendering Provider */  
  
RenderingProviderLast varchar2(35) NULL,  
  
RenderingProviderFirst varchar2(25) NULL,  
  
RenderingProviderMiddle varchar2(25) NULL,  
  
RenderingProviderSuffix varchar2(10) NULL,  
  
RenderingProviderIDQual char(2) NULL,  
  
RenderingProviderID varchar2(35) NULL,  
  
RenderingProviderOtherIDQual char(2) NULL,  
  
RenderingProviderOtherID varchar2(30) NULL,  
  
/* PurchasedService Provider */  
  
PurchasedSrvProviderLast varchar2(35) NULL,  
  
PurchasedSrvProviderFirst varchar2(25) NULL,  
  
PurchasedSrvProviderMiddle varchar2(25) NULL,  
  
PurchasedSrvProviderSuffix varchar2(10) NULL,  
  
PurchasedSrvProviderIDQual char(2) NULL,  
  
PurchasedSrvProviderID varchar2(35) NULL,
```

```
PurchasedSrvProvOtherIDQual char(2) NULL,  
PurchasedSrvProvOtherID varchar2(30) NULL,  
  
/* Facility Provider */  
FacilityName varchar2(35) NULL,  
FacilityIDQual char(2) NULL,  
FacilityID varchar2(35) NULL,  
FacilityAddress1 varchar2(25) NULL,  
FacilityAddress2 varchar2(25) NULL,  
FacilityCity varchar2(30) NULL,  
FacilityState char(2) NULL,  
FacilityZip varchar2(15) NULL,  
FacilityOtherIDQual char(2) NULL,  
FacilityOtherID varchar2(30) NULL,  
  
/* Supervising Provider */  
SupervisingProviderLast varchar2(35) NULL,  
SupervisingProviderFirst varchar2(25) NULL,  
SupervisingProviderMiddle varchar2(25) NULL,  
SupervisingProviderSuffix varchar2(10) NULL,  
SupervisingProviderIDQual char(2) NULL,  
SupervisingProviderID varchar2(35) NULL,  
SupervisingProviderOtherIDQual char(2) NULL,  
SupervisingProviderOtherID varchar2(30) NULL,  
  
/* Ordering Provider */  
OrderingProviderLast varchar2(35) NULL,  
OrderingProviderFirst varchar2(25) NULL,  
OrderingProviderMiddle varchar2(25) NULL,
```

```
OrderingProviderSuffix varchar2(10) NULL,  
OrderingProviderIDQual char(2) NULL,  
OrderingProviderID varchar2(35) NULL,  
OrderingProviderOtherIDQual char(2) NULL,  
OrderingProviderOtherID varchar2(30) NULL,  
  
/* Referring Provider */  
  
ReferringProviderLast varchar2(35) NULL,  
ReferringProviderFirst varchar2(25) NULL,  
ReferringProviderMiddle varchar2(25) NULL,  
ReferringProviderSuffix varchar2(10) NULL,  
ReferringProviderIDQual char(2) NULL,  
ReferringProviderID varchar2(35) NULL,  
ReferringProviderOtherIDQual char(2) NULL,  
ReferringProviderOtherID varchar2(30) NULL,  
  
/*COB 1 */  
  
OtherPayer1ID varchar2(30) NULL,  
OtherPayer1Paid number(19,4) NULL,  
OtherPayer1PaidProcedure varchar2(30) NULL,  
OtherPayer1PaidRevenueCode varchar2(30) NULL,  
OtherPayer1PaidQuantity number(19) NULL,  
OtherPayer1BundledLine int NULL,  
OtherPayer1AdjReasonGroup1 char(2) NULL,  
OtherPayer1AdjReason1 varchar2(5) NULL,  
OtherPayer1AdjAmount1 number(19,4) NULL,  
OtherPayer1AdjQuantity1 number(19) NULL,  
OtherPayer1AdjReasonGroup2 char(2) NULL,
```

```
OtherPayer1AdjReason2 varchar2(5) NULL,  
OtherPayer1AdjAmount2 number(19,4) NULL,  
OtherPayer1AdjQuantity2 number(19) NULL,  
OtherPayer1AdjReasonGroup3 char(2) NULL,  
OtherPayer1AdjReason3 varchar2(5) NULL,  
OtherPayer1AdjAmount3 number(19,4) NULL,  
OtherPayer1AdjQuantity3 number(19) NULL,  
OtherPayer1AdjReasonGroup4 char(2) NULL,  
OtherPayer1AdjReason4 varchar2(5) NULL,  
OtherPayer1AdjAmount4 number(19,4) NULL,  
OtherPayer1AdjQuantity4 number(19) NULL,  
--OtherPayer1AdjReasonGrp5Plus varchar2(77) NULL,  
--OtherPayer1AdjReason5Plus varchar2(155) NULL,  
--OtherPayer1AdjAmount5Plus varchar2(493) NULL,  
--OtherPayer1AdjQuantity5Plus varchar2(415) NULL,  
OtherPayer1PaidDate date NULL,  
  
/*COB 2 */  
  
OtherPayer2ID varchar2(30) NULL,  
OtherPayer2Paid number(19,4) NULL,  
OtherPayer2PaidProcedure varchar2(30) NULL,  
OtherPayer2PaidRevenueCode varchar2(30) NULL,  
OtherPayer2PaidQuantity number(19) NULL,  
OtherPayer2BundledLine int NULL,  
OtherPayer2AdjReasonGroup1 char(2) NULL,  
OtherPayer2AdjReason1 varchar2(5) NULL,  
OtherPayer2AdjAmount1 number(19,4) NULL,
```

```
OtherPayer2AdjQuantity1 number(19) NULL,
OtherPayer2AdjReasonGroup2 char(2) NULL,
OtherPayer2AdjReason2 varchar2(5) NULL,
OtherPayer2AdjAmount2 number(19,4) NULL,
OtherPayer2AdjQuantity2 number(19) NULL,
OtherPayer2AdjReasonGroup3 char(2) NULL,
OtherPayer2AdjReason3 varchar2(5) NULL,
OtherPayer2AdjAmount3 number(19,4) NULL,
OtherPayer2AdjQuantity3 number(19) NULL,
OtherPayer2AdjReasonGroup4 char(2) NULL,
OtherPayer2AdjReason4 varchar2(5) NULL,
OtherPayer2AdjAmount4 number(19,4) NULL,
OtherPayer2AdjQuantity4 number(19) NULL,
--OtherPayer2AdjReasonGrp5Plus varchar2(77) NULL,
--OtherPayer2AdjReason5Plus varchar2(155) NULL,
--OtherPayer2AdjAmount5Plus varchar2(493) NULL,
--OtherPayer2AdjQuantity5Plus varchar2(415) NULL,
OtherPayer2PaidDate date NULL

)NOLOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;

create sequence edi_claimdetail_seq start with 1 increment by 1;

create or replace trigger edi_claimdetail_trigger
```

```
before insert on EDI_ClaimDetail
REFERENCING NEW AS newrow
for each row
begin
  select edi_claimdetail_seq.nextval into :newrow.ID from dual;
end;
```

5.3.4 Microsoft Access Script to Create Tables

We do not recommend Access.

It has limitations and problems that make it unacceptable for business solutions. If price is an issue, we recommend the free MySQL software. This is a free, industrial strength Open Source database available at www.MySQL.com

```
CREATE TABLE edi_claims (
  ID INT NOT NULL AUTO_INCREMENT KEY,
  ClaimNo varchar (50) NOT NULL ,
  HIPAAsuiteID varchar (9) NULL ,
  TradingPartnerIDType char (2) NULL ,
  TradingPartnerID varchar (15) NOT NULL ,
  SubmitterName varchar (35) NULL ,
  SubmitterID varchar (30) NULL ,
  SubmitterContact varchar (60) NULL ,
  SubmitterTel varchar (20) NULL ,
  SubmitterTelExt varchar (10) NULL ,
  SubmitterFax varchar (20) NULL ,
  SubmitterEmail varchar (80) NULL ,
  ReceiverName varchar (35) NULL ,
  BillProvIDType char (2) NULL ,
```

BillProvID varchar (80) NULL ,
BillProvLast varchar (35) NULL ,
BillProvFirst varchar (25) NULL ,
BillProvMiddle varchar (25) NULL ,
BillProvSuffix varchar (10) NULL ,
BillProvSpecialty varchar (10) NULL ,
BillProvAddress varchar (55) NULL ,
BillProvAddress2 varchar (55) NULL ,
BillProvCity varchar (30) NULL ,
BillProvState char (2) NULL ,
BillProvZip varchar (15) NULL ,
BillProvContact varchar (60) NULL ,
BillProvTel varchar (80) NULL ,
BillProvTelExt varchar (80) NULL ,
BillProvFax varchar (80) NULL ,
BillProvEmail varchar (80) NULL ,
PayToProvIDType char (2) NULL ,
PaytoProvID varchar (80) NULL ,
PaytoProvLast varchar (35) NULL ,
PaytoProvFirst varchar (25) NULL ,
PayToProvMiddle varchar (25) NULL ,
PayToProvSuffix varchar (10) NULL ,
PayToProvSpecialty varchar (10) NULL ,
PaytoProvAddress varchar (55) NULL ,
PayToProvAddress2 varchar (55) NULL ,
PaytoProvCity varchar (30) NULL ,

PaytoProvState char (2) NULL ,
PaytoProvZip varchar (15) NULL ,
RendProvIDType char (2) NULL ,
RendProvID varchar (80) NULL ,
RendProvLast varchar (35) NULL ,
RendProvFirst varchar (25) NULL ,
RendProvMiddle varchar (25) NULL ,
RendProvSuffix varchar (10) NULL ,
RendProvSpecialty varchar (10) NULL ,
AttendProvLast varchar (35) NULL ,
AttendProvFirst varchar (25) NULL ,
AttendProvMiddle varchar (25) NULL ,
AttendProvSuffix varchar (10) NULL ,
AttendProvIDType char (2) NULL ,
AttendProvID varchar (80) NULL ,
AttendProvSpecialty varchar (10) NULL ,
OperatingProvLast varchar (35) NULL ,
OperatingProvFirst varchar (25) NULL ,
OperatingProvMiddle varchar (25) NULL ,
OperatingProvSuffix varchar (10) NULL ,
OperatingProvIDType char (2) NULL ,
OperatingProvID varchar (80) NULL ,
OperatingProvSpecialty varchar (10) NULL ,
OtherProvLast varchar (35) NULL ,
OtherProvFirst varchar (25) NULL ,
OtherProvMiddle varchar (25) NULL ,

OtherProvSuffix varchar (10) NULL ,
OtherProvIDType varchar (100) NULL ,
OtherProvID varchar (100) NULL ,
OtherProvSpecialty varchar (100) NULL ,
FacilityType char (2) NULL ,
FacilityIDType char (2) NULL ,
FacilityID varchar (80) NULL ,
FacilityName varchar (80) NULL ,
FacilityAddress varchar (55) NULL ,
FacilityAddress2 varchar (55) NULL ,
FacilityCity varchar (30) NULL ,
FacilitySpecialty varchar (100) NULL ,
FacilityState char (2) NULL ,
FacilityZip varchar (15) NULL ,
OutsideLab char (1) NULL ,
LabCharge varchar (10) NULL ,
SubscriberLast varchar (35) NULL ,
SubscriberFirst varchar (25) NULL ,
SubscriberMiddle varchar (25) NULL ,
SubscriberSuffix varchar (10) NULL ,
SubscriberAddress varchar (55) NULL ,
SubscriberAddress2 varchar (55) NULL ,
SubscriberCity varchar (30) NULL ,
SubscriberState char (2) NULL ,
SubscriberZip varchar (15) NULL ,
SubscriberIDType char (2) NULL ,

SubscriberID varchar (80) NULL ,
SubscriberDOB varchar (10) NULL ,
SubscriberSex char (1) NULL ,
InsuranceType char (2) NULL ,
GroupNo varchar (50) NULL ,
GroupName varchar (60) NULL ,
FilingIndicator char (2) NULL ,
PatientLast varchar (35) NULL ,
PatientFirst varchar (25) NULL ,
PatientMiddle varchar (25) NULL ,
PatientSuffix varchar (10) NULL ,
PatientAddress varchar (55) NULL ,
PatientAddress2 varchar (55) NULL ,
PatientCity varchar (30) NULL ,
PatientState char (2) NULL ,
PatientZip varchar (15) NULL ,
PatientIDType char (2) NULL ,
PatientID varchar (80) NULL ,
PatientDOB varchar (50) NULL ,
PatientSex char (1) NULL ,
PatientRelationship char (2) NULL ,
PatientPaid varchar (10) NULL ,
PregnancyIndicator char (1) NULL ,
OtherInsuredLast varchar (35) NULL ,
OtherInsuredFirst varchar (25) NULL ,
OtherInsuredMiddle varchar (25) NULL ,

OtherInsuredSuffix varchar (10) NULL ,
OtherInsuredPolicy varchar (50) NULL ,
OtherInsuredDOB varchar (10) NULL ,
OtherInsuredSex char (1) NULL ,
OtherInsuredRelationship char (2) NULL ,
OtherPayerName varchar (35) NULL ,
OtherPayerID varchar (80) NULL ,
OtherPayerPaid varchar (10) NULL ,
OtherPayerGroupName varchar (100) NULL ,
OtherPayerGroupNo varchar (100) NULL ,
PayerName varchar (35) NULL ,
PayerIDType char (2) NULL ,
PayerID varchar (80) NULL ,
PayerAddress varchar (55) NULL ,
PayerAddress2 varchar (55) NULL ,
PayerCity varchar (30) NULL ,
PayerState char (2) NULL ,
PayerZip varchar (15) NULL ,
Amount varchar (50) NULL ,
EstimatedAmountDue varchar (10) NULL ,
PlaceOfService char (2) NULL ,
SubmitReason char (1) NULL ,
ProviderSignature char (1) NULL ,
ProviderAcceptsAssignment char (1) NULL ,
BenefitAssignment char (1) NULL ,
InfoReleaseCode char (1) NULL ,

PatientSignatureCode char (1) NULL ,
RelatedCauses char (2) NULL ,
RelatedCausesState char (2) NULL ,
SpecialProgramCode char (3) NULL ,
ProviderParticipation char (1) NULL ,
ServiceDateFrom varchar (15) NULL ,
ServiceDateTo varchar (15) NULL ,
OnsetDate varchar (15) NULL ,
SimilarSymptomsDate varchar (10) NULL ,
DisabilityBegin varchar (10) NULL ,
DisabilityEnd varchar (10) NULL ,
HospitalizationBegin varchar (20) NULL ,
HospitalizationEnd varchar (10) NULL ,
AccidentDate varchar (20) NULL ,
AdmissionDate varchar (15) NULL ,
AdmissionHour varchar (10) NULL ,
AdmissionType char (1) NULL ,
AdmissionSource char (1) NULL ,
DischargeHour varchar (10) NULL ,
PatientStatus char (2) NULL ,
MedicalRecordNumber varchar (50) NULL ,
CoveredDays varchar (4) NULL ,
NonCoveredDays varchar (4) NULL ,
COBDays varchar (4) NULL ,
LifeTimeReserveDays varchar (4) NULL ,
PriorAuthorization varchar (50) NULL ,

OrigRefNo varchar (50) NULL ,
RefProvLast varchar (35) NULL ,
RefProvFirst varchar (25) NULL ,
RefProvMiddle varchar (25) NULL ,
RefProvSuffix varchar (10) NULL ,
RefProvIDType char (2) NULL ,
RefProvID varchar (80) NULL ,
RefProvSpecialty varchar (10) NULL ,
AdmitDiagnosis varchar (10) NULL ,
ECode varchar (100) NULL ,
PrincipalDiagnosis varchar (10) NULL ,
Diag2 varchar (10) NULL ,
Diag3 varchar (10) NULL ,
Diag4 varchar (10) NULL ,
Diag5 varchar (10) NULL ,
Diag6 varchar (10) NULL ,
Diag7 varchar (10) NULL ,
Diag8 varchar (10) NULL ,
Diag9 varchar (10) NULL ,
PrincipalProcedure varchar (10) NULL ,
PrincipalProcedureDate varchar (10) NULL ,
Proc2 varchar (50) NULL ,
Proc2Date varchar (10) NULL ,
Proc3 varchar (10) NULL ,
Proc3Date varchar (10) NULL ,
Proc4 varchar (10) NULL ,

```
Proc4Date varchar (10) NULL ,  
Proc5 varchar (10) NULL ,  
Proc5Date varchar (10) NULL ,  
Proc6 varchar (10) NULL ,  
Proc6Date varchar (10) NULL ,  
DRG varchar (15) NULL ,  
ValueCode1 char (3) NULL ,  
ValueAmount1 varchar (10) NULL ,  
ValueCode2 char (3) NULL ,  
ValueAmount2 varchar (10) NULL ,  
ValueCode3 char (3) NULL ,  
ValueAmount3 varchar (10) NULL ,  
ValueCode4 char (3) NULL ,  
ValueAmount4 varchar (10) NULL ,  
ValueCode5 char (3) NULL ,  
ValueAmount5 varchar (10) NULL ,  
ValueCode6 char (3) NULL ,  
ValueAmount6 varchar (10) NULL ,  
ValueCode7 varchar (10) NULL ,  
ValueAmount7 varchar (10) NULL ,  
ValueCode8 char (3) NULL ,  
ValueAmount8 varchar (10) NULL ,  
ValueCode9 char (3) NULL ,  
ValueAmount9 varchar (10) NULL ,  
ValueCode10 char (3) NULL ,  
ValueAmount10 varchar (10) NULL ,
```

ValueCode11 char (3) NULL ,
ValueAmount11 varchar (10) NULL ,
ValueCode12 char (3) NULL ,
ValueAmount12 varchar (10) NULL ,
ConditionCode1 varchar (30) NULL ,
ConditionCode2 varchar (30) NULL ,
ConditionCode3 varchar (30) NULL ,
ConditionCode4 varchar (30) NULL ,
ConditionCode5 varchar (30) NULL ,
ConditionCode6 varchar (30) NULL ,
ConditionCode7 varchar (30) NULL ,
ConditionCode8 varchar (30) NULL ,
FileName varchar (100) NULL ,
OccurrenceCode1 char (3) NULL ,
OccurrenceDate1 varchar (10) NULL ,
OccurrenceCode2 char (3) NULL ,
OccurrenceDate2 varchar (10) NULL ,
OccurrenceCode3 char (3) NULL ,
OccurrenceDate3 varchar (10) NULL ,
OccurrenceCode4 char (3) NULL ,
OccurrenceDate4 varchar (10) NULL ,
OccurrenceCode5 char (3) NULL ,
OccurrenceDate5 varchar (10) NULL ,
OccurrenceCode6 char (3) NULL ,
OccurrenceDate6 varchar (10) NULL ,
OccurrenceCode7 char (3) NULL ,

```
OccuranceDate7 varchar (10) NULL ,
OccuranceCode8 char (3) NULL ,
OccuranceDate8 varchar (10) NULL ,
OccuranceSpanCode1 char (3) NULL ,
OccuranceSpanFrom1 varchar (10) NULL ,
OccuranceSpanTo1 varchar (10) NULL ,
OccuranceSpanCode2 char (3) NULL ,
OccuranceSpanFrom2 varchar (10) NULL ,
OccuranceSpanTo2 varchar (10) NULL ,
Test_Prod char (1) NULL ,
ClaimType char (1) NULL ,
TypeofBill varchar (20) NULL ,
Remark1 varchar (90) NULL ,
Remark2 varchar (90) NULL ,
Remark3 varchar (90) NULL ,
Remark4 varchar (90) NULL ,
Claim_status char (1) NULL ,
TargetTP varchar (30) NULL ,
ClearingHouseID varchar (100) NULL,
    RepricedClaimNumber varchar (100) NULL,
    RepricingMethodology varchar (3) NULL,
    RepricedAmount varchar (15) NULL,
    SavingsAmount varchar (15) NULL,
    RepricerID varchar (30) NULL,
    RejectReason varchar (3) NULL
) ;
```

```
CREATE TABLE edi_claimdetail (  
  
    ID INT NOT NULL AUTO_INCREMENT KEY,  
  
    ClaimID varchar (50) NOT NULL ,  
  
    HIPAAsuiteID varchar (10) NULL ,  
  
    LineNumber decimal(18, 0) NOT NULL ,  
  
    LineID varchar (50) NULL ,  
  
    ServiceDateFrom varchar (50) NULL ,  
  
    ServiceDateTo varchar (50) NULL ,  
  
    FacilityCode varchar (50) NULL ,  
  
    RevenueCode varchar (50) NULL ,  
  
    ProcedureCode varchar (50) NULL ,  
    Description varchar (100) NULL,  
  
    Amount varchar (50) NULL ,  
  
    Unit varchar (50) NULL ,  
  
    Quantity varchar (50) NULL ,  
  
    UnitRate varchar (50) NULL ,  
  
    NonCovered varchar (50) NULL ,  
  
    Paid varchar (50) NULL ,  
  
    PlaceOfService char (3) NULL ,  
  
    Modifier1 char (3) NULL ,  
  
    Modifier2 char (3) NULL ,  
  
    Modifier3 char (3) NULL ,  
  
    Modifier4 char (3) NULL ,
```

```
DiagPointer1 char (1) NULL ,  
DiagPointer2 char (1) NULL ,  
DiagPointer3 char (1) NULL ,  
DiagPointer4 char (1) NULL ,  
ToothNumber char (2) NULL ,  
Surface varchar (10) NULL ,  
EmergencyIndicator char (1) NULL,  
RepricingMethodology varchar (3) NULL,  
RepricedAmount varchar (15) NULL,  
SavingsAmount varchar (15) NULL,  
RepricerID varchar (30) NULL,  
RejectReason varchar (3) NULL  
)  
;
```

5.3.5 Other Databases Scripts to Create Tables

If you have a database that is not mentioned here, you can do several things:

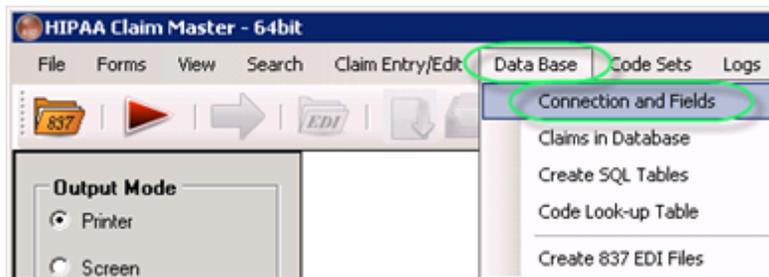
1. Have a database administrator change the scripts in order for them to function;
2. Create the fields in you database administration tool according to the specifications in the scripts.

5.4 Selecting Fields to Export

The HIPAA Claim Master can export all the fields that exist on the HCFA-1500 or the UB92 form. You might not want to export every field, but only those that you have in your database. You are really only limited by your database engine. In order to reassemble the claim from the database, it is better to have all the fields selected and available.

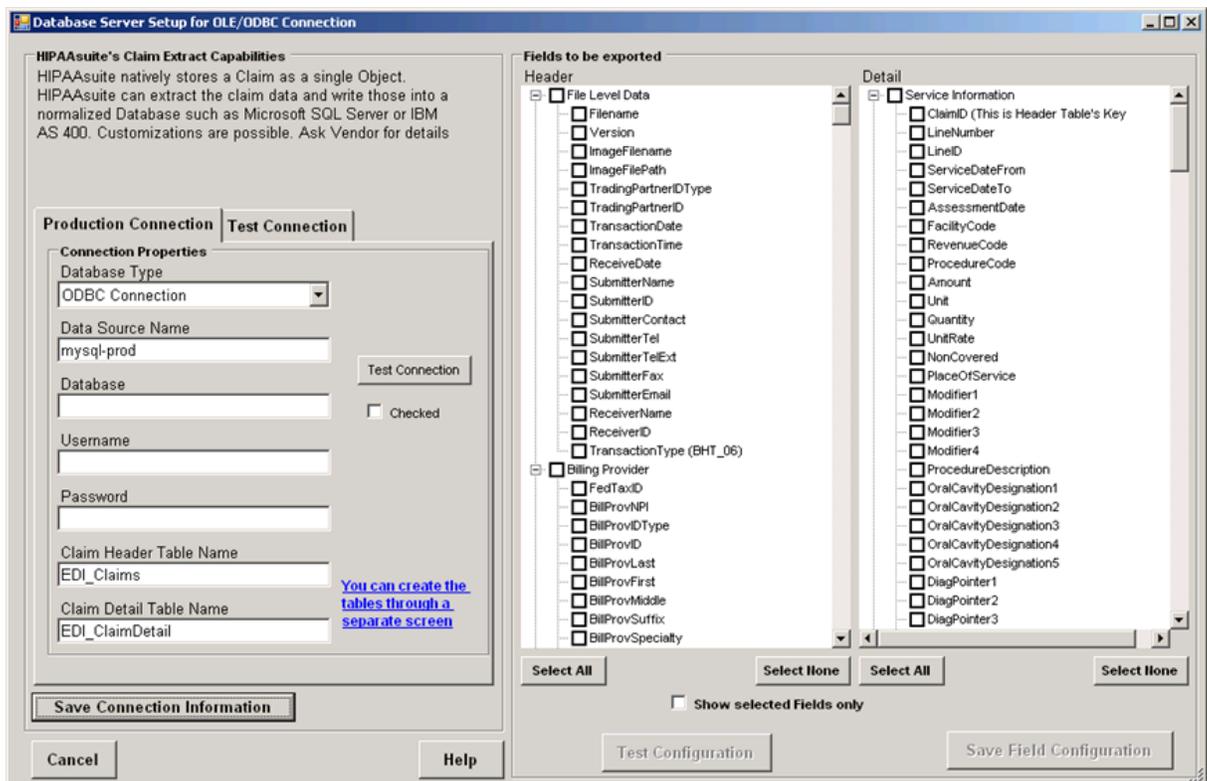
Once you have created [created your tables](#) and tested connection and table names, you can select which fields to export. Follow the instructions below.

1. Select *Data Base* ► *Connection and Fields* in the main menu.



The "Connection and Fields" menu

- You will be confronted with the following screen where you can set up the database connection and enter the names of the database tables.



The "Database Server Setup for OLE/ODBC Connection"

- The right side of the screen shows all the available database fields and columns. Choose fields to be exported by selecting checkboxes in front of the fields in the following tables:

Header

- File Level Data
- Billing Provider

- Pay-to Provider
- Pay-to Payer
- Subscriber
- Responsible
- Patient
- Rendering Provider
- Attending Provider
- Operating Provider
- Other Provider
- Facility
- Referring Provider
- Primary Care Provider
- Supervising Provider
- Assistant Surgeon
- Pick-up Location
- Drop-off Location
- Destination Payer
- COB Payer 1
- COB Payer 2
- Claim Data
- CN1 – Contact Information
- Dental Claims
- Diagnosis Information
- Procedure Information
- Value Code
- Condition Codes

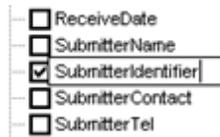
- Occurrence Codes
- CR1 – Ambulance Transport Information
- Ambulance Certification
- Repricing
- SQL to EDI

Detail

- Service Information
- Paperwork
- Repricing
- Drugs
- CR1
- AmbulanceCertification
- AttendingProvider
- OperatingProvider
- OtherProvider
- RenderingProvider
- PurchasedServiceProvider
- Facility
- SupervisingProvider
- OrderingProvider
- ReferringProvider
- COB1
- COB2

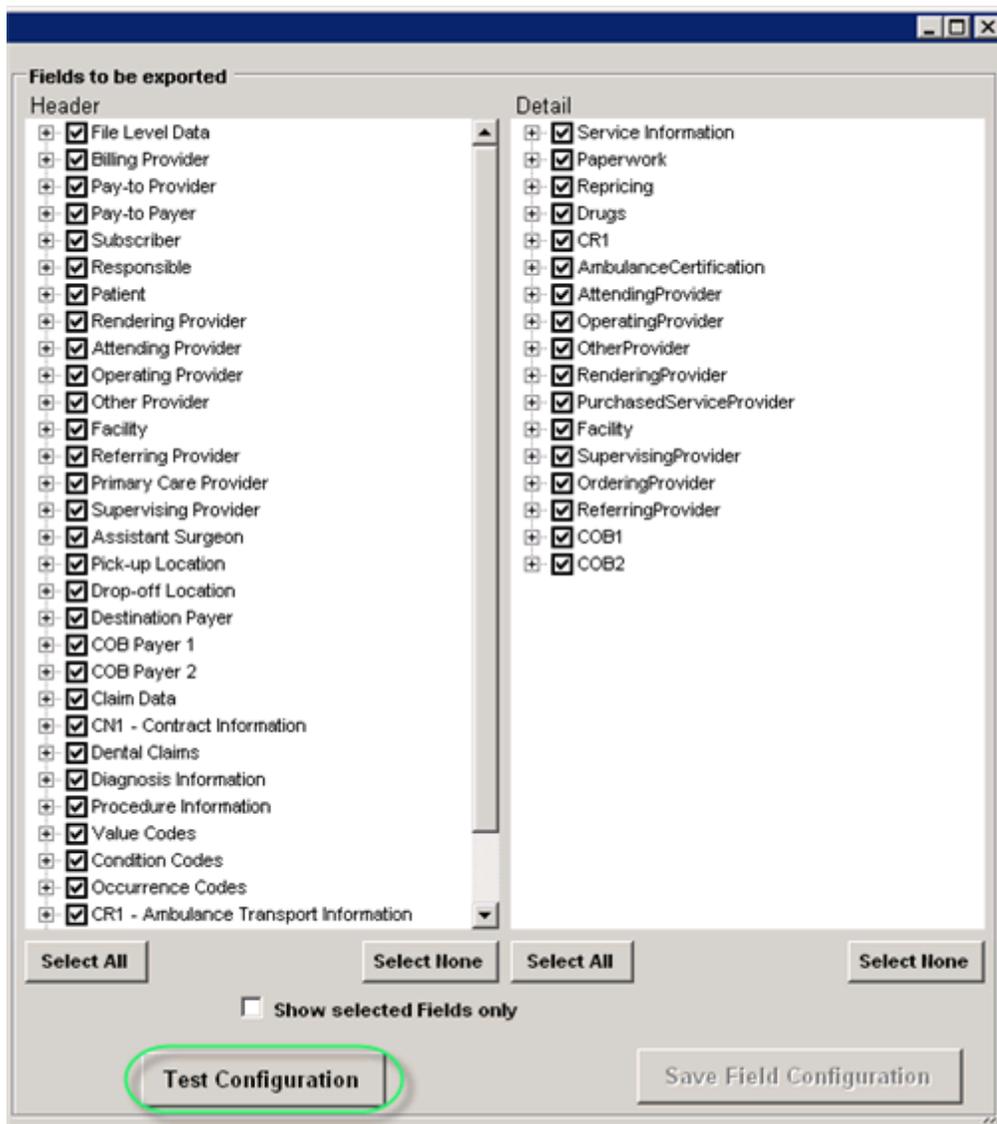
Tip: We recommend to select **all fields**. This makes it possible to reassemble all the information in the EDI file and create the image.

Tip: Sometimes it might be necessary to rename a field or fields. Some legacy database want upper case only or lower case only field names. The HIPAA Claim Master allows you to rename any field in any table. Click twice (not double-click) on a field and you are in "Edit" mode. Read more in the following "Renaming Fields" sub-topic.



The field selection list in "Edit" Mode

4. Once you have made your selection or selected all you have to test your setup, click on the "Test Configuration" button.



The "Test Configuration" button

The program will connect to the database and read the table definitions. Afterwards all the fields that you selected will be tested.

Notice: Before you can save your field configuration, you have to test the setup to make sure that all the fields that you selected exist in the tables that you specified. The table scripts that come with the HIPAA Claim Master do not have all the fields because some of the fields are very specific and not of use to many people.

You should receive a message like this. Click "OK."



The success message

If there are errors, refer to the following "Troubleshooting Data Fields" sub-section.

5. Only after the configuration have been checked out, are you allowed to save it. Click the "Save Field Configuration" button.

Important! After you have checked the setup and got the success message, you still need to save the configuration otherwise nothing will be exported.



The "Save Field Configuration" button

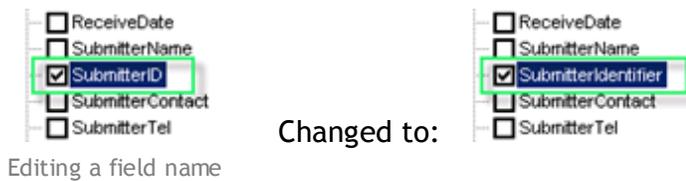
Renaming Fields

There are a few reasons why you would want to use different field names than the ones given by the HIPAA Claim Master:

- Your database cannot handle long column names such as "CA Clipper" or "IBM AS 400."
- You already have a staging table and want to import the data straight into your own table which has different field names.

HIPAA Claim Master lets you change the names of the columns that you selected for export. All you have to do is:

1. Check the box near the field for import.
2. Click the field name twice. Do not double-click, but slowly click twice. You will see that the field name is now editable.
3. Change the name to your value and click somewhere outside the field.
4. Your field name has now been changed.



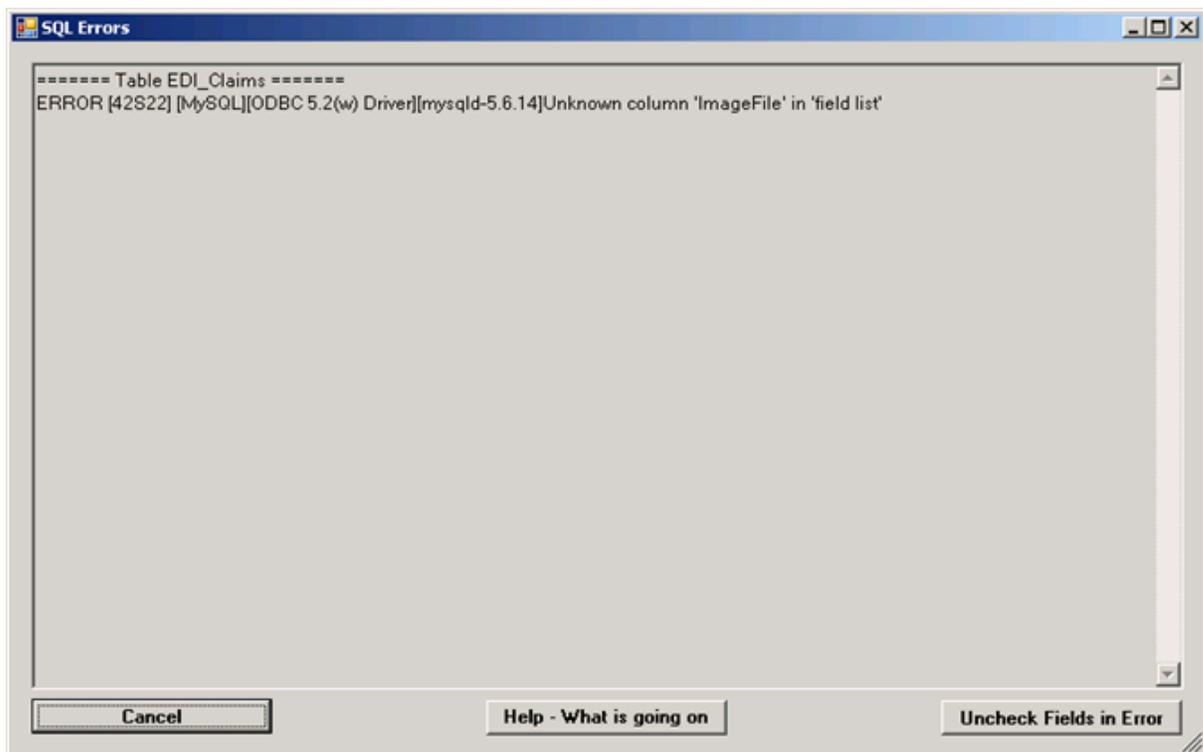
The new value will be saved as long as this field is selected for Import. If you unselect the field and save your selection, the field will revert to its original name.

Notice: Field names must not contain spaces.

Troubleshooting Data Fields

Before you can save your field configuration, you have to test the setup to make sure that all the fields that you selected exist in the tables that you specified. The table scripts that come with the HIPAA Claim Master do not have all the fields because some of the fields are very specific and not of use to many people.

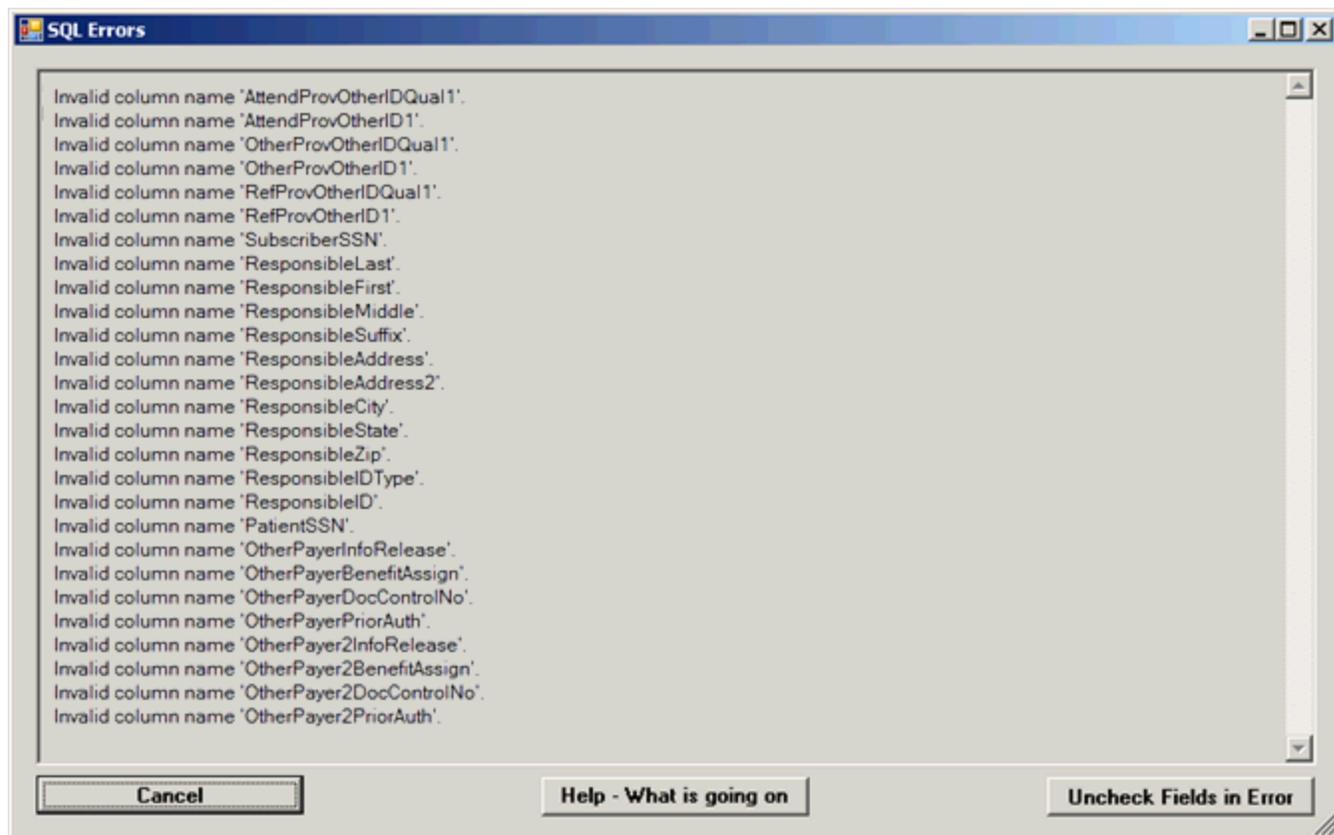
After clicking on the "Test Configuration" button, there maybe error messages. If the program finds errors, the "ODBC Error" screen will come up and tell you exactly which fields caused errors.



ODBC Setup Errors

Course

The table-created scripts that come with the HIPAA Claim Master do not contain all possible columns that the data export provides. You can add the fields manually in your database administration software. Some of the fields are very specialized and would not be suitable for general distribution and make the tables unnecessarily complicated. If the fields do not exist, an error message will come up and you have the option to deselect them.



ODBC Setup Errors

In the picture above, you can see the columns in error obviously do not exist in the table that was specified.

Solution

The possible solutions are:

1. To unselect the fields that course errors in the HIPAA Claim Master.
- Or:**
2. To create missing fields in the database and then try again.

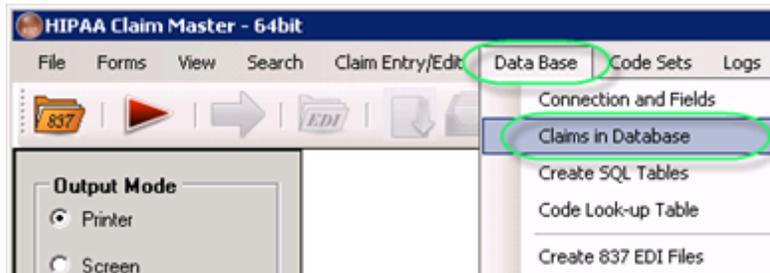
Important! After you checked the setup and got the "Setup checks Okay" message, you still need to

save the configuration otherwise nothing will be exported.

5.5 Browsing Exported Data

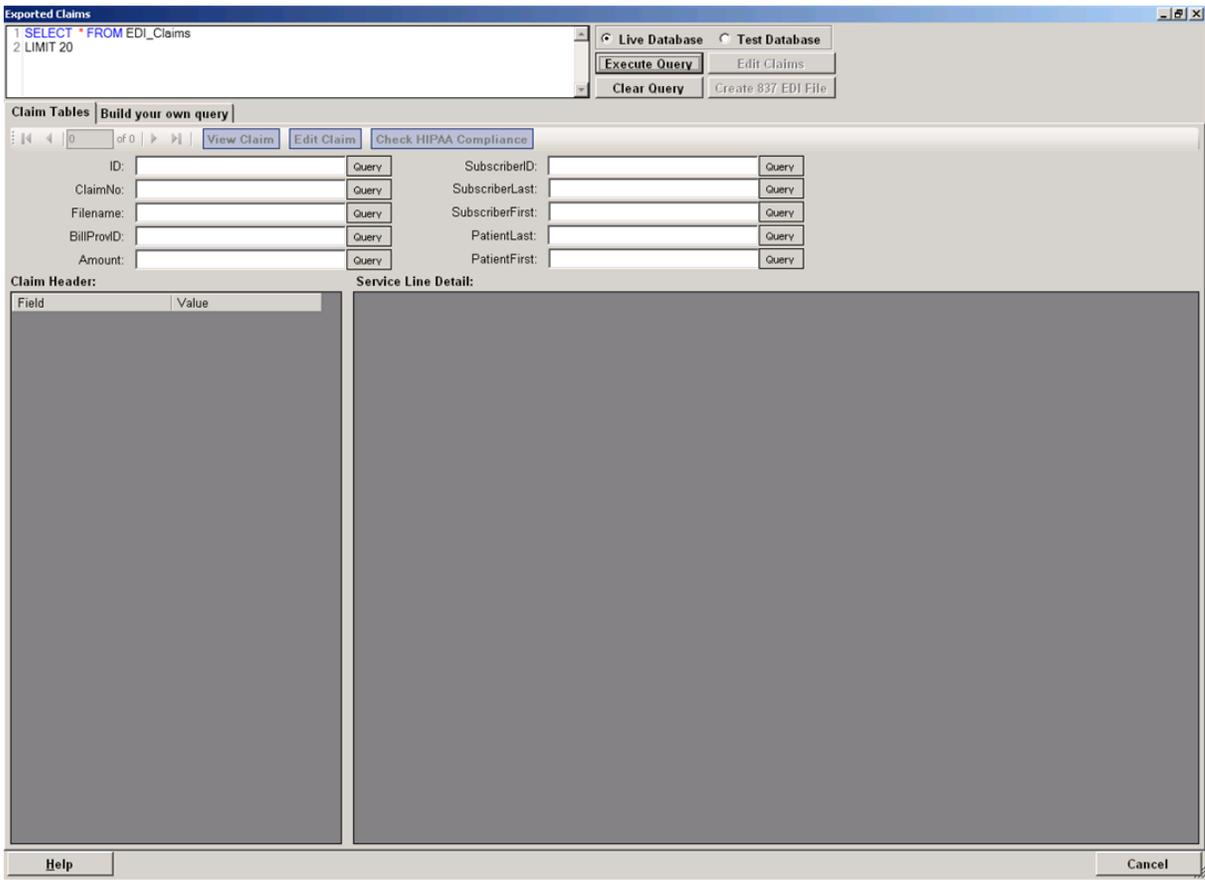
The HIPAA Claim Master can access the exported data and show it to the user. This might be helpful in tracking particular claims.

Select the *Data Base* ▶ *Claims in Database* in the main menu.



The "Claims in Database" menu

You will see the following screen where you can see all the claims and review them. In addition, you can query the database for any of the displayed fields. For example, if you want to check if all the claims from a specific file were imported, you could type the filename into the appropriate field and click the "Query" button. The results will be returned.



Click on a payment on the left to load the claims associated with this payment on the right side.

Exported Claims

```
1 SELECT * FROM EDI_Claims
2 LIMIT 20
```

Live Database Test Database

Execute Query Edit Claims

Clear Query Create 837 EDI File

Claim Tables Build your own query

1 of 20 View Claim Edit Claim Check HIPAA Compliance

ID: 1 Query SubscriberID: 021995432A Query

ClaimNo: 987654 Query SubscriberLast: SMITH Query

Filename: 837IPTST.edi Query SubscriberFirst: TESS Query

BillProvID: 097654321 Query PatientLast: SMITH Query

Amount: 4675.34 Query PatientFirst: TESS Query

Claim Header:

Field	Value
ID	1
Filename	837IPTST.edi
ImageFilePath	
ImageFilename	
Version	004010X096
TradingPartnerIDType	29
TradingPartnerID	987654
TransactionDate	3/29/2001
TransactionTime	13:10:00
ReceiveDate	10/24/2013
SubmitterName	HEALING HOSPITAL O...
SubmitterID	XZ54278
SubmitterContact	PHIL FIELD
SubmitterTel	7365554321
SubmitterTelExt	
SubmitterFax	
SubmitterEmail	
ReceiverName	XYZ INTERMEDIARY S...
ReceiverID	789654
TransactionType	CH
FedTaxID	097654321
BillProvDType	24
BillProvID	097654321
BillProvNPI	
BillProvLast	HEALING HOSPITAL O...
BillProvFirst	

Service Line Detail:

ID	ClaimID	LineNumber	LineID	ServiceDateFrom	ServiceDateTo	AssessmentDate	FacilityCode	RevenueCod
1	1	1						120
2	1	2						320
3	1	3						360
4	1	4						370
5	1	5						450
6	1	6						710

Help Cancel

Accessing the claims in the database

Editing Data

You can edit the data in the tables by typing in a cell. A dialog will ask you whether you want to save the changes.

pe	ZZ
	10/22/2012

Save Cancel

Editing the field value

Viewing Claims

Click on the "View Claim" button to render the claim.

Exported Claims

```
1 SELECT * FROM EDI_Claims
2 LIMIT 20
```

Live Database Test Database

Execute Query Edit Claims

Clear Query Create 837 EDI File

Claim Tables Build your own query

1 of 20 View Claim Edit Claim Check HIPAA Compliance

ID: 1 Query SubscriberID: 021995432A Query

ClaimNo: 987654 Query SubscriberLast: SMITH Query

Filename: 837IPTST.edi Query SubscriberFirst: TESS Query

BillProvID: 097654321 Query PatientLast: SMITH Query

Amount: 4675.34 Query PatientFirst: TESS Query

Claim Header:

Field	Value
ID	1
Filename	837IPTST.edi
ImageFilePath	
ImageFilename	
Version	004010X096
TradingPartnerIDType	29
TradingPartnerID	987654
TransactionDate	3/29/2001
TransactionTime	13:10:00
ReceiveDate	10/24/2013
SubmitterName	HEALING HOSPITAL O...
SubmitterID	XZ54278
SubmitterContact	PHIL FIELD
SubmitterTel	7365554321
SubmitterTelExt	
SubmitterFax	
SubmitterEmail	
ReceiverName	XYZ INTERMEDIARY S...
ReceiverID	789654
TransactionType	CH
FedTaxID	097654321
BillProvDType	24
BillProvID	097654321
BillProvNPI	
BillProvLast	HEALING HOSPITAL O...
BillProvFirst	

Service Line Detail:

ID	ClaimID	LineNumber	LineID	ServiceDateFrom	ServiceDateTo	AssessmentDate	FacilityCode	RevenueCod
1	1	1						120
2	1	2						320
3	1	3						360
4	1	4						370
5	1	5						450
6	1	6						710

Help Cancel

The following preview window will open.

Note: To preview the claims, you must have a printer installed on your PC.

HIPAA Preview

Print Close Next Claim

← 1 / 1 → Claim Add info COB info

Sender's ID Number: 987654

HEALING HOSPITAL OF ANYTOWN
1 HOSPITAL ROW
ANYTOWN, MD 21299

2 PAT 987654
3 MED REC N 69875-9
4 111

5 FEE TAX NO. 097654321 6 STATEMENT COVERED PERIOD FROM 02/21/01 THROUGH 02/26/01 7 Claim Days: 5

8 PATIENT NAME 021995432A 9 PATIENT ADDRESS 1 HOME STREET
10 SMITH, TESS T 11 HOME CITY MD 13746

12 BIRTHDATE 02/02/1930 13 SEX F 14 ADMISSION DATE 02/21/01 15 TIME 08:00 16 ICD9 2 17 ICD10 7 18 ICD9 PROC 14:00 19 ICD9 CODE 01

20 OCCURRENCE DATE 05 21 OCCURRENCE DATE 02/21/2001

22 SMITH, TESS T
23 1 HOME STREET
24 HOME CITY, MD 13746

25 01 26 VALUE CODED AMOUNT \$250.00

42 REV CD	43 DRG/PTG	44 ICD9 RATE	45 SERV. DATE	46 SERV. UNIT	47 TOTAL CHARGES	48 NO. COVS CHARGES	49
1	120	\$250.00		5	\$1,250.00		
2	320			5	\$1,006.00		
3	360			1	\$722.00		
4	370			1	\$310.00		
5	450			1	\$987.34		
6	710			1	\$400.00		

A claim image created from a database record

5.6 Displaying the Revenue or Procedure Code Descriptions

EDI files do not carry the description of codes since EDI is all about saving space. But since the codes are available commercially or from CMS, you can link tables with those codes in your database application and have the HIPAA Claim Master look up the description. These tables need to be in the database defined while [Setting up Database Connection](#).

Descriptions in the standard Forms

The UB92 and UB04 as well as the dental forms have the "Description" field for the line item revenue or procedure codes. For the CMS-1500 forms one can put the description in the upper half of the field locator 24 D.

0300	LAB - GENERAL CLASS	83690.TC	04/07/09	1	\$108.55		
0300	LAB - GENERAL CLASS	85025.TC	04/07/09	1	\$100.45		
0402	OTHR IMAG SERV-ULTRASOUND	76705.TC	04/07/09	1	\$946.30		
0450	ER - GENERAL CLASS	Z7502	04/06/09	1	\$930.00		

A part of a UB04 with revenue code descriptions

RECORD OF SERVICES PROVIDED							
	24. Procedure Date (MM/DD/YYYY)	25. Area of Oral Cavity (System)	26. Tooth Number(s) or Letter(s)	27. Tooth Surface	28. Procedure Code	29. Description	30. Fee
1	04/19/2004				D0120	PERIODIC ORAL EVALUATION	29.00
2	04/19/2004				D0274	BITEWINGS-FOUR FILMS	41.00
3	04/19/2004				D1110	PROPHYLAXIS-ADULT	56.00

Dental Procedure Code Descriptions

	24. A. DATE(S) OF SERVICE		B. PLACE OF SERVICE	C. EMG	D. PROCEDURES, SERVICES OR SUPPLIES (Specify Universal Circumstances)		E. DIAGNOSIS POINTER	F. \$CHARGES	G. DAYS OR UNITS	H. EPSDT Family Plan	I. ID QUAL	J. RENDERING PROVIDER ID.#
	From	To			CPT/HCPCS	MODIFIER						
1	06/07/1997	06/08/1997	23	Y	99285		AB	\$800.00	1		WPI	1417997107

A part of the CMS-1500 with the description filled in from the database

The CMS-1500 option is separate and can be selected in the Specific Options (the "CMS-1500: Display procedure description in 24D" check-box). Refer to [Configuring Specific Options](#).

Linking the Procedure Code Table

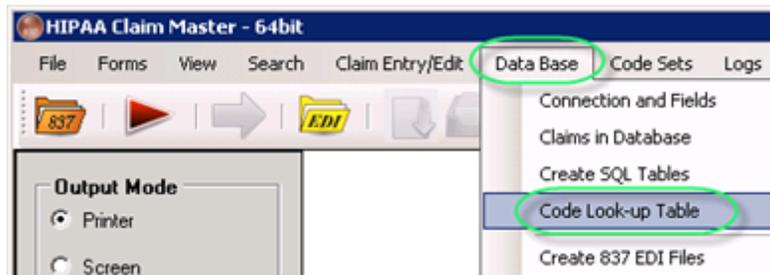
Acceptable Codes:

- HCPCS codes
- CPT procedure codes.
- All dental procedure codes.

Tip: Using the code look-up might slow down your application depending on the speed of your database. You can undo the code lookup by deleting the table name and clicking the "Save" button.

Follow the instructions below to show code descriptions.

1. Select the *Data Base* ▶ *Code Look-up Table* in the main menu.



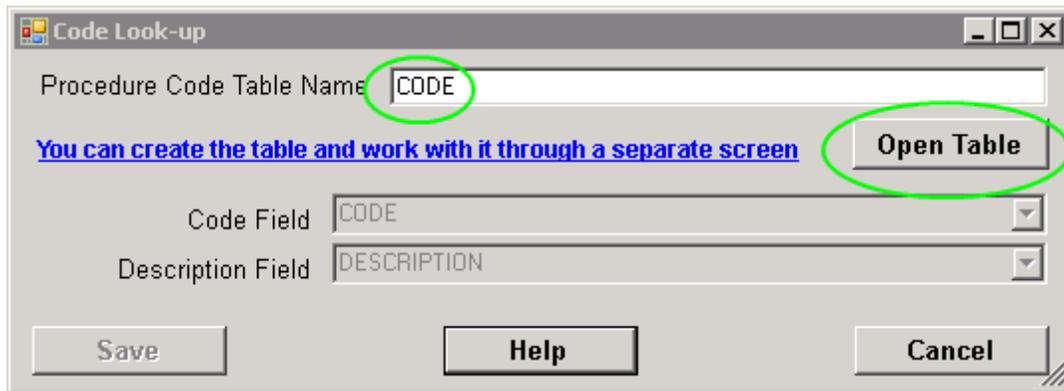
The "Code Look-up Table" menu

2. You will see the following screen:

The "Code Look-up" screen

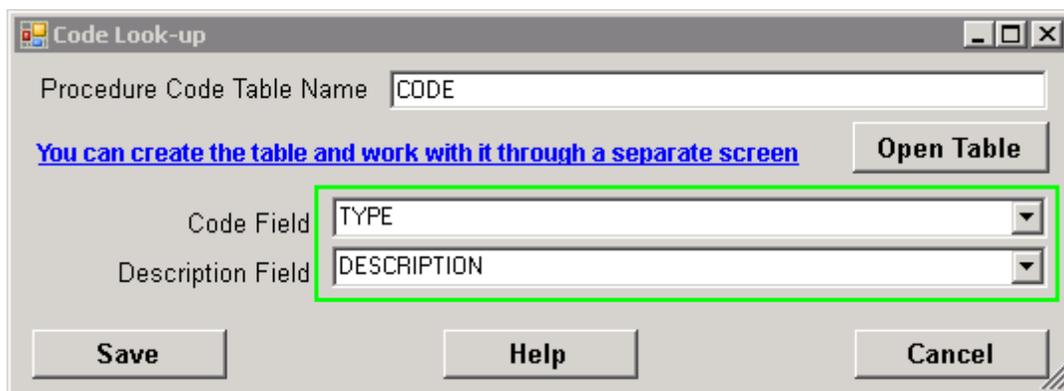
If you don't have a table already with the procedure codes, you can create it in the claim master application. You see the sentence in blue "You can create the table...". Click on this and the following form will come up.

3. Type the name of the table and click the "Open Table" button.



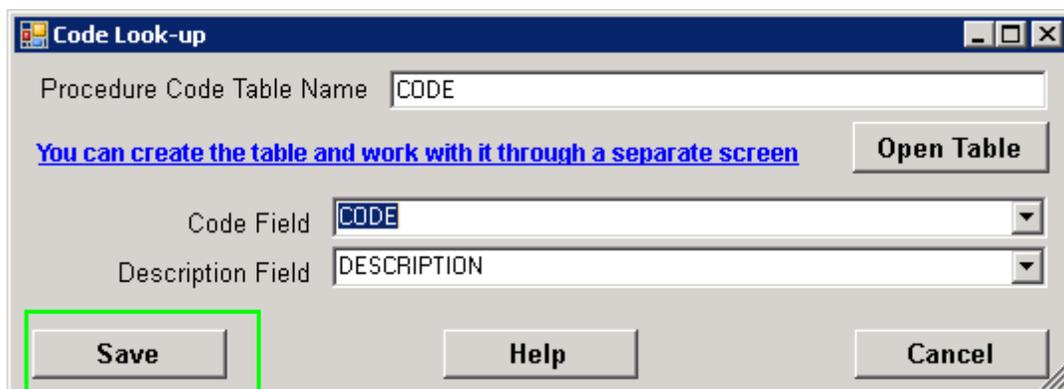
The "Open Table" button

4. If the table is found, then the drop-down boxes for "Code Field" and "Description Field" will be populated with all the available fields in your table. You can then choose the field that will carry the code and description.



The "Code Field" and "Description Field" values populated

5. Click "Save."



The "Save" button

From now on, the HIPAA Claim Master will fill in the code description fields.

5.7 Creating Code Look-up Tables

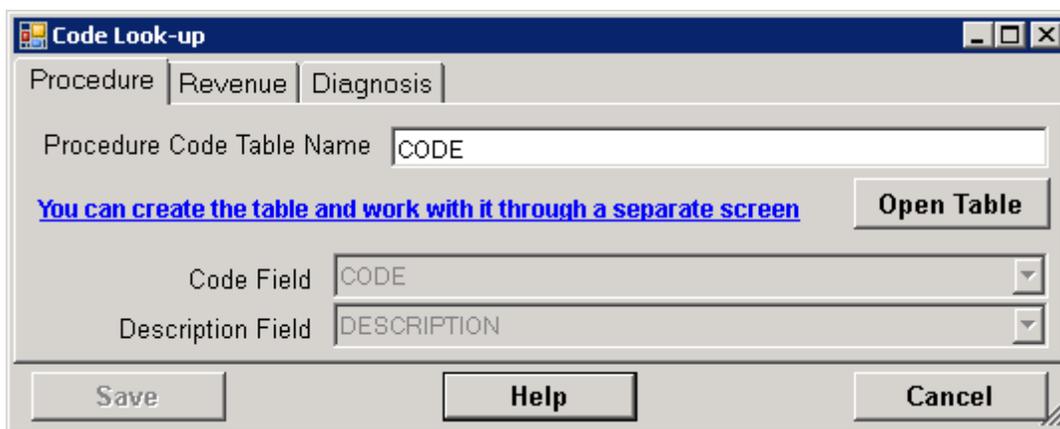
EDI is all about saving space and using code sets to represent more lengthy circumstances. But often it would be more convenient to have the descriptions of a revenue or procedure code visible instead of looking it up or knowing it by heart.

Important codes are

- Procedure Codes (HCPCS, CPT, Dental and ICD procedure codes)
- Revenue Codes
- Diagnosis Codes (ICD-9 and ICD-10)

There are tens of thousands of codes and they are mostly available at the CMS web site or other web searches. The HIPAA Claim Master provides a database schema to create cross reference tables for those codes.

When we go to the menu item Database --> Code Look-up tables we see this screen

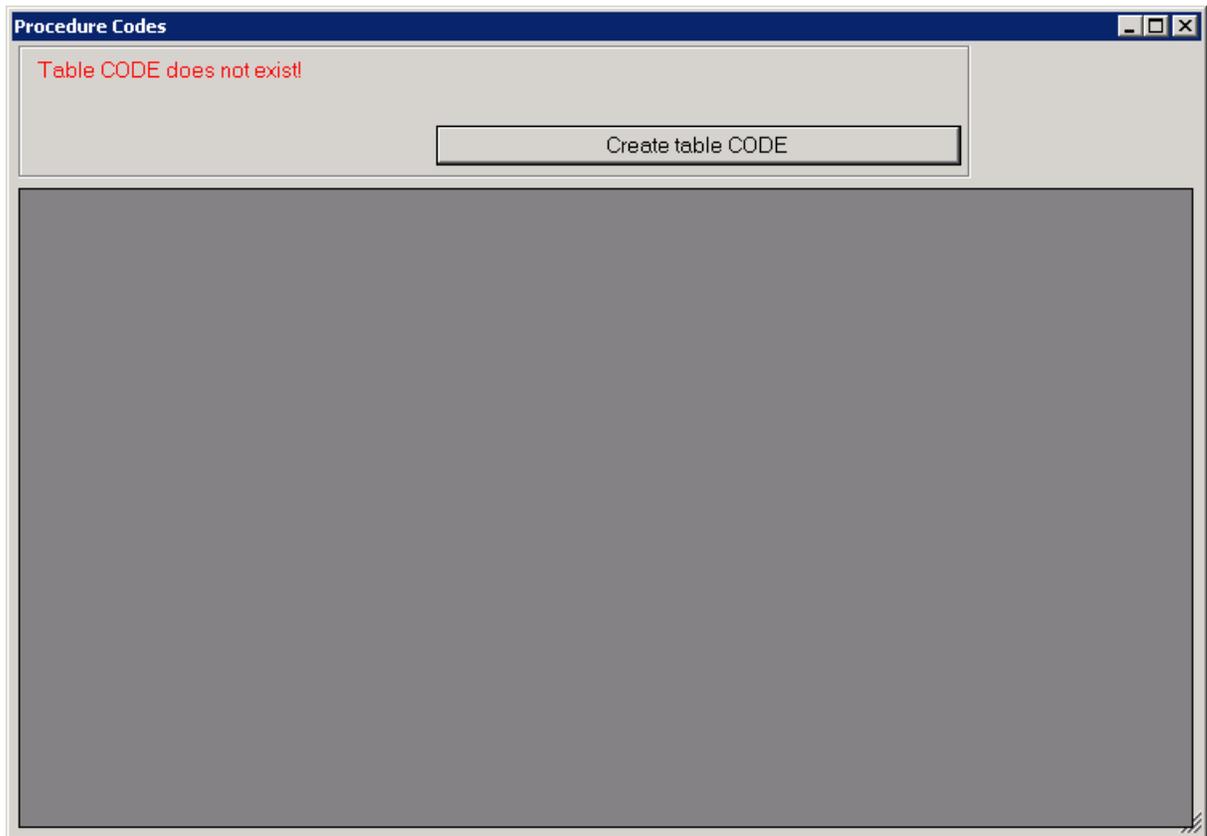


The Code Look-up form with tabs for Procedure, Revenue and Diagnosis codes

If you have already in your database a table that contains Procedure codes and their long and short descriptions, you can link to it in this form. Fill in your table name and open it and select the fields that contain the information.

If you don't have already a table with the procedure codes and want to create a new table, you can go to Database --> Code Look-up table and then click on the the blue sentence "You can create..."

The following form will come up



The form to create and fill the look up table

Since the table does not exist we press the button : "Create table CODE" and we will see this form

Database: Microsoft SQL Server MySQL/Generic Oracle

CODE Table script:

```
1 CREATE TABLE [dbo].[CODE](
2     [CODE] [varchar](15) NOT NULL,
3     [DESCRIPTION] [varchar](254) NULL,
4     [SHORT_DESC] [varchar](50) NULL,
5     [TYPE] [varchar](10) NULL
6     CONSTRAINT [PK_CODE] PRIMARY KEY CLUSTERED
7     (
8         [CODE] ASC
9     ) ON [PRIMARY]
10 ) ON [PRIMARY]
11
12
13
```

Compare Script with existing Table Save Execute Script

Help Close

The form with the script to create the look-up table

We click on the "Execute Script" button and create the table and close the form. Now the previous form has changed.

The form to manage procedure codes

The table is created but is empty, so now we can load the codes that we obtained separately. Maybe your company has a subscription to the code data as Ingenix, for example, sells, or you find them on the internet. We click on "Load from XLSX file" and navigate to the file. The spreadsheet has been created earlier and has the following layout:

First column the code, second the long description and third the short description.

Loading a previously prepared spreadsheet with the HCPCS codes.

We navigate to the file and select it. Then we select the correct code type: HCPCS in our

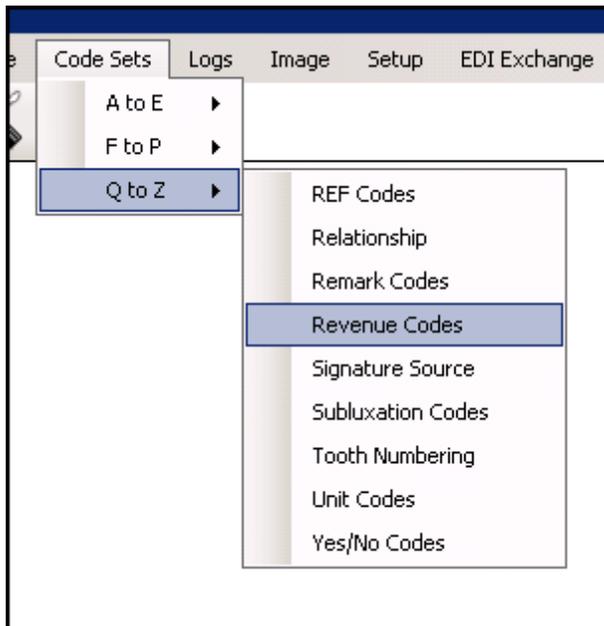
case and press on the button "Load Codes" Within a few seconds the file's codes are loaded into the database and now we can see the values.

CODE	DESCRIPTION	SHORT_DE
A0021	Ambulance service, outside state per mile, transport (medicaid only)	Outside state
A0080	Non-emergency transportation, per mile - vehicle provided by volunteer (individual or organization), with no vested...	Noninterest €
A0090	Non-emergency transportation, per mile - vehicle provided by individual (family member, self, neighbor) with vested...	Interest esc
A0100	Non-emergency transportation; taxi	Nonemerger
A0110	Non-emergency transportation and bus, intra or inter state carrier	Nonemerger
A0120	Non-emergency transportation: mini-bus, mountain area transports, or other transportation systems	Noner transp
A0130	Non-emergency transportation: wheel-chair van	Noner transp
A0140	Non-emergency transportation and air travel (private or commercial) intra or inter state	Nonemerger
A0160	Non-emergency transportation: per mile - case worker or social worker	Noner transp
A0170	Transportation ancillary: parking fees, tolls, other	Transport pa
A0180	Non-emergency transportation: ancillary: lodging-recipient	Noner transp
A0190	Non-emergency transportation: ancillary: meals-recipient	Noner transp
A0200	Non-emergency transportation: ancillary: lodging-escort	Noner transp
A0210	Non-emergency transportation: ancillary: meals-escort	Noner transp
A0225	Ambulance service, neonatal transport, base rate, emergency transport, one way	Neonatal em
A0380	Bls mileage (per mile)	Basic life sup
A0392	Bls routine disposable supplies	Basic supp

The screen after the codes have been loaded

Procedure codes are now loaded and in the database. We can add other codes such as CPT and Dental codes to it. If you have duplicate codes in other files that you load, you will see a message that the code is already in the table and will not be loaded again.

Revenue Codes and Diagnosis Codes work very similar. Go to the menu **Code Sets --> Revenue Codes**



The menu Revenue Codes

Now the same form that you used for the procedure codes comes up only that it now points to a table "RevenueCodes"

The third table that gets created and filled the same way is the diagnosis code table. You can reach it from the Code Sets menu under "Diagnosis Codes"

5.8 Troubleshooting Data Export

When you start exporting claims into the staging tables, you may encounter errors in the beginning. This usually results from table definition issues. The HIPAA Claim Master displays a very detailed error message that tells you what claim has failed, at what field and, if applicable, what line. This information should help you to troubleshoot the problems. Sometimes you must fulfill serious detective work to find and correct issues.

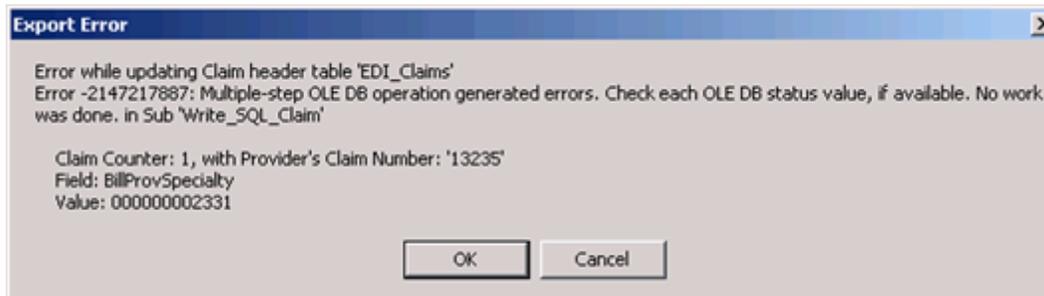
Below are some of the most common errors.

Item Cannot Be Found



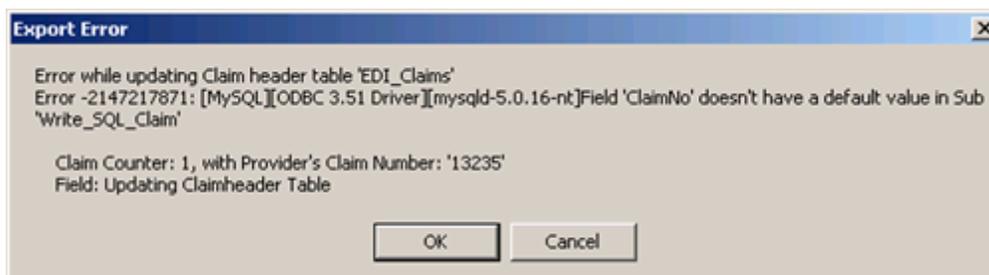
This error happens when a column that was selected for export does not exist in the Header table. In this particular case, it is the `claimNo` field. If you receive this error, go back to your table admin tool and check if the `claimNo` field is correct.

Multi-Step OleDB Operation Generated Errors



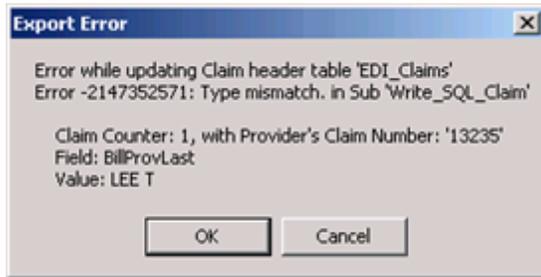
This is typically the message when the field definition in the database is different from the data. Here we see that the error occurred with the billing provider's specialty. The field definition in the table was set as `varchar(10)` and the value here "000000002331" has 12 digits causing the error.

Field Does Not Have a Default Value



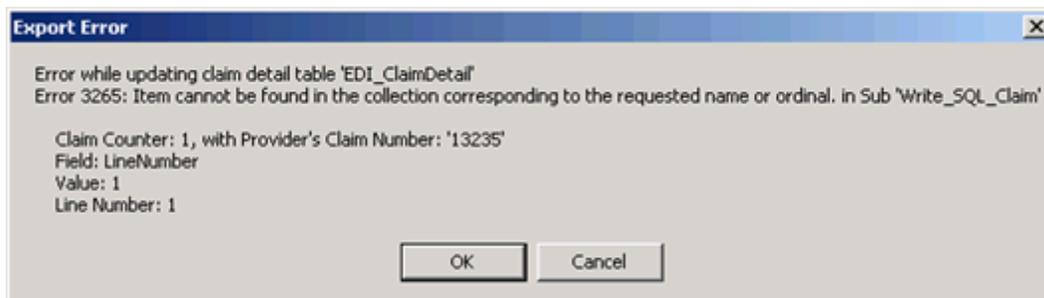
The field `claimNo` is defined as "Not Null." But somehow this data was not present and the HIPAA Claim Master tries to update the table with a null `claimNo`.

Type Mismatch



This error occurs when the HIPAA Claim Master tries to "write a letter" to a field that is defined as "Integer," for example. The data type of the column does not match the data type of the value.

Claim Detail Errors



The errors with the claim detail table are similar; the error message indicates the line number where the error occurred.

Chapter

VI

6 Displaying and Printing Additional Information

The X12 837 EDI file format allows for much more information than fits on a paper form. The creators of the 837 standard included many health care business scenarios besides only a professional or institutional environment. The 837 has segments for ambulance billing, Chiropractic care, home oxygen therapy, home health care, repricing information and a variety more. In addition the 837 has a much finer resolution to express complicated COB situations with several payers. The information in those segments is usually not displayed on the standard paper forms. For this reason the HIPAA Claim Master comes with additional pages that aim to display every possible piece of information that an 837 can contain.

The HIPAA Claim Master checks each claim for the existence of those special segments. You will see additional buttons in the claim view window to alert you of their existence. By clicking on "Addl" or "COB" the screens to display this information will show. Page through buttons will also show, if this information does not fit on one page.

6.1 Understanding Types of Additional Information

The 837 P and 837 I can contain different pieces of additional information which is expressed through the following segments. The HIPAAsuite parsing engine catches and stores each bit of information.

837 P

The 837 P can contain...

On the claim level:

- Loop 2010AB – Pay-To Provider's Information (Pay-To Address in 5010)
- Loop Loop 2010AC – Pay-To Plan
- Loop 2010BC – Responsible Party
- Loop 2010BD – Credit Card Holder
- Loop 2000B – Additional Subscriber Information
- Loop 2000C – Additional Patient Information
- Loop 2310A – Referring Provider's Information

- Loop 2310B – Rendering Provider's Information
- Loop 2310C – Purchased Service Provider
- Loop 2310D – Service Facility (2310C in 5010)
- Loop 2310E – Supervising Physician (2310D in 5010)
- Loop 2310E – Ambulance Pick-up Location (5010)
- Loop 2310F – Ambulance Drop-off Location (5010)
- Loop 2420 – Line Level Providers
- Loop 2300 – Additional Claim Information (DTP,REF,AMT etc.)
- Segment K3 File Information
- Segment PWK – Paperwork
- Segment NTE – Notes
- Segment HI*BF – Additional Diagnosis Codes
- Segment HI*BP/BO – Anesthesia Related Procedure Codes
- Segment HI*BG – Condition Codes
- Segment CN1 – Contract Information
- Segment CR1 – Ambulance Transport Information
- Segment CR2 – Spinal Manipulation
- Segment CR5 – Home Oxygen Therapie Info
- Segment CRC – Certificates, Conditions etc.
- Segment HCP – Repricing Information
- Segment CR7 – Home Healthcare Plan Information
- Segment HSD – Health Service Delivery

On the line level:

- Segment K3 File Information LineLevel
- Segment CRC – Certificates, Conditions Linelevel
- Segment QTY – Ambulance Patient Count, Anesthesia Add.Units Linelevel

- Loop 2420 – Line Level Providers
- Segment PWK – Paperwork Linelevel
- Segment HCP – Repricing Information Linelevel
- Segment HSD – Health Service Delivery Linelevel
- Segment PS1 Purchased Services Information Linelevel
- Segment MEA – Test Results Linelevel
- Loop 2440 Form Identification Code Linelevel
- Segment NTE – Notes LineLevel
- Segment CR2 – Spinal Manipulation Linelevel/[CHF 11-21-2011 *139]
- Segment CR1 – Ambulance Info Linelevel
- Segment CN1 – Contract Info Line
- Segment CR3 – DMERC – Line only
- Loop 2410 Drug Information Linelevel
- Loop 2400 – Additional Claim Information Linelevel(DTP,REF,AMT etc.)

837 I

The 837 I can contain...

On the claim level:

- EDI_Envelope: ISA – GS – ST – BHT
- Loop 2000A – Billing Provider's Information
- Loop 2010AB – Pay-To Provider's Information (Pay-To Address in 5010)
- Loop 2010AC – Pay-To Plan
- Loop 2010BD – Credit Card Holder
- Loop 2000B – Additional Subscriber Information
- Loop 2000C – Additional Patient Information
- Loop 2310A – Attending Physician's Information

- Loop 2310B – Operating Physician's Information
- Loop 2310C – Other Physician's Information
- Loop 2310D – Rendering Provider's Information (5010)
- Loop 2310E – Service Facility
- Loop 2310F – Referring Provider's Information (5010)
- Loop 2300 – Additional Claim Information (DTP, REF, AMT etc.)
- Segment K3 2300 – K3 File Information
- Segment PWK – PWK – Paperwork
- Segment NTE 2300 – NTE Notes
- Segment NTE 2400 – NTE – Notes Linelevel
- Segment HI*BF – HI*BF – Diagnosis Codes
- Segment HI*BN – HI*BN – E – Codes
- Segment HI*DR – HI*DR – Diagnosis Related Codes
- Segment HI*BI – HI*BI – Occurrence Span Codes
- Segment HI*BH – HI*BH – Occurrence Codes
- Segment HI*BE – HI*BE – Value Codes
- Segment HI*BG – HI*BG – Condition Codes
- Segment HI*TC – HI*TC – Treatment Codes
- Segment HI*BQ/BO – HI*BQ/BO – Procedure Codes
- Segment QTY – QTY – Claim Quantity
- Segment CN1 – CN1 – Contract Info
- Segment CRC – CRC – Certificates, Conditions
- Segment HCP – HCP – Repricing Information
- Segment CR7 – CR7 – Home Healthcare Plan Information
- Segment HSD – HSD – Healthcare Service Delivery
- Segment CR6 – CR6 – Home HealthCare Information

On the line level:

- Loop 2400 – Additional Claim Information Linelevel(DTP,REF,AMT etc.)
- Loop 2410 – Drug Information
- Loop 2420 – Line Level Providers
- Segment HCP – Repricing Information Linelevel
- Segment HSD – Healthcare Service Delivery Linelevel
- Segment CN1 – Contract Info Linelevel
- Segment CRC – Certificates, Conditions Linelevel
- Segment PWK 2400 – PWK – Linelevel

837 D

The 837 D can contain...

On the claim level:

- Loop 2000A Additional Billing Provider's Information
- Loop 2010AB Additional Pay-To Provider's Information
- Loop 2010BC – Credit Card Holder
- Loop 2000B – Additional Subscriber Information
- Loop 2000C – Additional Patient Information
- Loop 2310A – Referring Provider's Information
- Loop 2310B – Rendering Provider's Information
- Loop 2310C – Service Facility's Information
- Loop 300 – Additional Claim Information (DTP, REF, AMT etc.)
- PWK Claim Supplemental Information
- NTE Claim Note
- HCP Claim Pricing/Repricing Information

On the line level:

- Loop 2400 – Additional Claim Information Line level(DTP, REF, AMT etc.)
- NTE Claim Note
- DN2 – Tooth Status
- HCP Line Pricing/Repricing Information
- Loop 2420 – Line Level Providers

837 R

The 837 R can contain...

On the claim level:

- Loop Service Provider's Information
- Loop 2000B – Additional Subscriber Information
- Loop 2000C – Additional Patient Information
- Loop 2310A – Attending Provider's Information
- Loop 2310B – Operating Provider's Information
- Loop 2310C – Other Physician's Information
- Loop 2310D – Rendering Provider's Information
- Loop 2310F – Referring Provider's Information
- Loop 2300 – Additional Claim Information (DTP, REF, AMT etc.)
- K3 – File Information
- PWK Claim Supplemental Information
- NTE Claim Note
- HI*BF – Diagnosis Codes
- HI*BN – E Codes
- HI*DR – Diagnosis Related Codes
- HI*BQ/BO – Procedure Codes
- HI*BI – Occurrence Span Codes
- HI*BH – Occurrence Codes

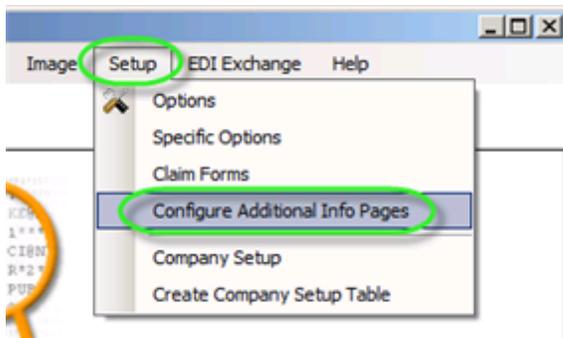
- HI*BE – Value Codes
- HI*BG – Condition Codes

On the line level:

- Loop 2400 – Additional Claim Information Linelevel(DTP, REF, AMT etc.)
- Segment NTE Claim Note
- Loop 2410 – Drug Information
- Loop 2420 – Line Level Providers

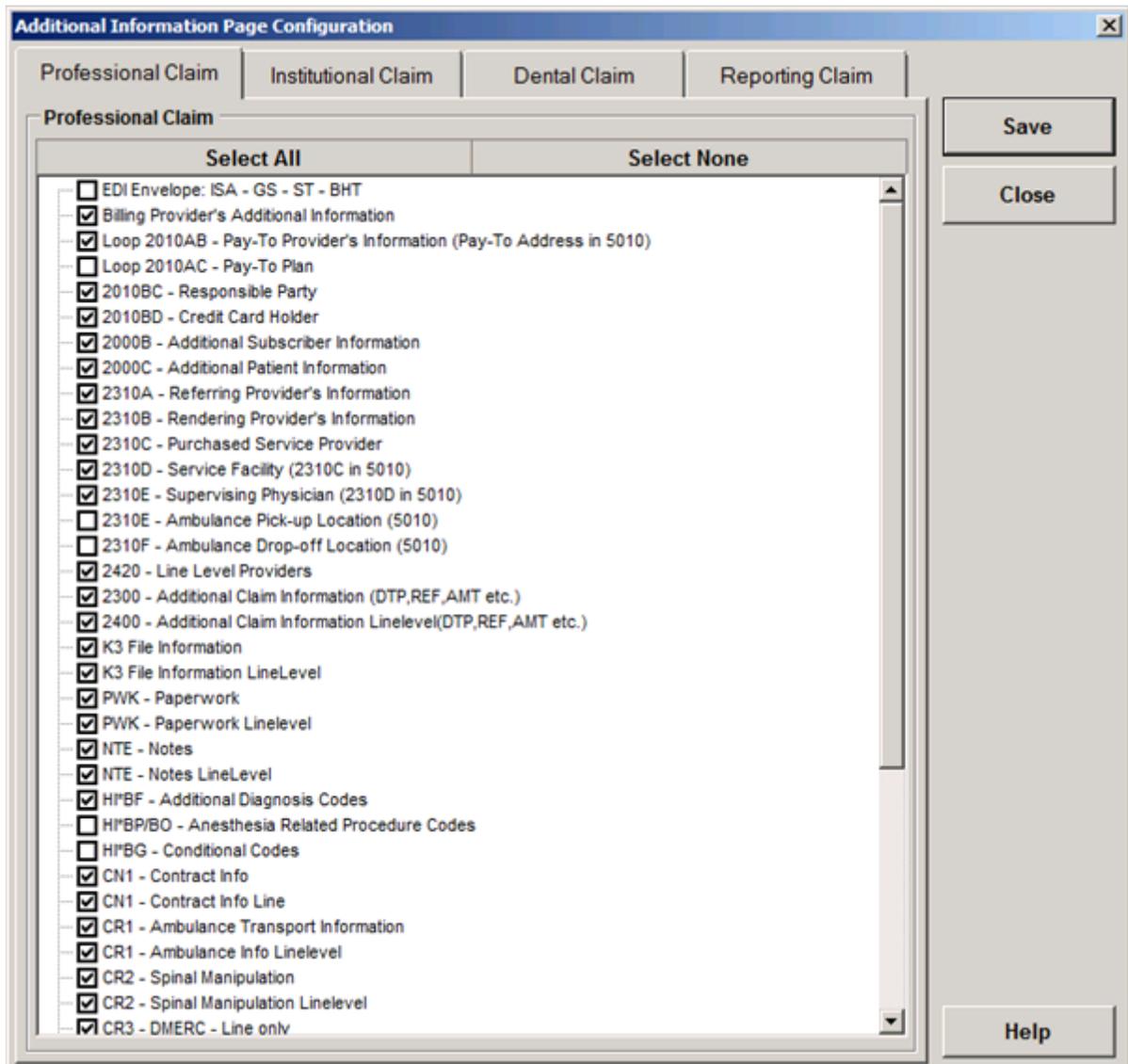
6.2 Configuring Additional Info Pages

To configure the additional information pages options, select *Setup* ▶ *Additional Info Pages* in the main menu.



The "Configuring Additional Info Pages" menu

The following screen will appear.



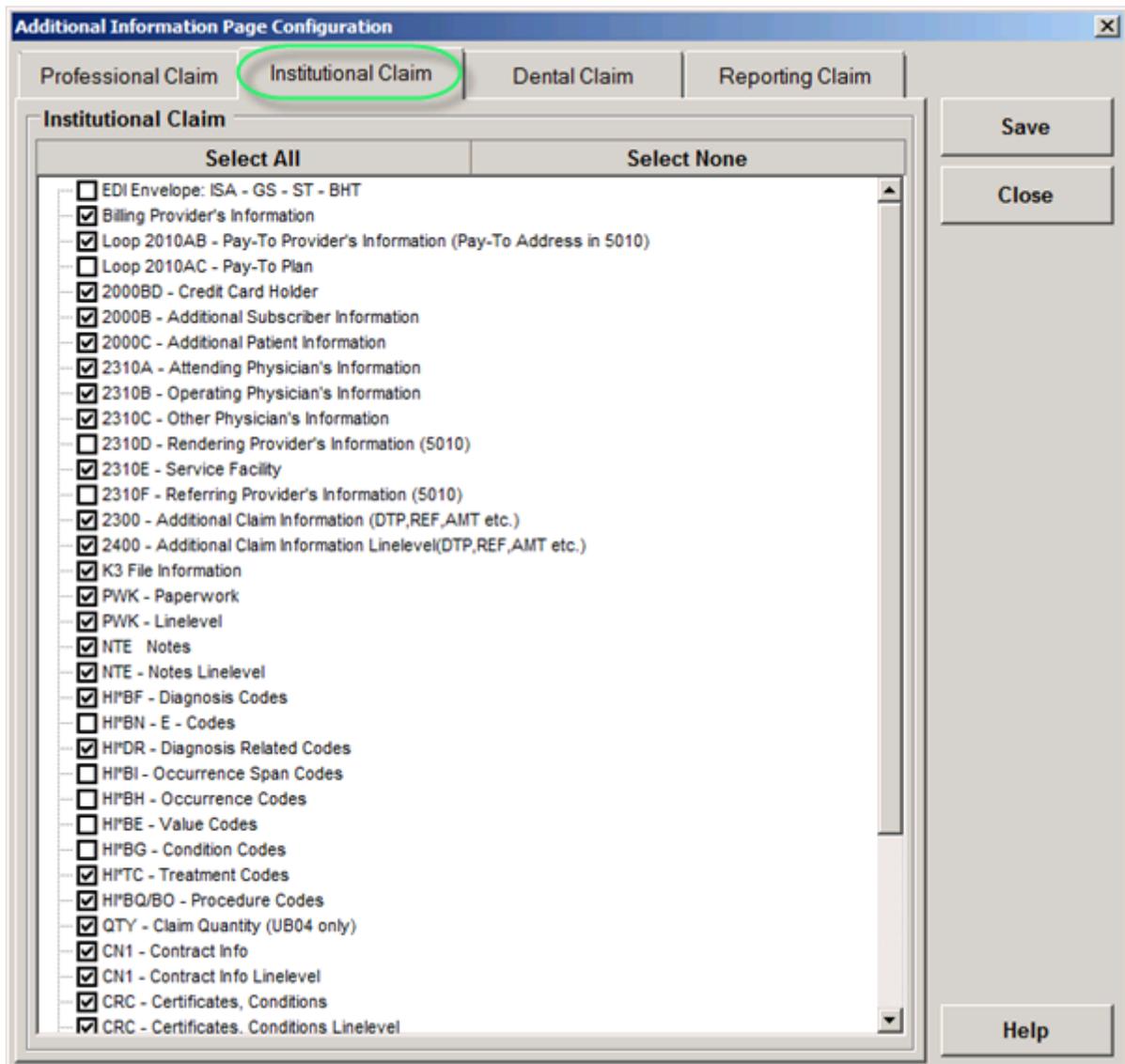
The "Additional Information Page Configuration" window

You can configure the following options:

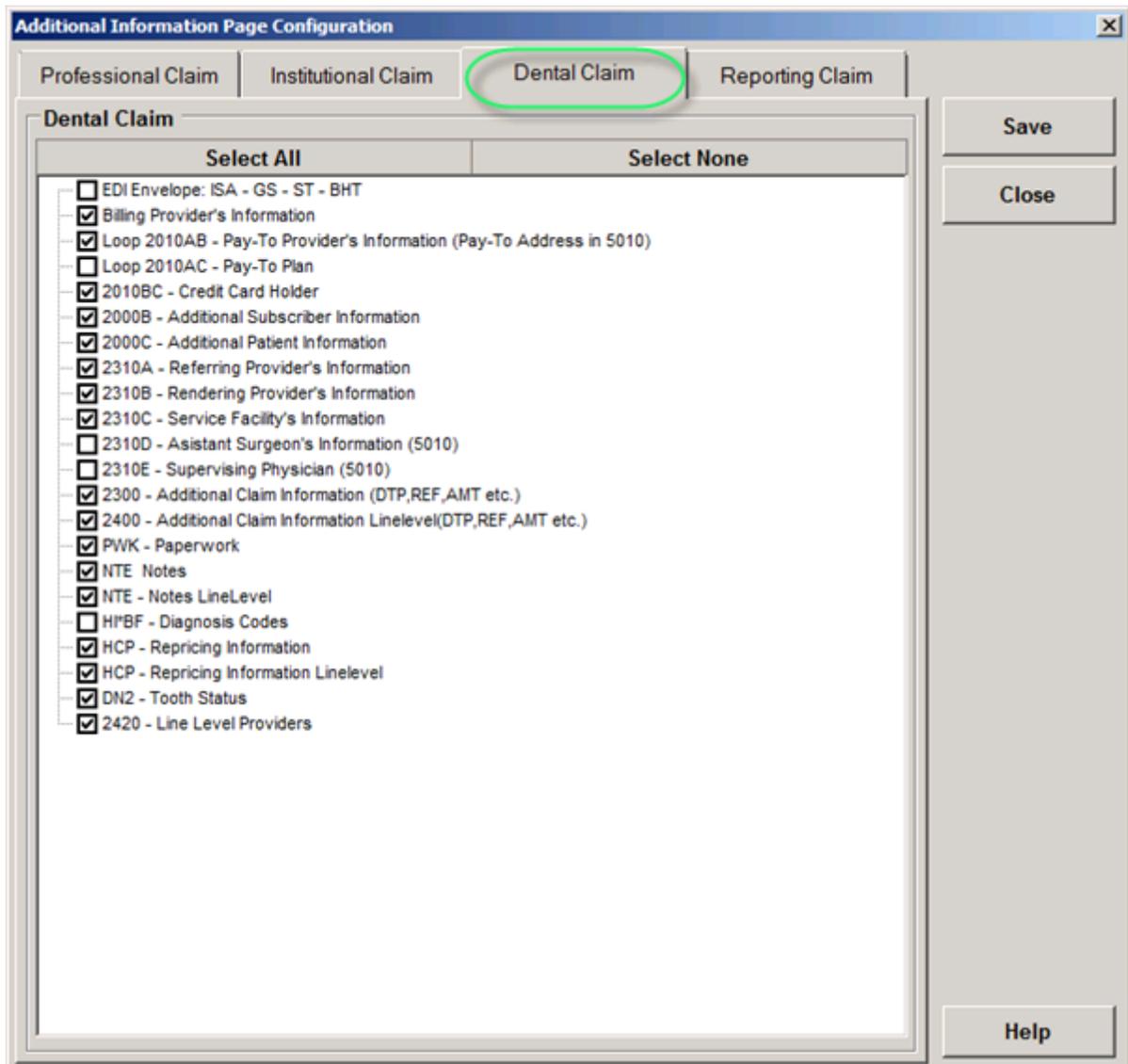
- Professional Claim
- Institutional Claim
- Dental Claim
- Reporting Claim

Once you have finished editing the options, click "Save."

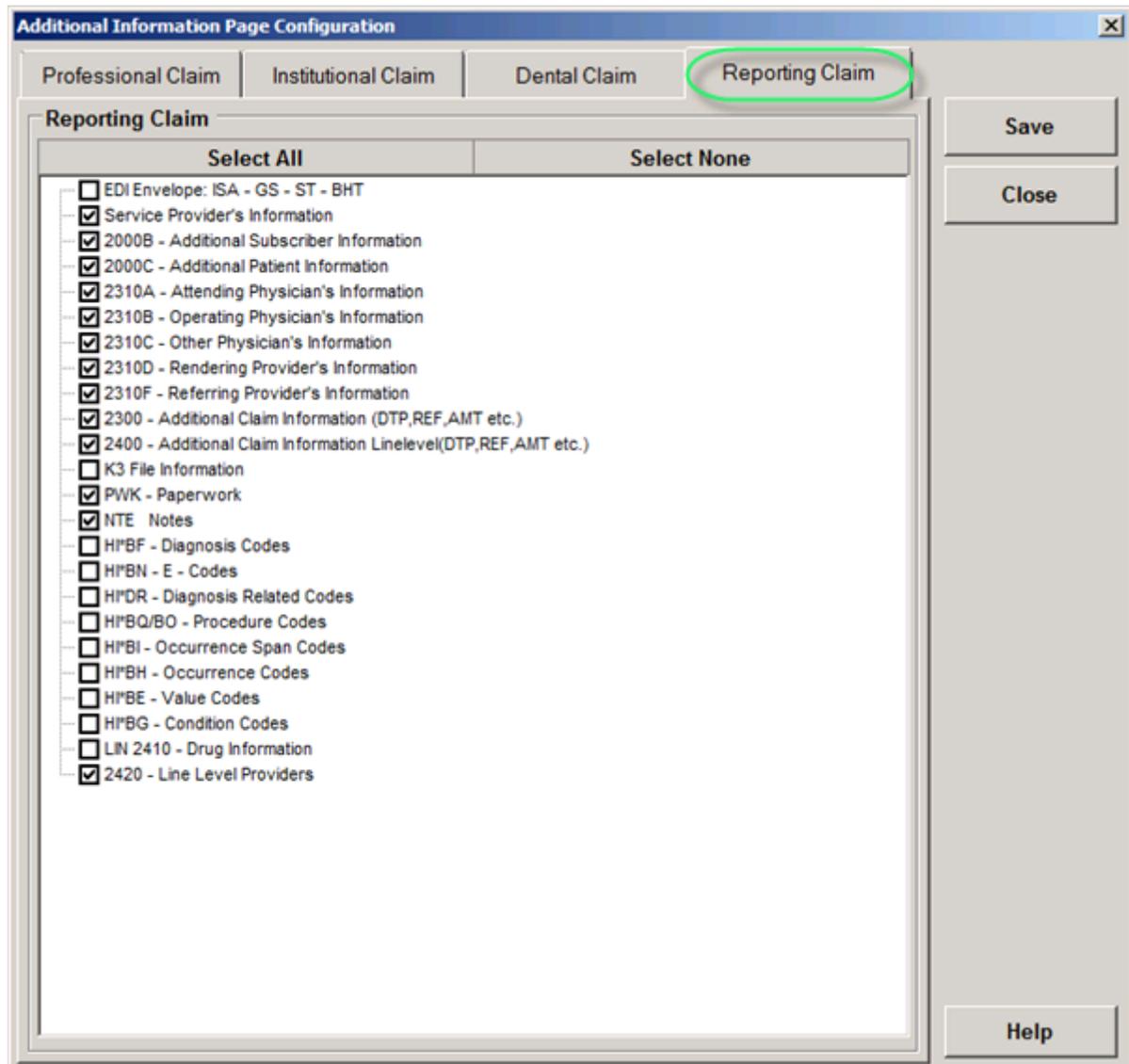
Below you can see screen-shots of the tabs.



The "Institutional Claim" tab



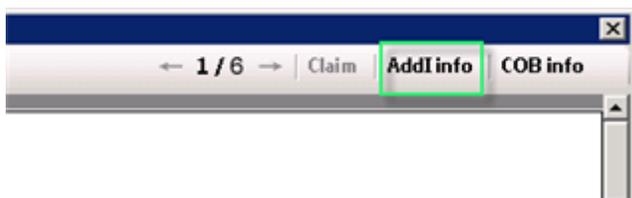
The "Dental Claim" tab



The "Reporting Claim" tab

6.3 Displaying Addl Info

The HIPAA Claim Master checks for additional information every time a claim is [displayed](#) or [printed](#). Should the claim contain one of the above mentioned segments, you will see the button "Addl Info" enabled.



The "Addl Info" button enabled

The arrow buttons indicate that several pages of claim information are available.

Click on the "Addl Info" button to show the additional information as shown on the following screen-shot.

HIPAA Preview

Print Close ← 1 / 6 → Claim Add Info COB Info

Additional Claim Information in the 837 Record

Pay To Provider

Name:	UVA HEALTH SERVICE		
Address:			
City:	ATLANTA	State	GA Zip 30384
Employer's Identification Number	541124769PP		

Referring Provider

Name:	MORGAN RAYMOND
National Provider Identifier	B07928

Service Location

Name:	UVA HEALTH SERVICE		
Address:	PO BOX 281183		
City:	ATLANTA	State	GA Zip 303841183

Claim: Additional Identifiers

Clearing House ID	122030731P01404
-------------------	-----------------

Supplemental Information/Paperwork:

Report Type:	Explanation of Benefits	Sent:	By Mail	Attachment No:	DMN0012
Report Type:	Radiology Reports	Sent:	By E - Mail	Attachment No:	DMNDER34312
Report Type:	Physician Order	Sent:	By Fax	Attachment No:	FRACTUREPICTUREN0012
Report Type:	Discharge Summary	Sent:	Available on Request	Attachment No:	DMN0012

Line Level - Supplemental Information/Paperwork

Line	Report Type	Report Transmission Code:
Line 1	Explanation of Benefits	
	Radiology Reports	
	Physician Order	
	Discharge Summary	
Line 2	Explanation of Benefits	
	Radiology Reports	
	Physician Order	
	Discharge Summary	

Claim Notes

Additional Information:	This is a Duplicate Claim and we are testing the 80 byte limit, I need 10 more
Additional Information:	This is a Duplicate Claim and we are testing the 80 byte limit, I need 10 more
Additional Information:	FNOC NONCONTRACTED PROVIDER
Additional Information:	P448 4 204341E01756600200
Certification Narrative:	P448 4 NEG RATE 0.00 SVNGS 0.00

Additional Claim information

The HIPAA Claim Master displays every element of a specific segment. Sometimes only the HIPAA codes are displayed, often they are translated into their descriptions. If you

need to find out about these codes, please consult the 837 implementation guide.

6.4 Displaying COB

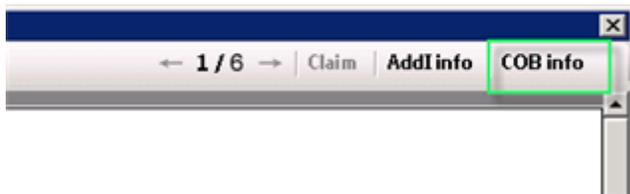
COB information can be very complex within the 837:

- Multiple payments of
 - Multiple lines, which contain
 - Multiple adjustments, done by
 - Multiple payers

...can be tracked in the nested structure of an 837. To display such a nested structure is a daunting task, as you can imagine. I still recommend that you fully understand the nesting of COB data.

On the Claim level, we find additional payers in the 2320 loops. Within the 837 EDI file, adjudication information is stored in either **CAS** segments or the Medicare, **MIA** (inpatient) and **MOA** (outpatient) segments. Specific amounts are similarly stored in the **AMT** segments. On the line level, each line can contain up to 25 different adjudication actions; **SVD** segments, which identify the payer and their amounts. Adjudication information is then written in multiple **CAS** segments and optionally, in the **DTP** segment, the adjudication date.

If the HIPAA Claim Master finds that a claim lists several payers, the claim preview screen will display the "COB" button.



The "COB Info" button

Click this button, and the following display appears:

HIPAA Preview

Print Close ← 1 / 6 → Claim Add Info COB Info

COB Information: Payer 2

Payer Sequence: Primary	Patient's Relationship to Insured: Self		
Insurance Group No:	Group Name:	Filing Code: Commercial Insurance Co.	
Assignment of Benefits: Yes	Release of Information: Yes, Provider has a Signed Statement Permitting Release		
Payer's Name: Aetna Ins	Payor ID: PR23	Claim Adjudication Date: 09/01/2003	

Insured Last Name: Coach	First: Dawn	Middle:	Suffix:
Subscriber's DOB: 04/24/1944	Subscriber's Gender: Female		
Member ID: 26654327902			

Prior Payment - Actual:	\$1,800.00	COB Allowed Amount:	\$2,075.00
Total submitted Charges:	\$2,325.00		

Claim Level Adjudication Information:

Adjustment Group	Adjustment Reason	Amt.	Qty
1 PR - Patient Responsibility	1 - Deductible amount	\$20.00	
2 PR - Patient Responsibility	2 - Coinsurance amount	\$255.00	
3 PI - Payor Initiated Reductions	45 - Charges exceed your contracted/ legislated fee arrangement.	\$250.00	

COB Information: Payer 3

Payer Sequence: Secondary	Patient's Relationship to Insured: Self		
Insurance Group No:	Group Name:	Filing Code: Commercial Insurance Co.	
Assignment of Benefits: Yes	Release of Information: Yes, Provider has a Signed Statement Permitting Release		
Payer's Name: United Healthcare Ins	Payor ID: 04523	Claim Adjudication Date: 09/01/2003	

Insured Last Name: Coach	First: Dawn	Middle:	Suffix:
Subscriber's DOB: 04/24/1944	Subscriber's Gender: Female		
Member ID: 26123456902			

Prior Payment - Actual:	\$10.00	COB Allowed Amount:	\$30.00
Total submitted Charges:	\$2,325.00		

Claim Level Adjudication Information:

Adjustment Group	Adjustment Reason	Amt.	Qty

COB Information: Payer 4

Payer Sequence: Tertiary	Patient's Relationship to Insured: Self		
Insurance Group No:	Group Name:	Filing Code: Commercial Insurance Co.	
Assignment of Benefits: Yes	Release of Information: Yes, Provider has a Signed Statement Permitting Release		
Payer's Name: Cigna Ins	Payor ID: NG451	Claim Adjudication Date: 09/01/2003	

Insured Last Name: Coach	First: Dawn	Middle:	Suffix:
Subscriber's DOB: 04/24/1944	Subscriber's Gender: Female		
Member ID: 26123457902			

Prior Payment - Actual:	\$10.00	COB Allowed Amount:	\$30.00
Total submitted Charges:	\$2,325.00		

Claim Level Adjudication Information:

Adjustment Group	Adjustment Reason	Amt.	Qty

COB Information: Payer 5

Payer Sequence: Tertiary	Patient's Relationship to Insured: Self		
Insurance Group No:	Group Name:	Filing Code: Commercial Insurance Co.	
Assignment of Benefits: Yes	Release of Information: Yes, Provider has a Signed Statement Permitting Release		

Information about additional payers

Each payer is in one block. Any claim level adjudication will be displayed, listing adjustment group and adjustment reason.

Line level information is displayed on the following pages.

HIPAA Preview												
Member ID		210005000										
COB PAYER PAID:				\$16.00			COB Allowed Amount:				\$110.00	
Claim Level Adjudication Information:												
Adjustment Group	Adjustment Reason						Amt.	Qty				
1	CO - Contractual Obligations	45 - Charges exceed your contracted/ legislated fee arrangement.						\$390.00				
2	PR - Patient Responsibility	1 - Deductible amount						\$90.00				
3	PR - Patient Responsibility	2 - Coinsurance amount						\$4.00				
Line Level COB Information												
Line:	Proc Code	Modifiers	Charge	From Date	To Date	Units	Facility	EMG	EPSDT	Fam Pla		
1	93016	TC	\$100.00	09/01/2003	09/01/2003	1 UN	11					
	Payer ID	Payer Name	Proc Cd	Mods	Rev Code	Payer Paid	Paid Qty	Bnd/Unb.	Adjud. Date			
	3670	Medicare	93016	TC		\$0.00	1	1	09/01/2003			
	Adjustment Group	Adjustment Reason						Amt.	Qty			
	1	CO - Contractual Obligations	45 - Charges exceed your contracted/ legislated fee arrangement.						\$80.00			
	2	PR - Patient Responsibility	1 - Deductible amount						\$20.00			
Line:	Proc Code	Modifiers	Charge	From Date	To Date	Units	Facility	EMG	EPSDT	Fam Pla		
2	93017	TC	\$100.00	09/01/2003	09/01/2003	1 UN	11					
	Payer ID	Payer Name	Proc Cd	Mods	Rev Code	Payer Paid	Paid Qty	Bnd/Unb.	Adjud. Date			
	3670	Medicare	93017	TC		\$0.00	1	2	09/01/2003			
	Adjustment Group	Adjustment Reason						Amt.	Qty			
	1	CO - Contractual Obligations	45 - Charges exceed your contracted/ legislated fee arrangement.						\$45.00			
	2	PR - Patient Responsibility	1 - Deductible amount						\$55.00			
Line:	Proc Code	Modifiers	Charge	From Date	To Date	Units	Facility	EMG	EPSDT	Fam Pla		
3	93018	TC	\$100.00	09/01/2003	09/01/2003	1 UN	11					
	Payer ID	Payer Name	Proc Cd	Mods	Rev Code	Payer Paid	Paid Qty	Bnd/Unb.	Adjud. Date			
	3670	Medicare	93018	TC		\$0.00	1	3	09/01/2003			
	Adjustment Group	Adjustment Reason						Amt.	Qty			
	1	CO - Contractual Obligations	45 - Charges exceed your contracted/ legislated fee arrangement.						\$85.00			
	2	PR - Patient Responsibility	1 - Deductible amount						\$15.00			
Line:	Proc Code	Modifiers	Charge	From Date	To Date	Units	Facility	EMG	EPSDT	Fam Pla		
4	80048		\$100.00	09/01/2003	09/01/2003	1 UN	11					
	Payer ID	Payer Name	Proc Cd	Mods	Rev Code	Payer Paid	Paid Qty	Bnd/Unb.	Adjud. Date			
	3670	Medicare	80048			\$8.00	1	4	09/01/2003			
	Adjustment Group	Adjustment Reason						Amt.	Qty			
	1	CO - Contractual Obligations	45 - Charges exceed your contracted/ legislated fee arrangement.						\$90.00			

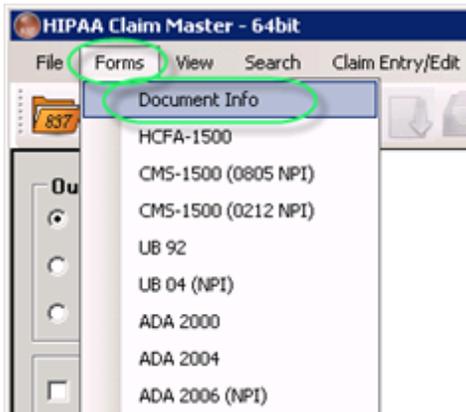
COB information on the line level

Notice the nested structure of the display. For each line, we see a recap of the line information, followed by payer information and possibly several lines of adjudication information.

6.5 Displaying Document Info

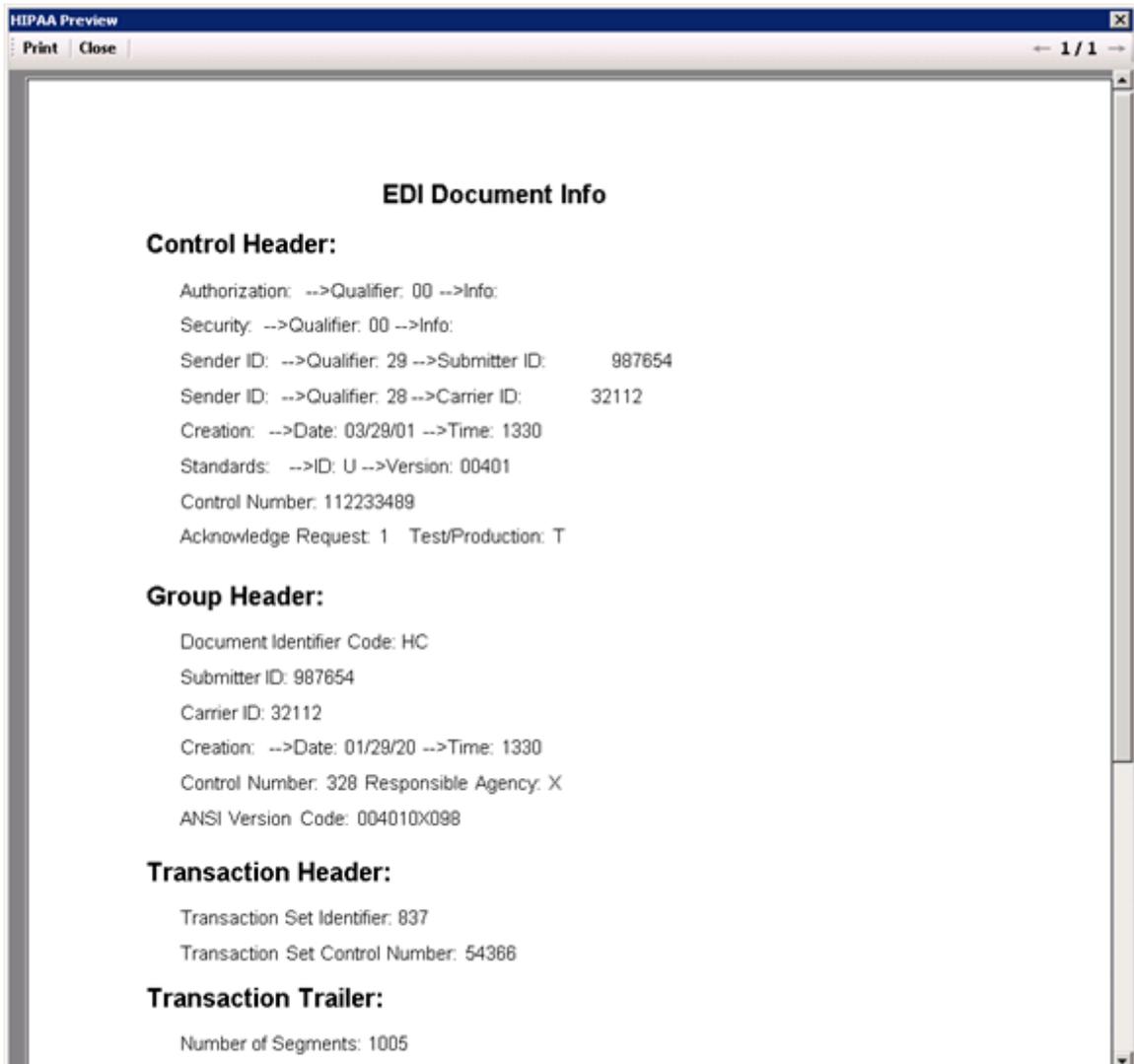
The EDI envelope contains information about the transaction set, the sender, the receiver, dates and other pieces that are not printed on medical forms.

1. Select *Forms* ▶ *Document Info* in the main menu.



The "Document Info" menu

2. The following screen will be displayed.

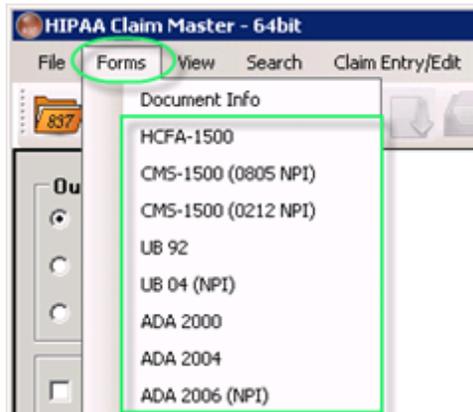


Displaying the EDI envelope information

6.6 Printing Blank Forms

Follow the instructions below to print a blank form.

Select the "Forms" menu.



The "Form" menu and its sub-menus

Select one of the following claim forms that the HIPAA Claim Master can display:

- HCFA-1500
- CMS-1500 (0805 NPI)
- CMS-1500 (0212 NPI)
- UB 92
- UB 04 (NPI)
- ADA 2000
- ADA 2004
- ADA 2006 (NPI)

Clicking on this menu displays a blank form depending on the choice of claim type. This form can be sent to the Printer by clicking on the "Print" button.

HIPAA Preview

Print Close

← 1 / 1 →

Sender ID:
Orig. Ref. No:

Dental Claim Form
©American Dental Association, 1999 version 2000

<input type="checkbox"/> 1. Benefit pre-treatment estimate Specialty		3. Center Name	
<input type="checkbox"/> Benefit statement of actual services		4. Center Address	
<input type="checkbox"/> 2. Medicaid Claim Prior Authorization #		5. City	
<input type="checkbox"/> Electronic Claim #		6. State	7. Zip

8. Patient Name (Last, First, Middle)		9. Address		10. City		11. State	
12. Date of Birth (MM/DD/YYYY)		13. Patient ID #		14. Sex <input type="checkbox"/> M <input type="checkbox"/> F		15. Phone Number	
16. Zip Code		17. Relationship to Subscriber/Employer <input type="checkbox"/> Self <input type="checkbox"/> Spouse <input type="checkbox"/> Child <input type="checkbox"/> Other		18. Employer/School Name		Address	

19. Subscriber ID/IDBI #		20. Group Name		21. Group #		22. Policy #	
22. Subscriber/Employer Name (Last, First, Middle)		23. Address		24. Phone Number		25. Zip Code	
25. City		26. State		27. Zip Code		28. Date of Birth (MM/DD/YYYY)	
29. Marital Status <input type="checkbox"/> Married <input type="checkbox"/> Single <input type="checkbox"/> Other		30. Sex <input type="checkbox"/> M <input type="checkbox"/> F		31. Is Patient covered by another plan for services listed on this form? <input type="checkbox"/> No (Skip 32-37) <input type="checkbox"/> Yes <input type="checkbox"/> Dental or <input type="checkbox"/> Medical		32. Policy #	
33. Other Sub-Owner's Name (if different)		34. Date of Birth (MM/DD/YYYY)		35. Sex <input type="checkbox"/> M <input type="checkbox"/> F		36. Program Name	
37. Employer/School Name		38. Employer/School Address		39. I have been involved in the treatment planned as indicated here. I agree to be responsible for all charges for dental services and materials provided by my dental benefit plan, unless the treating dentist or dental practice has a contractual agreement with my plan prohibiting all or a portion of such charges. To the extent permitted under applicable law, I authorize release to the carrier of any information and documentation relating to this claim and/or any request for pre-treatment estimate will find my further authorization in the future.		40. Employee/School Name	
41. I hereby authorize payment of the dental benefit otherwise payable to me directly to the below named dental entity.		42. Employee/School Address		43. Signed (Patient/Quarant)		Date (MM/DD/YYYY)	

41. Name of Billing dentist or dental entity		42. Phone Number		43. Fax		44. Provider ID #		45. Dentist (Doc, Sec, or T, J, N)	
46. Address		47. Dentist License #/State		48. First (last) of current office		49. Place of Treatment <input type="checkbox"/> Office <input type="checkbox"/> Hosp. <input type="checkbox"/> BOI <input type="checkbox"/> Other		50. Is treatment for orthodontics? <input type="checkbox"/> Yes <input type="checkbox"/> No	
50. City		51. State		52. Zip Code		53. Radiographs or models enclosed? <input type="checkbox"/> Yes - how many? _____ <input type="checkbox"/> No		54. Is treatment for orthodontics? Is restor already commenced: <input type="checkbox"/> Yes <input type="checkbox"/> No	
55. Implants (crowns, bridges, dentures), is this initial placement? <input type="checkbox"/> Yes <input type="checkbox"/> No		56. If treatment for orthodontics, is this initial placement? <input type="checkbox"/> Yes <input type="checkbox"/> No		57. If treatment result of: <input type="checkbox"/> Auto accident? <input type="checkbox"/> Other accident? <input type="checkbox"/> Other		58. If treatment result of: <input type="checkbox"/> Auto accident? <input type="checkbox"/> Other accident? <input type="checkbox"/> Other		59. If treatment result of: <input type="checkbox"/> Auto accident? <input type="checkbox"/> Other accident? <input type="checkbox"/> Other	
60. If treatment result of: <input type="checkbox"/> Auto accident? <input type="checkbox"/> Other accident? <input type="checkbox"/> Other		61. If treatment result of: <input type="checkbox"/> Auto accident? <input type="checkbox"/> Other accident? <input type="checkbox"/> Other		62. If treatment result of: <input type="checkbox"/> Auto accident? <input type="checkbox"/> Other accident? <input type="checkbox"/> Other		63. If treatment result of: <input type="checkbox"/> Auto accident? <input type="checkbox"/> Other accident? <input type="checkbox"/> Other		64. If treatment result of: <input type="checkbox"/> Auto accident? <input type="checkbox"/> Other accident? <input type="checkbox"/> Other	

60. Diagnostic Code Index		61. Date of procedure		62. Date of procedure		63. Date of procedure		64. Date of procedure	
65. Examination and treatment plan - List teeth in order		66. Date service complete (Mo, Day, Year)		67. Date service complete (Mo, Day, Year)		68. Date service complete (Mo, Day, Year)		69. Date service complete (Mo, Day, Year)	
Teeth	Surface	Diagnosis Index #	Procedure Code	Qty	Description	Fee	Admin. Use Only		
1.			1.			1.			
2.			2.			2.			
3.			3.			3.			
4.			4.			4.			

Display of a blank ADA 2000 form

Chapter

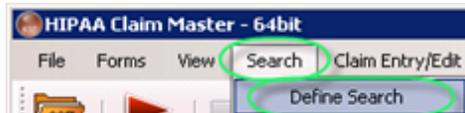
VII

7 Using Search

7.1 Searching through a Claim File

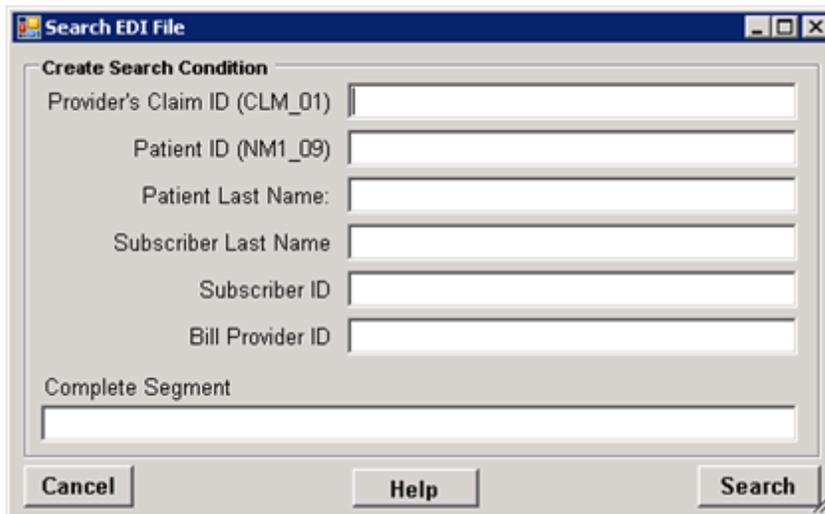
Follow the instructions below to search through the claim file.

1. Select *Search* ▶ *Define Search* in the main menu.



The "Define Search" menu

2. The following screen will appear. It will let you define search conditions.



The "Search EDI File" Screen

3. Specify your search criteria. You can search for:

- The Provider's Claim Number or Patient's Record number that is stored in CLM_01 segment
- The Patient Control Number or Patient's ID
- Patient Last Name
- Subscriber ID
- Subscriber Last Name
- Billing Provider's ID
- A complete EDI segment

You can specify multiple search conditions.

4. Click "Search."
5. The HIPAA Claim Master parse through the file and display the claim that fulfills this criterion. If there are many claims, it will display each claim that has this condition.

Chapter



8 Editing and Entering Claims

8.1 Using the Claim Entry/Edit Function

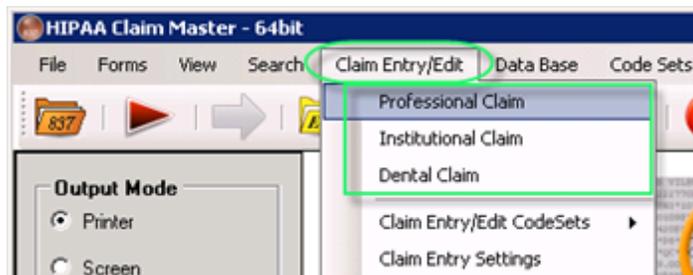
You can enter either a Professional, Institutional or Dental claim. Claim entry requires more transaction code sets and you can access and edit them from those menus as well as setup the component.

The following topics describe settings for the Claim Entry/Edit function:

- [Selecting a Claim Entry/Edit CodeSet](#)
- [Defining Claim Entry Settings](#)
- [Creating Diagnosis Table](#)

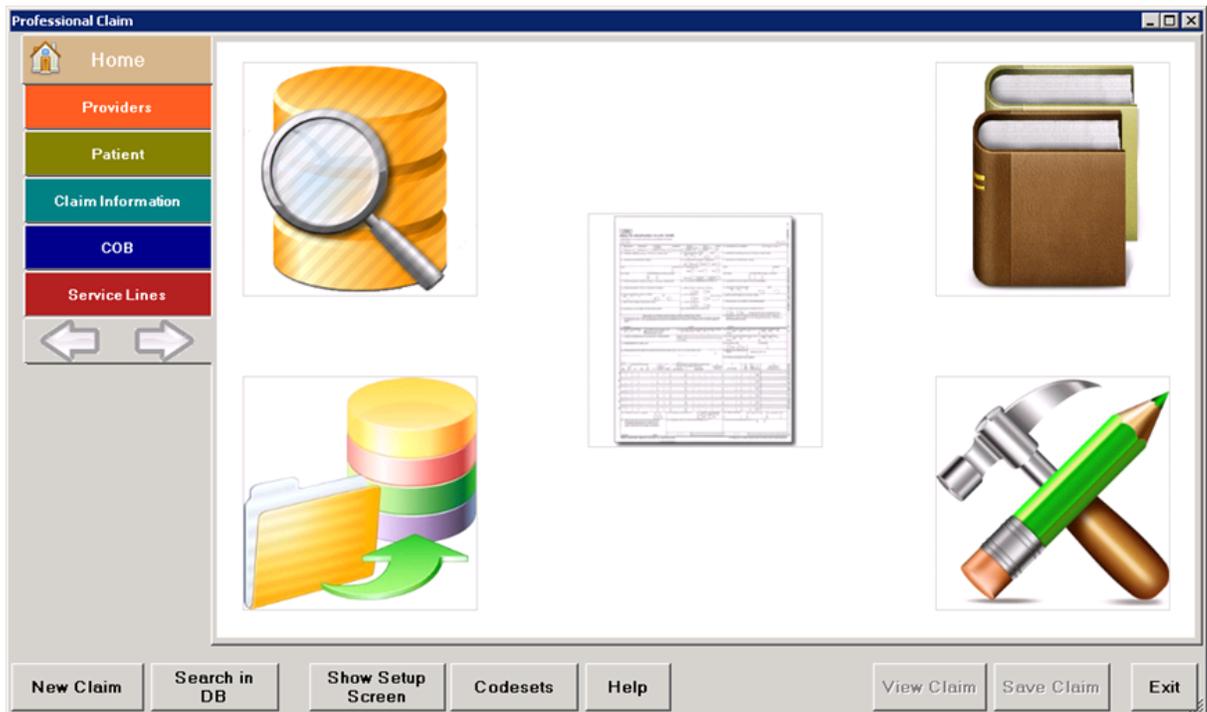
Follow the instructions below to edit or entry claims.

1. Select *Claim Entry/Edit* ▶ *Professional / Institutional / Dental Claim* in the main menu.



The "Claim Entry/Edit" menu

2. A screen like the one pictured below will be displayed. The following screen-shots will use the "Professional Claim" window.

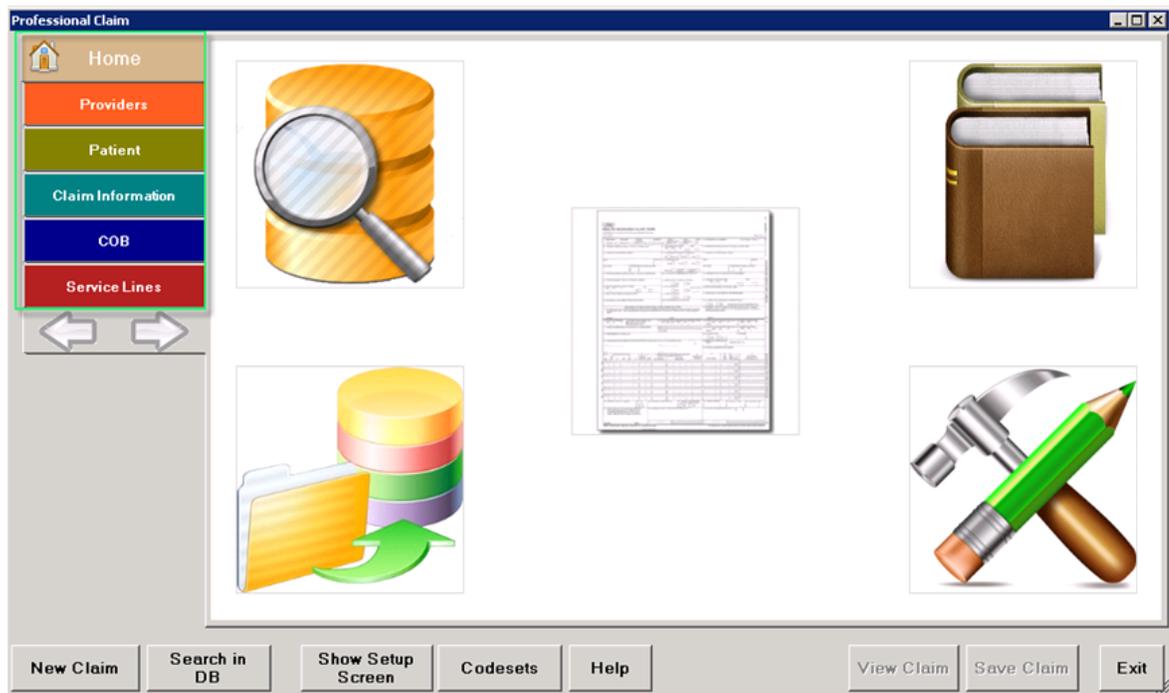


The "Professional Claim" window

3. Each segment or loop has its own set of editing screens listed on the left; they can optionally have different colors. The five (5) main data groups are:

- Providers
- Patient
- Claim Information
- COB
- Service Lines

Click one of these menu items to open to the relevant screen.



The left menu in the "Professional Claim" window

4. Once you have accessed a main category screen, you will see the top menu to access the individual screen. Use it to navigate along the claim.

Below you can see the billing provider information from a claim that was loaded from the database. To see and edit other provider types use the tabs at the top of the screen. The following image shows all the provider types that are part of the Professional Claim implementation.

Professional Claim

Home Billing Provider Pay-To Provider Referring Provider Rendering Provider Purchased Services Service Facility Supervising Provider

Providers Patient Claim Information COB Service Lines

Field 33 in the HCFA 1500 Form holds this information

Billing Provider Details Other ID's and Charge Info

Last/Organization Name: GRUPO EMPRESAS DE SALUD CAROLINA

First Name: Middle: Suffix:

Provider ID: 660614786 ID Type: 24 - Employer's Identification Number

Address: PO BOX 193044

Address2:

City: SAN JUAN State: PR Zip: 00919

Provider Specialty: ...

Contact: Phone: () - - Extension: Fax () - - Email:

Save Clear Help

Show Setup Screen Codesets Help View Claim Save Claim Exit

The "Providers" tab selected in the "Professional Claim" window

Different sections may have a different range of tabs across the top. The following screen-shot shows top menu items for the "Service Lines" section selected in the left. This top menu differs from the top menu displayed on the previous screen-shot with the "Providers" section selected.

No	Service From	Service To	Place	Procedures, Service or Supplies, M...	Charges	Diag. Poi...	Units
1	08/28/2007		11	HC 93000	18.43	1.	UN

The "Service Lines" tab is selected in the "Professional Claim" window

- To preview your changes, click "View Claim." A claim form will be shown on the screen.

The "View Claim" button

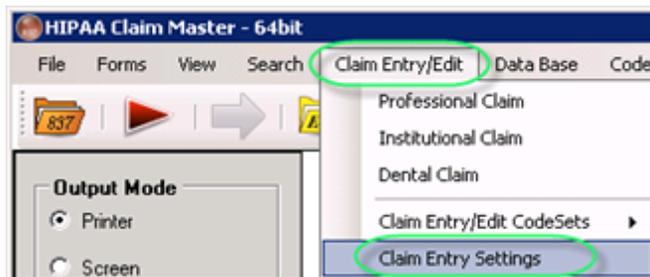
- To save the changes, click "Save Claim."

The "Save Claim" button

8.2 Selecting a Claim Entry/Edit CodeSet

8.3 Defining Claim Entry Settings

- To configure the claim entry options, select *Claim Entry/Edit* ▶ *Claim Entry Settings* in the main menu.



The "Claim Entry Settings" menu

Alternatively, you can access the setup window from the Claim entry/edit window. Click the "Show Setup Screen" button. Read more in [Using the Claim Entry/Edit Function](#).

No	Service From	Service To	Place	Procedures, Service or Supplies, M...	Charges	Diag. Poi...	Units
1	08/21/2007		11	HC 99214	76.81	1.	UN

The "Show Setup Screen" button on the "Professional Claim" window

2. The following screen will appear.

Claim Entry Setup

Colors for Tabs

Do not use colors for tabs

Use colors for tabs

Restore default colors

	Main Color	Selected Color
<i>Providers:</i>	<input type="color"/>	<input type="color"/>
<i>Patient:</i>	<input type="color"/>	<input type="color"/>
<i>Claim Information:</i>	<input type="color"/>	<input type="color"/>
<i>COB:</i>	<input type="color"/>	<input type="color"/>
<i>Service Lines:</i>	<input type="color"/>	<input type="color"/>
<i>Font Color:</i>	<input type="color"/>	
<i>Home Tab:</i>	<input type="color"/>	

Auto-Complete Mode

Turn off Auto-Complete for Provider, Payer and Subscriber names

Turn off Auto-Complete for Provider, Payer and Subscriber ID's

New Claims

Version in New Claims

Use 4010

Use 5010

Information in New Claims

Production

Test

Save Claims

Validate claims before save

Don't save claim if warnings

Save Entities

Save Providers

Save Payers

Save Subscribers/Patients

Close **Help** **Save**

The HIPAA "Claim Entry Setup" window

3. The claim entry setup screen is divided into the following groups:

- Colors for Tabs
- Auto-Complete Mode
- New Claims
- Save Claims
- Save Entities

Read the detailed descriptions further.

4. Once you have finished editing the options, click the *Save* button to save any changes or click the *Close* button do discard them.

Colors for Tabs

Unlike other screens in the HIPAA Claim Master, the window for claim entry has a lot of

colors (see [Using the Claim Entry/Edit Function](#)). We decided to do this to give a clear distinction between the different groups of information. There are so many screens in this application that it is easily confusing to the data entry person. To develop a sense of intuition of where one is in the process and flow of the program we decided to add colors. You also have the choice to not use colors for the tabs:

- Do not use colors for tabs
- Use colors for tabs

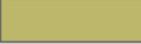
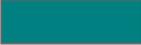
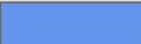
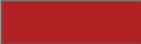
You are free to customize the colors. Click the button with the three dots to the right of each color to display the color selection screen where you can choose the color for a particular segment.

Colors for Tabs

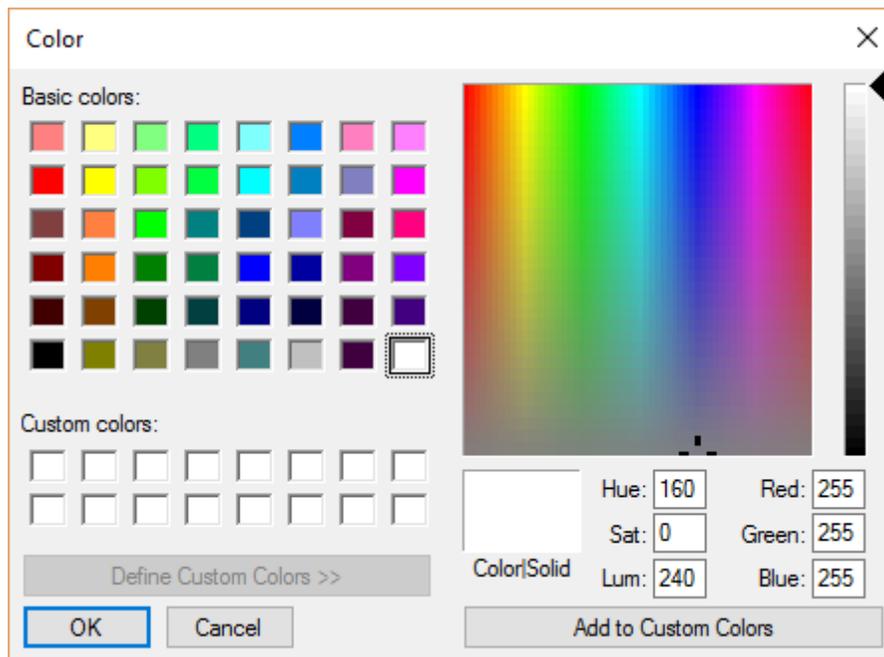
Do not use colors for tabs

Use colors for tabs

Restore default colors

	<u>Main Color</u>	<u>Selected Color</u>
<i>Providers:</i>	 	 
<i>Patient</i>	 	 
<i>Claim Information:</i>	 	 
<i>COB:</i>	 	 
<i>Service Lines:</i>	 	 
<i>Font Color:</i>	 	
<i>Home Tab:</i>	 	

The color palette selection button



The color selection window

New Claims

HIPAA Claim Entry lets you predefine the version of the claim that you enter. This choice will influence the screens that you are presented with. Claims you create will be written to the database according to the implementation guide for the version that you choose.

New Claims

Version in New Claims

Use 4010

Use 5010

Information in New Claims

Production

Test

Claim version options

Version in New Claims

Set new claims to the X12 4010 or 5010 standards. A claim loaded into the claim editor will preserve the standard it was saved in originally.

- Use 4010

- Use 5010

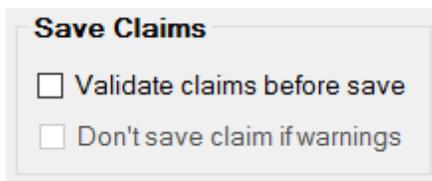
Information in New Claims

Set new claims to use Production or Test data.

- Production
- Test

Save Claims

Before saving a claim to the database, it can be validated against the X12 standard. The compliance engine will warn users of errors that would make the claim fail compliance, such as missing segments, incorrect datatypes, unrecognized qualifiers or excessive characters in a particular field.



Save Claims

Validate claims before save

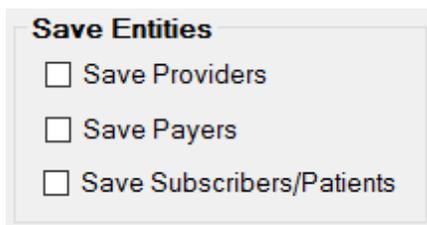
Don't save claim if warnings

Claim compliance validation options

- Validate claims before save - runs the compliance engine to check for potential errors in a claim file
- Don't save claim if warnings - if a claim fails the compliance check, it is not saved to the database

Save Entities

Saving entities to the database is an option for facilitating data entry. Providers, patients and payers are saved into a database after being entered for later retrieval. Entering a last name or ID will automatically fill the entity's remaining fields in the claim editor. They can be individually selected. To view currently saved entities, see entity editing.



Save Entities

Save Providers

Save Payers

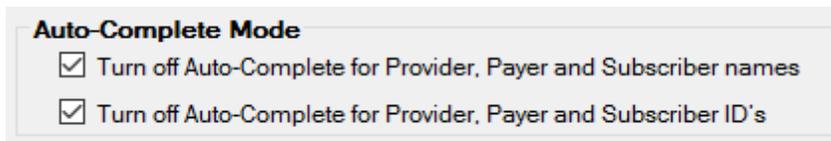
Save Subscribers/Patients

Save entity to DB options

Checking an entity will allow a user to autofill a provider, payer, patient or subscriber in the claim editor by filling in the entity's ID.

Auto-Complete Mode

Entities saved to the database can be loaded into a provider, payer or patient form by typing out the relevant entity's name or ID. The ID and Name textboxes will suggest an entity to enter into the form and accepting will autocomplete the remaining details. To disable this capability for the ID or Name, check the appropriate box. To learn more about autocomplete for providers, payers and subscribers, see [Provider, Patient and Payer Autocomplete](#).



Auto-Complete Mode

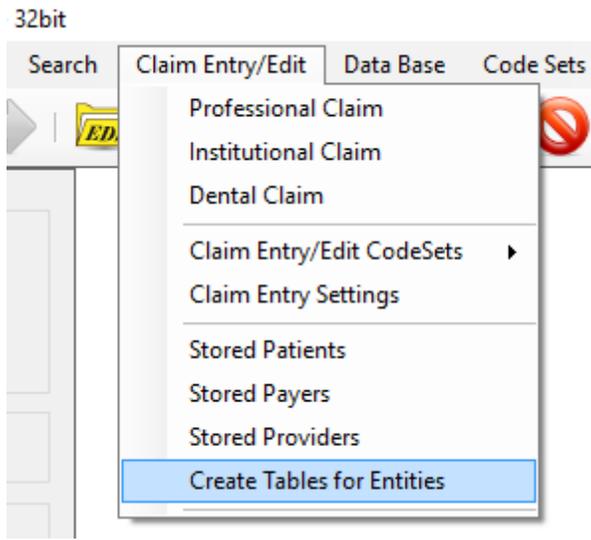
- Turn off Auto-Complete for Provider, Payer and Subscriber names
- Turn off Auto-Complete for Provider, Payer and Subscriber ID's

Autocomplete options

8.4 Provider, Patient and Payer Tables

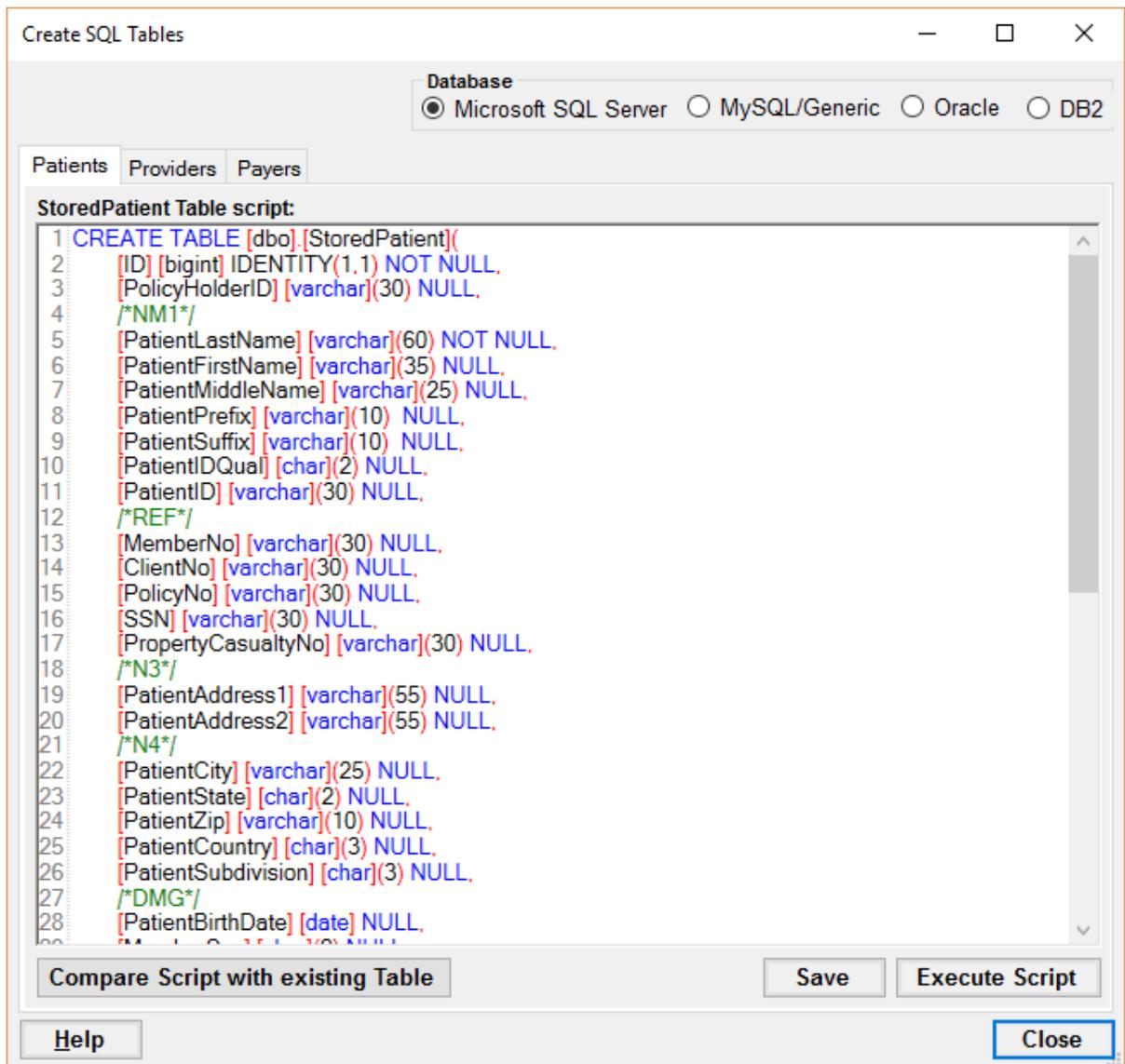
Filling in provider, patient, and payer details multiple times is a repetitive task that can be alleviated by storing the relevant entity in a table for later retrieval. For example, entering a provider's ID number will suggest a stored provider to select from and automatically fill the remaining fields in the provider form with the relevant data. To create the provider, patient or payer tables follow these steps:

1. Click *Claim Entry/Edit* ▶ *Create Tables for Entities*



Create tables for entities menu

2. This opens the *Create SQL Tables* window. Here you can create the *StoredPatient*, *StoredProvider*, and *StoredPayer* tables. Check the appropriate database is selected and click *Execute Script* to create a table. If the table has been changed in a new release, clicking *Compare Script with existing table* will create a script to adjust the table. Changes to database tables do not occur often.



Create SQL Tables window

3. Click *Execute Script*. A new table will be created.

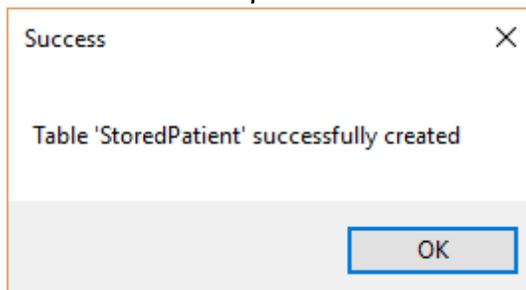
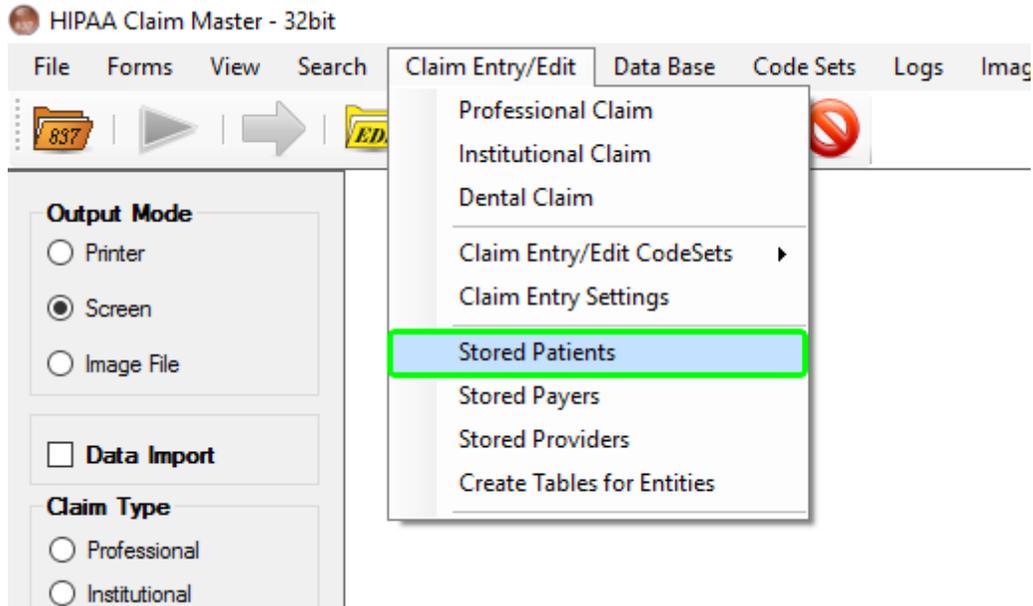


Table creation success

8.5 Stored Patients

Stored patients can be viewed and edited in this screen. To open the Stored Patients editor, click *Claim Entry/Edit* and then *Stored Patients*:



Stored patients menu option

A new window containing all patients, both subscribers and dependents, will be opened. To create the *StoredPatient* table, see [Provider, Patient and Payer Tables](#).

Stored Patient editing screen

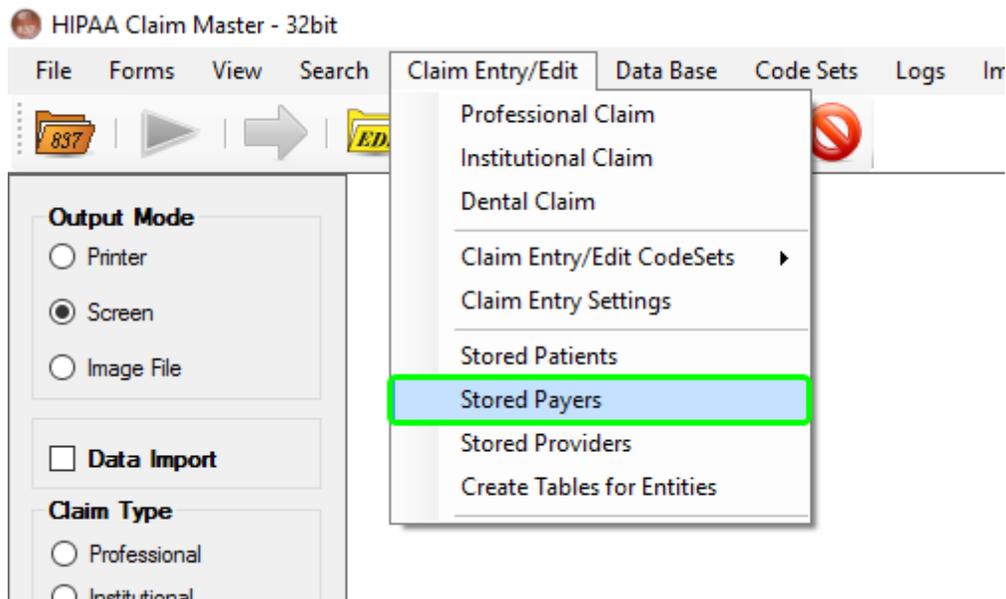
The table contains all patients present in the StoredPatient table. Selecting a patient in the table will fill out the Patient Information form. Editing the fields and clicking *Save Patient into Database* will modify the contents of the selected patient's row to match your changes.

To create a new patient, click the *New Patient* button below the Stored Patients table, fill in the Patient Information, and click the *Save Patient into Database* button to save the new patient.

To delete an existing patient, select the patient in the Stored Patients table and click the *Delete Patient* button.

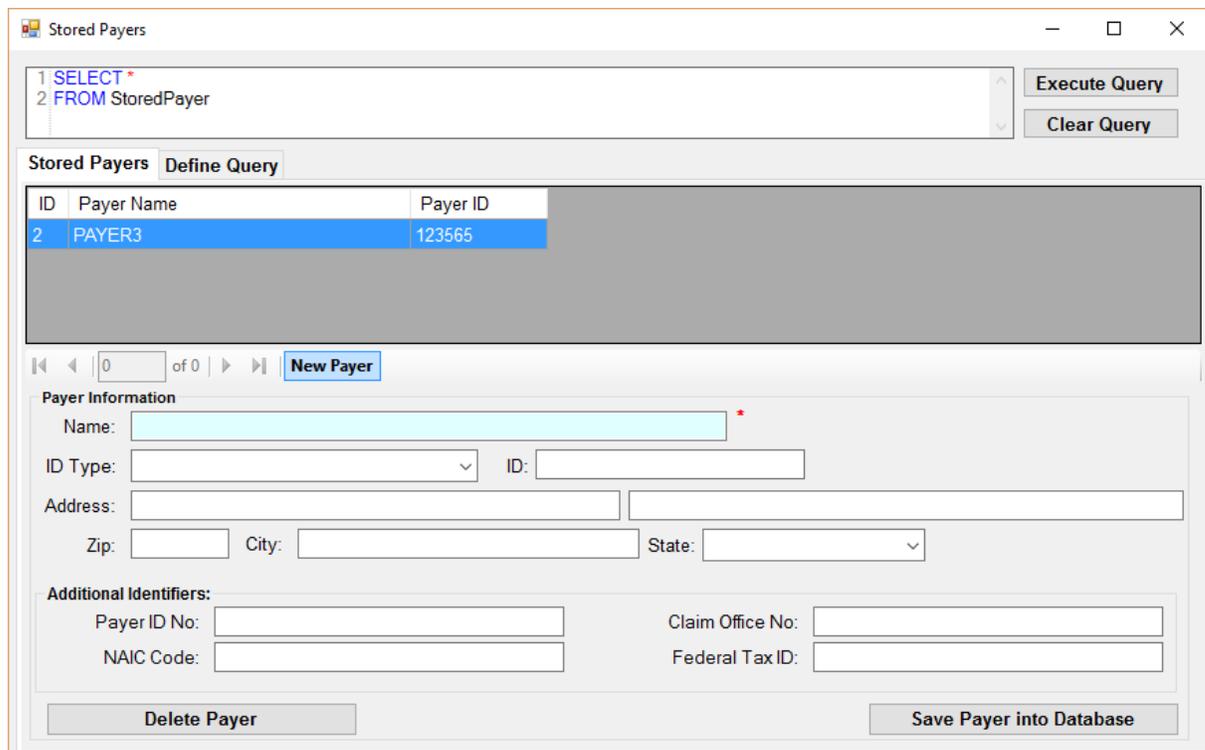
8.6 Stored Payers

Stored payers can be viewed and edited in this screen. To open the Stored Payers editor, click *Claim Entry/Edit* and then *Stored Payers*:



Stored payers menu option

A new window containing all payers stored in the *StoredPayer* table will be opened. To create the *StoredPayer* table, see [Provider, Patient and Payer Tables](#).



Stored Payer editing screen

The table contains all payers present in the *StoredPayer* table. Selecting one will fill out

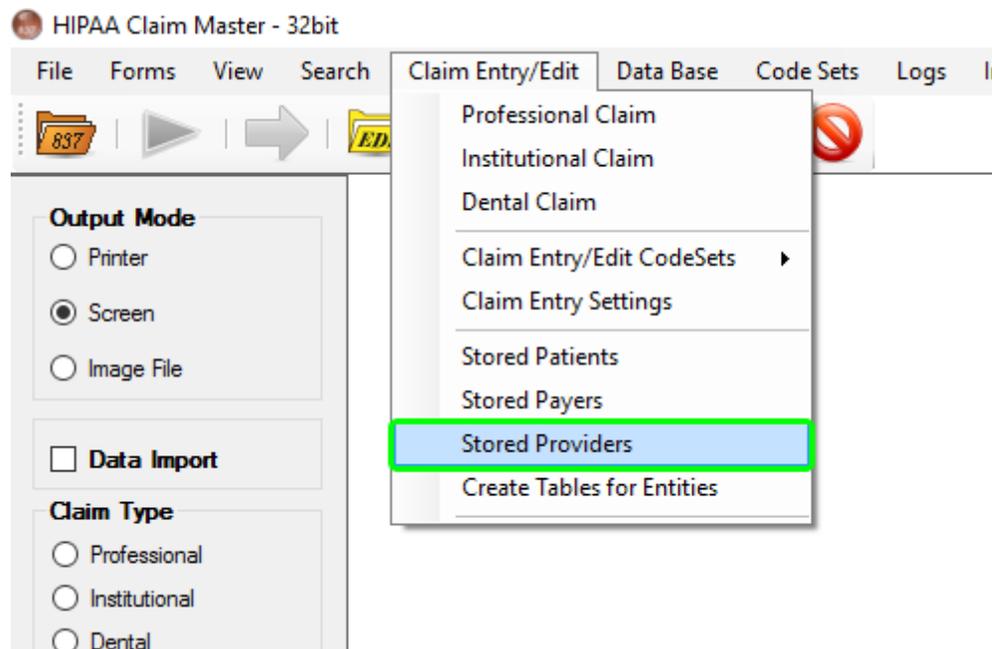
the Payer Information form beneath it. Editing the payer fields and clicking *Save Payer into Database* will modify the contents of the selected payer's row to match your changes.

To create a new payer, click the *New Payer* button below the Stored Payers table, fill in the Payer Information, and click the *Save Payer into Database* button to save the new Payer.

To delete an existing payer, select the payer in the Stored Payers table and click the *Delete Payer* button.

8.7 Stored Providers

Providers stored in the database can be viewed and edited in the Stored Providers editing screen. To open the Providers editor, click *Claim Entry/Edit* and then *Stored Providers*:



Stored providers menu option

A new window containing all payers stored in the *StoredProvider* table will be opened. To create the *StoredProvider* table, see [Provider, Patient and Payer Tables](#).

Stored Providers editing screen

Stored Provider editing screen

The table contains all providers present in the *StoredProvider* table. Selecting one will fill out the Provider Information form to the right of the window. Editing these fields and later clicking *Save Provider into Database* will modify the contents of the selected provider's row to match your changes.

To create a new provider, click the *New Provider* button below the Stored Provider table, fill in the Provider Information, and click the *Save Provider into Database* button to save the new provider.

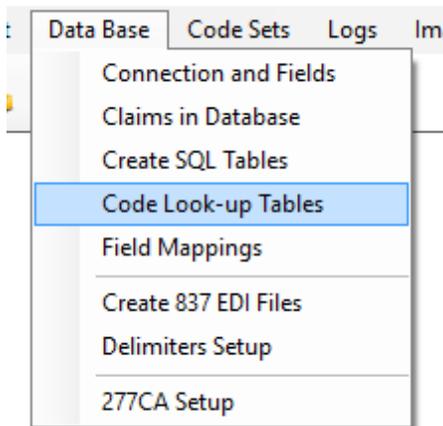
To delete an existing provider, select the provider in the Stored Providers table and click the *Delete Provider* button.

8.8 Creating CodeSet Tables

Some codesets are so numerous that it does not make sense to store them in an XML file. There are more than 25,000 procedure and revenue codes and more than 15,000 diagnosis codes in ICD-9 alone. For that reason we decided to store Procedure Codes, Diagnosis Codes, Revenue Codes and ZIP codes in the database. In the following example a Diagnosis Code table will be created.

Note: You need to configure the connection to the database first. See [Setting up Database Connection](#).

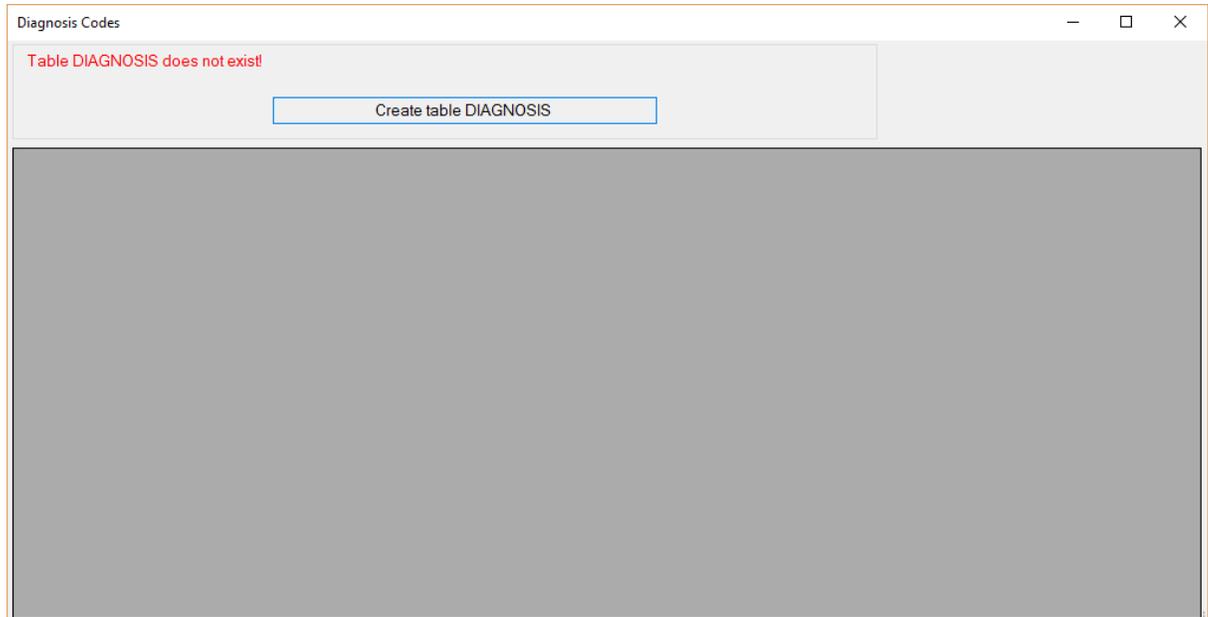
1. Select *DataBase* ► *Code Look-up Tables* in the main menu.



The Code Look-up Tables menu option from the main menu bar

Alternately, a specific codeset's editor can be opened by clicking the appropriate Procedure, Diagnostic or Revenue Code option from either the *Code Sets* menu or the *Claim Entry/Edit* ▶ *Claim Entry/Edit CodeSets* menu option.

2. The Code Set edit screen will appear.



The "Diagnosis Code" window

3. Initially the table does not exist in your database. To create the table, click the "Create table DIAGNOSIS" button.

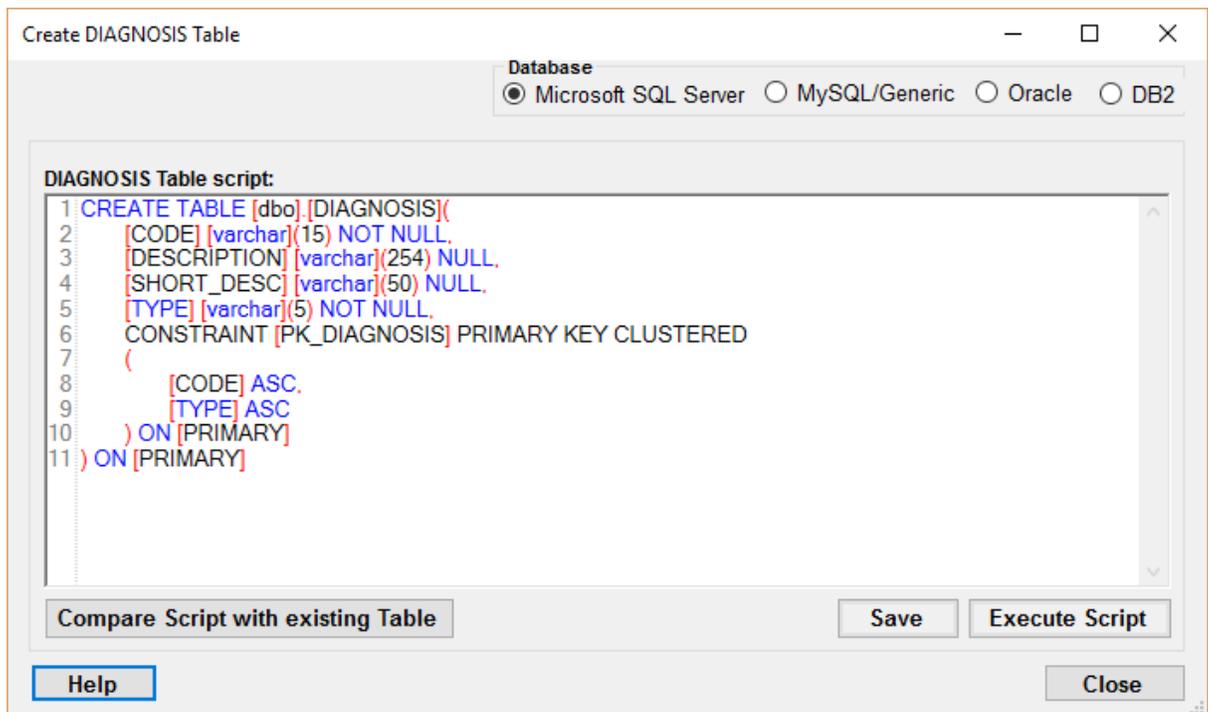
Diagnosis Codes

Table DIAGNOSIS does not exist!

Create table DIAGNOSIS

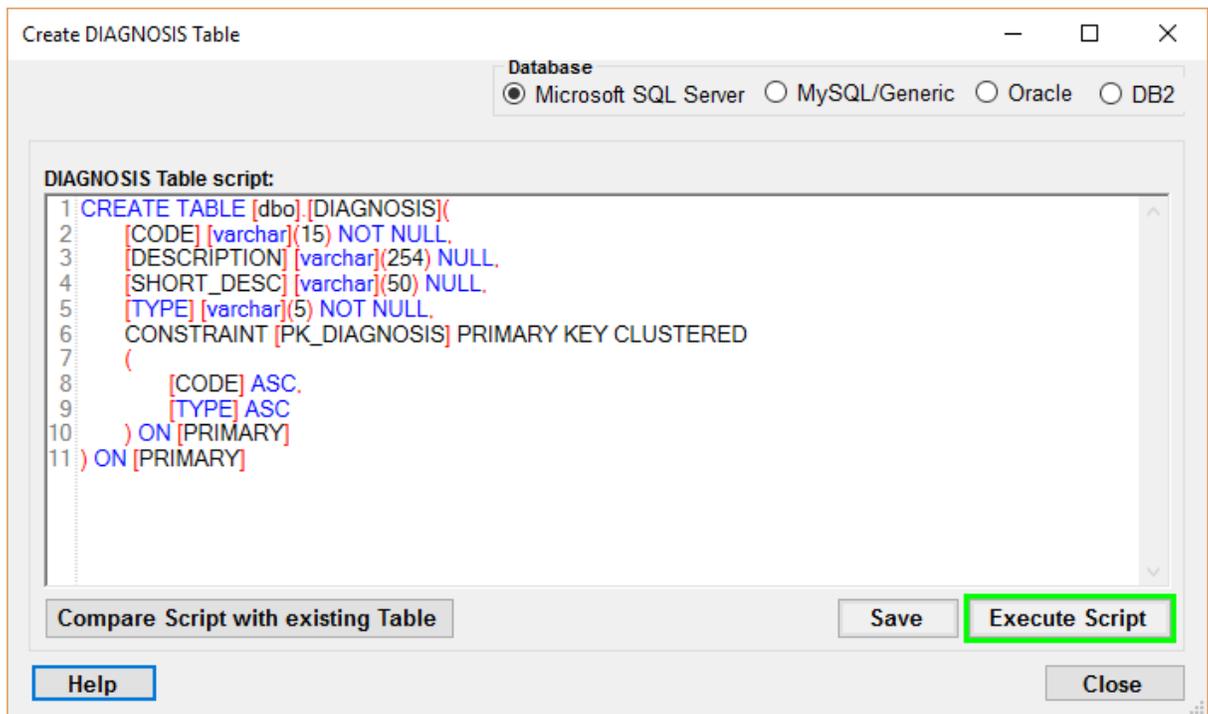
The "Create table DIAGNOSIS" button

4. A table creation window will be displayed.



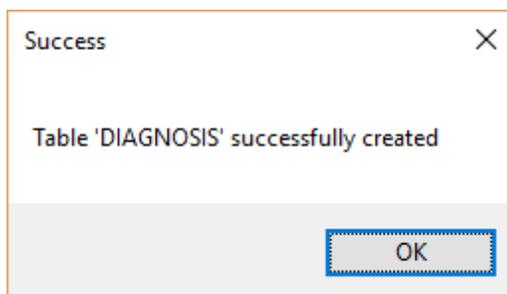
The "Create DIAGNOSIS Table" window

5. Select the appropriate database and click the "Execute script" button and the table will be created.



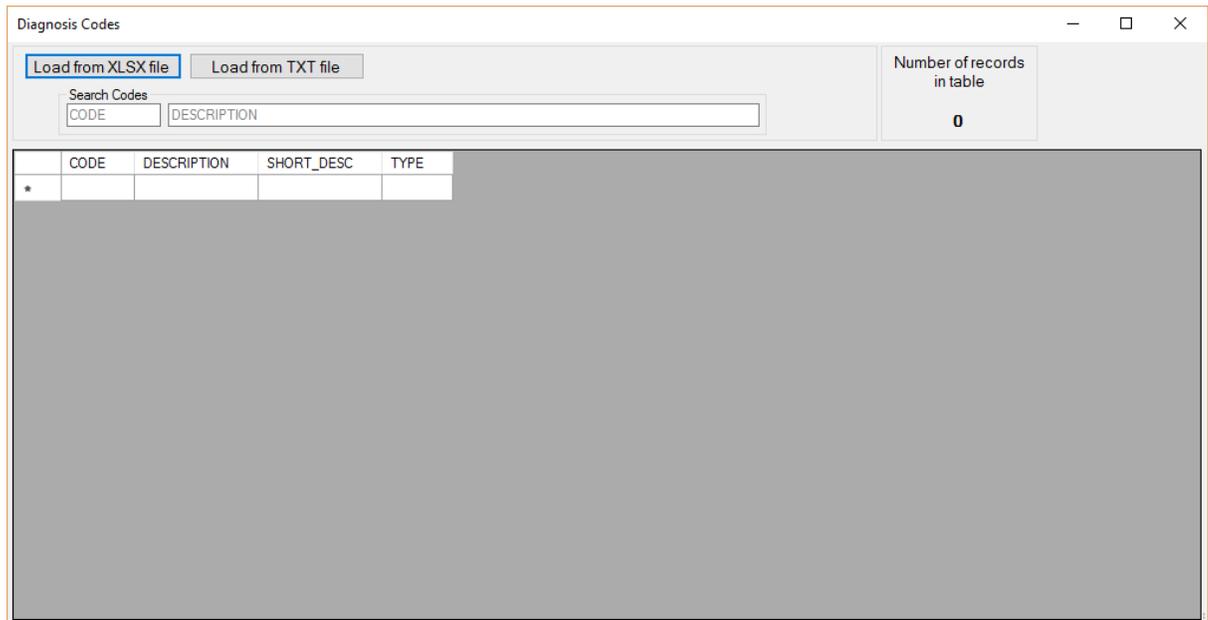
The "Create DIAGNOSIS Table" with the "Execute" button highlighted

6. You will receive a success message like this. Click OK.



The success popup

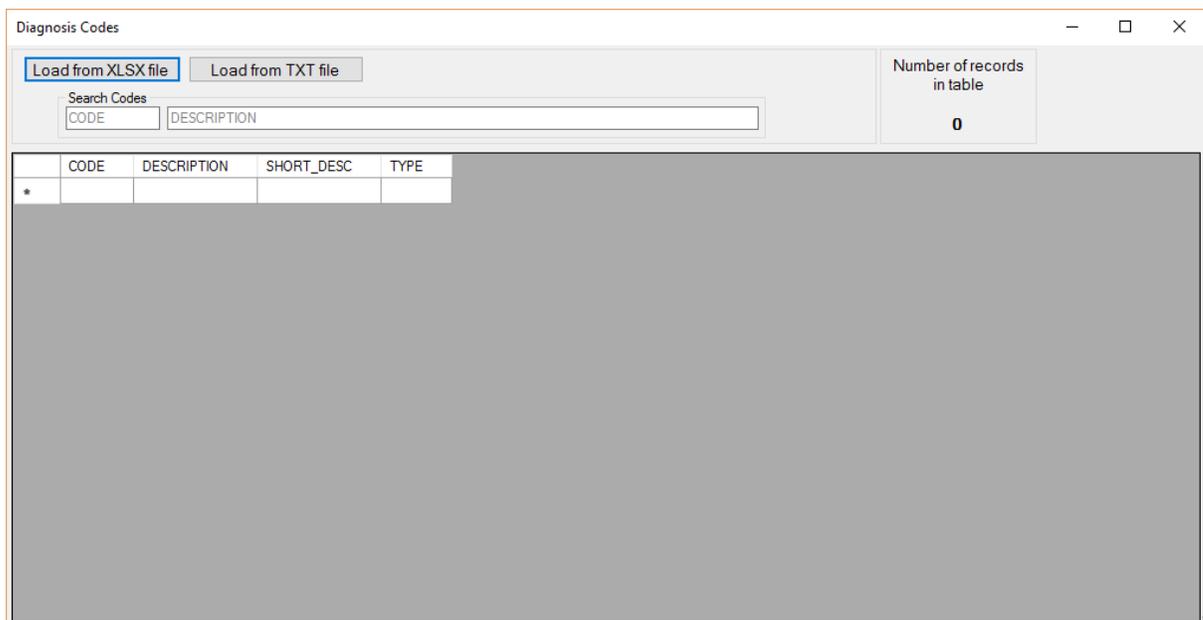
7. Once the table has been created, you are redirected back to the "Diagnosis Codes" window. It looks different now and indicates that the newly created table exists.



The "Diagnosis Codes" window with the empty table

8.9 Filling CodeSet Tables

Loading a Codeset Table can be done using a .xlsx spreadsheet, comma separated value (csv) textfile, or space delimited textfile. The following examples assume you have already opened a codeset's edit screen and will use the Diagnostic Codes codeset as an example.



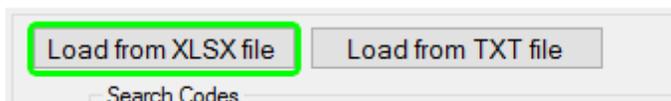
The Diagnosis Codes codeset editing window.

Tip: The diagnosis codes according to the International classification of Diseases (ICD) in the 9th revision are published by Ingenix or the AMA for example. You can also download them from the [CDC-National Center for Health Statistics](#).

Loading from XLSX

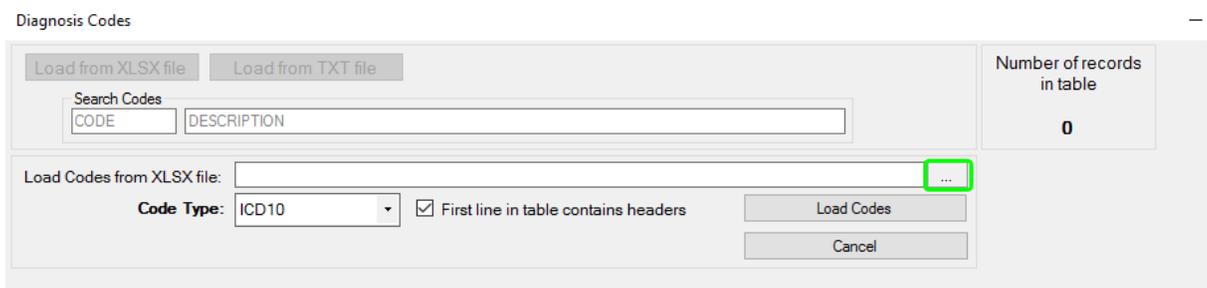
1. Click the *Load from XLSX file* button.

Diagnosis Codes



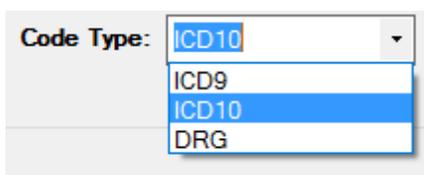
The "Load ICD-9 Codes" button

2. Select an XLSX file to use by writing its full filepath or clicking the file selection button.



File selection button

3. Select a code type if applicable.



Code type selection for diagnosis codes.

4. Click the *Load Codes* button.

Loading from Text File

There are two ways to determine column position from a text file in order to build a table: with a delimiter such as a comma or semicolon in a character delimited file or knowing the each column's position in a fixed width file.

1. Click the *Load from TXT file* button.

Diagnosis Codes

Load from XLSX file Load from TXT file

Search Codes

Click the "Load ICD9 Codes" button.

2. Select a filetype to load the codesets from. Select *File with positions* if the codeset text file you wish to save to the database is a fixed width file or *File with delimiters* if the codeset file is character-delimited.

Load from TXT file

File with positions

File with delimiters

Text file type selection.

- a. **File with positions / fixed width** - Select fixed width file that you downloaded or have available on CD/DVD.

Diagnosis Codes

Load from XLSX file Load from TXT file

Search Codes

CODE

Load Codes from text file: ...

Code Type: ICD10

First line in file contains headers

Codes: from position 7 to position 14

Long Description: from position 78 to position to end of line

Short Description: from position 17 to position 77 to end of line

Try One Line Load Codes Cancel

The layout of the text file is usually the code in position 1 through 12, and the description in position 13 through 44. If the first line contains headers, you can step over the headers by checking the box labeled *First line in file contains headers*.

Diagnosis Codes

Load from XLSX file Load from TXT file

Search Codes

CODE

Load Codes from text file: ...

Code Type: ICD10

First line in file contains headers

Codes: from position 7 to position 14

Long Description: from position 78 to position to end of line

Short Description: from position 17 to position 77 to end of line

Try One Line Load Codes Cancel

Clicking *Try One Line* will open a message box with a single example row from the text file.

Click *Load Codes* to fill the database table.

- b. **File with delimiters / character delimited** - Select fixed width file that you downloaded or have available on CD/DVD.

Diagnosis Codes

The delimited text file is usually delimited by a comma. The layout usually contains the codes in field 1 and full description in field 3. If the text file includes the column headers, check the box labeled *First line in file contains headers*.

Diagnosis Codes

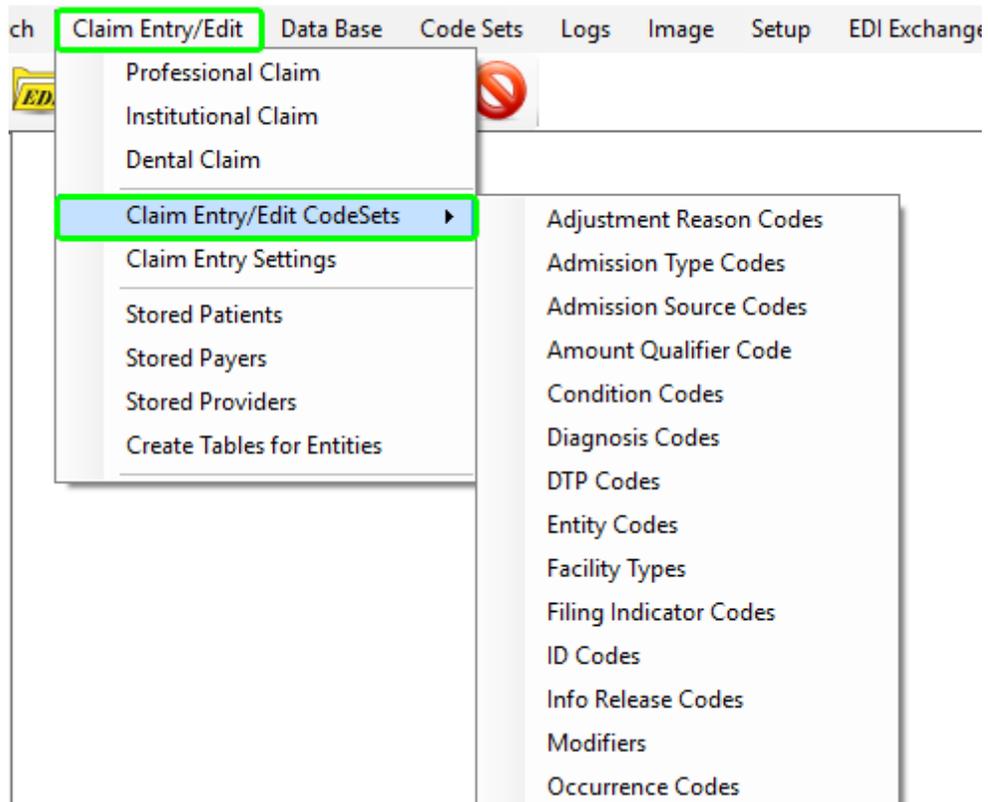
Clicking the *Try One Line* button will open a message box with a single row from the text file as an example.

Click the *Load Codes* button to fill the database table.

8.10 Viewing and Editing CodeSets

To view/edit code sets in HIPAA Claim Master, follow the instructions below.

1. Go to *Entry/Edit* ► *Claim Entry/Edit CodeSets* in the main menu.



The additional code sets that are required for Claim Entry/Edit

Alternatively, you can access the codesets setup from the Claim entry/edit window. Click the *Codesets* button. Read more in [Using the Claim Entry/Edit Function](#).

The "Codesets" button on the "Professional Claim" window

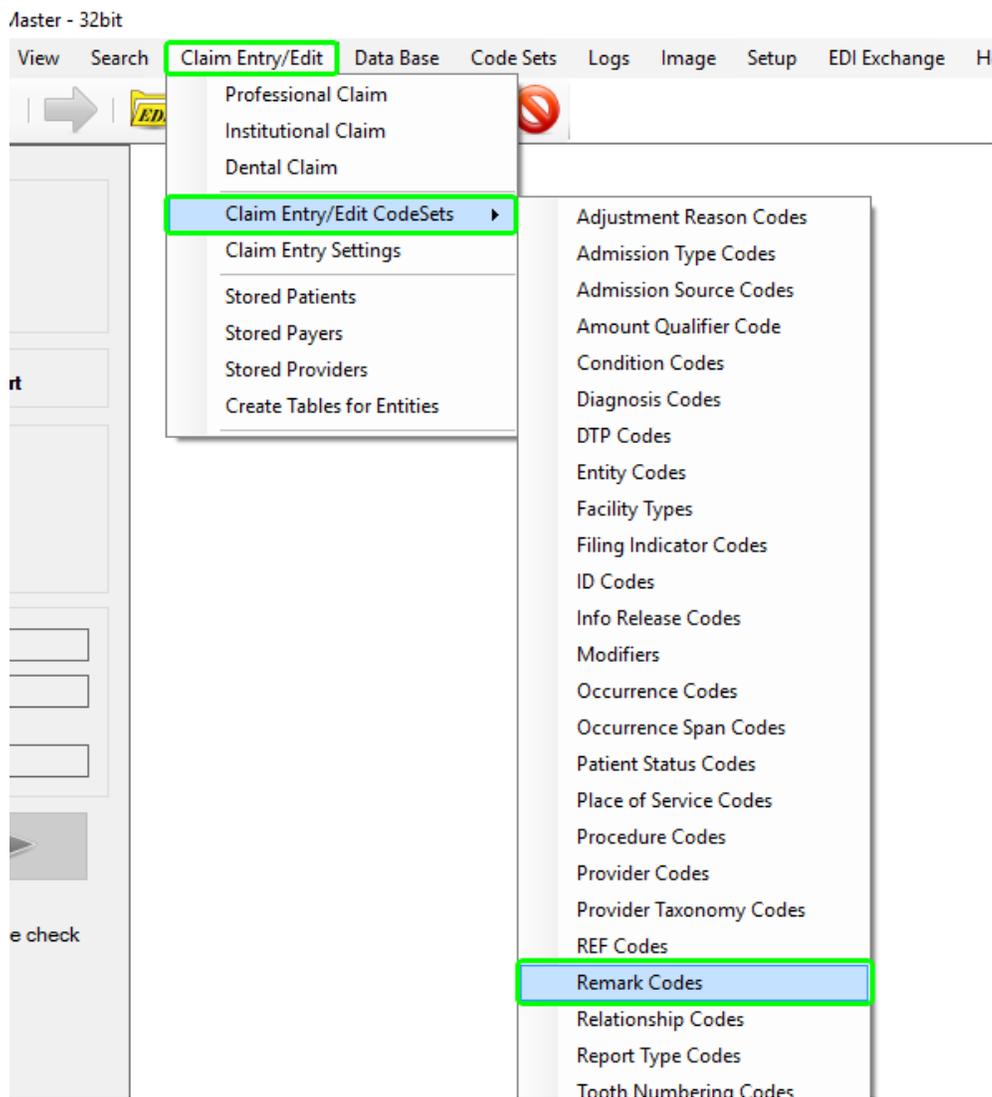
2. The Code Sets - From the sub-menu, select a code set name. The following menu items are available via the "Code Sets" menu:

- Adjustment Reason Codes
- Admission Type Codes
- Admission Source Codes
- Amount Qualifier Code
- Condition Codes
- Diagnosis Codes – See [Creating Codeset Tables](#)
- DTP Codes
- Facility Types
- Filing Indicator Codes
- Info Release Codes
- Occurrence Codes
- Occurrence Span Codes

- Patient Status Codes
- Place of Service Codes
- Procedure Codes - See [Creating Codeset Tables](#)
- Provider Taxonomy Codes
- REF Codes
- Remark Codes
- Relationship Codes
- Report Type Codes
- Revenue Codes - See [Creating Codeset Tables](#)
- Tooth Numbering Codes
- Treatment Codes
- ZIP Codes - See [Creating Codeset Tables](#)

Diagnosis, Procedure, Revenue and ZIP code sets are too large to store in memory. They can be edited like any other codeset but the database tables to store them are created separately.

3. To select, for example, the "Remark Codes" codeset for display, click the *Claim Entry / Edit* menu item and from the *Claim Entry/Edit CodeSets* dropdown menu options select *Remark Codes*:



The "Claim Entry/Edit CodeSets"

4. The Code Set configuration window for the selected codeset, in this case Remark Codes, will displayed. You can add additional codes by typing it into the blank row at the end of the list or edit the existing ones by double-clicking the field to edit it and type in the new value. Click the Save button to save the changes.

Claim Edit/Entry Codeset: Remark Codes

Claim Edit/Entry Codeset: Remark Codes

Save Close Help

Number of records in codeset
842

	Code	Description
▶	3D	OBGYN
	M1	X-ray not taken within the past 12 months or near enough to the start of treatment.
	M2	Not paid separately when the patient is an inpatient.
	M3	Equipment is the same or similar to equipment already being used.
	M4	This is the last monthly installment payment for this durable medical equipment.
	M5	Monthly rental payments can continue until the earlier of the 15th month from the first re...
	M6	You must furnish and service this item for as long as the patient continues to need it. We ...
	M7	No rental payments after the item is purchased, or after the total of issued rental paymen...
	M8	We do not accept blood gas tests results when the test was conducted by a medical sup...
	M9	This is the tenth rental month. You must offer the patient the choice of changing the rent...
	M10	Equipment purchases are limited to the first or the tenth month of medical necessity.
	M11	DME, orthotics and prosthetics must be billed to the DME carrier who services the patie...
	M12	Diagnostic tests performed by a physician must indicate whether purchased services are...
	M13	No more than one initial visit may be covered per specialty per medical group. Visit may ...
	M14	No separate payment for an injection administered during an office visit, and no paymen...
	M15	Separately billed services/tests have been bundled as they are considered components ...
	M16	Please see the letter or bulletin of (date) for further information.

The "Remark Codes" code set configuration window

As mentioned above, the procedure, diagnosis, revenue and ZIP codes are saved to a database. These four codesets have a slightly different codeset editing screen (shown below) but they behave in the same way. Since these codesets can grow very large, search boxes are provided to filter through codes or descriptions.

CODE	DESCRIPTION	SHORT_DESC	TYPE
0001	Therapeutic ultrasound of vessels of head and neck	Ther ult head & neck ves	ICD9
0001T	ENDOVASCULAR REPAIR OF INFREARENAL ABDOMINAL AORTIC ANEURYSM OR DISSECTION; MODULA...	ENDOVS REP INFRARNL AAA/DISCTN	CPT
0002	Therapeutic ultrasound of heart	Ther ultrasound of heart	ICD9
0002T	ENDOVASCULAR REPAIR OF INFREARENAL ABDOMINAL AORTIC ANEURYSM OR DISSECTION; AORTO...	ENDOVS REP INFRARNL AAA/DISCTN	CPT
0003	Therapeutic ultrasound of peripheral vascular vessels	Ther ult peripheral ves	ICD9
0003T	CERVICOGRAPHY	CERVICOGRAPHY	CPT
0005T	TRANSCATHETER PLACEMENT OF EXTRACRANIAL CEREBROVASCULAR ARTERY STENT(S). PERCUTA...	TRANSCATHETER PLACEMENT	CPT
0006T	TRANSCATHETER PLACEMENT OF EXTRACRANIAL CEREBROVASCULAR ARTERY STENT(S). PERCUTA...	TRANSCATHETER PLACEMENT	CPT
0007T	TRANSCATHETER PLACEMENT OF EXTRACRANIAL CEREBROVASCULAR ARTERY STENT(S). PERCUTA...	TRANSCATHETER PLACEMENT	CPT
0008T	UPPER GASTROINTESTINAL ENDOSCOPY INCLUDING ESOPHAGUS, STOMACH, AND EITHER THE DUO...	UGI ENDOSCOPY	CPT
0009	Other therapeutic ultrasound	Other therapeutic ultsnd	ICD9
0009T	ENDOMETRIAL CRYOABLATION WITH ULTRASONIC GUIDANCE	ENDOMETRIAL CRYOABLATION	CPT
001	Central Nervous System, Bypass	Central Nervous System, Bypass	ICD10
0010	Implantation of chemotherapeutic agent	Implant chemothera agent	ICD9
00100	ANESTHESIA FOR PROCEDURES ON SALIVARY GLANDS, INCLUDING BIOPSY	ANESTH, SALIVARY GLAND	CPT
00102	ANESTHESIA FOR PROCEDURES ON PLASTIC REPAIR OF CLEFT LIP	ANESTH, REPAIR OF CLEFT LIP	CPT

The slightly different "Procedure Codes" code set configuration window.

The codeset windows provide the following information:

- **Number of records in codeset** – The total number of codes stored in the database.
- **Codes and their corresponding descriptions.** A short description, standards version (ICD), city and state (ZIP) may also be included, depending on the loaded code table. The codes are indicated in the source EDI files. The descriptions are displayed instead of codes in the result easy-to-read files. For example, for Remark Codes there are the following codes and corresponding descriptions:
 - 3D - OBGYN
 - M1 - X-ray not taken within the past 12 months or near enough to the start of treatment.
 - M2 - Not paid separately when the patient is an inpatient.
 - M3 - Equipment is the same or similar to equipment already being used.
 - etc.

4. Click on the "Save" button to save the changes. Changes to the procedure, diagnosis, revenue and ZIP codes are saved automatically to the database.

Chapter

IX

9 Creating 837 EDI Files

9.1 About Creating EDI Files

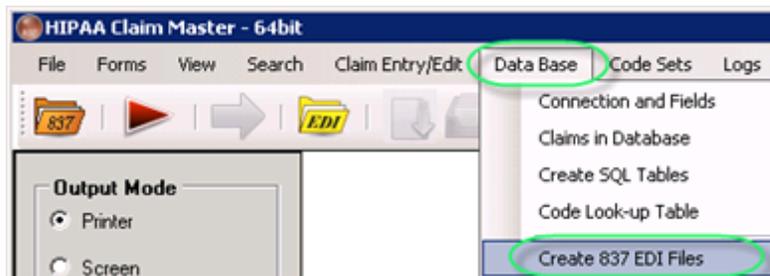
With the version 4.1, the creation of EDI files from database records was introduced. This is a powerful and important addition to the capabilities of the HIPAA Claim Master. Now you can create EDI files from your claim records whether they were imported into the database via EDI files or originated from your own claim system. This is an add-on component of the HIPAA Claim Master and needs to be licensed in order to use it.

The licensing of the database integration component is required to use this feature.

Note: HIPAA compliant EDI files are not easily created from records that don't originate from EDI. It might take a while to produce compliant records for several reasons:

- You might not have every field that is required by the implementation guides
- The data that you have is not formatted according to the HIPAA guides
- Data might have to be assembled from across your enterprise's applications

We recommend that you also purchase and employ the EDI Exchange module. This way you can check each individual claim record before it is written to the EDI file and you can even suppress records that are not created without compliance warnings.



The menu item to create EDI files under the "Data Base" menu

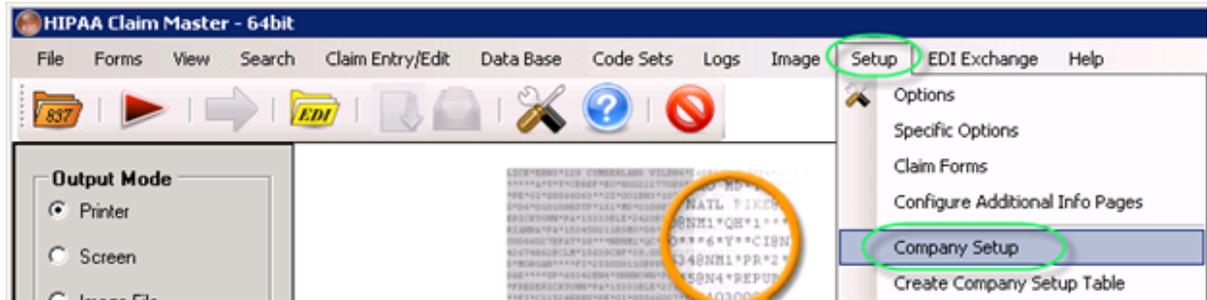
9.2 Setting up Company

Creating EDI files will include a lot of information on you and your trading partners. The company setup screen is designed to enter all the necessary information for the EDI creation. When you create EDI files there are a few elements needed to identify your company as the sender, the producer of this EDI file. For this matter, use the company setup screen. You can specify your company information and other setting related to your company there.

If you have the EDI Exchange component licensed then all this information is stored in the data file "COMPANY_SETUP". If you don't have this component then this information is stored in the Windows registry.

Note: You need to configure the database first. See [Setting up Database Connection](#).

1. To configure the company, select *Setup* ▶ *Company Setup* in the main menu.



The "Company Setup" menu

2. The following screen will appear.

The "Company Setup" window

3. You can configure the following options:

- Company Information
- EDI Version
- Contact Information
- ISA 14 and 15

Read the detailed descriptions further.

4. Once you have finished editing the options, click "Save."

Company Information

- **Name** – Company name. Obligatory option.
- **Address 1**
- **Address 2**
- **City** – Obligatory option.
- **State** – Obligatory option.
- **ZIP**
- **Plus 4**
- **ISA Segment Sender Identification** – It is the most important option to choose. It identifies your organization for all EDI transactions. Most commonly the Qualifier "ZZ" and a free form ID up to 15 character in length is chosen. Other options include your tax ID and the Dun & Bradstreet ID.
- **Qualifier** – The ISA Identifier and qualifier are the unique key to the trading partner database file. The ISA identifier can be up to 15 bytes long, the qualifier has to be 2 bytes. Approved qualifiers are:
 - 01 – Duns (Dun & Bradstreet)
 - 14 – Duns Plus Suffix
 - 20 – Health Industry Number (HIN)
 - 27 – Carrier ID Number (Identification Number as assigned by HCFA)
 - 28 – Fiscal Intermediary ID Number (Identification Number as assigned by HCFA)
 - 29 – Medicare Provider and Supplier ID Number (Identification Number as assigned by HCFA)
 - 30 – U.S. Federal Tax ID Number (Identification Number)
 - 33 – NAIC Company Code (National Association of Insurance Commissioners Company Code)

- **ZZ – Mutually Defined**
- **Application Sender's Code GS_2** – The option represents the program or machine that created the file. Usually the same as the ISA Segment Sender Identification. You can select other ID if you need. Obligatory option. The GS_02 identifier is usually the same as the ISA identifier but you can also choose some other ID. It represents the program or machine that created the file, but the most common is a repetition of the ISA identifier.
- **Tax ID** – Obligatory option. Also called an Employer Identification Number (EIN).
- **Plan ID / Payor ID** – Fill in your Payer ID and Plan ID if you have one; this is only important for the 271 and 277 transactions. Optional setting.
- **Three letter identifier to prepend to all outgoing EDI files** – Optional setting. Makes your files easily identifiable.

EDI version

Specify EDI version:

- **only 4010**
- **4010 if in original otherwise 5010**
- **only 5010**

Contact Information

The contact information goes into the PER segment of outgoing transactions:

- **Contact Person**
- **Telephone**
- **Ext**
- **Fax**
- **E-mail**

ISA 14 and 15

- **EDI files will be** – ISA element 14 determines if the resultant file contains test or

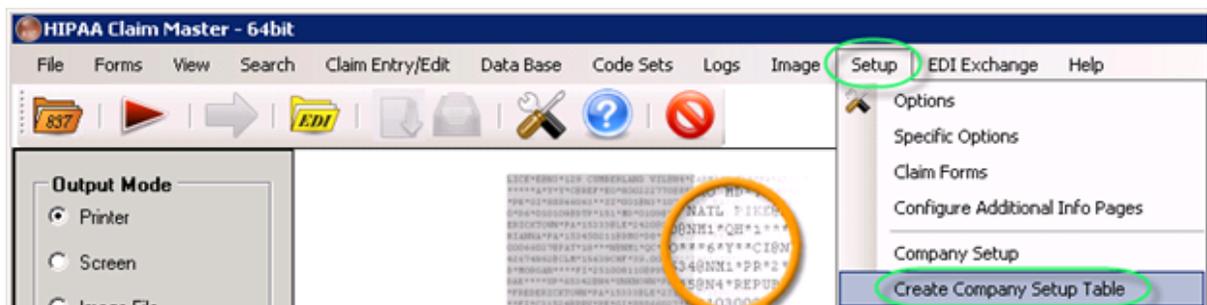
production data:

- Test
- Production
- **Acknowledgement (TA1) req.** – ISA element 15 indicates whether you want a TA1 acknowledgement for files that you send out.

9.3 Creating Company Setup Table

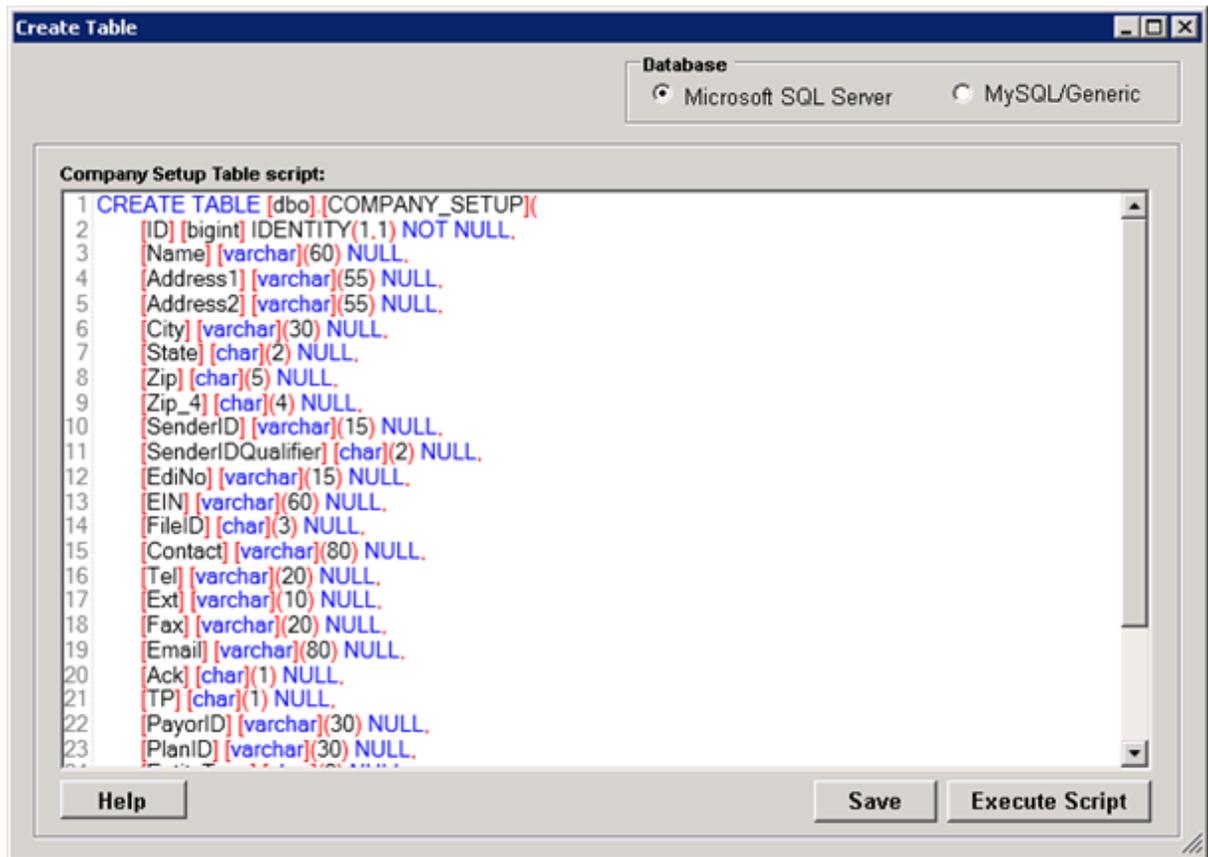
You can create the Company Setup Table by following the steps below in the HIPAA Claim Master.

1. To configure the program options, select *Setup* ▶ *Create Company Setup Table* in the main menu.



The "Create Company Setup Table" menu

2. The following screen will appear.

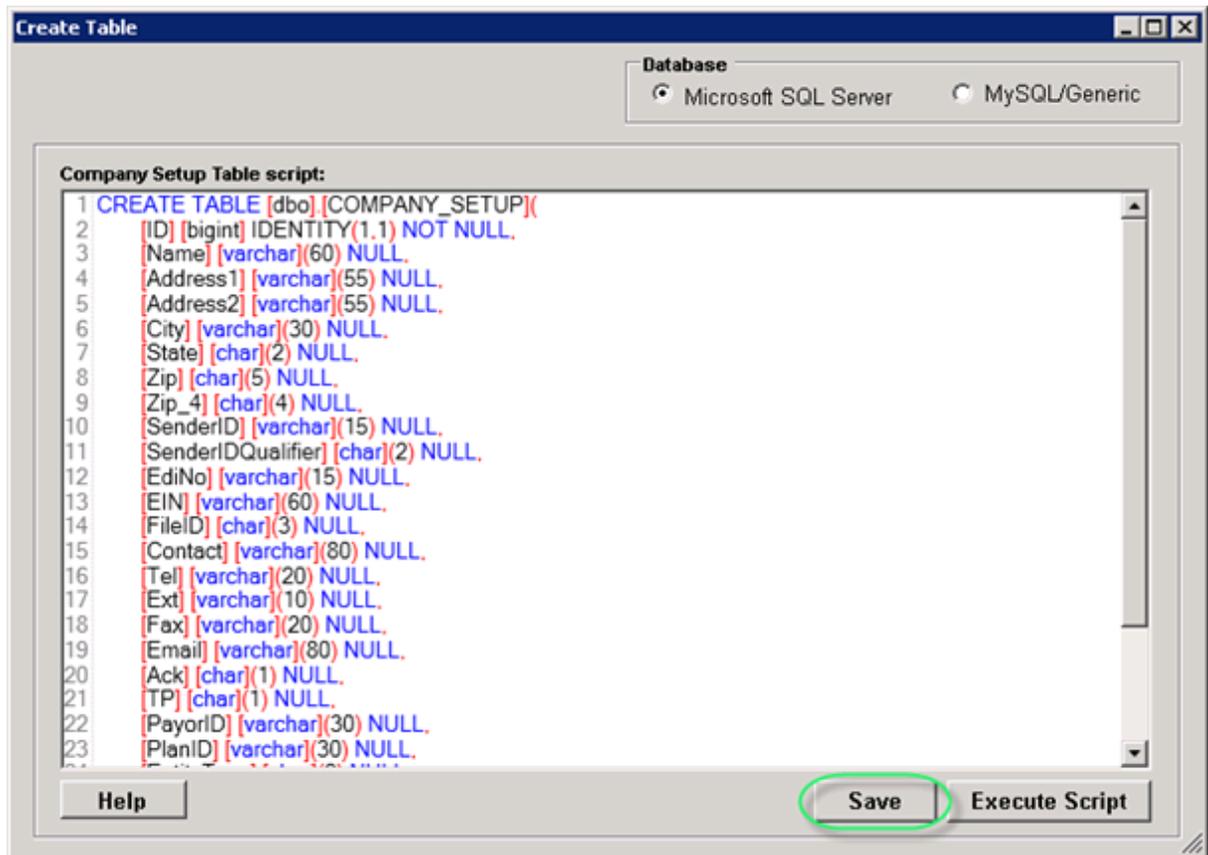


"Create Table" window

3. This window allows you to edit, save and run the table scripts.
4. Define the following options:
 - Database
 - Microsoft SQL Server
 - MySQL/Generic
 - Company Setup Table Script – SQL statements to create the Company table in your database.

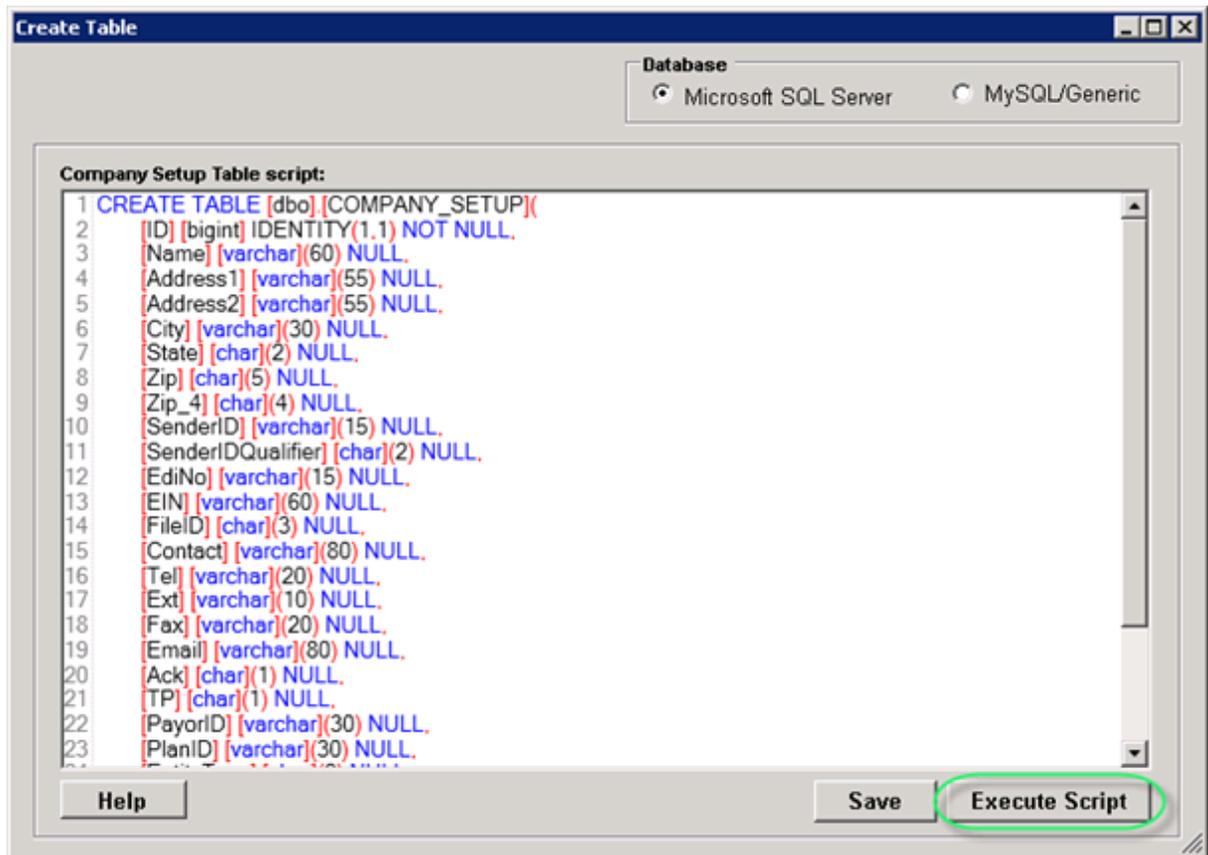
Tip: You can modify the scripts so that they run on your specific database.

5. Once you have you modified the script, click "Save."



The "Save" button

6. Click "Execute Script" to create the table in the database.



The "Execute Script" button

7. Once the table has been created successfully, you will see the following notification:



The success message

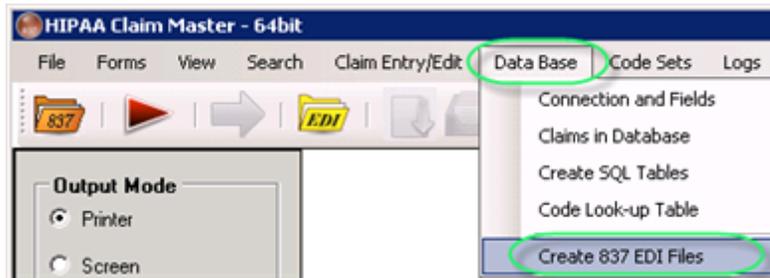
Warning: Executing the script will wipe out any previous table you have created. Remove the script files once you are satisfied so nobody can occasionally destroy the database table.

Make sure there are no any error messages for successful table creation.

9.4 Creating 834 EDI Files from the Attached Database

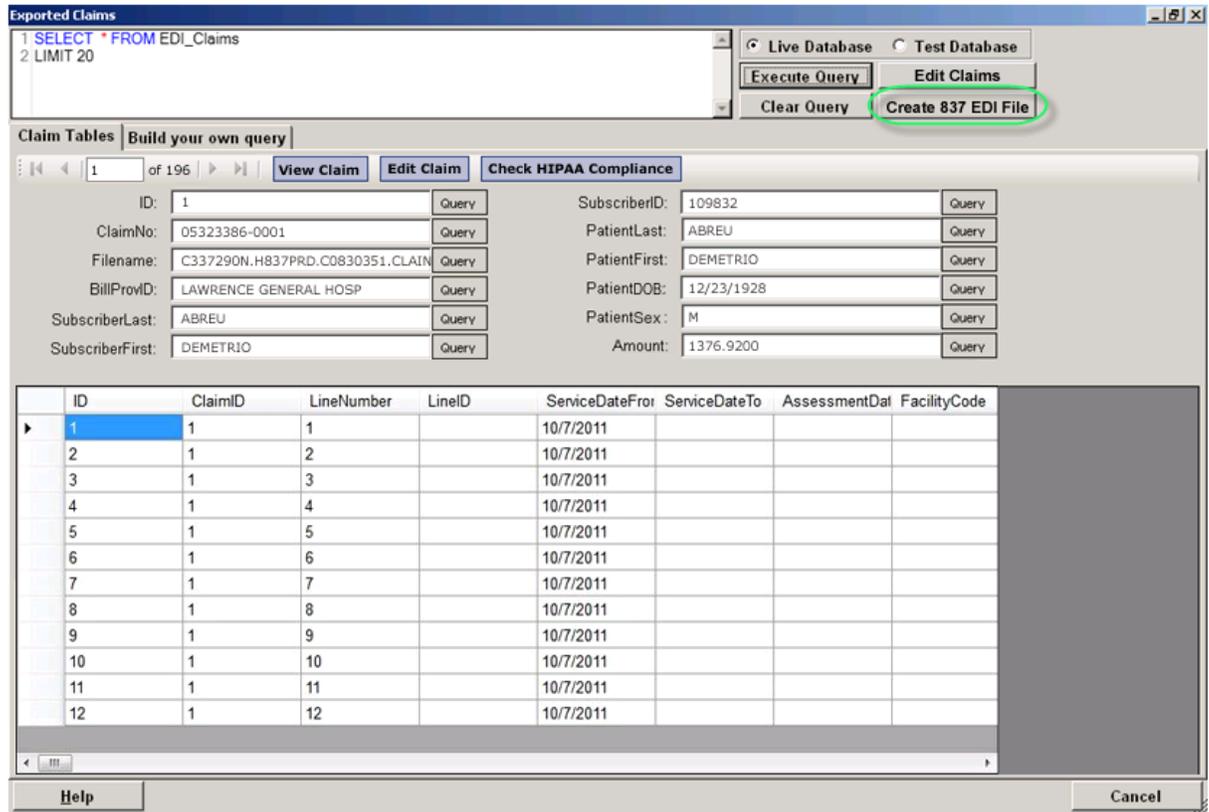
Once the [EDI creation](#) is enabled, you can create the EDI files. Follow the instructions below.

1. To configure the company, select *Data Base* ▶ *Create 837 EDI Files* in the main menu.



The menu item to create EDI files under the "Data Base" menu

Alternatively, you can click the "Create 837 EDI File" button in the query builder ("Exported Claims" window). Refer to [Exporting the Data](#).



The "Create 837 EDI File" button located on the "Exported Claims" screen

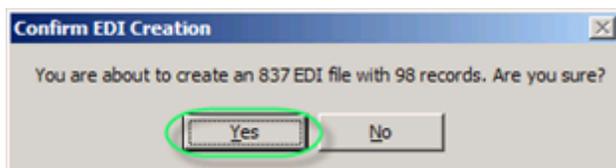
2. Clicking the button will bring up the "Create 837 Transaction Sets" window with the query that you designed.

The fields are described in [Understanding the "Create 837 Transaction Sets" Window](#).

3. After you have filled all necessary values, click "Go!"

The "Create 837 Transaction Sets" window with the highlighted "Go!" button

4. You will see a confirmation dialog like this. Click "Yes."



The verification message before the EDI records are created

5. After you click "Yes," the process of assembling the claims to the 837 files starts. You can see the progress bar on the bottom pane.



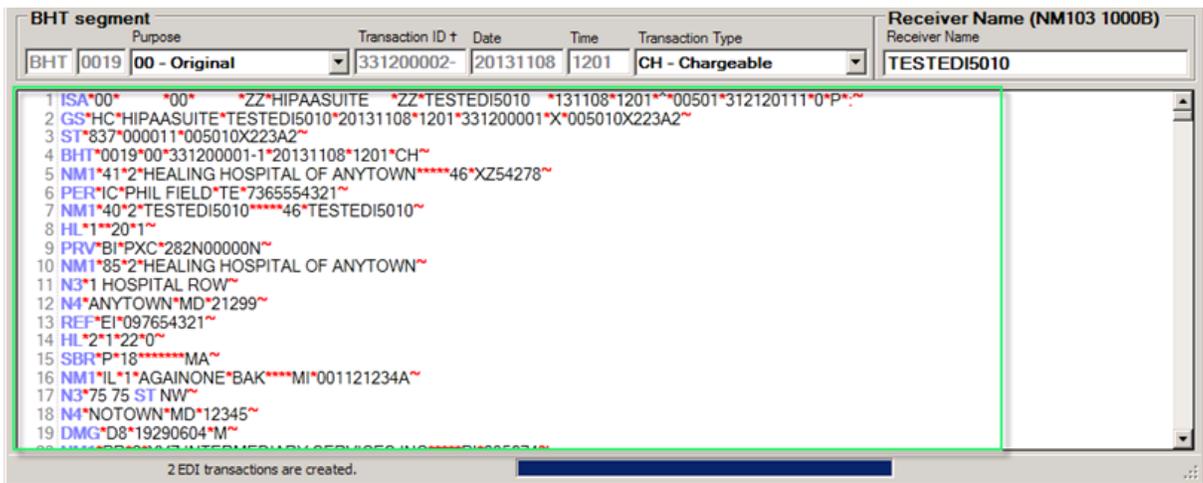
Progress bar at the bottom of the screen keeps you informed about the process

6. The bottom pane informs you about the number of transactions created.



The bottom pane displaying the number of created EDI transactions

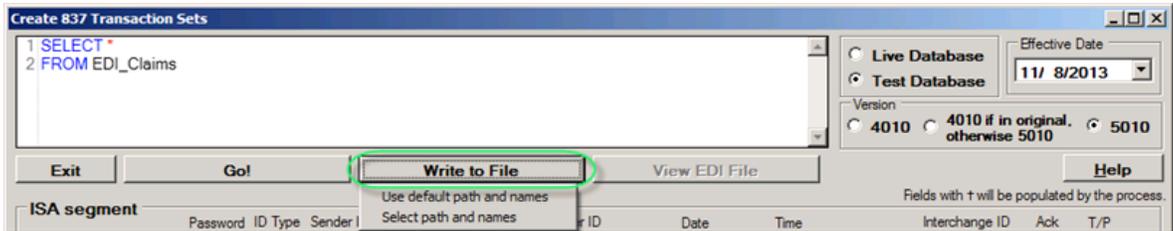
7. The EDI text is now displayed in the bottom part of the window.



The EDI file content displayed

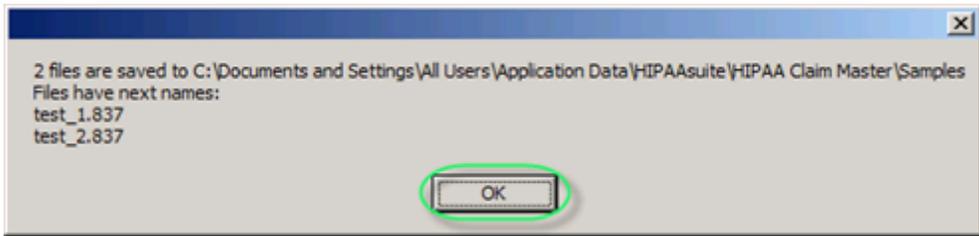
8. Now you can write down the EDI text that is still in memory and visible in the text box to a file. Click on the "Write to File" button and you will have the choice to write in the default location which is the trading partner's outbox, or specify any other location:

- Use default path and names
- Select path and names



The "Write to File" button

9. After successful writing, you get a message where the file was written to.



The confirmation message that the file was successfully written

Viewing the Saved Files

The button "View EDI File" is now enabled and you can bring up the EDI Editor to view it.

Create 837 Transaction Sets

1 SELECT *
2 FROM EDI_Claims

Live Database
 Test Database

Effective Date: 11/ 8/2013

Version: 4010 4010 if in original, otherwise 5010 5010

Exit Go! Write to File **View EDI File** Help

Fields with † will be populated by the process.

ISA segment

ISA	00	00	ZZ	HIPAAASUITE	ZZ	TESTEDI5010	131108	1206	00501	312120647	0	P
test_1_1.837												
test_2_1.837												

GS segment

GS	H	HIPAAASUITE	TESTEDI5010	20131108	1206	331200002	X	005010X222A1

ST segment

ST	837	000021	005010X222A

BHT segment

BHT	0019	00 - Original	331200002-	20131108	1206	CH - Chargeable

Receiver Name (NM103 1000B)

Receiver Name: TESTEDI5010

```

1 ISA*00*00*ZZ*HIPAAASUITE*ZZ*TESTEDI5010*131108*1206*00501*312120647*0*P*~
2 GS*HC*HIPAAASUITE*TESTEDI5010*20131108*1206*331200001*X*005010X223A2~
3 ST*837*000011*005010X223A2~
4 BHT*0019*00*331200001-1*20131108*1206*CH~
5 NM1*41*2*HEALING HOSPITAL OF ANYTOWN*****46*XZ54278~
6 PER*IC*PHIL FIELD*TE*7365554321~
7 NM1*40*2*TESTEDI5010*****46*TESTEDI5010~
8 HL*1*20*1~
9 PRV*BI*PXC*282N00000N~
10 NM1*85*2*HEALING HOSPITAL OF ANYTOWN~
11 N3*1 HOSPITAL ROW~
12 N4*ANYTOWN*MD*21299~
13 REF*EI*097654321~
14 HL*2*1*22*0~
15 SBR*P*18*****MA~
16 NM1*IL*1*AGAINONE*BAK****MI*001121234A~
17 N3*75 75 ST NW~
18 N4*NOTOWN*MD*12345~
19 DMG*D8*19290604*M~
  
```

2 EDI transactions are created.

The "View EDI File" button highlighted

The following preview window will open.

- **Query field** – This field is located in the top left corner. It is used to narrow the records to be written to the EDI files. You can create your own query or you can use the query builder in the [exported claims](#) screen.
- **Live/Test Database radio buttons** – They are located to the right of the query text box. They allow to choose between the Live database and the Test database as outlined in the [data connection](#) setup.
- **Effective date** – It is the effective date of the file. By default, the current date is selected.
- **Version** – Here you can select the version of the resulting EDI file. You have the choice of 4010 or 5010. It is generally possible to generate 4010 files from claims that came in 5010 files and vice versa, a third option is for claims that came from EDI files and that have been imported into the database. Here you have the option to preserve the original version so that a claim that came in an EDI file in 4010 format will also be output as 4010 files while a 5010 file will be output as 5010.

EDI Envelopes

The next part of the screen contains the EDI envelopes:

- **ISA Segment** – The ISA segment contains the outermost envelope of the EDI file and it represents data that is required to read an EDI file. Listed here are the sender and receiver, the data and time, whether a response is expected and which characters to use as segment, element and sub-element separator. The picture above shows this screen and you can see the individual data elements. The data in black you can change, other elements are determined by the Company Setup screen.
- **GS Segment** – The GS or Group Start segment contains transaction specific information, which transaction sets are in this file and also has information on the creation (Sender code) and consumption (Receiver Code) of the EDI files. Group control number and version will be filled in by the process since this is dependent on the claim type.
- **ST Segment** – The ST or Start of Transaction Set segment again lists the transaction set code and a unique identifier for the transaction.
- **BHT Segment** – The BHT segment further qualifies the claim in the 837 transaction set by specifying whether this is an original submission and whether the claims are chargeable or just encounter records.

If you have the EDI Exchange component licensed, you can select the trading partner and

fill the form with a specific trading partner's information.

- **Select Trading Partner** – Once this button has been clicked, a popup window appears where you can select your defined trading partner.

Chapter



10 Creating 277CA EDI Files

10.1 What are 277CA files

The CA stands for Claim Acknowledgment. The 277CA was formerly known as the unsolicited claim status response. This transaction is used to tell the sender of a claim in detail whether the claim was rejected, if there is additional information necessary and maybe even if the claim is already adjudicated. This transaction is unsolicited, meaning that the trading partner did not send a 276 status request to receive this 277CA. This transaction is usually sent by the receiver after a 837 claim file came in and has been pre-processed.

10.2 277CA in the HIPAA Claim Master

The creation of the 277CA is a feature in the HIPAA Claim Master that has to be licensed and enabled. If you see the menu items for the 277CA disabled, please contact HIPAAsuite's sales department for a quote.

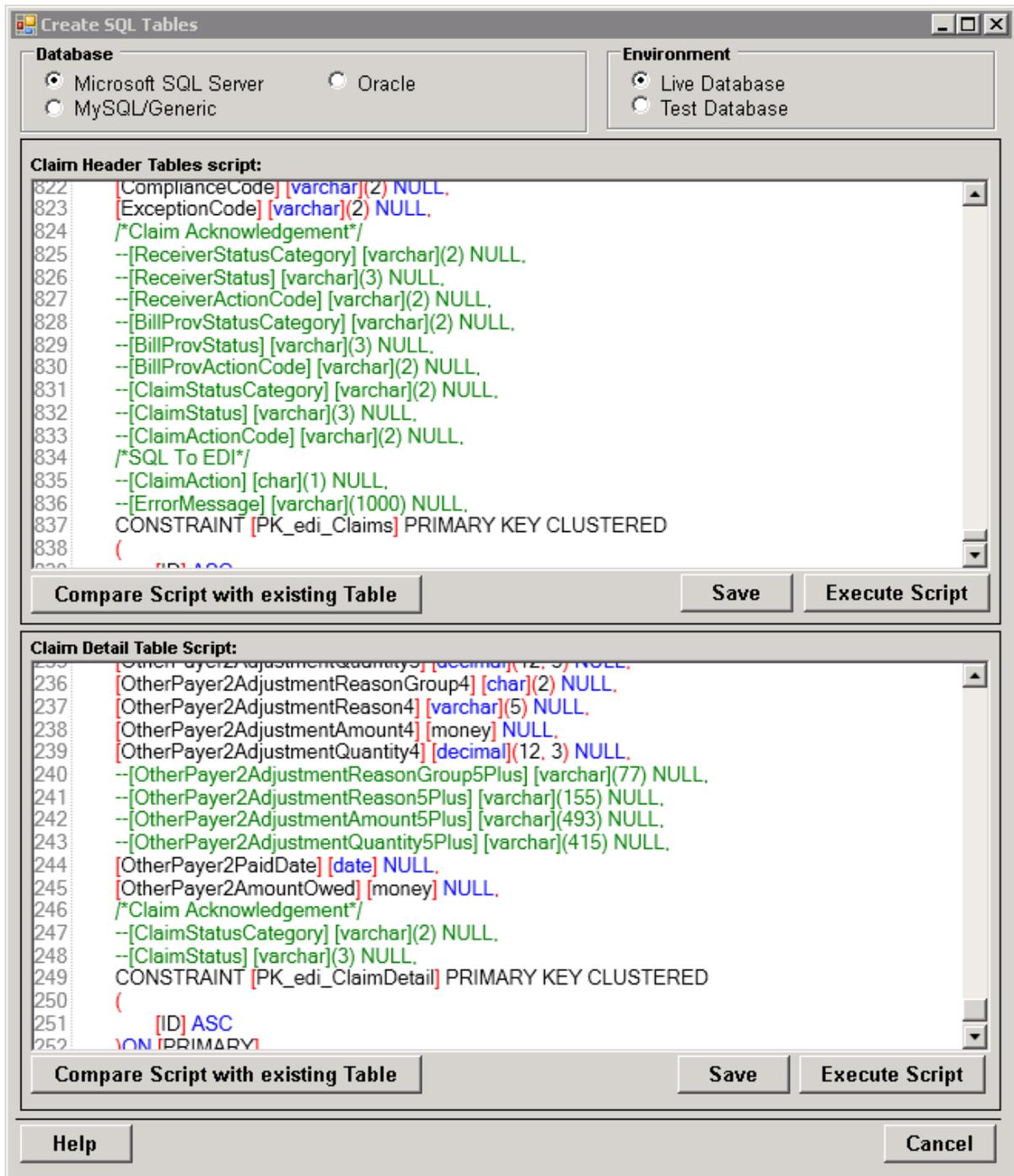
The HIPAA Claim Master allows two mechanism to create the 277CA.

1. When a file is processed and the trading partner is set up to receive the 277CA, then after each claim is parsed from the 837, a 277CA record is created and filled with a standard claim status and status category that is determined in the 277CA setup screen
2. The claims are exported to the database and a separate process enters values in so-called feed back fields with the results of processing or even adjudication. Then triggered by a query a 277CA record can be created for each claim and packaged into a file to be send back to the claim originator.

The second method is much more precise since it returns actual process data while the first process has only a canned message to convey.

These feed-back fields can also be used by the HIPAA Claim Status Responder to return the status of a claim by finding it in the claim tables and then reading the status from the feed-back fields.

The feedback fields are included in the scripts to create the EDI_Claims and EDI_ClaimDetail tables but they are commented out. If you open the screen to [create the tables](#) and scroll both scripts down to the bottom you will see the commented out area of claim acknowledgment fields.

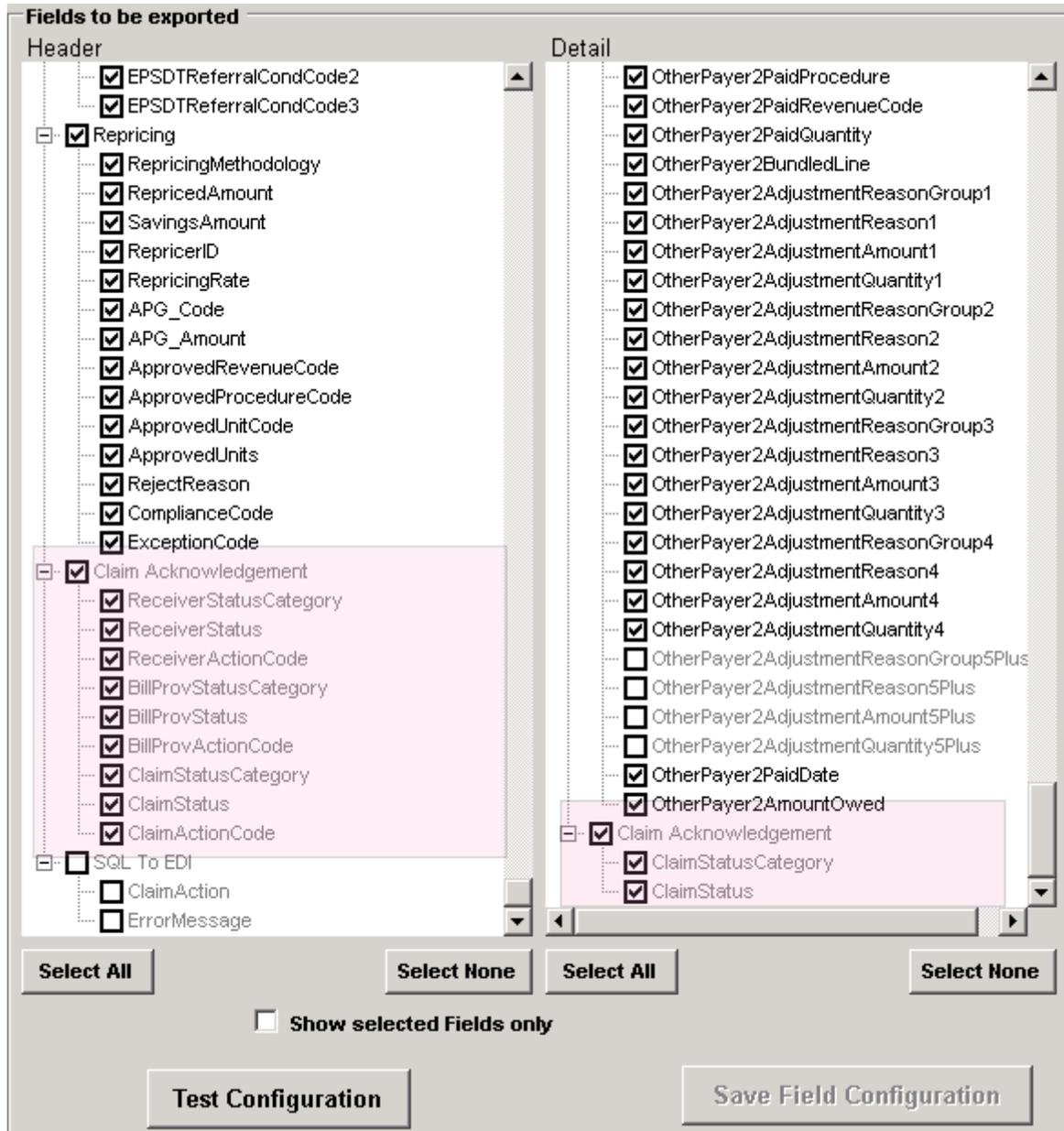


The feedback fields in the SQL scripts

To create the fields you have to uncomment those rows and then save the script. Then you can click on the button "Compare Scripts with existing Table" and the fields will be added to your data tables.

Now you still have to select those fields. For that you go into the screen to set up the

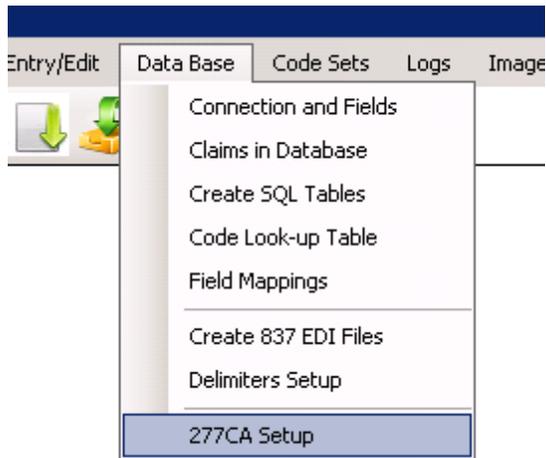
[database connection and fields](#) and again scroll to the bottom of the field lists and check all the fields in the Claim Acknowledgment section.



The feed back fields checked so they are included in the processing.

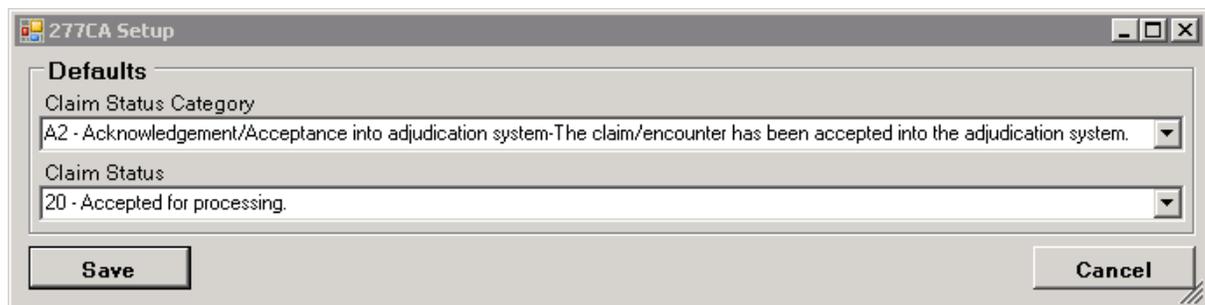
10.3 Setting up the Auto generated 277CA

The Database menu item has now a sub menu item "277CA Setup".



The menu for the 277CA Setup screen

When we click on it the following screen comes up:



The 277CA Setup screen

Here we set which are the standard messages for claim status after the 837 file has been received.

The default values are:

for the Claim Status Category: A2- Acknowledgment/Acceptance into adjudication system. - The claim/encounter has been accepted into the adjudication system

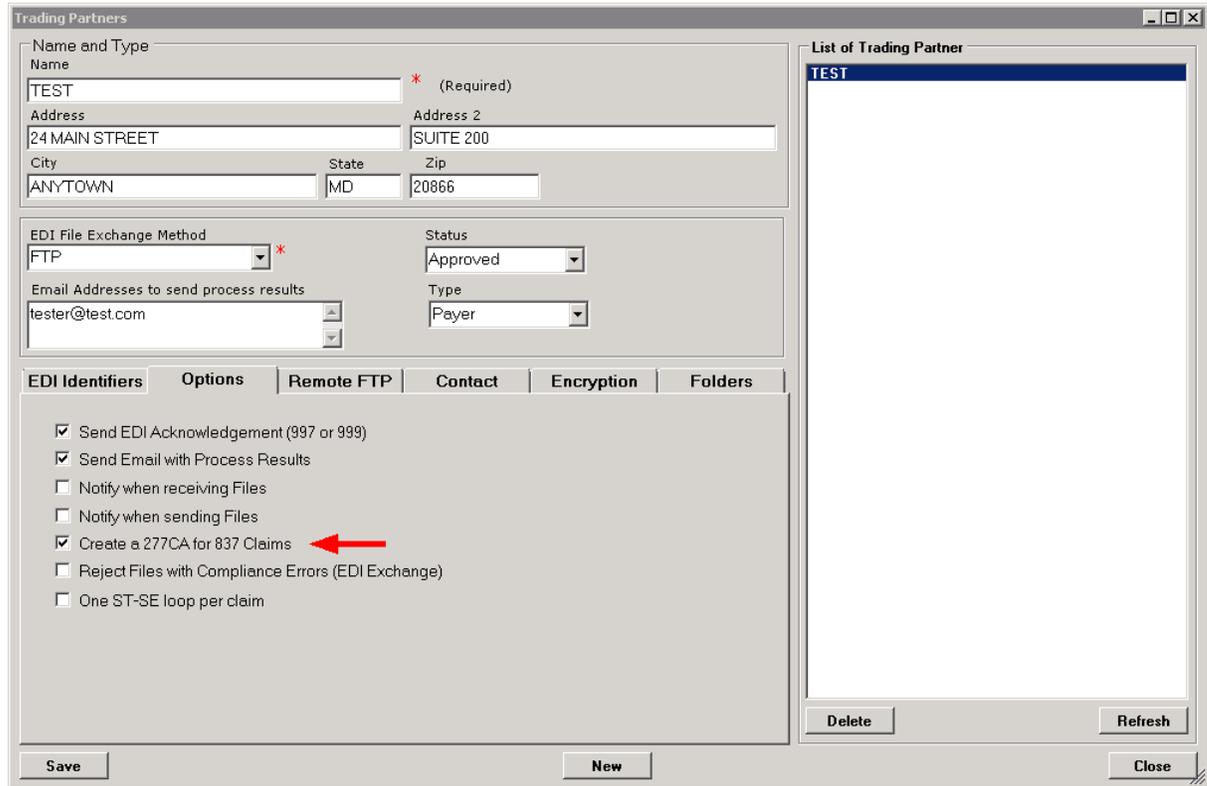
for the Claim Status: 20-Accepted for processing.

10.4 Setting up trading partner for 277

The creation of the 277CA is usually instigated by the trading partner who sends the 837 claim file. Though the 277CA is not part of the original HIPAA transactions it is eyed very

positively as an important part of lowering the costs of healthcare.

In order to set up a trading partner to receive the 277CA we open the trading partner screen and select the tab with 'Options'



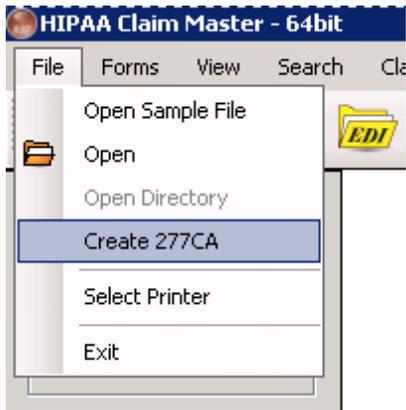
The trading partner setup with the option to generate 277CA's

Here we see the option to 'Create a 277CA for 837 claims'. Once the option is checked the 277CA will be generated every time a 837 EDI file from this trading partner is processed and send back according to the settings in the EDI file Exchange Method.

10.5 Creating a 277CA file manually

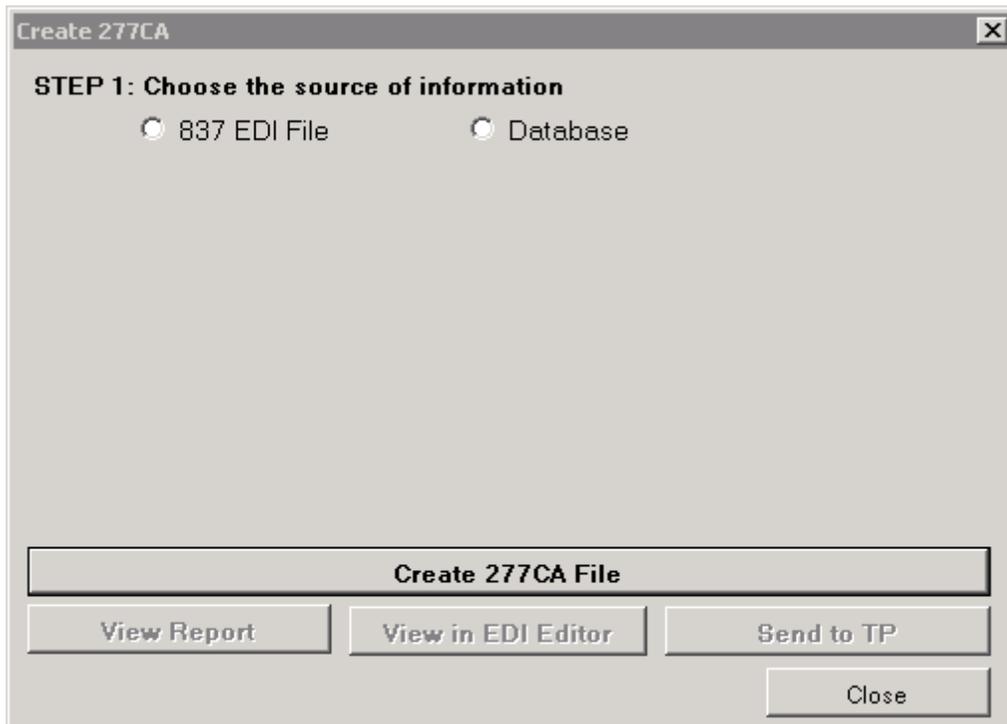
The HIPAA Claim Master can create 277CA's manually.

For this we go to the File menu and select "Create 277CA"



The menu to create 277CA's manually

We will see the following screen come up



Step 1 in the screen to create 277CA's manually

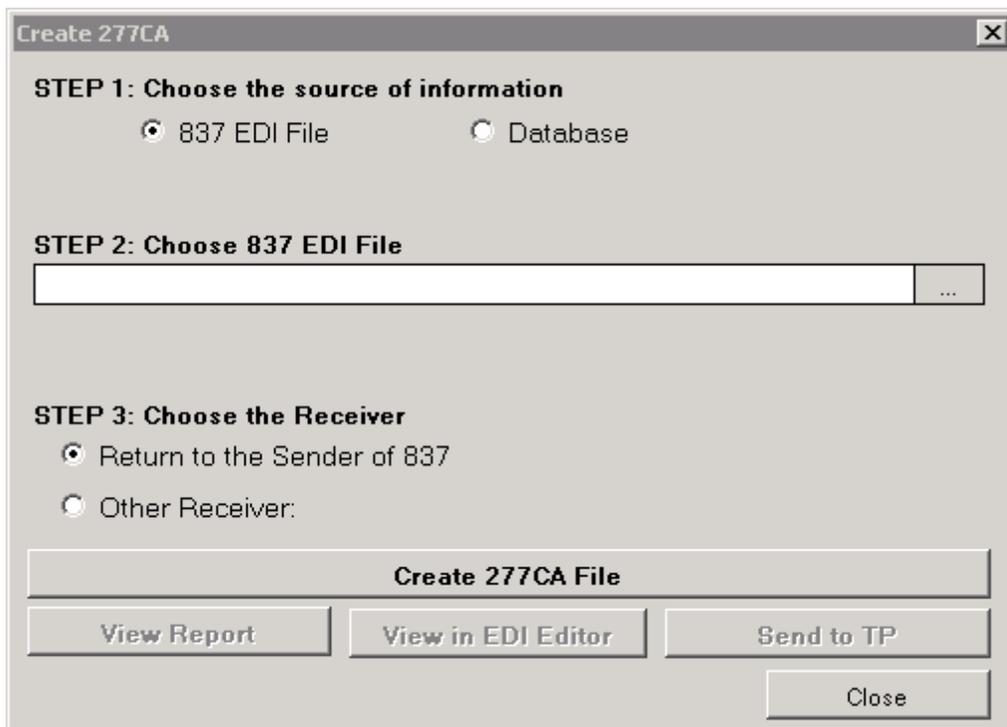
Step 1 on this form is to choose the source for the 277CA creation.

1. you select an 837 claim file and create a 277CA file from the claims it contains. The status that is assigned comes from default status values that are set in the [Setup](#) screen.
2. Database records in the claim tables. The status information comes from the feed-

back fields in the database

10.5.1 From file

We now select the 837 EDI file option



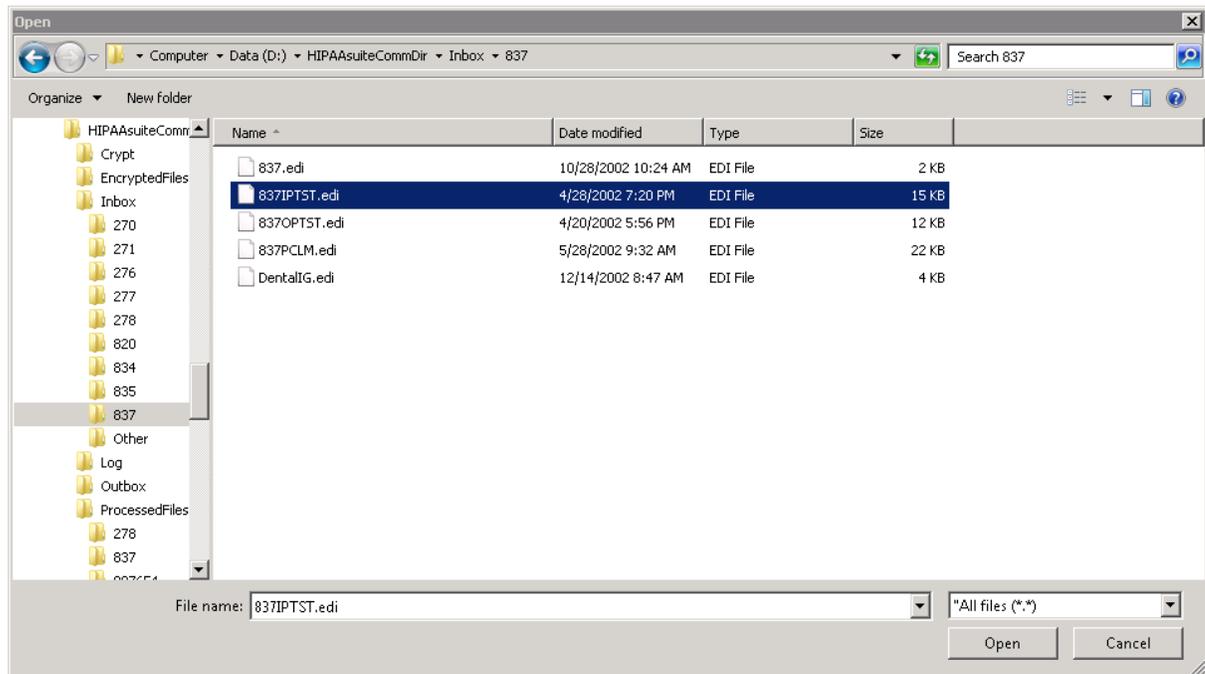
The screenshot shows a dialog box titled "Create 277CA" with a close button (X) in the top right corner. The dialog is divided into three steps:

- STEP 1: Choose the source of information**
 - 837 EDI File
 - Database
- STEP 2: Choose 837 EDI File**
 - A text input field with a browse button (three dots) to the right.
- STEP 3: Choose the Receiver**
 - Return to the Sender of 837
 - Other Receiver:

At the bottom of the dialog, there are four buttons: "Create 277CA File", "View Report", "View in EDI Editor", and "Send to TP". A "Close" button is located in the bottom right corner.

Step 2 in the process to generate a 277CA from an 837 EDI file

and now the Step 2 is revealed, we have to select the file with the Windows file chooser



Selecting an 837 EDI file for processing

And now Step 3 comes forward and we have to choose the receiver of the 277CA file. We have two options

1. to send the 277CA file back to where the 837 came from, this means the ISA and GS

STEP 3: Choose the Receiver

- Return to the Sender of 837
 Other Receiver:

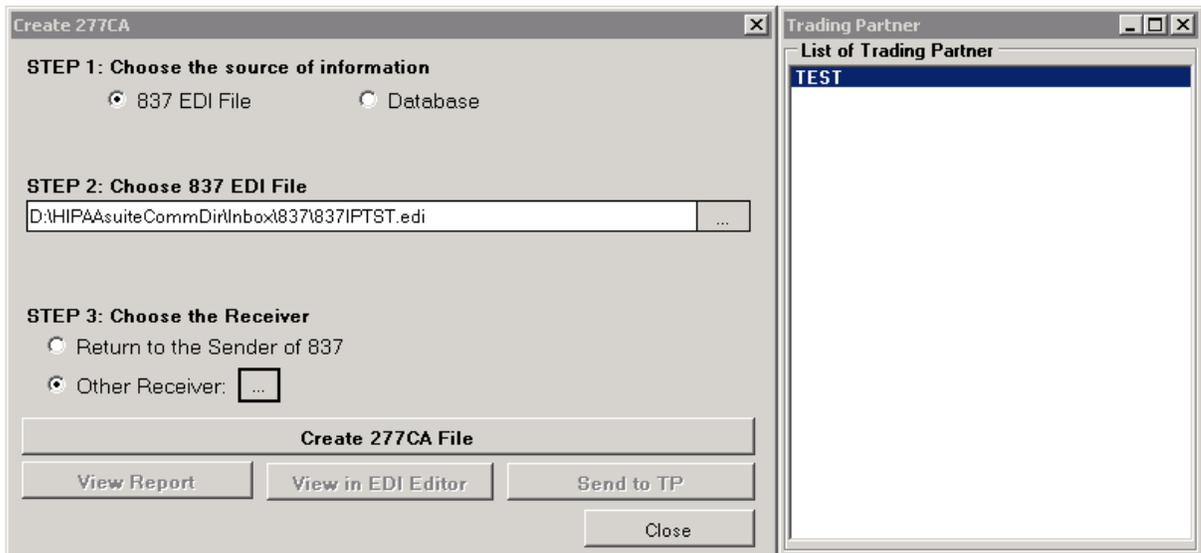
headers are reversed or

2. to choose a trading partner from the list of trading partners in the database.

STEP 3: Choose the Receiver

- Return to the Sender of 837
 Other Receiver:

If we click on the button next to the 'Other Receiver' option a list comes up on the right side of the screen and we can choose the trading partner.



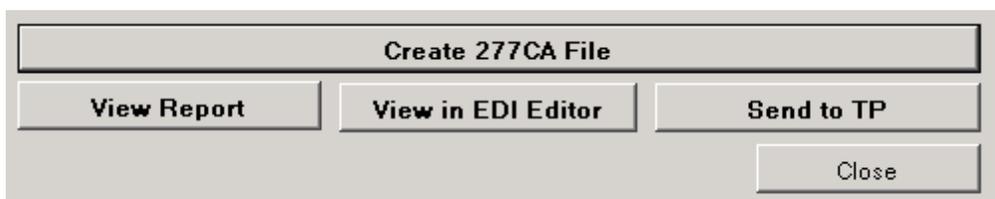
Selecting the trading partner who will receive the 277CA

Once we choose the desired trading partner, the button 'Create 277CA File' is enabled. We press it and the following process result window will show us where the system created the file.



Process Results after creating the file

We close this window and have now several options enabled



Further processing options are now enabled

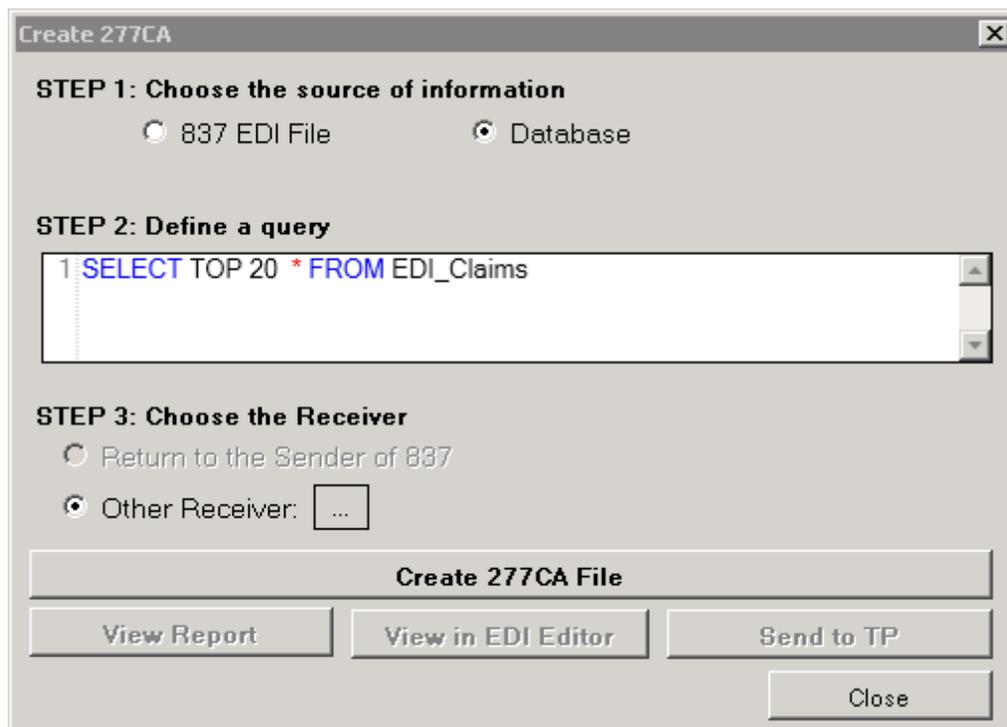
View Report We can view the 277 as printable report

View in EDI Editor: We see the file displayed in the EDI Editor

Send to TP: Send the file and if the trading partner has FTP or other transports selected and configured the file will be sent to the trading partner and a success message is visible.

10.5.2 From Database

If we choose the Database options we will see the form change to this



The screenshot shows a dialog box titled "Create 277CA" with a close button (X) in the top right corner. The dialog is divided into three steps:

- STEP 1: Choose the source of information**
 - 837 EDI File
 - Database
- STEP 2: Define a query**
 - A text box contains the SQL query: `1 SELECT TOP 20 * FROM EDI_Claims`
- STEP 3: Choose the Receiver**
 - Return to the Sender of 837
 - Other Receiver:

At the bottom of the dialog, there are four buttons: "Create 277CA File", "View Report", "View in EDI Editor", and "Send to TP". A "Close" button is located in the bottom right corner.

Step 2 is revealed, entering the SQL query to select the claims.

Now we see a text box where we can enter a SQL query that will be passed on to the database. Through your query will you have full control over the records that you select. The option to send the records back to the sender is in this case disabled. There is no

guarantee that database claims are from the same trading partner so we disabled this option. You will have to choose the trading partner by clicking on the button with the three dots and the screen to choose a trading partner attaches next to this form.

The screenshot shows two windows. The main window is titled 'Create 277CA' and has three steps:

- STEP 1: Choose the source of information**
 - 837 EDI File
 - Database
- STEP 2: Define a query**
 - Text box containing: `1 SELECT TOP 20 * FROM EDI_Claims`
- STEP 3: Choose the Receiver**
 - Return to the Sender of 837
 - Other Receiver:

Buttons at the bottom of the 'Create 277CA' window: 'Create 277CA File', 'View Report', 'View in EDI Editor', 'Send to TP', and 'Close'.

The 'Trading Partner' window is open to the right, showing a 'List of Trading Partner' with one entry: 'TEST'.

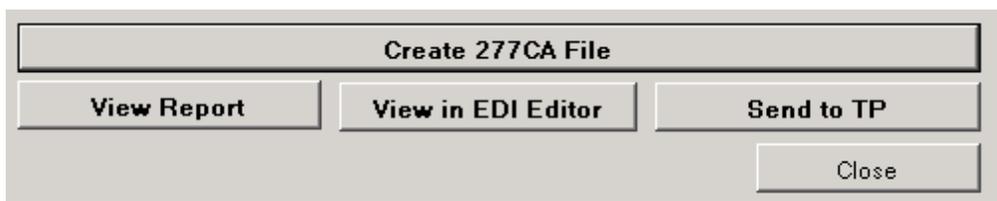
Selecting the trading partner to receive the 277CA

Once we choose the desired trading partner, the button 'Create 277CA File' is enabled. We press it and the following process result window will show us where the system created the file.



The process results after generating the 277CA

We close this window and have now several options enabled



Options after the file is created

View Report We can view the 277 as printable report

View in EDI Editor: We see the file displayed in the EDI Editor

Send to TP: Send the file and if the trading partner has FTP or other transports selected and configured the file will be sent to the trading partner and a success message is visible.

10.5.3 From the 'Claim in Database' screen

The previous chapter explained how to create a 277CA from the database via the special screen to create a 277CA manually. We can arrive at this screen in a second way.

If we open the menu Database --> Claims in Database we see a button 'Create 277CA'.

The screenshot shows the 'Exported Claims' interface. At the top, a query editor contains the text: `1 SELECT TOP 20 * FROM EDI_Claims`. To the right of the query editor are buttons for 'Execute Query', 'Edit Claims', 'Clear Query', 'Create 837 EDI File', and 'Create 277CA File'. A red arrow points to the 'Create 277CA File' button. Below the query editor, there are navigation buttons: 'View Claim', 'Edit Claim', and 'Check HIPAA Compliance'. The main area displays claim details for ID 1, including fields for ClaimNo, Filename, BillProvID, Amount, SubscriberID, SubscriberLast, SubscriberFirst, PatientLast, and PatientFirst. At the bottom, there are two tables: 'Claim Header' and 'Service Line Detail'.

Claim Header:		Service Line Detail:					
Field	Value	ID	ClaimID	LineNumber	LineID	ServiceDateFrom	ServiceDateTo
ID	1	1	1	1	C1	3/15/2001	
Filename	837PCLM.edi						

The 'Claims in Database' screen

We click this button and also arrive at the screen

Create 277CA

STEP 1: Choose the source of information

837 EDI File Database

STEP 2: Define a query

1 SELECT TOP 20 * FROM EDI_Claims

STEP 3: Choose the Receiver

Return to the Sender of 837

Other Receiver: ...

Create 277CA File

View Report **View in EDI Editor** **Send to TP**

Close

The same screen shows that is accessed from the File menu

Chapter



XI

11 Automating

11.1 Running the Application from the Command Line

The HIPAA Claim Master can be used through other applications. A single 837-claim file can be read and displayed on the screen by using the HIPAA Claim Master executable with the EDI filename as an argument. This can be an important feature for a work flow application where a claim can be called up to the screen.

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
C:\>E:\HIPAA\suite\HIPAAClaimPrinter.exe F:\Claims\Inbox\837PCLM.edi_
```

Calling the HIPAA Claim Master from the DOS prompt with the Claim EDI file as argument

The result is that the claim will be displayed on the screen without the usual frame of the HIPAA Claim Master application.

The screenshot shows a 'Preview Claim' window with a 'HEALTH INSURANCE CLAIM FORM'. The form is filled out with the following information:

- Sender's EDI No:** 987654
- Claim Identification No:** 765992
- HEALTH INSURANCE CLAIM FORM**
- 1. Patient's Name (Last, First, Middle Initial):** Nosmith, Herman
- 2. Patient's Birthdate:** 12/25/1920
- 3. Patient's Sex:** M
- 4. Insured's Name (Last, First, Middle Initial):** Nosmith, Herman
- 5. Patient's Address (No., Street):** 10 9th Avenue
- 6. Patient's Relationship to Insured:** Self
- 7. Insured's Address (No., Street):** 10 9th Avenue
- 8. Patient's Status:** Single
- 9. City:** Nanachut
- 10. State:** MD
- 11. Insured's Policy Group or FECA Number:** 654621234A
- 12. ZIP Code:** 13747
- 13. Insured's Date of Birth:** 12/25/1920
- 14. Other Insured's Name (Last, First, Middle Initial):** (Blank)
- 15. Employment? (Current or Previous):** No
- 16. Insured's Date of Birth:** 12/25/1920
- 17. Other Insured's Policy or Group Number:** (Blank)
- 18. Auto Accident:** No
- 19. Other Insured's Date of Birth:** (Blank)
- 20. Employer's Name or School Name:** XYZ Carrier Services Inc
- 21. Insurance Plan Name or Program Name:** XYZ Carrier Services Inc
- 22. Is there another Health Benefit Plan?:** No
- 23. Patient's or authorized Person's Signature:** on File
- 24. Source:** Y
- 25. Date:** (Blank)
- 26. Payment Assignment Cert. Indicator:** Y
- 27. Date of current Illness, Injury, Pregnancy:** (Blank)
- 28. If Patient has had same or similar Illness Give first Date:** (Blank)
- 29. Dates Patient unable to work in current Occupation From To:** (Blank)
- 30. Name of referring Physician or other Source:** (Blank)
- 31. I.D. Number of referring Physician:** (Blank)
- 32. Hospitalization Dates related to current Services From To:** (Blank)
- 33. Outside Lab? YES NO:** NO
- 34. Diagnosis or Nature of Illness or Injury (Relate Items 1,2,3 or 4 to Item 24e by Line):** 1... 274.9
- 35. Medicoled Resubmission Code:** (Blank)
- 36. Original Ref. No.:** (Blank)
- 37. Prior Authorization Number:** (Blank)

The claim displayed on the screen without the HIPAA Claim Master program frame.

If a "P" is used as a second argument, the claim will be printed without ever being displayed.

Claim Master returns -1 if any errors occurred in the process, and 0 if there was no error messages. It is really triggered by the error logging. So, any error that would make it to the log will cause the program to return -1.

11.2 Using Command Line Arguments (CLI)

The HIPAA Claim Master can also accept command line arguments. This is a very important feature if you want to put the HIPAA Claim Master on a scheduler and automate the processing of files or directories. Refer to [Running the Application via Scheduler](#).

The following is the list of the command line arguments, they must be separated by commas.

1. The first command line argument is the **filename** or the **directory path**. If not applicable, leave blank.
2. The second argument or a combination of arguments contains the processing options.
 - **P** – Print.

```
C:\>C:\Program Files\HIPAAsuite\HIPAA Claim Master\HIPAAClaimMaster.exe C:\Test.edi,P
```

Using the "P" option as a Command Line Argument

- **T** – TIFF File. If "T" is used as an argument, the claim will be saved as a TIFF image file.

See [Creating Image Files from EDI](#) for details on how to set up the Image printing capabilities.

Be sure that you have tested the Image printing, before trying the command line mode.

```
C:\>C:\Program Files\HIPAAsuite\HIPAA Claim Master\HIPAAClaimMaster.exe C:\Test.edi,T
```

Using the "T" option as a Command Line Argument

- **F** – PDF File. If "F" is used as an argument, the claim will be saved as a PDF image. ("P" could not be used again as it is already taken by print option).

Again, be sure that you have tested the Image printing, before trying the command line mode.

- **X** – Export to database
 - **S** – Search. See examples further.
 - **Q** – Query against database. See examples further.
 - **E** – Create EDI in conjunction with **Q** you can create an EDI file, either using the default settings from the last screen or the database depending whether the module EDI Exchange is licensed.
 - **P** – If a "P" is used as a second argument, the claim will be printed without ever being displayed.
3. The third command line argument is the image output filename or the EDI output filename when you create an 837 EDI file.

If your options specify single file name, then we will have a multi-page image file.

If your options specify one file per image, then passing the image filename in the command line really only makes sense in conjunction with a query.

Search condition when you try to find one specific claim within an EDI file and create the image for it. If there are multiple matches the program creates an enumeration in the filename such as "Filename," Filename_1, Filename_2 and so on.

A SQL query against the database where you expect a limited number of results (a single record preferably).

4. The fourth argument is your Query; you should have "Q" as second argument.
5. The fifth argument is the destination trading partner that you send an EDI file to. This works only with the EDI Exchange module enabled.
6. In the sixth argument you specify whether to use "T" Test or "P" Production database connection if you have "Q" for query as the second argument.

Using Several Arguments

You can have a combination of the above arguments. For example, "XF" will export the data and create a pdf file.

Example

If you want to export and image an EDI file, you can use "XF" as the second option.

```
C:\>C:\Program Files\HIPAAsuite\HIPAA Claim Master\HIPAAClaimMaster.exe C:\Test.edi,%F
```

Using the "XF" option for export and output as PDF file

The "X" option will turn on the translator functionality and claims will be exported to the defined data source. You can only do one imaging option such as printing or image file at a time.

"PF" is not valid.

Searching an EDI File

Here is the syntax for the command line arguments:

1. First: Filename to be processed.

2. Second: Command options:

S for search

F for PDF

3. Third: Image file full path and filename.

4. Fourth: Search conditions in the form of Pipe (|) separated key-value pairs. (This is the back slash key in Shift mode).

Keys that can be searched at this time are:

- **ClaimNO** – CLM_01 (you can look for several claim ID's at once separating the ID's with '&' symbols)
- **PatientID** – NM1*QC_09
- **PatientLast** – NM1*QC_3
- **SubscriberID** – NM1*IL_09
- **SubscriberLast** – NM1*IL _03
- **BillProvid** – NM1*85_09
- **Segment** – A complete segment such as "HI*BF*3544"

Example:

H PAA ai nPr i nt er . exe I : \ H PAA s u i t e C o m m D r \ I n b o x \ 837 \ H C F A _ 1 , S F , C \ T e m p \ T e s t . p d f , C a i n N b = 6073123 G W D B i l l P r o v I D = 123414 | S u b s c r i b e r L a s t = S M T H

Working with the Database

Example 1

Here is an example of using a database search to create an 837 EDI file.

```
C:\Users\Martin>"C:\Program Files\HIPAAsuite\HIPAA Claim Master\HIPAAClaimMaster.exe" .QE. SELECT * FROM EDI_CLAIMS WHERE EDI_FILENAME = '3899EGD.837', BEECHSTREET, I
```

An example of using a SQL query to create an EDI file.

The above example has:

- A blank first argument since no file or directory are to be processed.
- The second argument is 'QE' meaning that an EDI file is to be created using the results from a SQL query.
- Argument 3 is blank, meaning we use the default file naming convention
- The fourth argument lists the query.
- The fifth argument specifies the ISA identifier of the destination trading partner. This works only with the EDI Exchange module enabled. If the fifth argument is blank and EDI Exchange disabled then the last setting from the Create 837 screen is used as trading partner identifier.
- The sixth argument determines whether the production or test database connection is to be used.

Example 2

Another example creating an image file from a query

```
C:\Users\Martin>"C:\Program Files\HIPAAsuite\HIPAA Claim Master\HIPAAClaimMaster.exe" .QE.C:\Temp\ImageFile.SELECT * FROM EDI_CLAIMS WHERE EDI_FILENAME = 'test.837', B
```

Creating a PDF image file from the results of a query

The file C:\Temp\ImageFile will contain all the images from claims that fulfill the query.

Example 3

Here is an example of using a database search to create an 277CA EDI file. This works only with the EDI Exchange module enabled.

```
C:\>"C:\Program Files\HIPAAsuite\HIPAA Claim Master\HIPAAClaimMaster.exe" 277,"S  
ELECT TOP 20 * FROM EDI_Claims",133052274,T
```

Creating a 277CA EDI file from the results of a query

The above example has:

- A first argument is 277 meaning that an 277 EDI file is to be created using the results from a SQL query.
- The second argument lists the query.
- The third argument specifies the ISA identifier of the destination trading partner.
- The fourth argument is optional and determines whether the production or test database connection is to be used. By default production connection is used.

Processing EDI Files with the EDI Exchange Module Enabled

The only argument you need in this case is "Auto". The directory to look for new files and the processing options are defined in the setup screens for EDI Exchange. No further arguments are needed.

11.3 Printing and imaging from the database

In January 2015 we added the capability to print or create image files from database records.

The command line arguments for this are as follows:

1st argument: Should be empty

2nd argument can be now:

QP - print records

QT - tiff files

QF - pdf files

3rd argument - Image File name- used only if program is set to create one result file (the same argument we use when create image files from file)

4th argument - Your query

5th argument - Custom printer name, can be omitted if the default printer is used

6th argument - Database, Test or Production, with the default of Production.

Example:

```
"C:\Program Files\HIPAAsuite\HIPAA Claim Master\HIPAAClaimMaster.exe" ,QF,,Select *
from EDI_Claims where ClaimID = '1357'
```

This example will create an image file with the default naming convention from the claim with the ID of '1357'

```
"C:\Program Files\HIPAAsuite\HIPAA Claim Master\HIPAAClaimMaster.exe" ,QP,,Select *
from EDI_Claims where SubscriberLast = 'SMITH',Lexmark X 500 Series,T
```

This example will print all claims in the Test database where the subscriber's last name is 'SMITH' to the Lexmark X500 Printer

11.4 Changing the Default Company

The default company is the identity you assume as the sender when creating EDI files. This affects the ISA, GS, and NM1 sender segments. A single entity may have a need to act as multiple sender identities and so has a need to change the information in the EDI envelopes.

The default company can be changed using command line arguments, affecting the ISA and GS sender ID segments.

ISA Segment Sender Identifier	HIPAA SUITE1	*
Qualifier	ZZ - Mutually Defined	*
Application Sender's Code GS_2	HIPAA SUITE1	*
Tax ID	9876543210	*

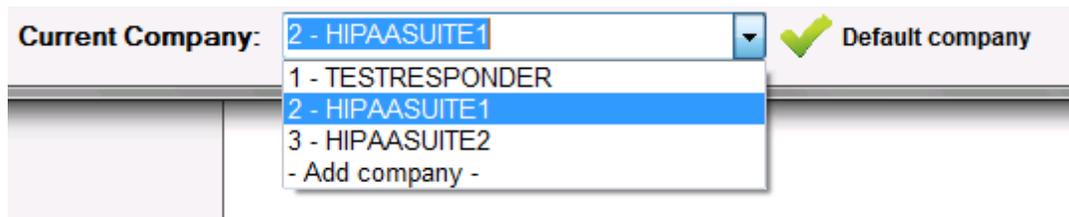
Sender ID segments in Company Setup

To change the default company, a single argument is needed. **"Setcompany"** followed by the company ID of the company you wish to set.

```
"setcompany <ID>"
```

The Company's ID in this case is not its EDI identifier, but the value of the ID row in the COMPANY_SETUP table. It can also be found in the Company Setup window's Current

Company selector:



Company selection in Company Setup

Example

The command argument `setcompany 3`

```
C:\Program Files\HIPAAsuite\HIPAA Claim Master>HIPAAClaimMaster.exe setcompany 3
```

will set the default company to ID = 3, which is HIPAASUITE2 in the example picture above. The log entry for this command will be

```
Set default company with ID 3
Company # 3 is set as a default company
```

Excerpt from log. Default company changed.

11.5 Running the Application via Scheduler

The HIPAA Claim Master can be automated through a scheduler. A scheduler is software that can be programmed to execute a certain task at a certain time repeatedly. Windows has such a scheduler built in.

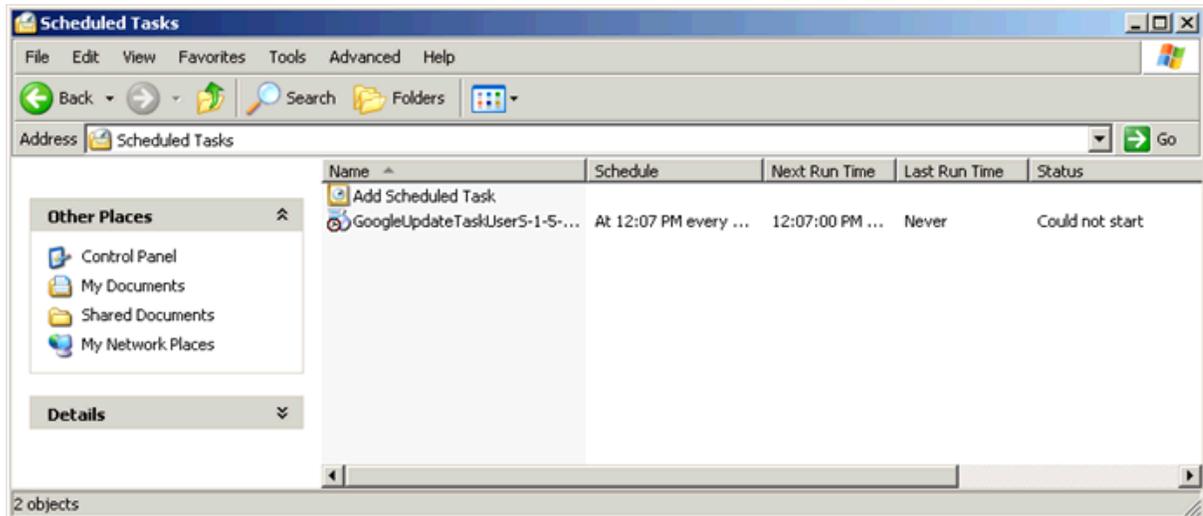
The Windows scheduler can be used to let the HIPAA Claim Master execute in regular intervals without user intervention. Usually one would let the HIPAA Claim Master scan a directory and then print, create image files or export the data and remove the files to prevent duplicate processing.

Use the command line with the appropriate options as described in [Using Command Line Arguments \(CLI\)](#).

Note: Please consult Windows help files for further information on setting up the scheduler.

1. If the task scheduler is installed on your system, you will find it in the "Control Panel". To open Scheduled Tasks, click Start ► Control Panel ► Scheduled Tasks.

Note: In different Windows versions, the way to access the Task Scheduler can differ. Consult your Windows version documentation.



The Windows Task Scheduler

2. Click on "Add Scheduled Task" and the "Scheduled Task Wizard" will guide you through the setup process.
3. In the first window, read the welcome message and click "Next."



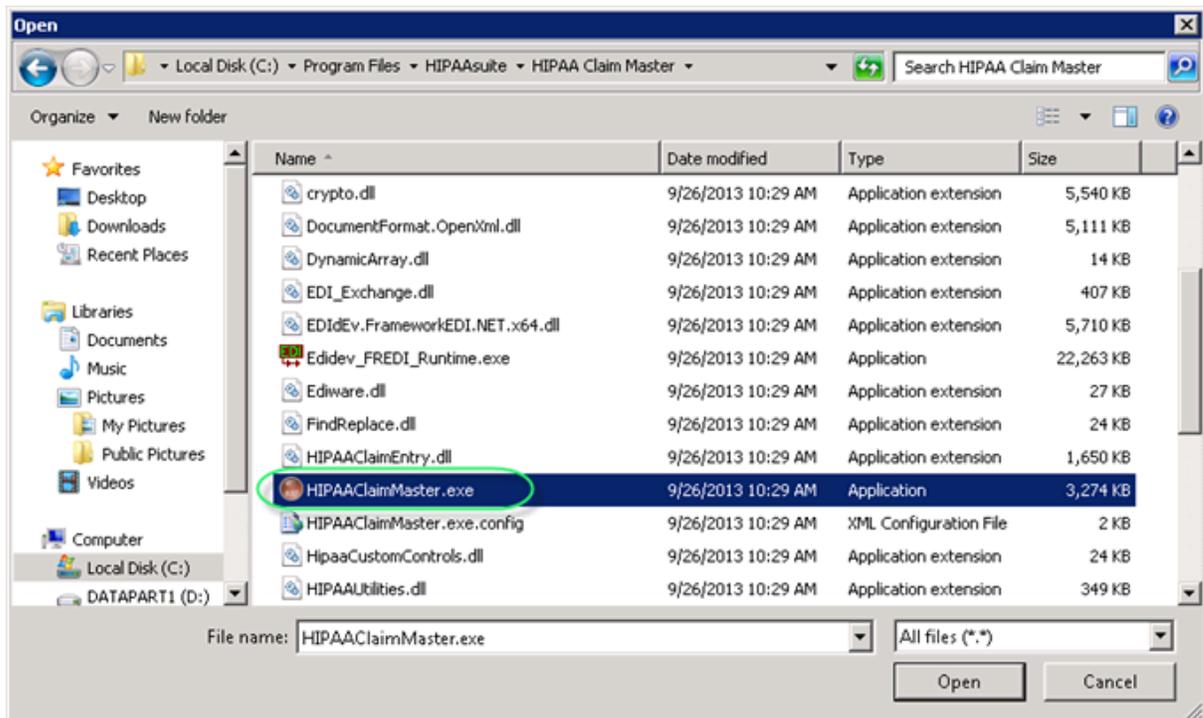
The "Scheduled Task Wizard" window

4. Browse to the `HIPAAClaimMaster.exe` executable file.



The "Scheduled Task Wizard" window

By default, the application is located in `c:\Program Files\HIPAAsuite\HIPAA Claim Master\HIPAAClaimMaster.exe`



Selecting a program to schedule

Click "Next."

5. Type a name for this task. Choose a time interval:

- Daily

- Weekly
- Monthly
- One time only
- When my computer starts
- When I log on

Click "Next."



The "Scheduled Task Wizard" window

6. Enter the name and password of a user. The task will run as if it were started by that user. Click "Next."



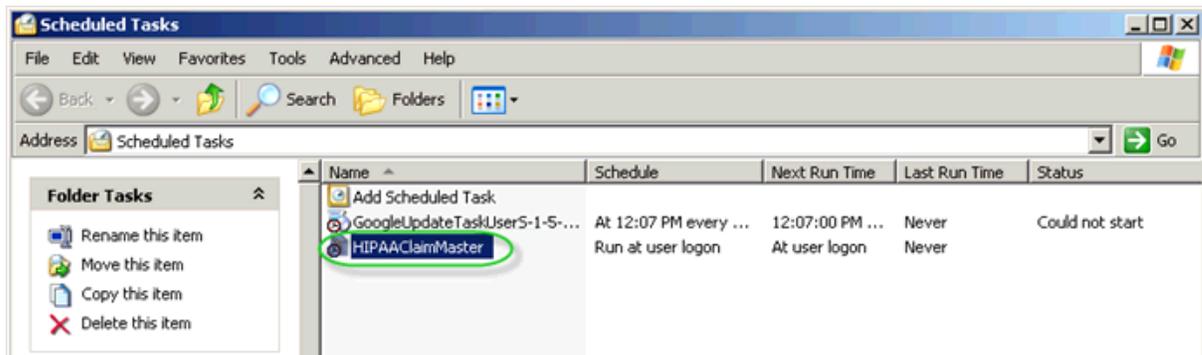
The "Scheduled Task Wizard" window

7. In the last screen, make sure you have specified all data correctly. Click "Finish" to save your task.



The "Scheduled Task Wizard" window

8. The scheduled task has been added to the system. Now let's edit the new task and add the desired command line arguments with the appropriate options. Double-click the newly created task in the "Scheduled Tasks" window.



The Windows Task Scheduler

9. Enter parameters to the "Run" field after the application path and click "OK." Refer to [Using Command Line Arguments \(CLI\)](#) for more information.

Tip: Logs may be helpful when running the program automatically through the scheduler. See [Accessing Daily File Log](#).

Chapter

XIII

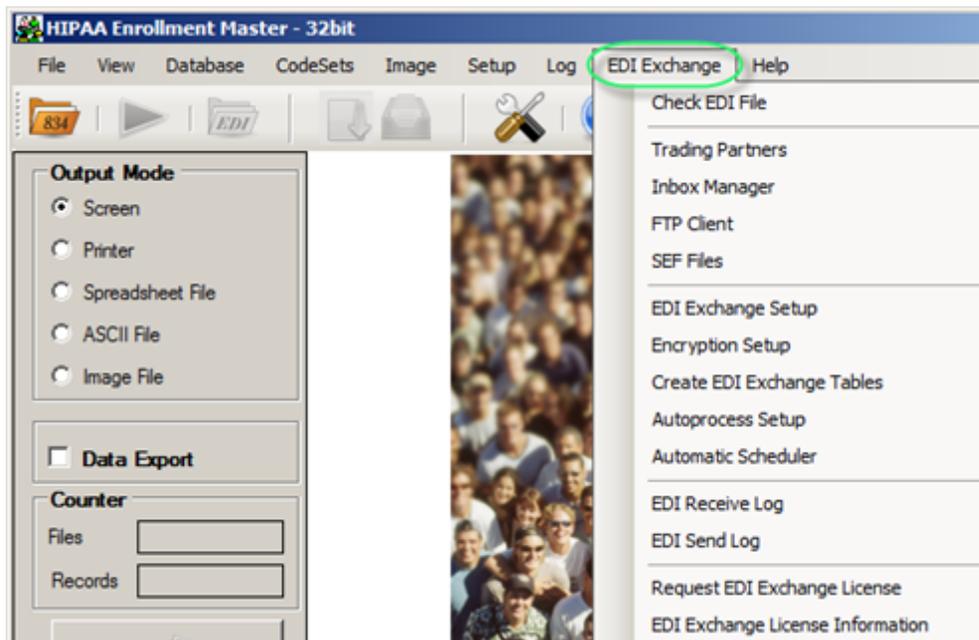
12 Using EDI Exchange Features

12.1 Getting Started

12.1.1 About EDI Exchange

EDI Exchange is a module available in most HIPAAsuite EDI applications. It is an option that you can purchase for an additional cost. Some of our products, such as HIPAA Claim Master, process EDI files but do not receive or send EDI files to and from your trading partners. EDI Exchange is created to do that. EDI Exchange is designed for those organizations that have a large volume of EDI files, need more order and automation and adhere to tougher compliance rules. The EDI Exchange is an EDI pre-processor that handles FTP transport, encryption, HIPAA compliance check, trading partner management, etc. Outgoing EDI files can be checked for compliance; individual records that do not pass the check can be withheld.

HIPAAsuite products with EDI Exchange module have a main menu item called "EDI Exchange" with sub-menus to call the module's functions.



The "EDI Exchange" menu in HIPAA Enrollment Master

EDI Exchange performs the following functions:

- **Trading Partners Management** – The following Trading Partner's parameters can be stored and transparently managed with the help of EDI Exchange: name, address, EDI identifiers, delivery methods, encryption parameters, FTP servers,

CORE-Compliant server addresses and credentials, communication numbers and folders to keep files separated, special requirements specific to this trading partner. Read more in [Setting up Trading Partners](#).

- **File Transport** – EDI Exchange has a built-in FTP client that can securely connect to your trading partner's FTP servers. If you employ your own FTP server, you can utilize the folder structure that EDI Exchange uses to manage incoming files, users, home directories and permissions so that your Trading Partners can drop off and pick up EDI files. Supported are:

- Simple FTP
- FTP Secure
 - Implicit FTPS
 - Explicit FTPS
- Secure Shell FTP or sFTP

Read more in [Using FTP Client](#).

- **Encryption** – Many healthcare-related companies use encryption to cloak the content of their EDI files. The prevalent method of encryption is **PKI** (Private Key Infrastructure) that uses the product of two incredibly large prime numbers as cipher. EDI Exchange supports **PGP** (Pretty Good Privacy), the leader in PKI products as well as the open source **GPG** project with its [Windows sub project](#) PGP4Win. Both are implementations of the same encryption mechanism. Read more in [Using Encryption](#).
- **File Management** – EDI Exchange uses a clear directory structure to store EDI files. The structure is based on root directories for incoming files, outgoing files, processed files and suspended files. Below these root directories, there are subdirectories for each trading partner and then each transaction set. Read more in [Defining Root Directory](#) and [Initializing EDI Exchange](#).
- **EDI Compliance Check** – EDI standards are strict and precise; adherence to the standards is very important so that any organization can work with them regardless of their backend system software. EDI Exchange has a built-in compliance engine that checks incoming files for compliance. The engine also generates a report listing each problem with the exact location. Outgoing EDI files can also be checked and you have an option to withhold individual records that violate the rules. Read more in [Checking EDI Files](#).
- **EDI Control for Transactions** – The EDI protocols have a few supporting transaction sets that are useful to the smooth functioning of EDI exchanges. They provide the

sender with an instant feedback on receipt. The following transaction sets are available:

- **TA1 Acknowledgment**
- **997/999 Functional Acknowledgment**
- **277U/277CA Unsolicited Claim Status Response** (in case of Claims)
- **Logging** – EDI Exchange has several logs that are instrumental to keep processing in order and allows to forensically investigate mishaps. There are three logs in EDI Exchange:
 - **Incoming file log** – See [Accessing EDI Receive Log](#).
 - **Outgoing file log** – See [Accessing EDI Send Log](#).
 - **Daily transaction log**

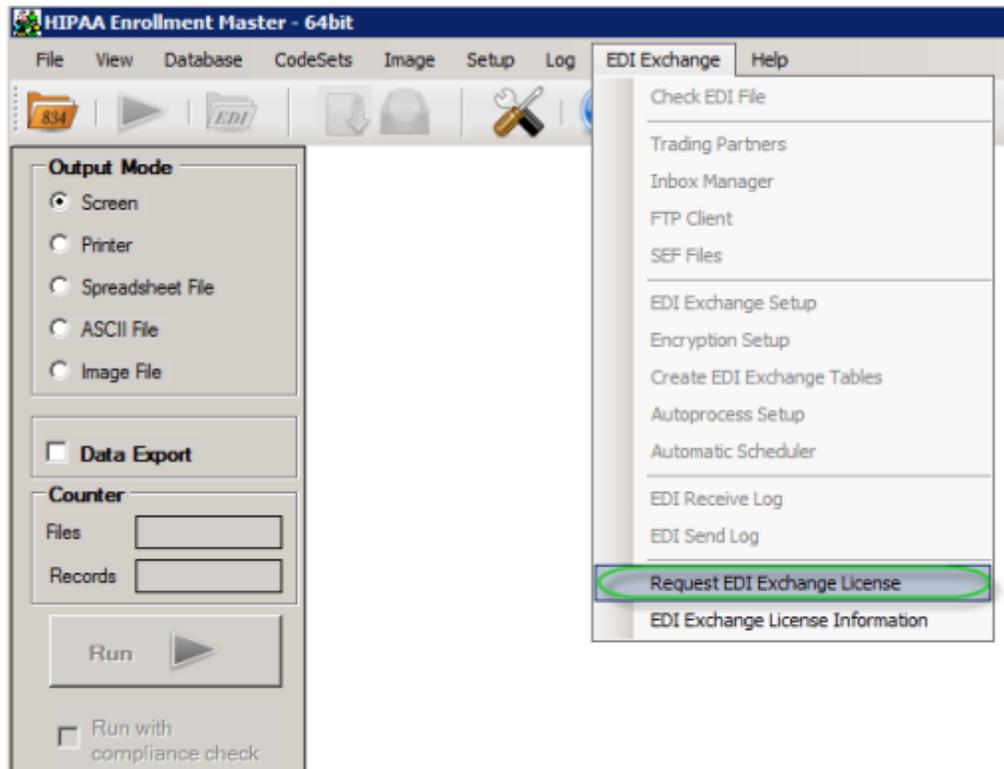
12.1.2 Requesting EDI Exchange License

If your trial has expired, you can request an extension to the trial.

If you purchased the product and need a final license key, you should request an EDI Exchange license.

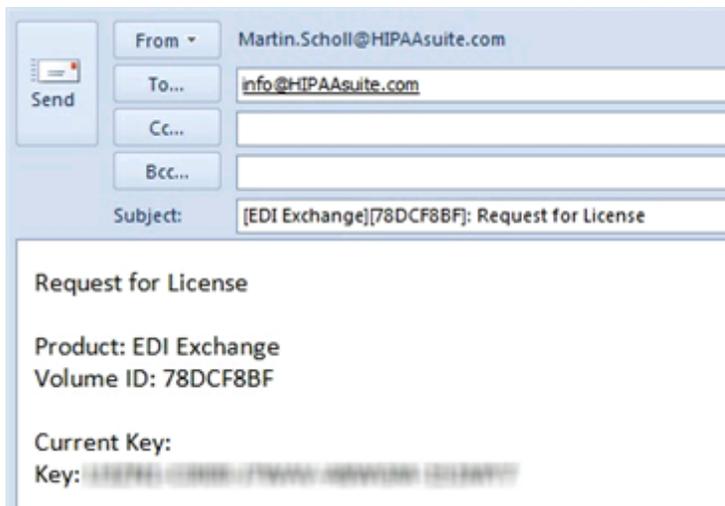
Follow the instructions below to request a trial or final license key.

1. Select "Request EDI Exchange license" under the "EDI Exchange" menu item.



A menu item to request a license key

2. Once you have clicked this menu item, your default email application appears. In our case, it is Microsoft Outlook. All information necessary to produce the key is automatically filled out.



Email message created by EDI Exchange

3. You can add a trial extension or a final key after purchasing or relocating the software.

Once you receive the response with the key for EDI Exchange, you can bring the "Registration Form" screen up again and click on "Register". Enter the key to unlock EDI Exchange. In the Result area, you will see that EDI Exchange has been registered.



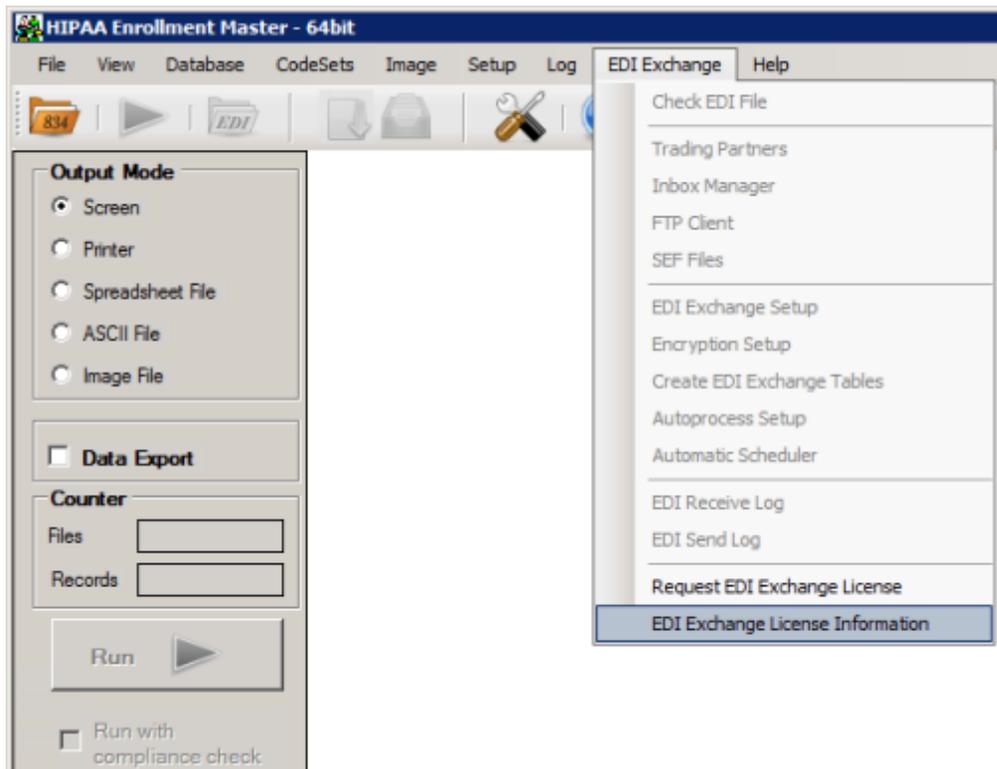
The screenshot shows a window titled "Registration Form" with a close button in the top right corner. The main heading is "License Registration Form" followed by "EDI Exchange". Below this is a text input field labeled "License Key" with a blurred key value. To the right of the input field are two buttons: "Register" and "Close". Below the input field is a section labeled "Result" containing a message box with a green border. The message reads: "Thank you for evaluating EDI Exchange! EDI Exchange is fully functional with certain trial limitations. This evaluation version expires in 15 days".

Entering the license key

12.1.3 Registering EDI Exchange

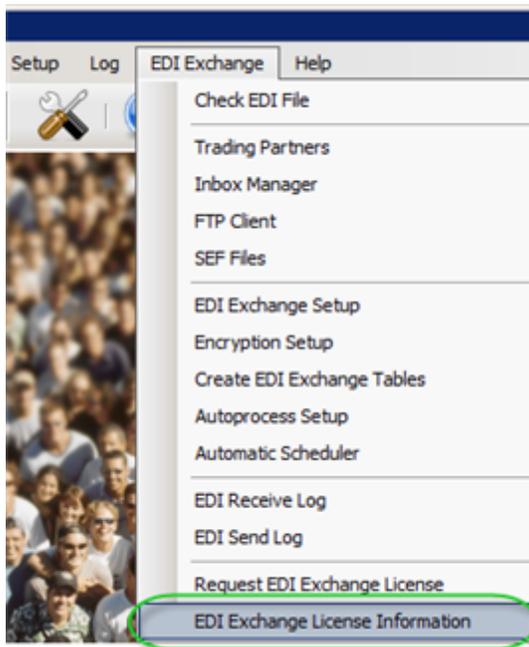
EDI Exchange is licensed separately from the host application, HIPAA Claim Master, for example. The reason is that EDI Exchange will work on all HIPAAsuite Products that are installed on your particular computer. For example, if you have HIPAA Claim Master and HIPAA Enrollment Master licensed, only one license of EDI Exchange is needed and the module will work across two products.

When you first install a HIPAAsuite product of your choice, a 15-day EDI Exchange trial is included. Once the trial expires, EDI Exchange loses its functionality. The menu items under "EDI Exchange" become disabled except the last ones that allow you to license and enable the product.



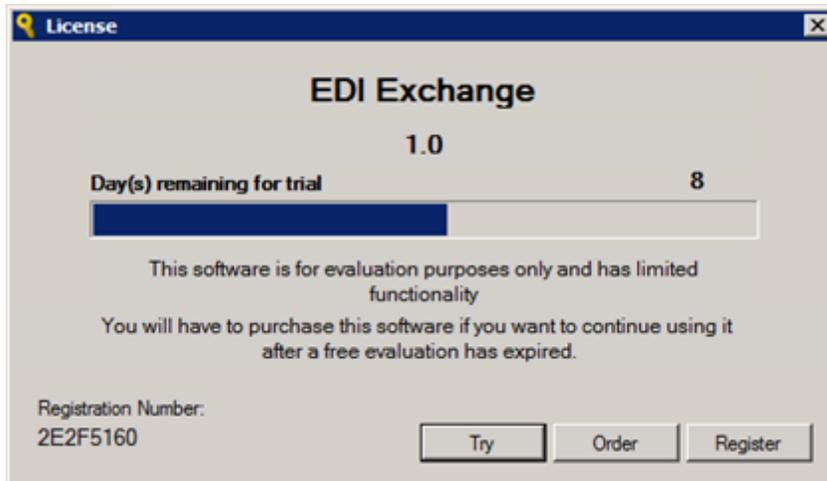
The "EDI Exchange" menu with menu items disabled

You can register the product by clicking on the "EDI Exchange License Information" option under the "EDI Exchange" menu.



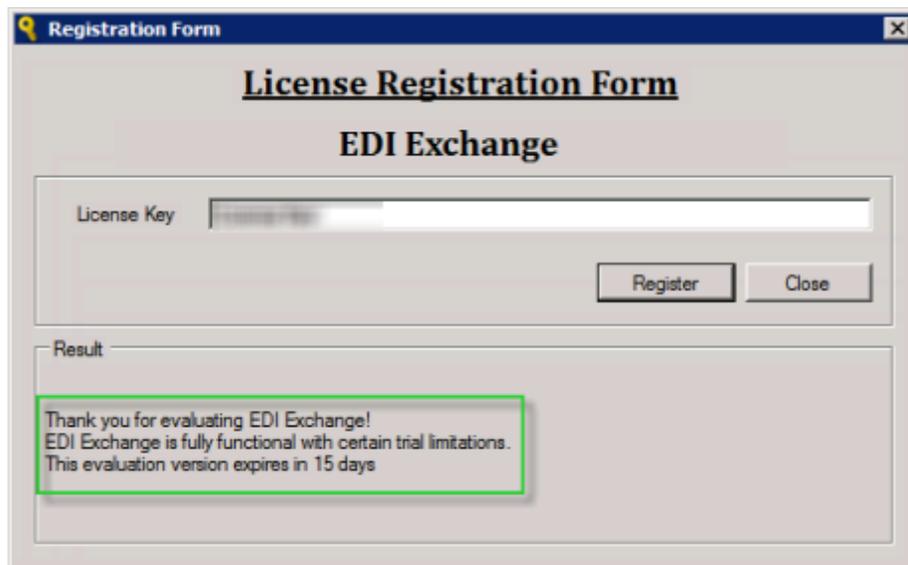
"EDI Exchange License Information" option under the "EDI Exchange"

Then the license screen appears. In the lower left corner you can find the unique registration number needed to create either trial extensions or final licensing.



The license information screen

Once you click on "Register," you can enter the license key that you have previously received via email from us (see [Requesting EDI Exchange License](#).) Click on "Register" and you will see the registration message in the "Result" area.



Extending the trial by entering a license key

Close the "Registration Form" and continue using the EDI Exchange.

12.2 Configuring EDI Exchange (Obligatory Settings)

12.2.1 1 Setting up Database Connection

EDI Exchange work is based on the database connection that you define under *Database Connection and Data Fields* in the main menu of the HIPAA host application. Make sure the connection has already been set up and tested before proceeding with EDI Exchange.

Then proceed to the next step: [Creating Database Tables](#).

12.2.2 2 Creating Database Tables

Once you have configured the database connection ([Setting up Database Connection](#)), follow the instructions below.

1. Select *EDI Exchange* ► *Create EDI Exchange Tables* in the main menu.

CREATE 837 TRANSACTION SETS

SELECT top 30 *
FROM EDI_Claims

Live Database Effective Date: 12/10/2012
 Test Database

Version:
 4010 4010 if in original, otherwise 5010 5010

Exit Go! Write to File View EDI File Help

ISA segment

ISA	00	00	ZZ	HIPAASUITE	ZZ	TESTEDI5010	121210	1215	00501	345121558	0	P
ISA	00	00	ZZ	HIPAASUITE	ZZ	TESTEDI5010	121210	1215	00501	345121558	0	P

GS segment

GS	HC	HIPAASUITE	TESTEDI5010	20121210	1215	234500001	X	005010X222A1	ST	837	000012	005010X222A
GS	HC	HIPAASUITE	TESTEDI5010	20121210	1215	234500001	X	005010X222A1	ST	837	000012	005010X222A

BHT segment

BHT	0019	00 - Original	234500001-2	20121210	1215	CH - Chargeable
BHT	0019	00 - Original	234500001-2	20121210	1215	CH - Chargeable

ST segment

ST	837	000012	005010X222A
ST	837	000012	005010X222A

Select Trading Partner

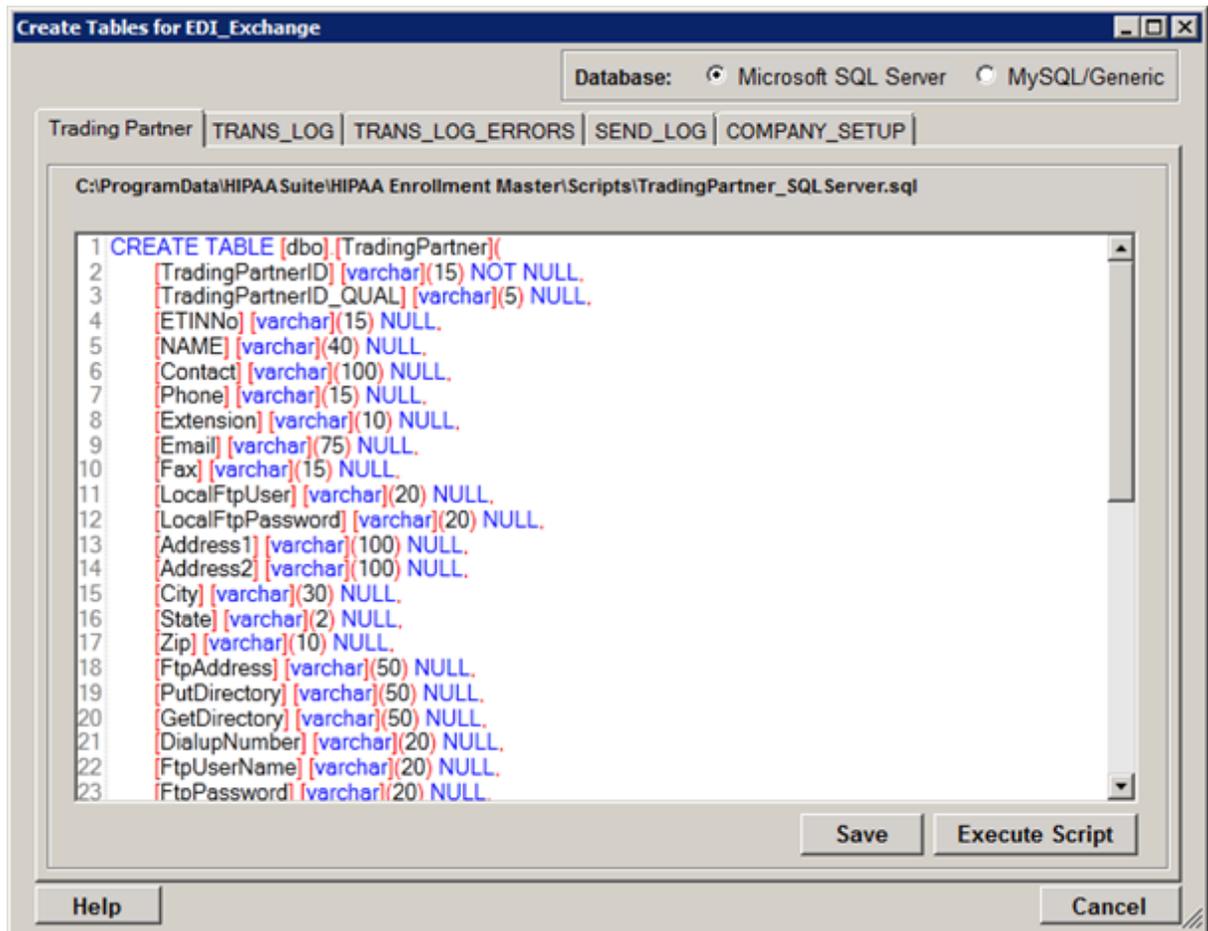
Fields with † will be populated by the process.

```
ISA*00* *00* *ZZ*HIPAASUITE *ZZ*TESTEDI5010 *121210*1215***00501*345121558*0*P::~~
GS*HC*HIPAASUITE*TESTEDI5010*20121210*1215*234500002*X*005010X222A2~
ST*837*000021*005010X222A2~
BHT*0019*00*234500002-1*20121210*1215*CH~
NM1*41*2*MCHS INC DBA MINERAL COMM HOSPITAL*****46*1902990005~
PER*IC*Jane Doe*TE*9005555555~
NM1*40*2*TEST****46*TESTEDI5010~
HL*1**20*1~
PRV*BI*PXC*282NC0060X~
NM1*85*2*MCHS INC DBA MINERAL COMM HOSPITAL*****XX*1902990005~
N3*225 Main Street~
N4*Centerville*PA*17111~
REF*EI*810421823~
HL*2*1*22*1~
SBR*P***Group Name*****98~
NM1*IL*1*Doe*John****MI*123456~
```

2 EDI transactions are created.

The menu item to create the necessary tables.

2. The "Create Tables for EDI_Exchange" screen will appear. Table creation and/or modification for your database is handled here.



The screen to create the tables

3. Select the database type you use for your host HIPAA application.

- **Database**

- **Microsoft SQL Server** (SQL Server 2008 and above)

- **MySQL**

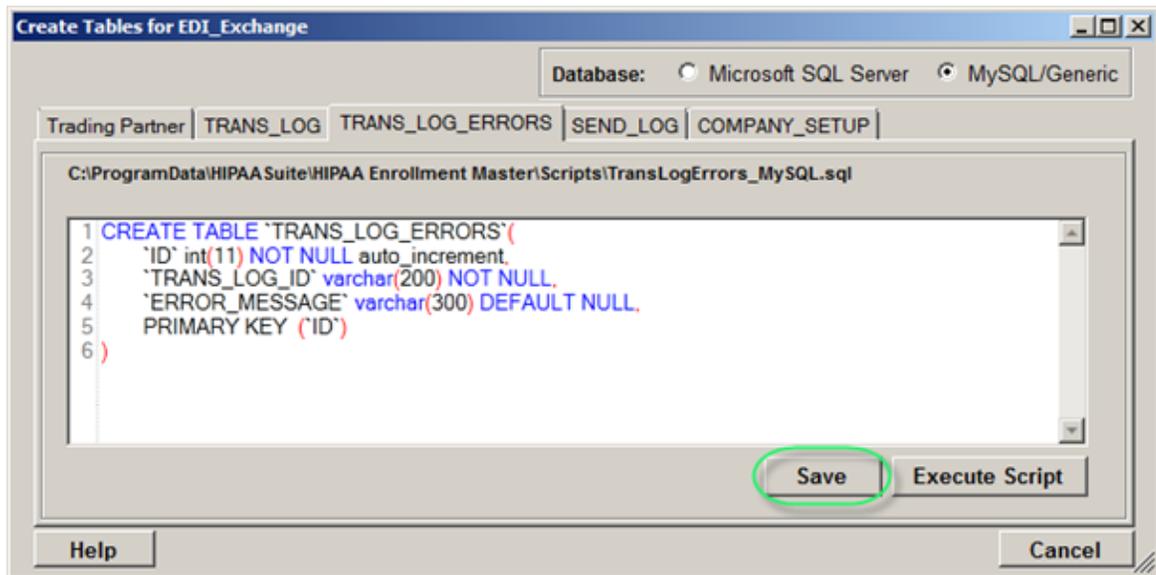
Note: In case your database is not listed, modify the scripts or ask your database administrator to make the necessary modifications.

4. The following tables are part of EDI Exchange:

- **TradingPartner** – SQL statements to create the "TradingPartner" table in your database. This table contains information about trading partners.
- **Trans_Log** – SQL statements to create the "TRANS_LOG" table in your database. This table contains incoming file information, keeps track of all EDI files that you receive and the compliance check report.

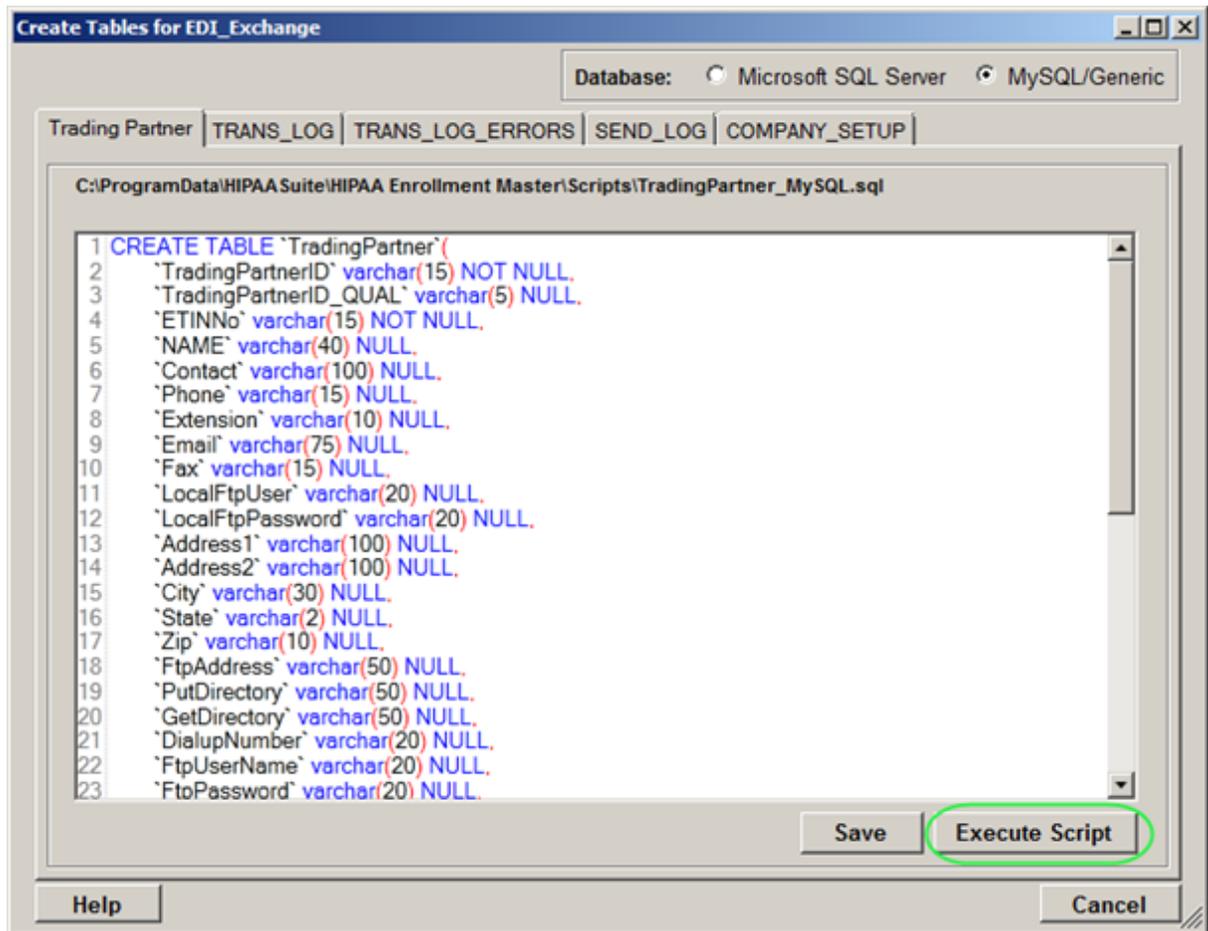
- **Trans_Log_Errors** – SQL statements to create the "TRANS_LOG_ERRORS" table in your database. This table collects the results of the compliance check and keeps track of all sent files.
 - **Send_Log** – SQL statements to create the "SEND_LOG" table in your database. The table contains information about EDI files created and sent to trading partners.
 - **Company_Setup** – SQL statements to create the "COMPANY_SETUP" table in your database. This table collects information about you, the sender of EDI information.
5. You can modify the scripts so that they run on your specific database. Once you have you modified the script, click "Save."

Tip: Every database system has their own little syntax idiosyncrasies and the scripts might require tweaking. You can edit the table scripts in this screen and save your modified scripts. One example are 'date' and 'time' or 'money' data types that do not exist in SQL Server 2005. You can just rename those types to 'datetime' and save you script and it will run fine.



The "Save" button

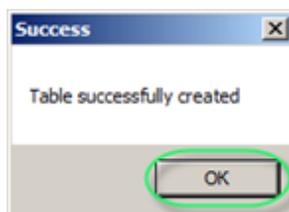
6. For **each** script on every tab, click "Execute Script" to create the corresponding table in the database.



The "Execute Script" button

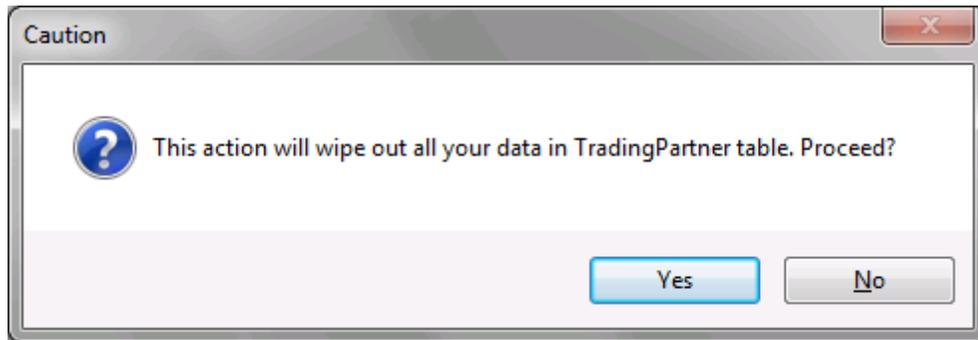
Notice: Creating tables means clicking the "Execute Script" button in all five tabs of the "Create Tables for EDI_Exchange" window. Then close this window.

7. Once the table has been created successfully, you will see the following notification:



The Create Table script success message

Warning: Double-execution of a script wipes out the previous table you have created. A prompt will warn you before deleting an existing table. To Add/Remove fields use the "Compare Script..." button. Remove the script files once you have created the tables so nobody can destroy the tables by accident.



The double-execution warning message.

Make sure there are no error messages and the table creation has been completed successfully.

Compare Script with existing Table

HIPAAsuite products go through continual development and improvements. Often these changes lead to new fields in the database. While it is easy to drop a table and regenerate it with the new fields, you will lose all the data in the table. To avoid this trouble there is the button "Compare Script with existing Table". If you click this, the table structure in your database will be compared with the script. There are two possible outcomes. Your table is up to date

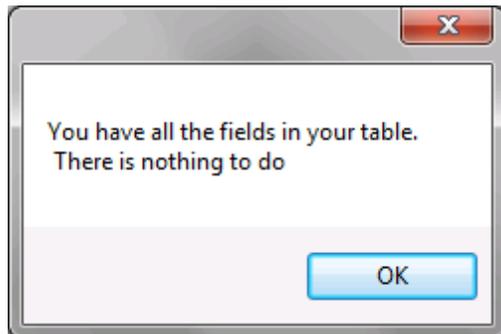
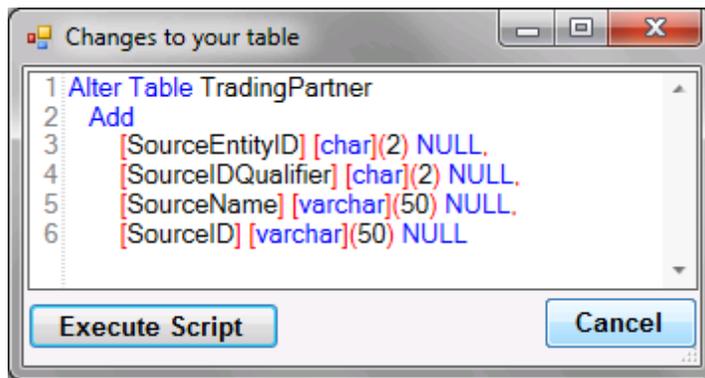


Table is up to date

or if your table is missing recently added fields, you will see a window pop up that shows an 'Alter Table' script with which you can add those fields to the table without interfering with existing data.



The 'Alter Table' script that shows as a result of missing fields

You can now click the "Execute Script" button and the field will be added and a message will confirm your changes



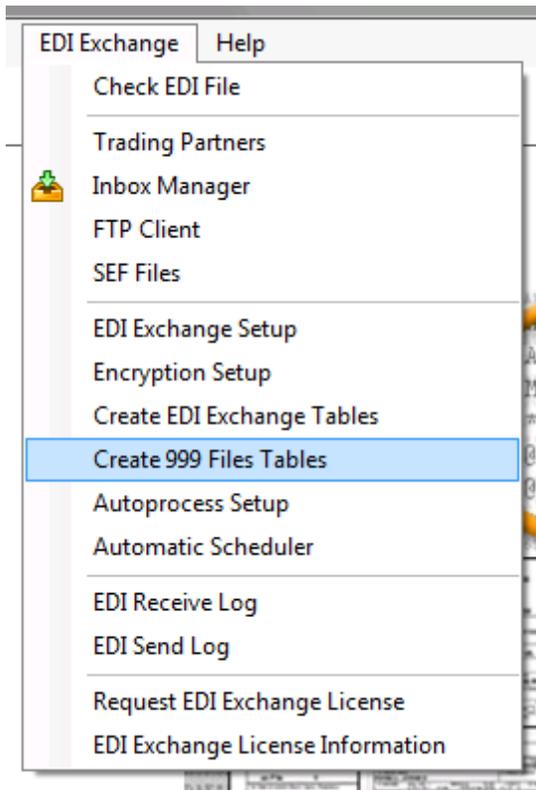
Alter Table statement successfully executed.

Once you have created the tables, you can start setting up the other application options. See the next step: [Defining Auto-Processing Options](#).

12.2.3 2b Creating 999 File Tables

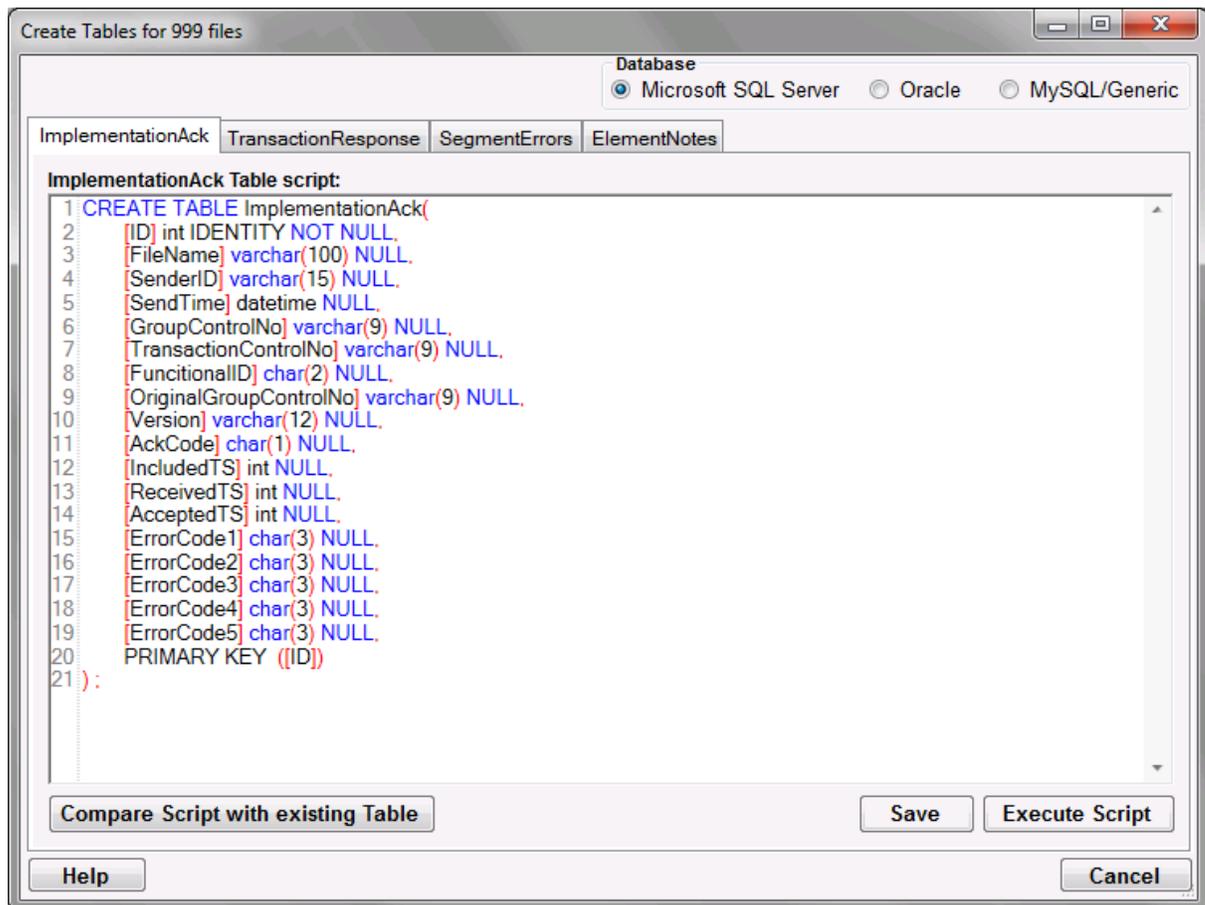
Once you have configured the database connection ([Setting up Database Connection](#)), follow the instructions below.

1. Select *EDI Exchange* ► *Create 999 Files Tables* in the main menu.



The menu item to create the necessary tables

2. The "Create Tables for 999 Files" screen will appear. Table creation and/or modification for your database is handled here.



The screen to create the tables

3. Select the database type you use for your host HIPAA application.

- Database
 - Microsoft SQL Server (SQL Server 2008 and above)
 - Oracle
 - MySQL

Note: In case your database is not listed, modify the scripts or ask your database administrator to make the necessary modifications.

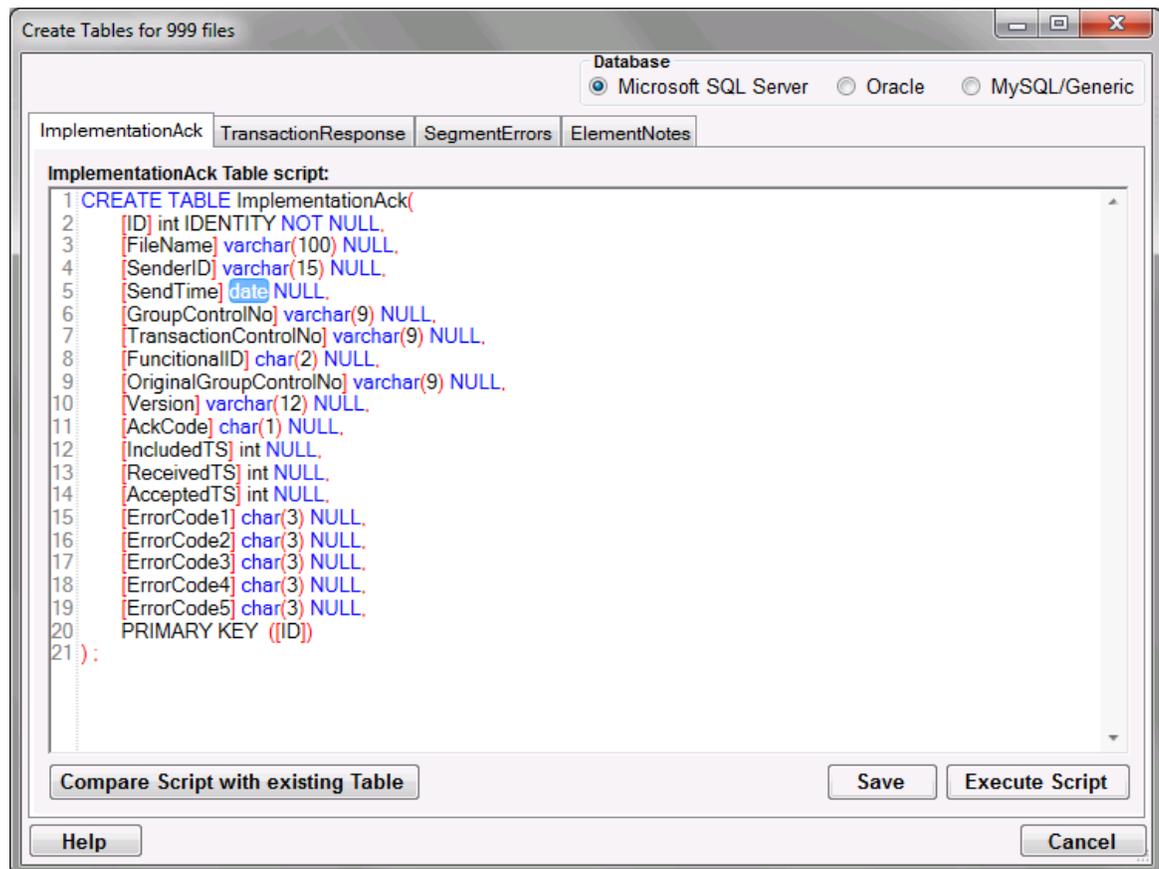
1. The following tables are part of EDI Exchange:

- **ImplementationAck** – SQL statements to create the "ImplementationAck" table in your database. This table contains information about Acknowledgments.
- **TransactionResponse** – SQL statements to create the "TransactionResponse" table in your database. This table contains individual transactions contained in 999 files.

- **SegmentErrors** – SQL statements to create the "SegmentErrors" table in your database. This table contains individual segments in error contained in 999 transactions.
- **ElementNotes** – SQL statements to create the "ElementNotes" table in your database. The table contains the elements in error in a specific segment.

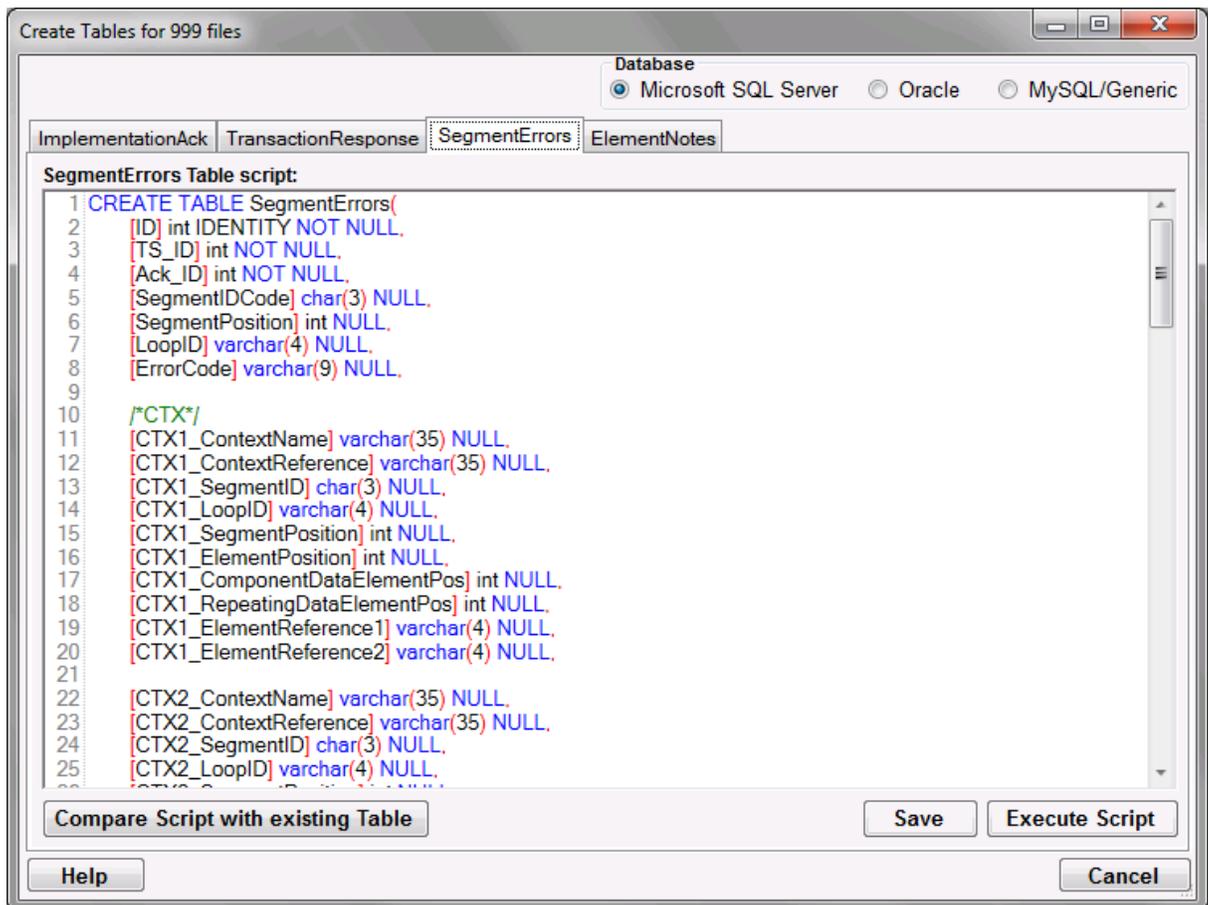
5. You can modify the scripts so that they run on your specific database. Once you have you modified the script, click "Save."

Tip: Every database system has their own little syntax idiosyncrasies and the scripts might require tweaking. You can edit the table scripts in this screen and save your modified scripts. One example are 'date' and 'time' or 'money' data types that do not exist in SQL Server 2005. You can just rename those types to 'datetime' and save you script and it will run fine.



The "Save" button

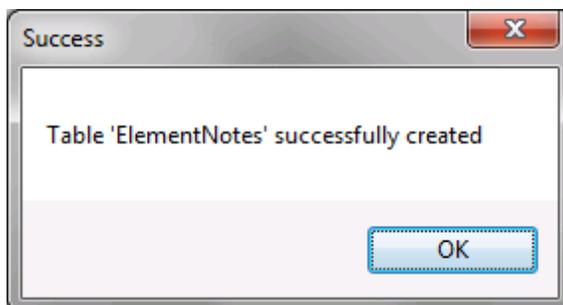
6. For **each** script on every tab, click "Execute Script" to create the corresponding table in the database.



The "Execute Script" button

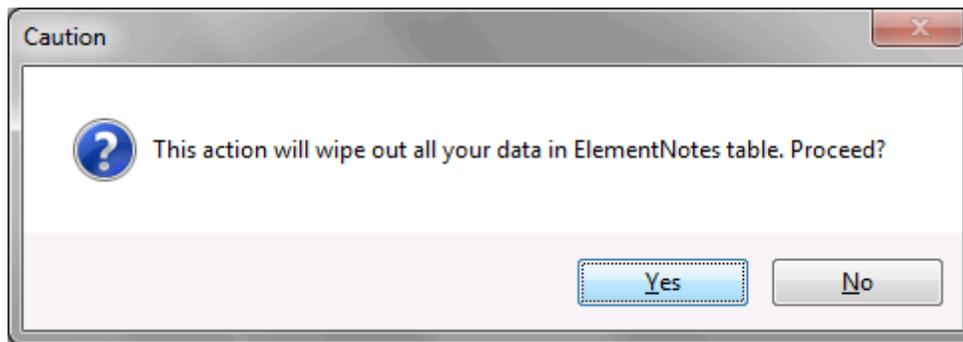
Notice: Creating tables means clicking the "Execute Script" button in all four tabs of the "Create Tables for 999 Files" window. Then close this window.

7. Once the table has been created successfully, you will see the following notification:



The Create Table script success message

Warning: Double-execution of a script wipes out the previous table you have created. A prompt will warn you before deleting an existing table. To Add/Remove fields use the "Compare Script..." button. Remove the script files once you have created the tables so nobody can destroy the tables by accident.



The double-execution warning message.

Make sure there are no error messages and the table creation has been completed successfully.

Compare Script with existing Table

HIPAAsuite products go through continual development and improvements. Often these changes lead to new fields in the database. While it is easy to drop a table and regenerate it with the new fields, you will lose all the data in the table. To avoid this trouble there is the button "Compare Script with existing Table". If you click this, the table structure in your database will be compared with the script. There are two possible outcomes. Your table is up to date

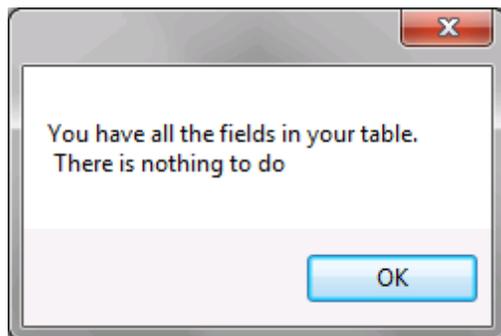


Table is up to date

or if your table is missing recently added fields, you will see a window pop up that shows an 'Alter Table' script with which you can add those fields to the table without interfering with existing data. In the latter case, you can click the "Execute Script" button and the field will be added and a message will confirm your changes

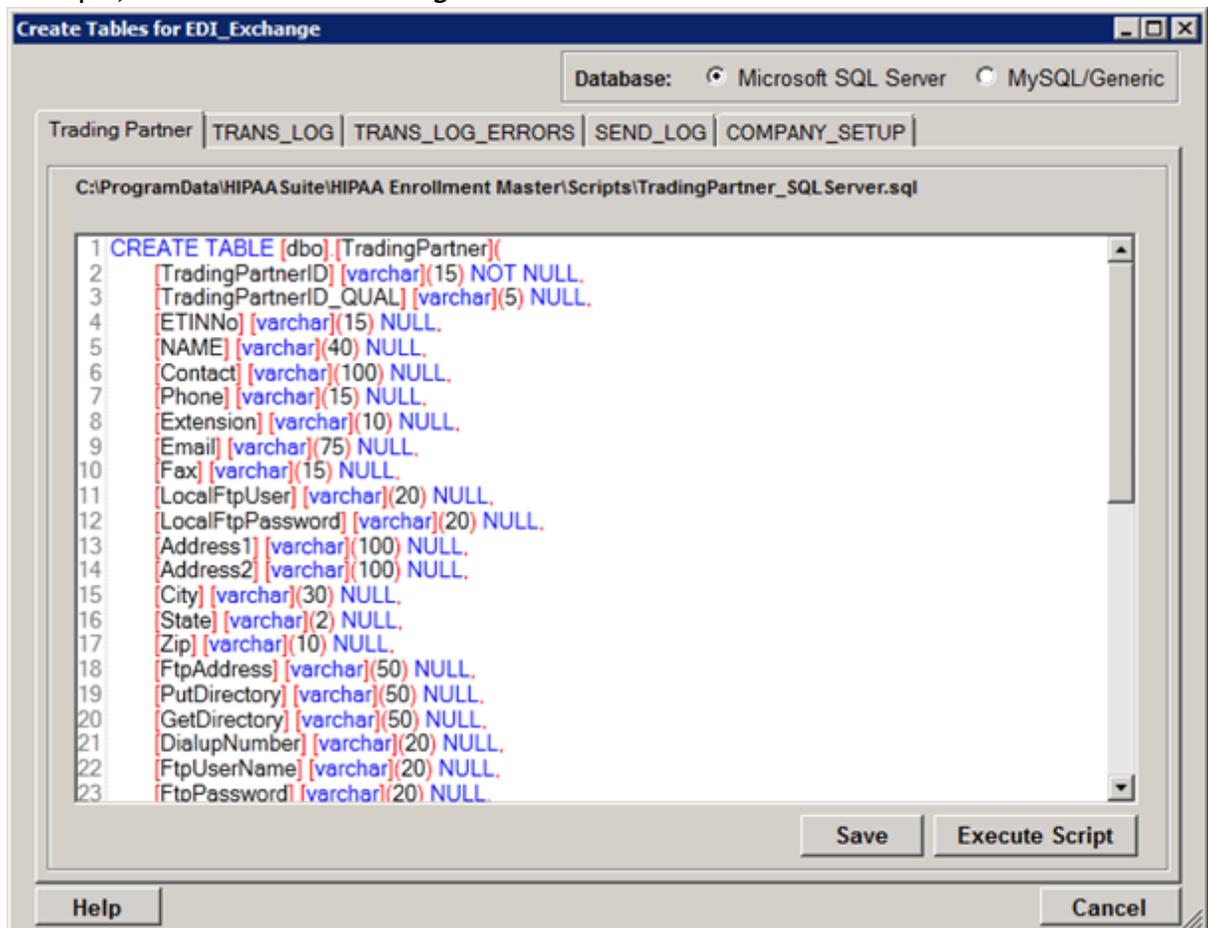


Alter Table statement successfully executed.

12.2.4 2c Updating Database Tables

To update an existing table (in the event of an update, for example), follow the instructions below.

1. Start with the table creation script window of the table you want to update. In this example, we will use the *Trading Partner* table.

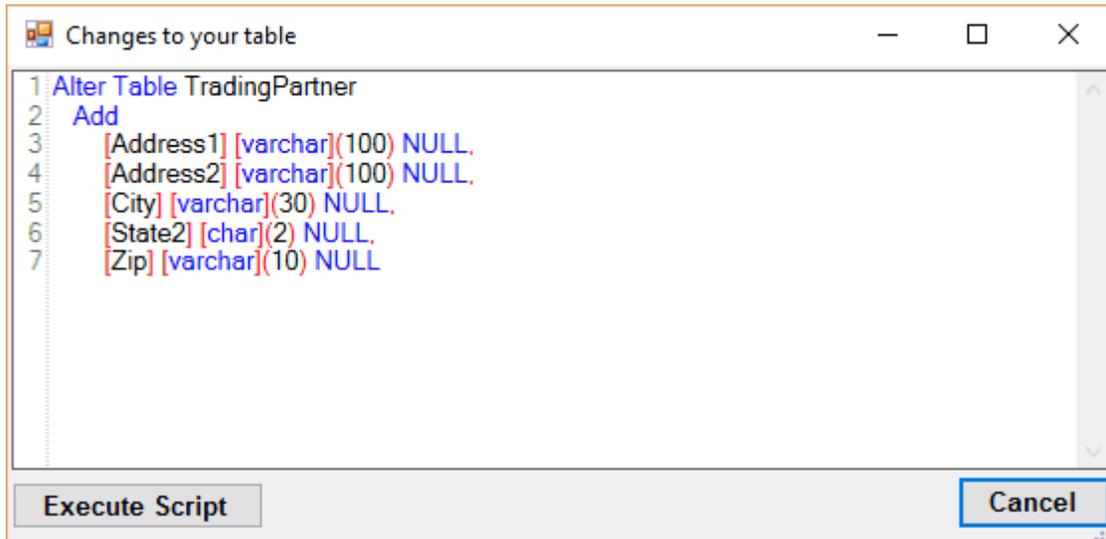


The table creation script for the Trading Partner table.

2. In the case of modifying a table to include/exclude/change a field or fields required by a program update, the script will have been updated for you and clicking the *Compare Script with existing Table* button will bring you to the next step. To modify the table yourself in order to conform to your particular database, first edit the script text to suit your database system, click the *Save* button, then the *Compare Script with existing Table* button, and proceed to the next step.

Tip: Every database system has their own little syntax idiosyncrasies and the scripts might require tweaking. You can edit the table scripts in this screen and save your modified scripts. One example are 'date' and 'time' or 'money' data types that do not exist in SQL Server 2005. You can just rename those types to 'datetime' and save you script and it will run fine.

3. Having clicked *Compare Script with existing Table*, the script will be compared to the existing table and any additional fields will be presented. In this example, the Trading Partner table's Address fields will be added.



Changes to be made to Trading Partner table.

4. Click *Execute Script*. This will perform the additions/changes stated in the alter table script and a prompt will appear informing you of the change. The table has now been modified.

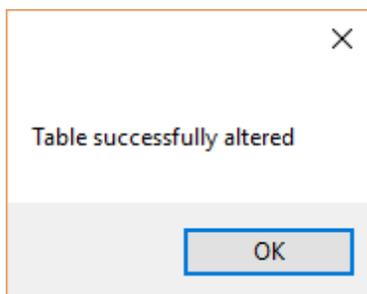


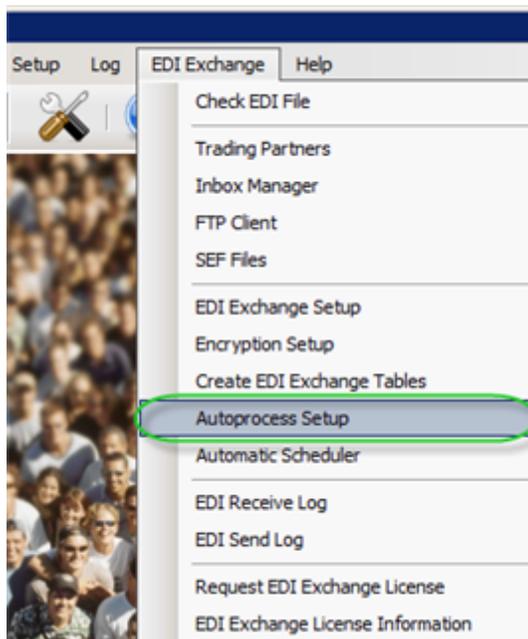
Table has been altered.

12.2.5 3 Defining Auto-Processing Options

In the "Auto-Process Setup" you can instruct the program on what to do after analyzing and decrypting the received files in the Inbox Manager. The auto processing enables you to combine and run multiple fulfillment steps together (for example, export, saving, printing.) These options are important for the hand-over from EDI Exchange to the other HIPAAsuite program that hosts EDI Exchange.

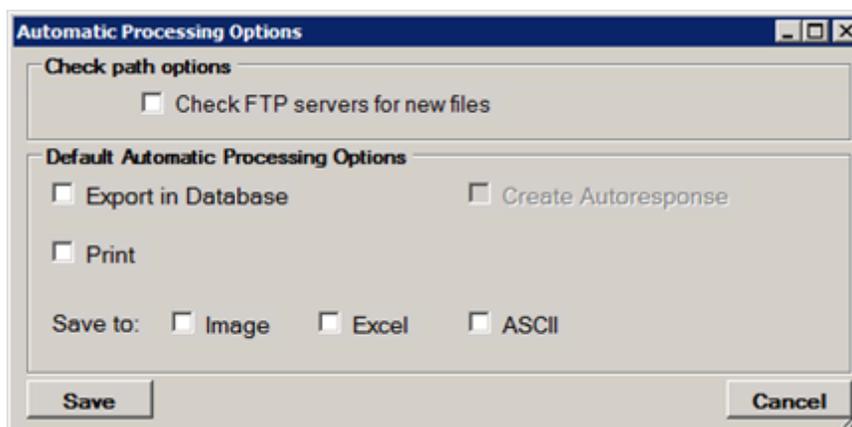
Follow the instructions below to specify the Auto Process Options.

1. Select *EDI Exchange* ▶ *Autoprocess Setup* in the main menu.



The "Autoprocess Setup" menu item

2. The following screen will appear if the host HIPAA application is Enrollment Master.



Defining the Auto Processing Options

3. The following options can be specified:

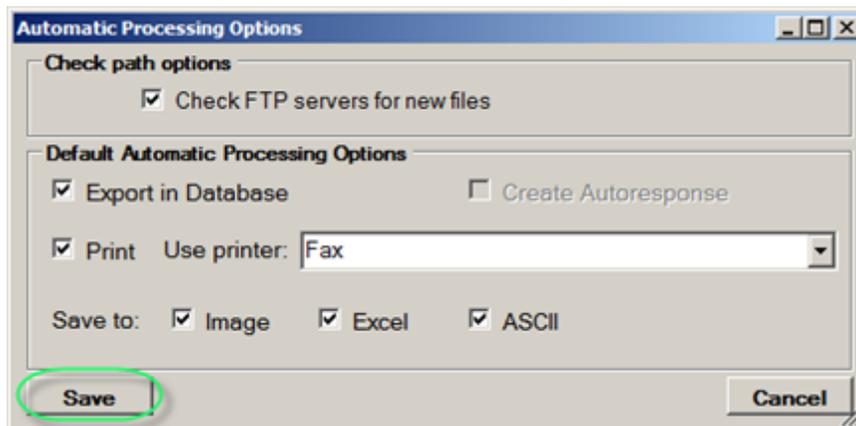
Check Path Options

- **Check FTP servers for new files** – If checked, EDI Exchange automatically looks for new files for all trading partners that have FTP connection set up. Then the program gets all the waiting files and puts them into the Inbox. In the second step, it goes through every file.

Default Automatic Processing Options

- **Export in Database** – If selected, the module exports new files to the database.
- **Print** – If selected, the module prints files using the selected printer.
- **Create Auto-Response** – If selected, the module creates an auto-response to the received files.
- **Save to** – If selected, the system automatically saves files as:
 - **Image**
 - **Excel**
 - **ASCII**

4. Click "Save."



The "Save" button

Once you have saved the auto-processing options, the files will not only be analyzed but also processed according to the defined settings. Proceed to the next step: [Defining Communications Directory](#).

12.2.6 4 Defining Communications Directory

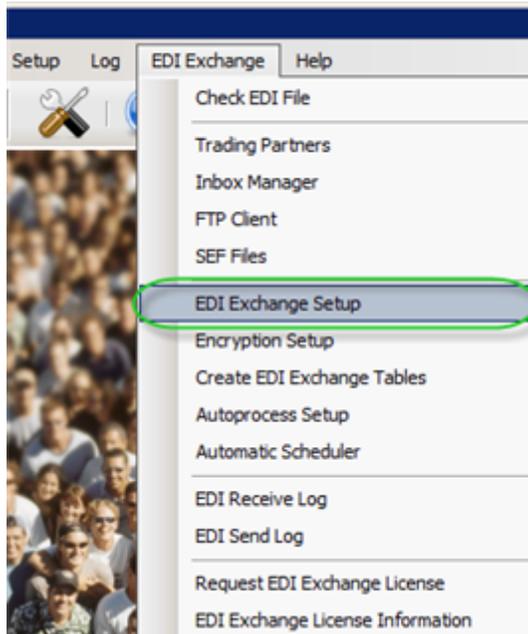
EDI file directory is an obligatory setting you need to set up before starting using the EDI Exchange. In order to keep track of the thousands of EDI files that accumulate over time, EDI Exchange uses a folder structure which we call the "HIPAAsuite Communications Directory" or "HIPAAsuiteCommDir" in short. In it, you will find all your EDI files sorted into several categories:

- **Inbox**
- **Outbox**
- **ProcessedFiles**
- **EncryptedFiles**
- **SuspendedFiles**

Within these directories, there will be folders for each Trading Partner and type of transaction. The location and names of the sub-folders are handled in the "Trading Partner Setup." See [Setting up Trading Partners](#).

Follow the instructions below to specify EDI communications root directory.

1. Select "EDI Exchange Setup" under the "EDI Exchange" menu item.



The "EDI Exchange Setup" menu item

2. The following window will appear.

The "EDI Exchange Setup" window

Note: The icon in the lower left corner indicates that EDI Exchange has not been initialized yet.

3. Under "EDI Exchange Directory Setup" specify the following setting:

- **Root Directory Path** – Define the root path in the "Root Directory Path" text field. The root path is the folder where all your EDI files reside. EDI Exchange will later create sub-directories required to operate.

The root communications directory setup

4. Click "Save."

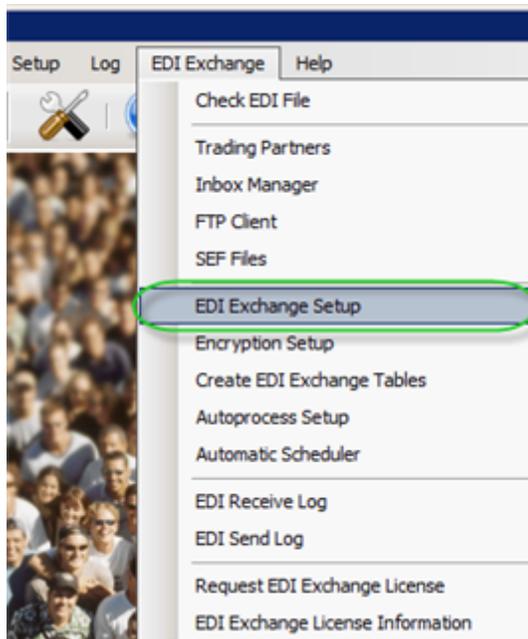
The "Save" button

After setting the root directory, you can click the "Initialize EDI Exchange" button. Read more in [Initializing EDI Exchange](#).

12.2.7 5 Initializing EDI Exchange

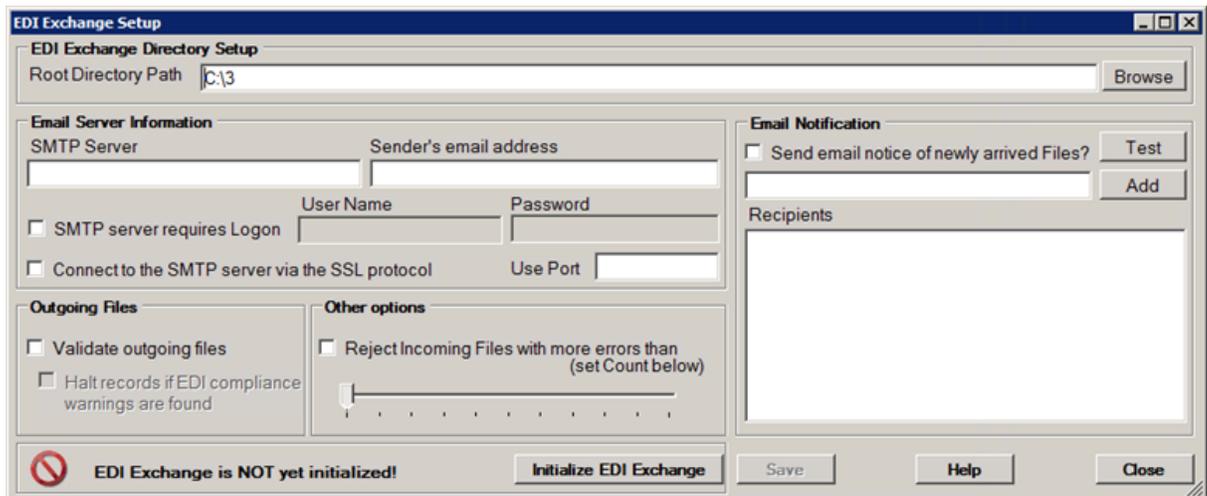
Before you can use EDI Exchange, and after you have configured the obligatory settings, you have to perform the initialization. Follow the instructions below.

1. Select "EDI Exchange Setup" under the "EDI Exchange" menu item.



The "EDI Exchange Setup" menu item

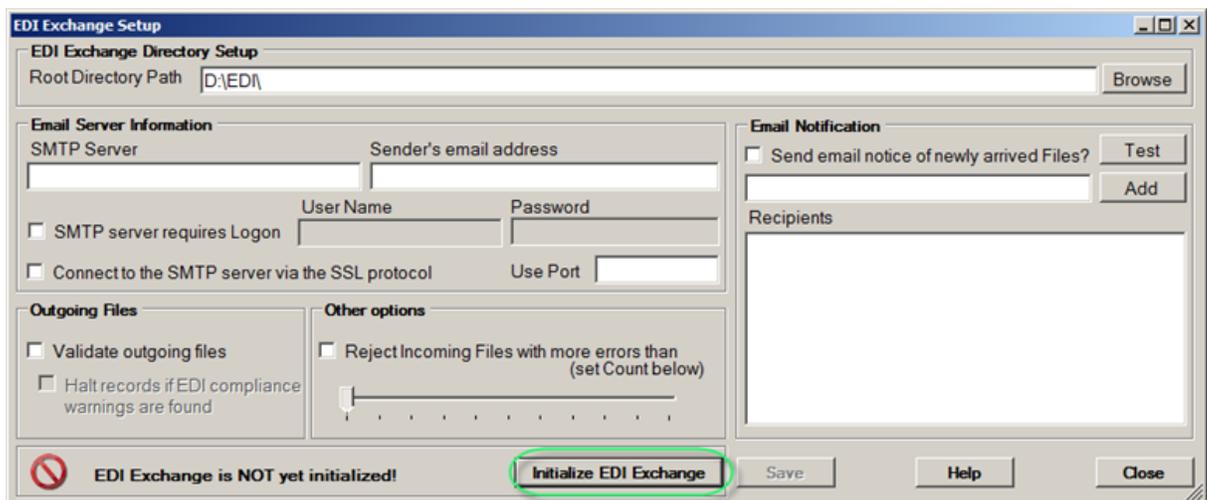
2. The following window will appear.



The "EDI Exchange Setup" window

Note: The icon in the lower left corner, indicating that EDI Exchange has not been initialized yet.

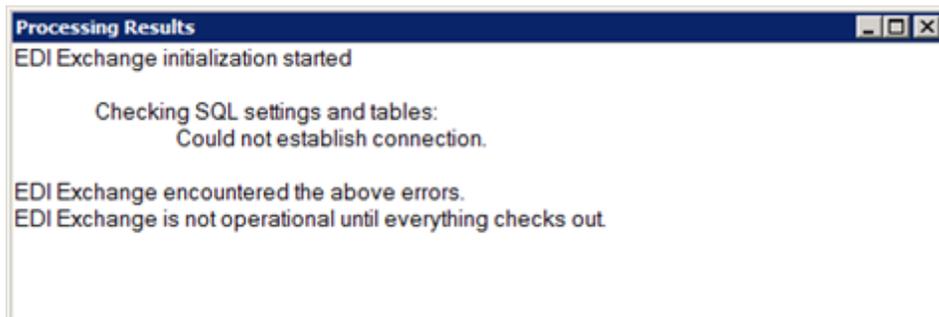
3. After setting the root directory (see the previous step [Defining Communications Directory](#)), you can initialize EDI Exchange module. Click the "Initialize EDI Exchange" button to accomplish the process.



The "Initialize EDI Exchange" button

Once the "Initialize EDI Exchange" button is clicked, the system checks if all settings have been configured correctly.

1. The first thing the initialization process checks is the connection to the database and the presence of the necessary tables. EDI Exchange relies on the database connection that is part of the HIPAAsuite application that you are using. EDI Exchange needs Database Connectivity licensed and enabled. If this part is not yet set up, then you will get an error like this:



Initialization failed because of SQL connection problems

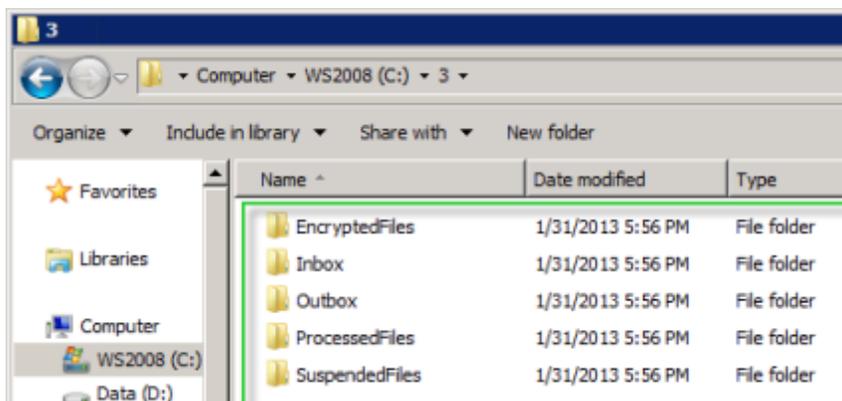
Read more in [Setting up Database Connection](#).

2. Once the connection is established, the program checks if the correct tables exist in the database. See [Creating Database Tables](#). Once the SQL part checks out, you will see the following message.

```
Checking SQL settings and tables:
  Connection settings are checked. Trading Partners table is checked.
  TRANS_LOG table is checked.
  TRANS_LOG_ERRORS table is checked.
  SEND_LOG table is checked.
```

3. The next step of the initialization processes – the program checks and, if necessary, creates the root directory and five sub-directories. Within these root directories, there will be folders for each Trading Partner and type of transactions. The location and names of the sub-folders are handled in the "Trading Partner" setup. The root folder is specified via the "EDI Exchange Setup" screen. Read more in [Defining Root Directory](#).

The sub-directories that EDI Exchange creates are as follows:



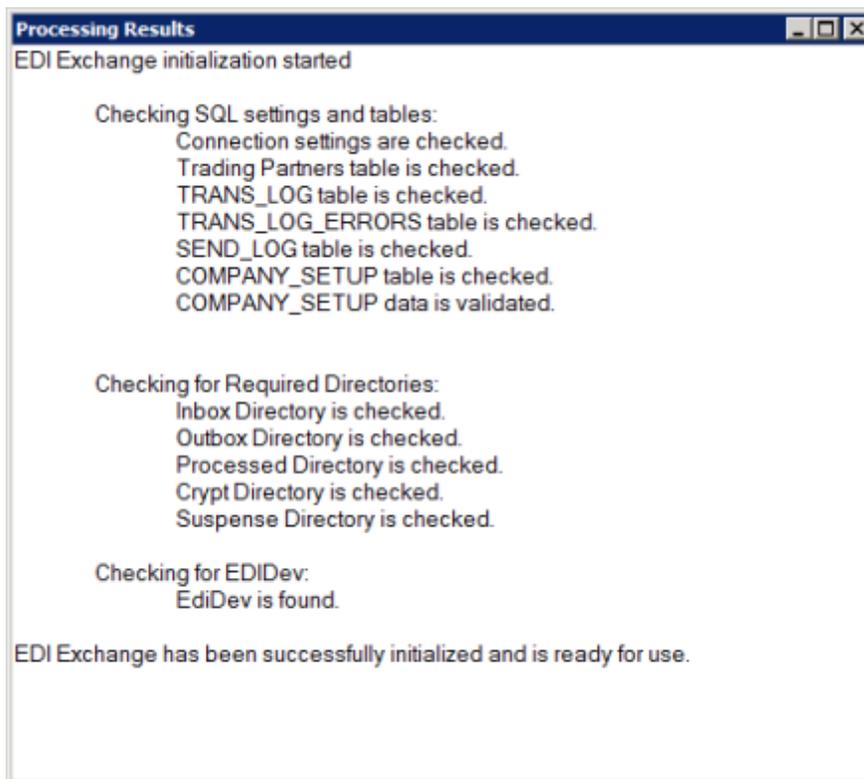
The directory structure of EDI Exchange

4. The next step of the initialization – the program checks if the HIPAA EDI compliance

engine is properly installed.

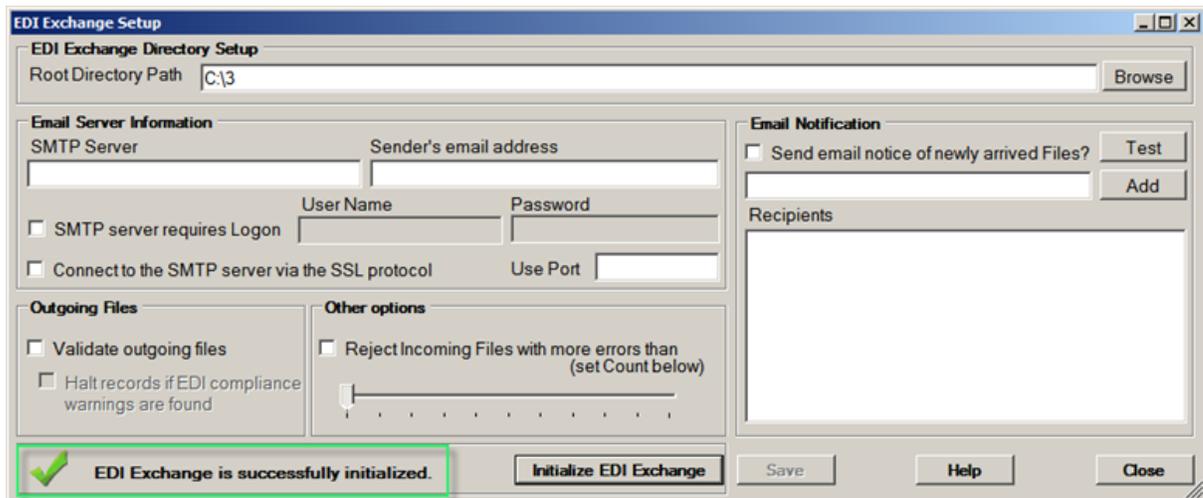
5. The last thing checked by the system is if the Automatic File Processing options have been set up. Read more in [Defining Auto-Processing Options](#). The options are important for the hand over from EDI Exchange to the other HIPAAsuite program that hosts EDI Exchange.

Once all verifications have been completed successfully, you will see the following message:



Successful initialization of EDI Exchange

After that your EDI Exchange is initialized.



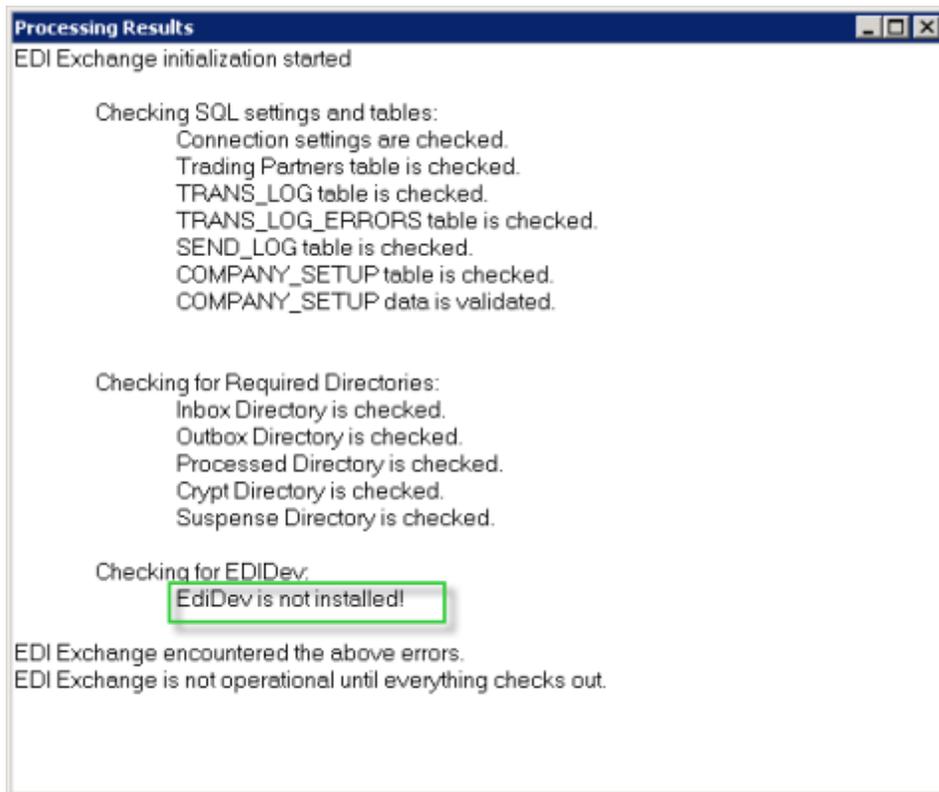
The "EDI Exchange is successfully initialized" message on the bottom of the "EDI Exchange Setup" window

Troubleshooting Initialization

When you are going through the Initialization process of the EDI Exchange, you can encounter the following message in the "Processing Results" window:

Checking for EDIDev:

EDIDev is not installed!



The EdiDev component is not installed

To resolve this issue, do the following.

1. Go to the <http://www.edidev.com/ediregis.htm> site.
2. Some systems may require Microsoft Redistributable Package to be installed first. In this case, download one of the following components according to your OS' bit depth.

The recommended minimum system requirement for Framework EDI:

- 256 MB RAM
- 1GB available disk space
- Windows 2000/2003/2008/XP/Vista/7
- Prerequisites: Some systems may require Microsoft Redistributable Package to be installed first to support:
 - Framework EDI.NET (32-bit) - download [Microsoft Visual C++ 2005 Redistributable Package \(x86\)](#)
 - Framework EDI.NET4 (32-bit) - download [Microsoft Visual C++ 2010 Redistributable Package \(x86\)](#)
 - Framework EDI.NET (64-bit) - download [Microsoft Visual C++ 2005 Redistributable Package \(x64\)](#)
 - Framework EDI.NET4 (64-bit) - download [Microsoft Visual C++ 2010 Redistributable Package \(x64\)](#)

3. Download one of the following components you need according to your OS' bit depth:

- Framework EDI Enterprise evaluation 32-bit

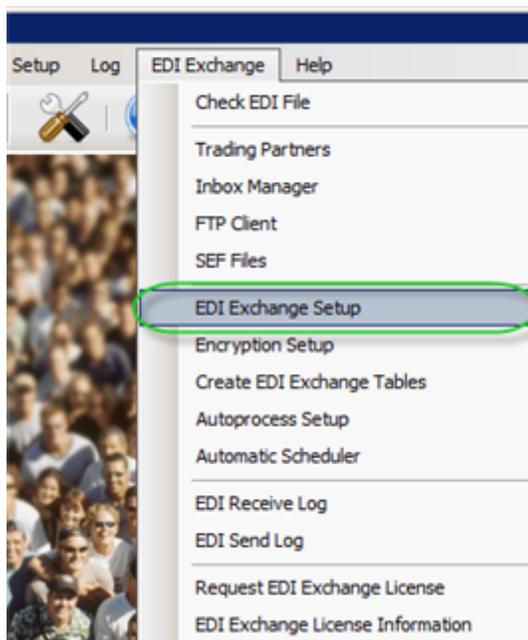
- Framework EDI Professional evaluation 64-bit
4. Install downloaded components and start again the EDI Exchange initialization procedure.

12.3 Configuring EDI Exchange (Optional Settings)

12.3.1 Setting up Email Notifications

EDI Exchange can send emails to operators and trading partners to notify them about files and processing results. This functionality of EDI Exchange lies beyond the most basic setup that the initialization checks for. For this feature to work properly, you need to set up an email server which EDI Exchange communicates with. You can do this in the "Email Server Information" frame of the setup screen. Follow the instructions below.

1. Select "EDI Exchange Setup" under the "EDI Exchange" menu item.



The "EDI Exchange Setup" menu item

2. The following window will appear.

The "EDI Exchange Setup" window

3. Specify the following email options:

Email Server Information

- SMTP server
- Sender's email address
- SMTP server requires logon
- Username
- Password
- Connect to the SMTP server via the SSL protocol
- Use port

Setting up the email server

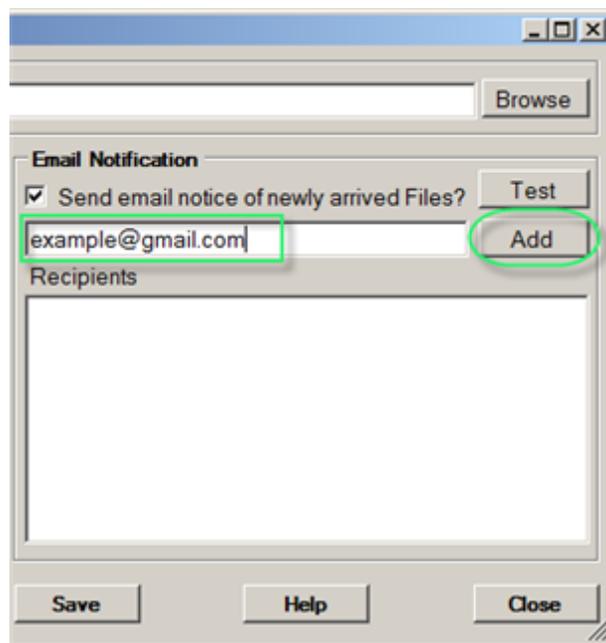
Note: If you do not have this information, please ask your administrator to set this portion up for you.

4. To configure email notifications, specify the following options:

Email Notification

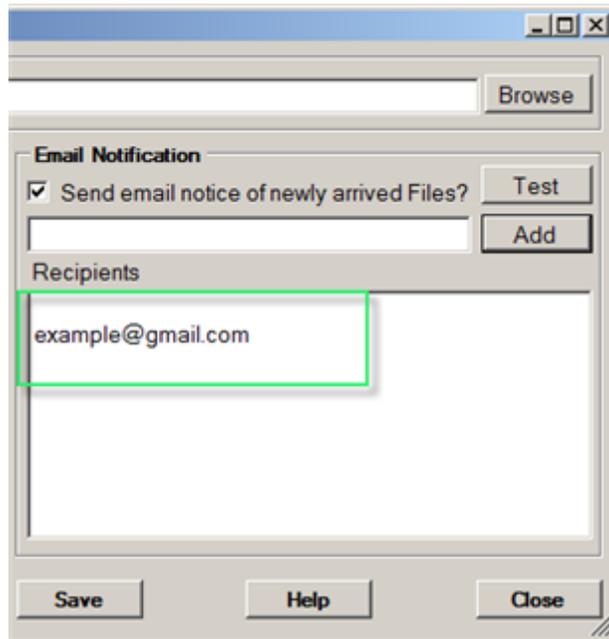
- **Send email notice of newly arrived files?** — Use this checkbox to define if you want to send email notice once new files have arrived.
- **Recipients** — You can add your staff's email(s) and the processing messages will be sent to these emails.

Enter an email address and click "Add."

The image shows a software dialog box titled "Email Notification". At the top, there is a "Browse" button. Below it, the "Email Notification" section contains a checked checkbox labeled "Send email notice of newly arrived Files?" and a "Test" button. A text input field contains the email address "example@gmail.com", which is highlighted with a green box. To the right of this field is an "Add" button, also highlighted with a green circle. Below the input field is a section labeled "Recipients" with an empty list box. At the bottom of the dialog are three buttons: "Save", "Help", and "Close".

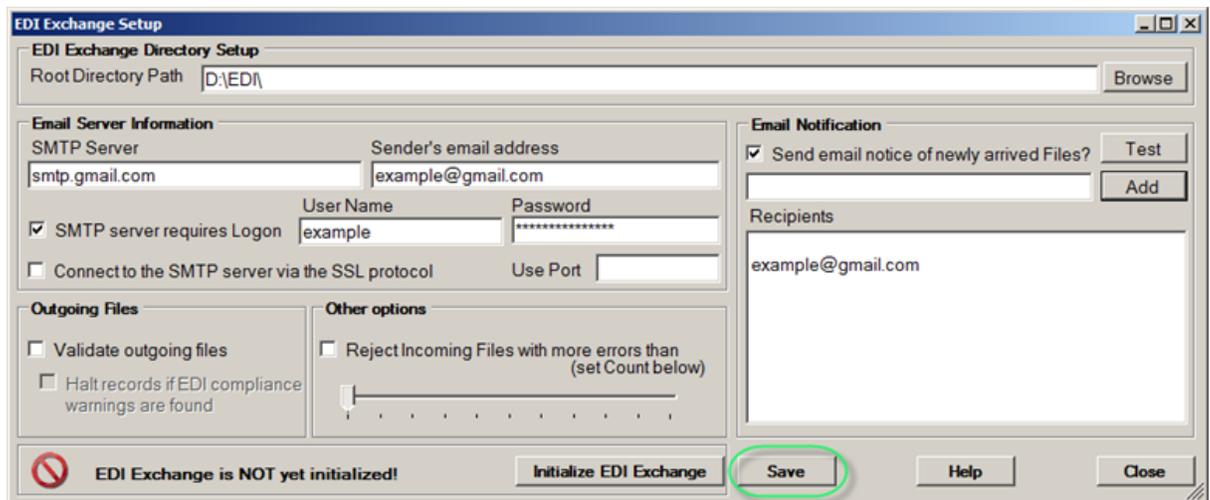
Setting up the email recipients

The email address will appear in the list.



Added email recipient

5. Click on the "Test" button to verify your settings.
6. Click on the "Save" button.



The "Save" button

Troubleshooting Email Settings

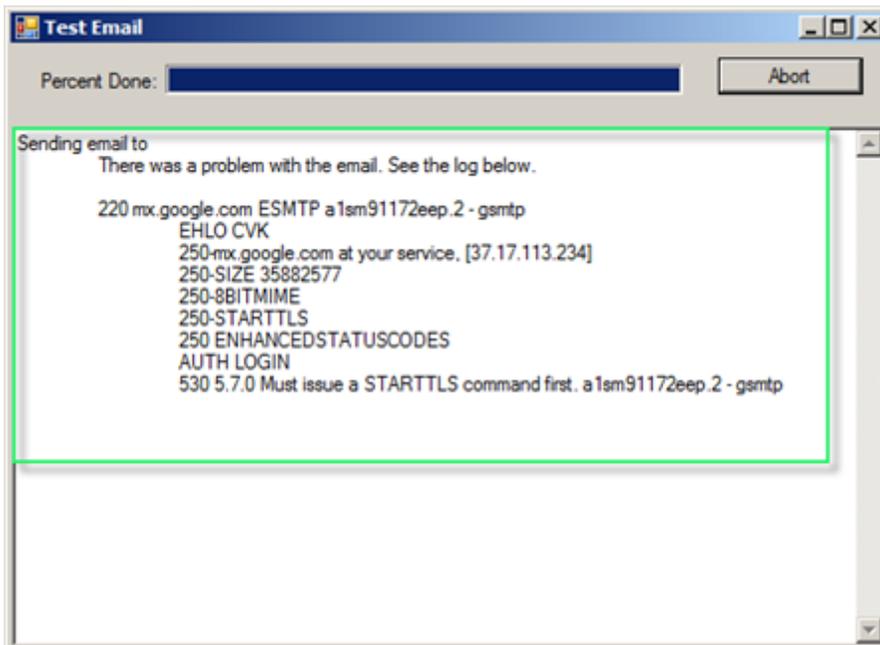
If you have problems with setting up the email server, please contact your administrator. He/She should know values to specify and how to test the settings.

Below is an example of what happens when the email server does not respond.



After a time out, you get a failure notice

After acknowledging the failure, you get a more detailed error message in the process result screen.



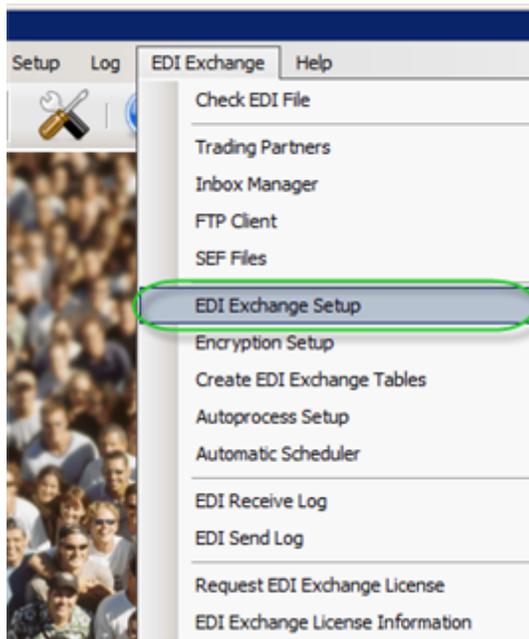
The process result screen with a detailed error message

12.3.2 Setting up Incoming and Outgoing Files Options

To check if your files are HIPAA-compliant, EDI Exchange can run a compliance check on outgoing EDI files. For incoming files, you can specify an acceptable error level. These settings are especially important with new trading partner relationships or with new processes since it always takes a while until an EDI process runs without issues and problems.

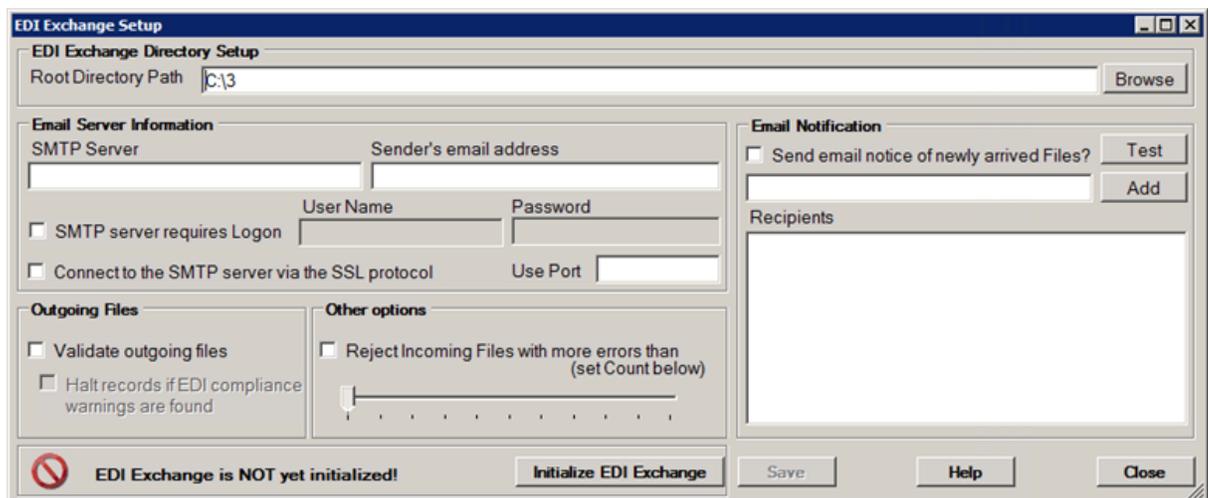
Follow the instructions below.

1. Select "EDI Exchange Setup" under the "EDI Exchange" menu item.



The "EDI Exchange Setup" menu item

2. The following window will appear.



The "EDI Exchange Setup" window

3. Specify the following outgoing files options:

Outgoing Files

- **Validate outgoing files** – Select this checkbox to validate if the outgoing files are HIPAA-compliant.
- **Halt records if EDI compliance warnings are found** – Select this checkbox to suppress the sending of files with warnings or errors.

EDI Exchange Setup

EDI Exchange Directory Setup
Root Directory Path: D:\EDI\

Email Server Information
SMTP Server: smtp.gmail.com
Sender's email address: example@gmail.com
User Name: example
Password: *****
 SMTP server requires Logon
 Connect to the SMTP server via the SSL protocol
Use Port:

Email Notification
 Send email notice of newly arrived Files?
Recipients: example@gmail.com

Outgoing Files
 Validate outgoing files
 Halt records if EDI compliance warnings are found

Other options
 Reject Incoming Files with more errors than (set Count below)

EDI Exchange is NOT yet initialized! Initialize EDI Exchange Save Help Close

Validating outgoing files

4. Specify the following incoming files options:

Other Options

- **Reject incoming files with more errors than (set count below)** – If checked, the incoming files with more errors than defined will not be placed into the "Inbox" folder.

EDI Exchange Setup

EDI Exchange Directory Setup
Root Directory Path: D:\EDI\

Email Server Information
SMTP Server: smtp.gmail.com
Sender's email address: example@gmail.com
User Name: example
Password: *****
 SMTP server requires Logon
 Connect to the SMTP server via the SSL protocol
Use Port:

Email Notification
 Send email notice of newly arrived Files?
Recipients: example@gmail.com

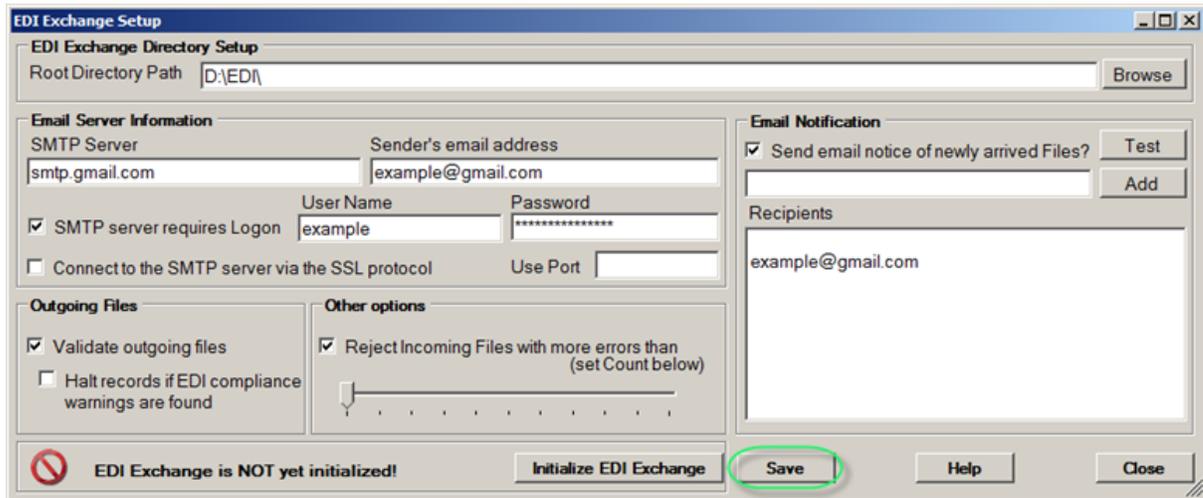
Outgoing Files
 Validate outgoing files
 Halt records if EDI compliance warnings are found

Other options
 Reject Incoming Files with more errors than (set Count below)

EDI Exchange is NOT yet initialized! Initialize EDI Exchange Save Help Close

The "Reject incoming files with more errors than" option

5. Click on "Save."

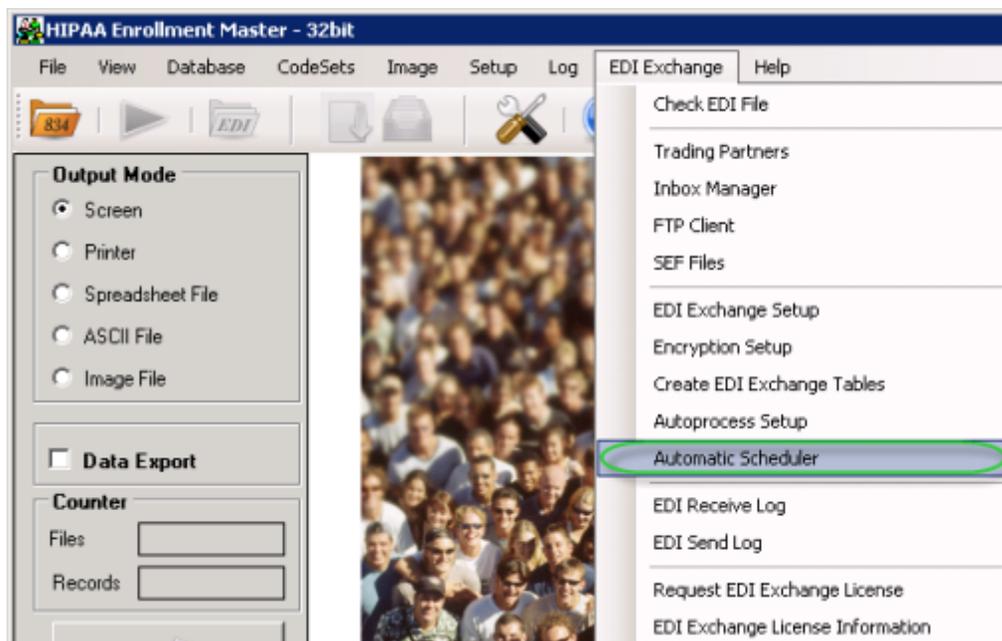


12.3.3 Running the Application via Scheduler

EDI Exchange integrates with the Windows Scheduler to allow the automation of the EDI file exchange process. You can set up EDI Exchange to go out to the trading partner's FTP server, download files, decrypt and compliance check them and further process them with the appropriate HIPAAsuite application, for example load claims into a SQL database.

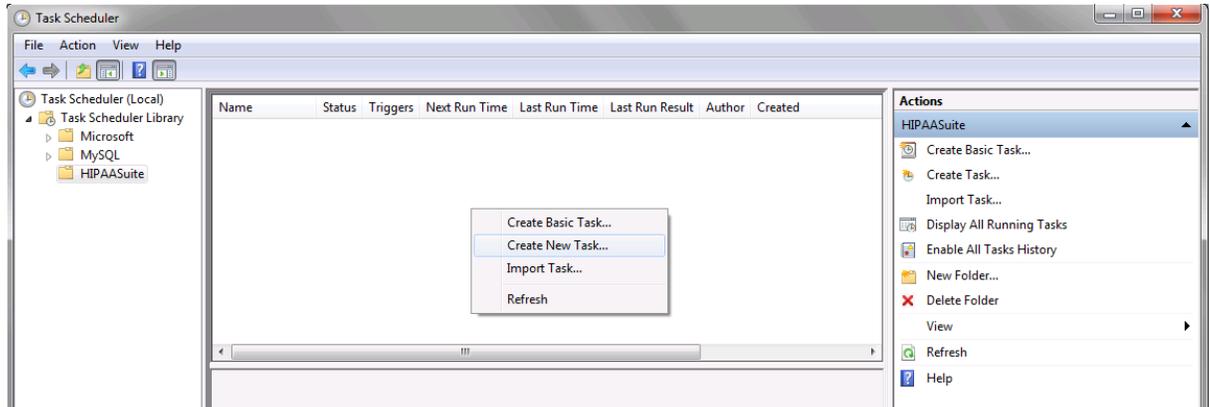
Follow the instructions below to schedule the EDI files exchange process.

1. Select "Automatic Scheduler" under the "EDI Exchange" menu.



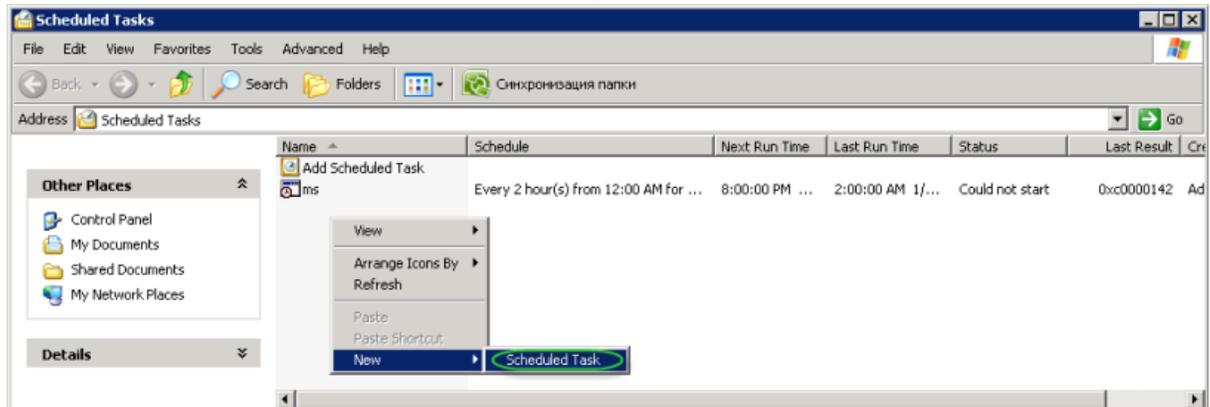
The "Automatic Scheduler" menu item

2. In the opened window, right click and choose the "Scheduled Task" menu item.



Task Scheduler in Windows Vista onwards.

In Windows XP, the Scheduled tasks directory looks like this:



The "Scheduled Task" menu item in Windows XP.

See "Running the Application via Scheduler" in the help of the host HIPAA application for detailed instructions on how to schedule a task.

12.3.4 Using the Command Line Arguments (CLI)

The only EDI Exchange specific command line argument is "Auto"

The Auto processing options are defined in another [screen](#) and, with the command line argument "Auto," they will be exercised. Make sure that you have configured them according to your needs.

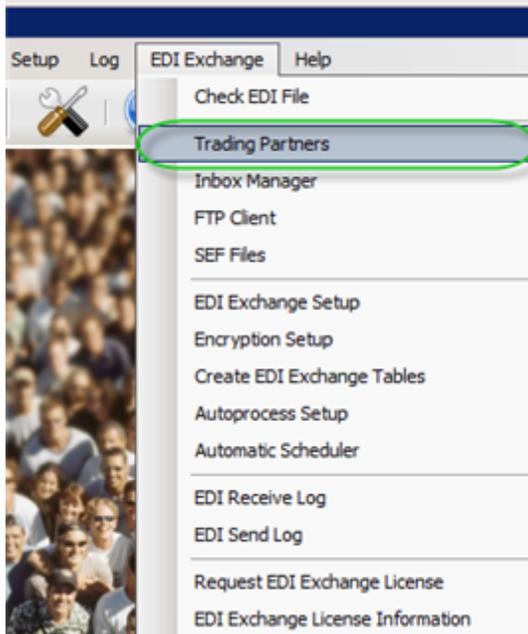
12.4 Working with Trading Partners

12.4.1 Setting up Trading Partners

With EDI Exchange you can keep track of your trading partners. You can set up their identifiers to send them EDI files or 999 acknowledgment, send email notifications and compliance check results, encryption keys and file transport mechanisms.

Once you have created the trading partner table (see [Creating Database Tables](#)) and initialized EDI Exchange (see [Initializing EDI Exchange](#)), you can set up the relationships with your trading partner. Follow the instructions below.

1. Select "Trading Partners" under the "EDI Exchange" menu.



The "Trading Partners" menu item

2. The following screen will appear.

The "Trading Partners" window

3. Click the "New" button to start entering the trading partner information.

4. Define the necessary options. They are described further.

The screenshot shows the 'Trading Partners' application window. The left pane contains the following fields and options:

- Name and Type:** Name (Required), Status (dropdown), Type (dropdown), Address, City, State, Zip.
- EDI File Exchange Method:** (dropdown, Required), with checkboxes for:
 - Send EDI Acknowledgement (997 or 999)
 - Send Email with Process Results
 - Notify when receiving Files
 - Notify when sending Files
 - Create a 277U for 837 Claims
- Email Addresses:** (text area)
- EDI Identifiers:** Remote FTP, Contact, Encryption, Folders (tabs)
- EDI Version:** Radio buttons for 4010 (selected) and 5010.
- EDI Identifiers and Qualifier:** ISA Identifier and Qualifier (Record Key) (Required), ETIN Number and Qualifier (with '46' entered), Application Sender Code (GS02) (Required), Application Receiver Code (GS03).

The right pane, 'List of Trading Partner', is empty. At the bottom of the window are buttons for 'Save', 'New', 'Delete', 'Refresh', and 'Close'.

5. Click on "Save."



The "Save" button

6. The newly added Trading Partner's name will appear in the right pane.

Trading Partner Options

The company information of a trading partner can be specified on the top of the form.

The screenshot shows the 'Trading Partners' window with the following fields and options:

- Name and Type**
 - Name: Text input field with a red asterisk and '(Required)' label.
 - Address: Text input field.
 - City: Text input field.
 - State: Text input field.
 - Zip: Text input field.
 - Status: Dropdown menu.
 - Type: Dropdown menu.
- EDI File Exchange Method**: Dropdown menu with a red asterisk.
- Email Addresses**: Text area for entering email addresses.
- Checkboxes**:
 - Send EDI Acknowledgement (997 or 999)
 - Send Email with Process Results
 - Notify when receiving Files
 - Notify when sending Files
 - Create a 277U for 837 Claims

Top area of the "Trading Partners" window

Name and Type

- **Name** – Trading partner's company name. Required field.
- **Address** – Trading partner's company address.
- **City**
- **State**
- **Zip**
- **Status** – Trading partner's status. Choose one of the available options:
 - **Inactive** – No upload into a database system through ODBC will be done.
 - **Test Only** – All outgoing EDI messages will be stamped with "Test" (ISA_15). Records will only be exported to the test environment.
 - **Approved** – All outgoing EDI messages will be stamped with "Production" (ISA_15). Records will be exported to the Live system with ODBC.
- **Type** – There are five types of trading partners, select the necessary one:
 - **Providers** – Hospitals, doctors or other health care providers.
 - **VANs** – Value Added Networks like clearing houses or EDI Networks.
 - **Service bureaus** – Third party entities such as repricing organizations.
 - **Sponsors** – Entities that sponsor the benefits of subscribers such as Medicaid, government agencies or large employers.

- **Payers** – Entities that pay for health care benefits such as health insurers.

The next block on the form lists the communication methods and preferences.

- **EDI File Exchange Method** – Obligatory setting. HIPAAsuite supports three communication methods. Choose a preferred mode of sending EDI communications to the trading partner:
 - **FTP** – Allows you to transmit files actively to the Trading Partner or his Clearinghouse.
 - **Outbox** – All files for Trading Partner are stored locally. The Trading Partner is responsible for picking up files in his special directory of the local FTP or HTTP server.
 - **SOAP** - Allows you to transmit files to the Trading Partner or his Clearinghouse using SOAP 1.2 (CORE) by default.
 - **SOAP Version** - Defaults to CAQH CORE "SOAP 1.2". "SOAP 1.1" is also provided as a compatibility option for Trading Partners that are not yet CORE-Certified but use SOAP services.
 - **MIME** - Allows you to transmit files to the Trading Partner or his Clearinghouse using MIME (CORE). This is one of two CORE options.
- **Email Addresses** – Enter the email address(es) into the text field.

EDI Identifiers Tab

This tab relates to the EDI identifiers and EDI Version.

The screenshot shows a software interface with a tabbed menu at the top. The 'EDI Identifiers' tab is selected and highlighted with a green circle. Below the tabs, there are two main sections. The first section, 'EDI Version', has two radio buttons: '4010' (which is selected) and '5010'. The second section, 'EDI Identifiers and Qualifier', contains several input fields. The 'ISA Identifier and Qualifier (Record Key)' field has 'TXEBS' in the first box and 'ZZ' in the second box, with a red asterisk to the right. Below this, there are three rows of fields: 'ETIN Number and Qualifier' with '46' in the second box and a red asterisk to the right; 'Application Sender Code (GS02)' with 'TXEBS' in the box and a red asterisk to the right; and 'Application Receiver Code (GS03)' and 'Trading Partner Assigned ID' with empty boxes.

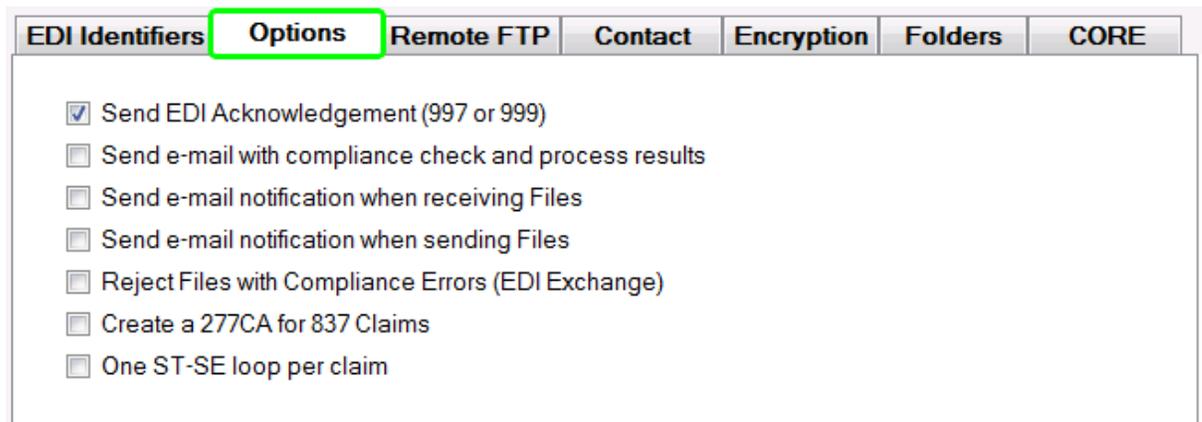
The "EDI Identifiers" tab

- **EDI Version** – There are two standards for HIPAA:
 - 4010 – This standard was introduced in the original transaction from 2003 to 2011.
 - 5010 – From 2012 on all HIPAA transactions must be conducted in the 5010 version.
- **EDI Identifiers and Qualifier**
 - **ISA Identifier and Qualifier (Record Key)** – The ISA Identifier and Qualifier are the unique key to the trading partner database file. The ISA identifier can be up to 15 bytes long, the qualifier has to be 2 bytes. Approved qualifiers are:
 - 01 – Duns (Dun and Bradstreet)
 - 14 – Duns Plus Suffix
 - 20 – Health Industry Number (HIN)
 - 27 – Carrier Identification Number as assigned by HCFA
 - 28 – Fiscal Intermediary Identification Number as assigned by HCFA
 - 29 – Medicare Provider and Supplier Identification Number as assigned by HCFA
 - 30 – U.S. Federal Tax Identification Number
 - 33 – National Association of Insurance Commissioners Company Code (NAIC)
 - ZZ – Mutually Defined. Many organizations use the ZZ qualifier with their

name as the ID, for example ZZ and HIPAASUITE.

- **ETIN number** – The Electronic Transmitter Identification Number established by a Trading Partner Agreement. This number occurs only in the 837 transactions. Often, the ETIN is same as the ISA ID.
- **Application Receiver Code (GS_02)** – A code identifying a part that sends a transmission or the specific application within the sender's organization. Codes are agreed by Trading Partners. Again, usually this code is same as the ISA ID. This code is placed in the GS_02 element in the Functional Group Header (GS). Some Trading Partners want to send a specific code in GS_03, the application receiver code. You can enter it into the corresponding field. Most of the time it is not necessary.
- **Assigned ID** - Some trading partners, like health insurance exchanges will give a plan an ID that is different from the ISA identifier defined in the Company Setup screen of the application. Especially in the creation of filenames is this Assigned ID important.

Options Tab



EDI Identifiers	Options	Remote FTP	Contact	Encryption	Folders	CORE
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/>					
	<input type="checkbox"/>					
	<input type="checkbox"/>					
	<input type="checkbox"/>					
	<input type="checkbox"/>					
	<input type="checkbox"/>					
	<input type="checkbox"/>					

The "Options" Tab

You can choose one of the following options:

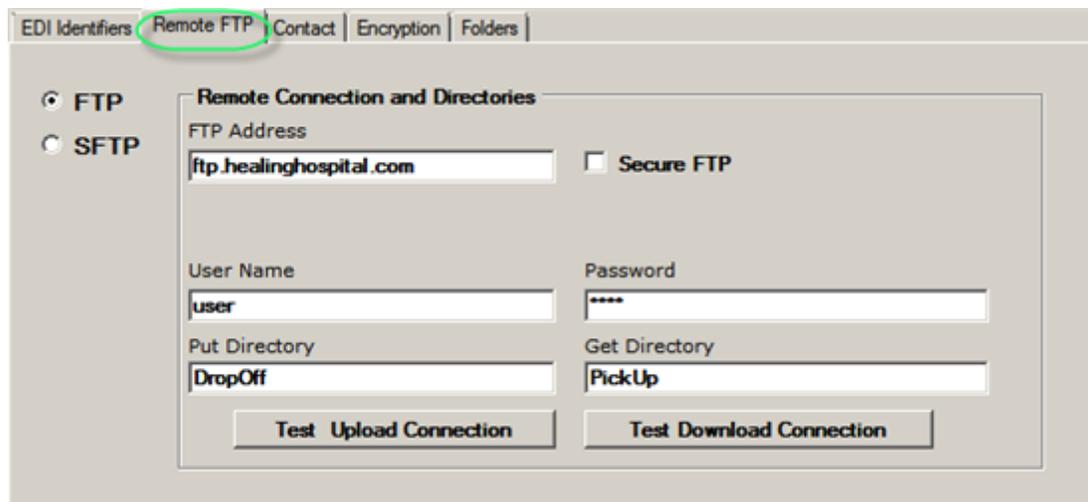
- **Send EDI Acknowledgment (997 or 999)** – This check-box allows sending Functional Acknowledgment transactions to the Trading Partner.
- **Send e-mail with compliance check and process results** – This option allows sending the compliance check results back to your contact at the Trading Partner via email. No PMI will be transmitted. Adding an email address is important, even when the Communication method is not "Email."
- **Send e-mail notification when receiving files** – This option allows sending an

acknowledgment email of EDI files. This option is not necessary when you choose 997 or 999 acknowledgments.

- **Send e-mail notification when sending files** – This option allows sending a file to the Trading Partner notifying them that a file has been created for them.
- **Reject Files with Compliance Errors** - This option will reject files that have compliance warnings and move them into the suspended files directory. It also determines whether the TA1 and 999 indicate acceptance or the 999 lists all the errors and warning.
- **Create a 277CA for 837 claims** – This option only applies to 837 Claims. Checking this option will produce a 277CA Claims Acknowledgment report for received 837 Claim files.
- **One ST-SE loop per claim** - This option separates all claims into individual transactions enclosed by their own ST and SE segments.

Remote FTP Tab

If your trading partner has an FTP Server, then you can set up here the connection information. Read more in [Using Built-in FTP Client](#).



The "Remote FTP" tab

The file transfer protocol (FTP) is one of the first internet protocols and goes back to the 1960's. Transporting electronic files was one of the great achievements of the internet. During the last 50 years a lot of improvements to this protocol have been made, mainly to increase the security of the transfer.

- **FTP** – For security reasons, EDI Exchange supports secure FTP or FTPs.
 - **Explicit FTPS Connection** – The explicit method is a legacy compatible

implementation where FTPS aware clients can invoke security with an FTPS aware server without breaking overall FTP functionality with non-FTPS aware clients. In explicit mode (also known as FTPS), an FTPS client must "explicitly request" security from an FTPS server and then step-up to a mutually agreed encryption method. If a client does not request security, the FTPS server can either allow the client to continue insecure or refuse/limit the connection.

- **Implicit FTPS Connection** – The implicit method requires that all clients of the FTPS server be aware that SSL is to be used on the session, and thus is incompatible with non-FTPS-aware clients. Negotiation is not allowed with implicit FTPS configurations. A client is immediately expected to challenge the FTPS server with a TLS/SSL ClientHello message. If such a message is not received by the FTPS server, the server should drop the connection. In order to maintain compatibility with existing non-TLS/SSL aware FTP clients, implicit FTPS was expected to listen on the IANA Well Known Port 990/TCP for the FTPS control channel and 989/TCP for the FTPS data channel. This allowed administrators to retain legacy compatible services on the original 21/TCP FTP control channel.
- **SFTP** – also known as FTP over SSH is deemed the most secure form of FTP and uses encryption certificates. There are 3 different ways to authenticate a SFTP connection,
 - With user name and password, just like a regular FTP connection,
 - User name and a certificate
 - User name, certificate and password

Remote Connection and Directories

You need the FTP address, the user name and password to establish the connection and the directory information where files are picked up and where dropped off. Fill in the following fields:

- **FTP Address** - This is usually the IP address of the server
- **User Name**
- **Password**
- **Put Directory** - This is the directory where you drop off files
- **Get Directory** - This is the directory where you download files from

It is possible that a trading partner has two FTP servers, one for 'put' and another one for 'get'. If so, check "Different Download Server" and additional fields will

become visible so you can specify the those connection parameters.

After Download: You have two choices. Files on the server will be either deleted or moved to another folder of your choice .

Contact Tab

The Contact Tab stores contact information for your selected Trading Partner.

Contact Information		Communication Numbers	
Name	<input type="text"/>	Type	Number
Communication Type	<input type="text"/>	Telephone	1234567
Number	<input type="text"/>	Extension	1111111
<input type="button" value="Add"/>		Fax	9876543

Local Access for FTP and HIPAA Suite Web		Registered Users	
User Name	<input type="text"/>	User	Password
Password	<input type="text"/>		
<input type="button" value="Add"/>			

The "Contact" tab

Contact Information

Enter the name and the number into the corresponding fields and select the communication type from the drop-down list.

- **Contact Name**
- **Communication Numbers** – Valid Communication number qualifiers are:
 - **TE** – Stands for Telephone.
 - **FX** – Stands for Fax.
 - **EX** – Stands for Extension.
 - **EM** – Stands for email.

Click the "Add" button and the contact will appear in the "Communication Numbers" table.

Note: The information that you fill in goes also into EDI files in the "PER" segment.

Local Access for FTP and HIPAA Suite Web

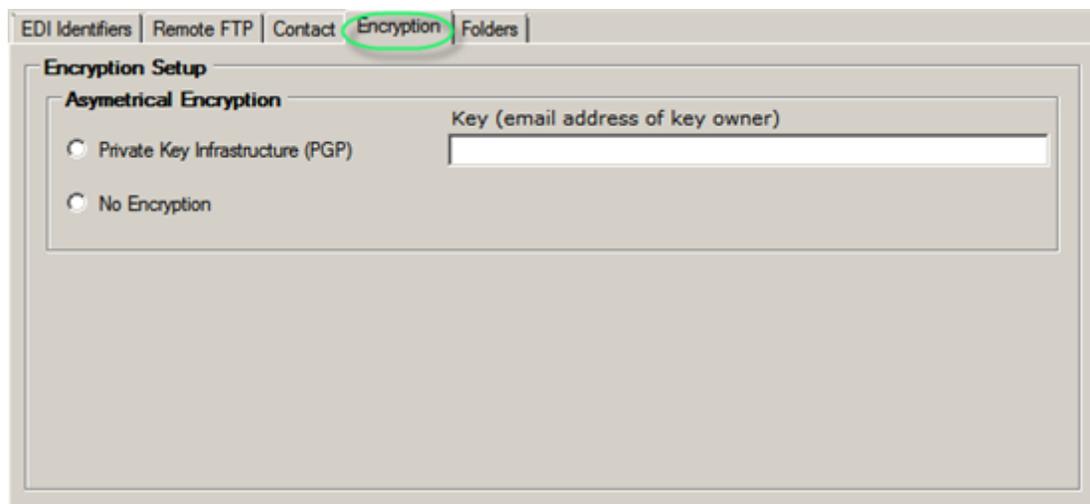
To register a user, enter the user name and the password into the corresponding fields and click the "Add" button. The user will appear in the "Registered Users" table.

- User Name
- Password
- Registered Users

Encryption Tab

EDI Exchange supports PKI encryption. Encryption keys are defined by the email address of the owner. Both supported products, PGP and GnuPG use this logic. Read more in [Using Encryption](#).

You can set up the encryption parameters for a Trading Partner on the "Encryption" tab.



The screenshot shows a software interface with several tabs: "EDI Identifiers", "Remote FTP", "Contact", "Encryption", and "Folders". The "Encryption" tab is selected and highlighted with a green circle. Below the tabs, the "Encryption Setup" section is visible. It contains a sub-section titled "Asymmetrical Encryption" with two radio button options: "Private Key Infrastructure (PGP)" and "No Encryption". To the right of these options is a text input field labeled "Key (email address of key owner)".

The "Encryption" tab

Encryption Setup

- Asymmetrical Encryption
 - Private Key Infrastructure (PGP)
 - Key (email address of key owner)
 - No Encryption

Folders Tab

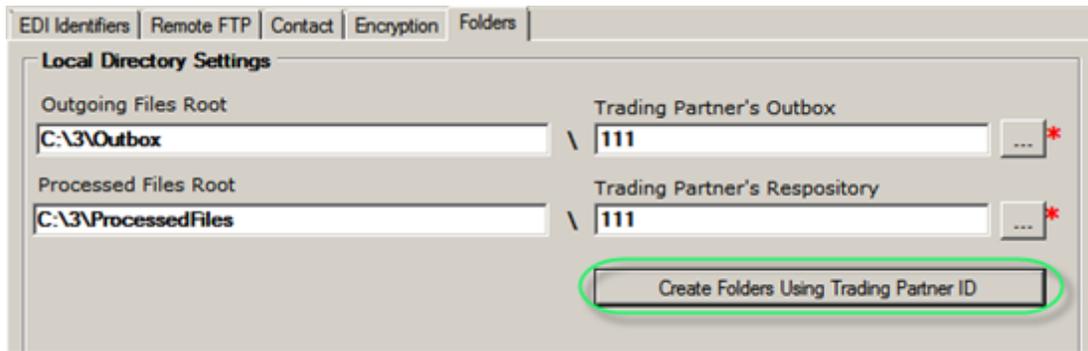
Within the HIPAAsuite Communications Directory, each Trading Partner has his own folder. This keeps files finely separated and in order. Here is where you can set this up. Read more in [Defining Communications Directory](#).

The screenshot shows the 'Folders' tab of the HIPAAsuite Communications Directory configuration. The 'Local Directory Settings' section is active. It contains four text input fields: 'Outgoing Files Root' with the value 'D:\HIPAAsuiteCommDir\Outbox', 'Processed Files Root' with 'D:\HIPAAsuiteCommDir\ProcessedFiles', 'Trading Partner's Outbox' with 'TXEBS', and 'Trading Partner's Repository' with 'TXEBS'. Each of the last two fields has a three-dot button to its right. Below these fields is a 'File Naming Convention' dropdown menu set to 'CMS'. At the bottom right of the settings area is a button labeled 'Create Folders Using Trading Partner ID'.

The "Folders" tab

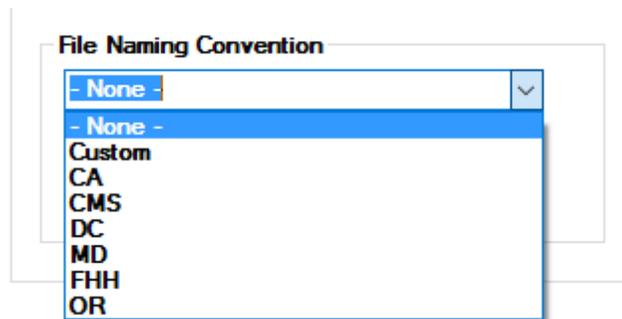
- **Outgoing Files Root** – This field has a pre-generated path. You can change this path by changing the EDI Root Directory.
- **Trading Partner's Outbox** – Mandatory setting. Click on the three-dots button to access the "Select Folder" window. There you choose an existing folder or create a new one.
- **Processed Files Root** – This field has a pre-generated path. You can change this path by changing the EDI Root Directory.
- **Trading Partner's Repository** – Mandatory setting. Click on the three-dots button to access the "Select Folder" window. There you choose an existing folder or create a new one.

Once you have specified the directory settings, click on the "Create Folders Using Trading Partner ID."



The "Create Folders Using Trading Partner ID" button

- **File Naming Convention** - Health Insurance Exchanges (HIX) demand that a carrier adheres to more or less complex File naming conventions. Since these conventions are often really complicated we decided to hard code several schemes. California, Maryland, DC and the CMS scheme are among those currently configured and we will add other schemes if needed.



File naming dropdown menu options

The *Custom* File Naming Convention option requires some additional setup. This is covered in [Custom File Naming Conventions](#).

CORE Tab

This tab stores settings for the use of CORE-Compliant SOAP- and MIME-enveloped transactions. When using a Requester or similar application, these settings apply to the information source. When using a Responder or similar application, these settings apply to the information requester/receiver. Soap 1.2 or 1.1 will use the same options.

The screenshot shows a software interface with a tabbed menu at the top. The tabs are: EDI Identifiers, Options, Remote FTP, Contact, Encryption, Folders, and CORE. The CORE tab is selected and highlighted with a green border. Below the tabs, the CORE Settings section is visible. It contains the following fields and buttons:

- CORE Settings**
 - UserName:
 - Password:
 - SSL Certificate:
- Real Time**
 - MIME Address: **Test**
 - SOAP Address:
- Batch**
 - MIME Submission Address: **Test**
 - MIME Retrieval Address:
 - SOAP Submission Address:
 - SOAP Retrieval Address:

CORE settings tab

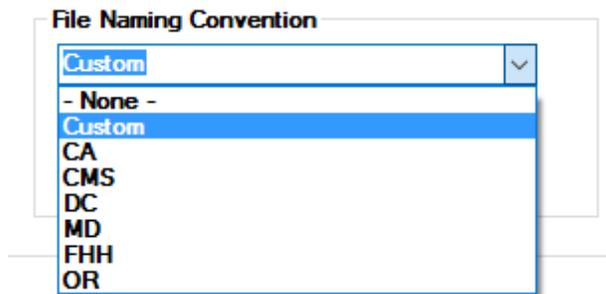
- **UserName** - UserName portion of the username authentication token. Used to verify a Trading Partner's Username token's Username or in your own Username token authenticate yourself to a Trading Partner's CORE-compliant service.
- **Password** - Password portion of the username authentication token. Used to verify a Trading Partner's Username token's Password or in your own Username token to authenticate yourself to a Trading Partner's CORE-compliant service.
- **SSL Certificate** - Instead of Username tokens, use an SSL certificate to verify a Trading Partner's identity or access a Trading Partner's CORE-compliant service. Not currently implemented.
- **RealTime**
 - **MIME Address** - Trading Partner's web address for MIME Real-Time transactions.
 - **SOAP Address** - Trading Partner's web address for SOAP Real-Time transactions.
- **Batch**
 - **MIME Submission Address** - Trading Partner's web address for MIME Batch transactions.
 - **MIME Retrieval Address** - Some Trading Partners may use a different address to submit or retrieve batch transactions. Use this field for a retrieval-specific address.
 - **SOAP Submission Address** - Trading Partner's web address for SOAP Batch

transactions.

- **SOAP Retrieval Address** - Some Trading Partners may use a different address to submit or retrieve batch transactions. Use this field for a retrieval-specific address.

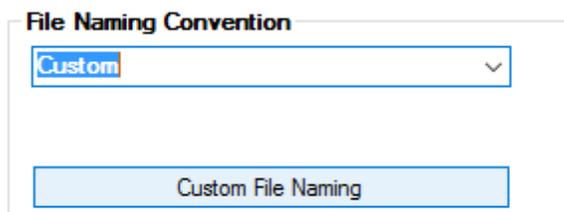
12.4.2 Custom File Naming Conventions

Health Insurance Exchanges (HIX) demand that a carrier adheres to File naming conventions. These file naming conventions enable a reader to quickly determine the date, time, sender, recipient, etc. of a particular file among other files without the need to open them individually. The custom File Naming Convention option enables you to create a file naming convention scheme tailored to your (or your trading partner's) requirements.



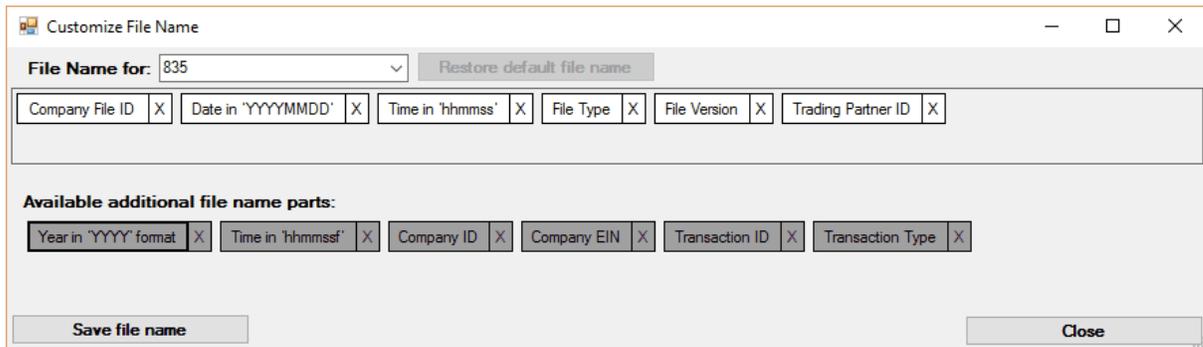
Custom File Naming Convention option highlighted

Selecting the Custom option in the File Naming Convention dropdown menu option and clicking the *Custom File Naming* button below it:



Custom File Naming button enabled

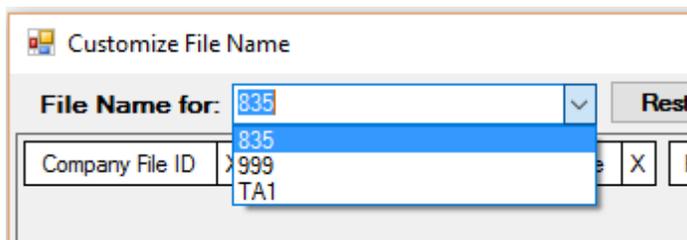
will bring up the screen pictured below. This screen is used to create file naming conventions for a particular trading partner. These examples will use the 835 file type as an example, but all file types the HIPAASuite product represents (as well as TA1s and 999s) can have their own file naming convention applied to each trading partner. The following picture shows the file name customization screen with the default building blocks for an 835 EDI file.



File Naming Customization screen with default custom file naming convention

Filetype

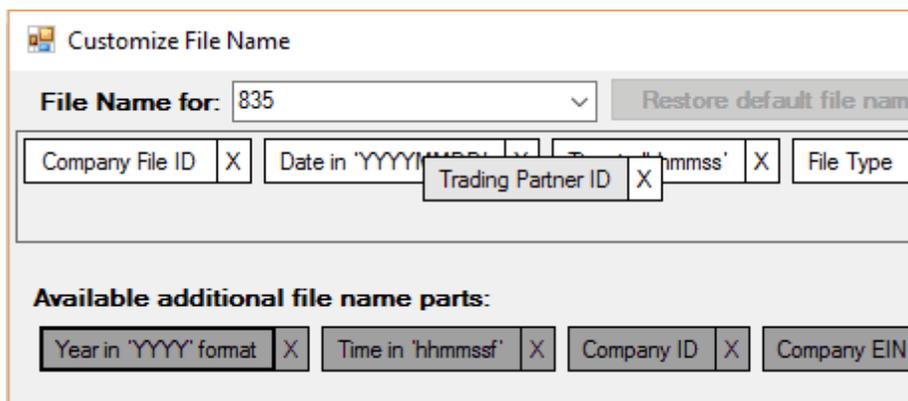
Here we can create a custom file naming convention for a specific file type. The file types available will depend on the HIPAASuite product used to create the file naming convention. To change the file type, select it from the file type dropdown menu:



Changing the filetype

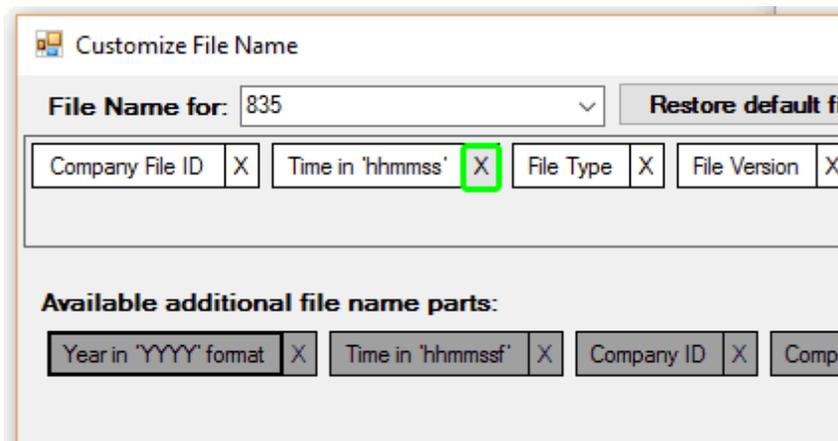
Creating the file naming convention

A coherent file naming convention can be created by clicking and dragging elements into order. Present elements can be reordered by clicking and dragging them into position:



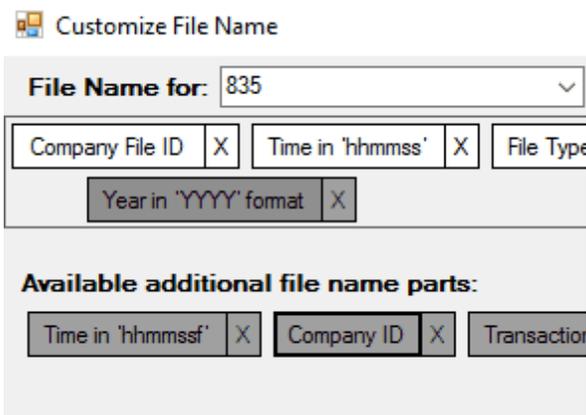
Click-dragging an element into position

Existing elements can be removed by clicking the X to the right of the element:



Deleting an element

The greyed-out elements towards the bottom of the screen can be added to the custom file naming convention by clicking and dragging them into place:



Adding a file name element

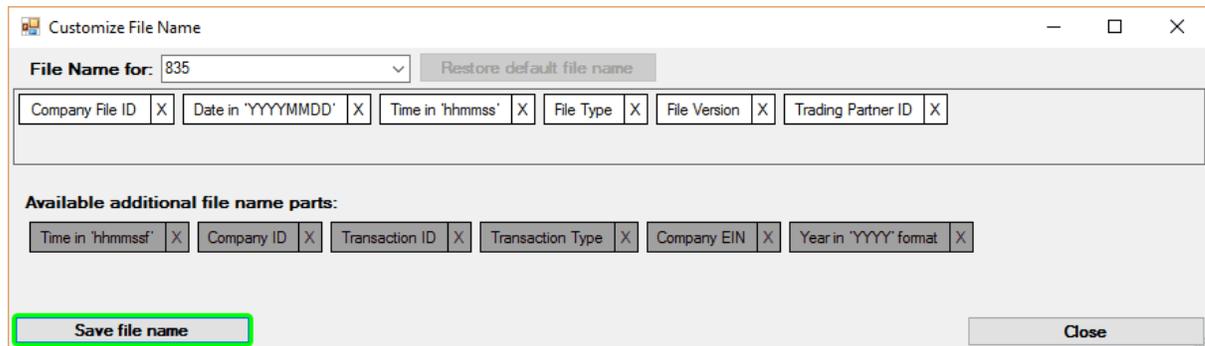
To discard the current changes and start over with the default file naming convention, click the *Restore default file name* button.



Restore default file name button

Saving or discarding

Finally, the custom file naming convention can be saved by clicking *Save file name* or discarded by clicking *Close*.



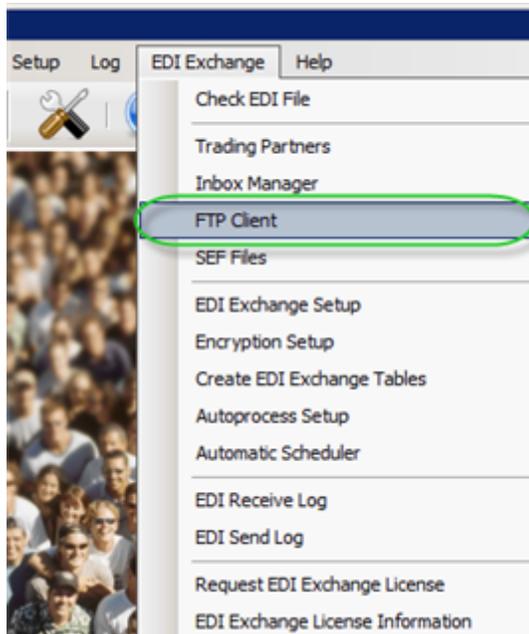
Save button on file name customization screen

12.4.3 Using Built-in FTP Client

EDI Exchange has a built-in FTP client. This utility allows you to drop-off and pick-up files from a trading partner that you have defined in the "Trading Partners" menu (see [Setting up Trading Partners.](#))

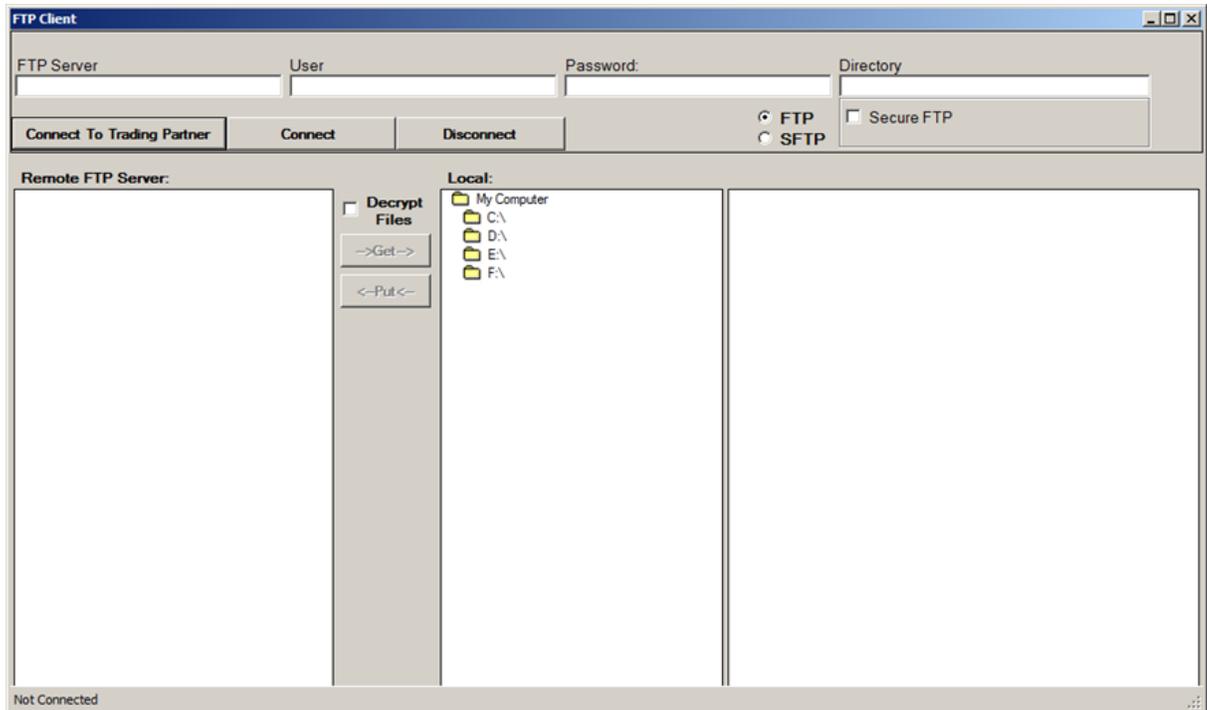
Follow the instructions below to exchange EDI files with your Trading Partner.

1. To access the FTP client, select "FTP Client" under the "EDI Exchange" menu item.



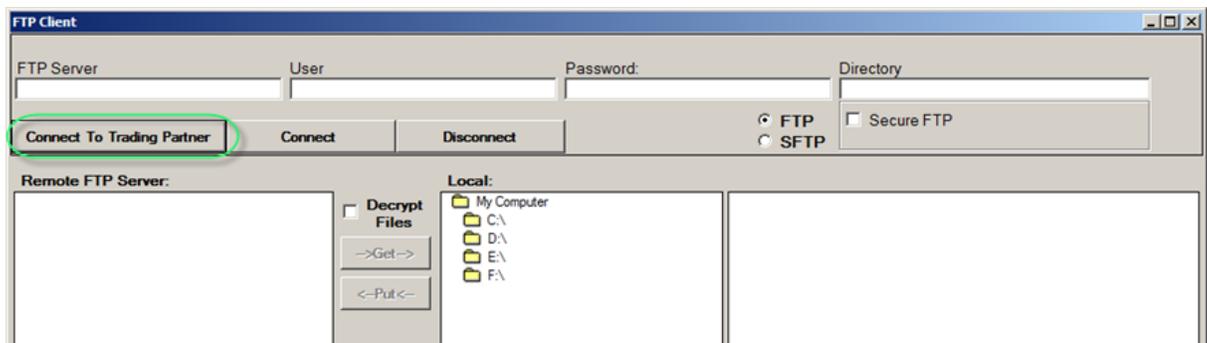
The "FTP Client" menu item

2. The following window will appear.



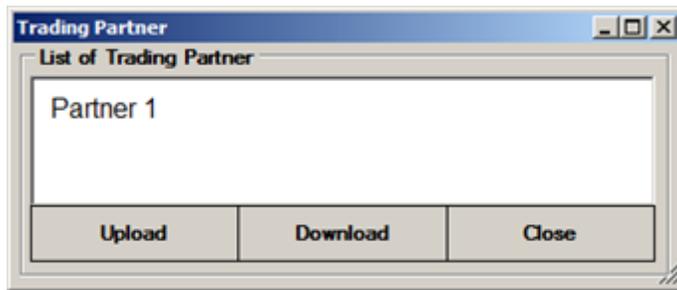
The built-in FTP client

3. Click the "Connect to Trading Partner" button on the FTP client window.



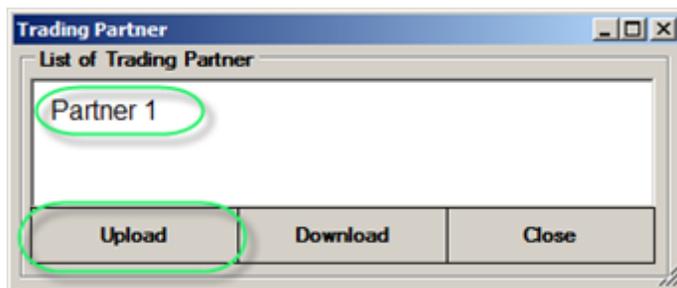
The "Connect to Trading Partner" button

4. The following screen opens.



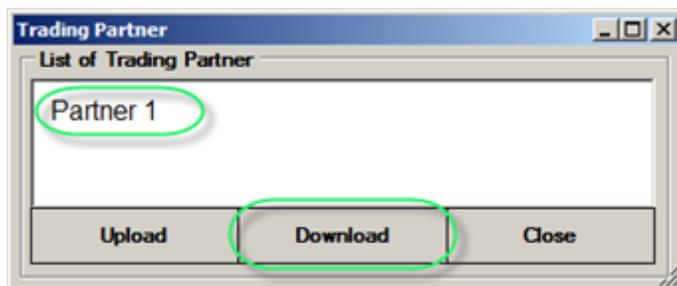
Selecting a Trading Partner for FTP transfer

5. Highlight the trading partner that you want to connect to.
6. Click on the "Upload" button to connect to the "Put" directory that you set up in the trading partner screen.



The "Upload" button

Or click on the "Download" button to connect to the "Get" directory that you set up in the trading partner screen.

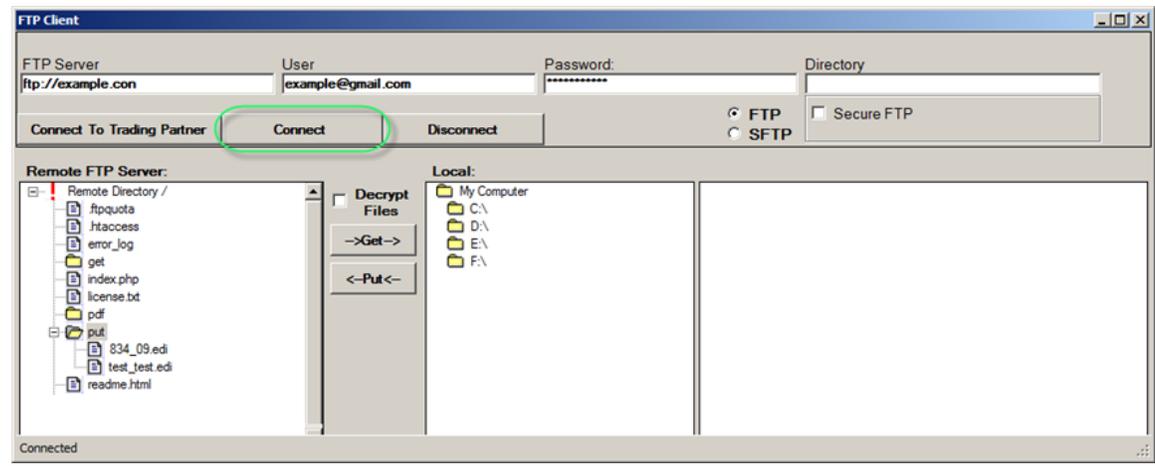


The "Download" button

5. Once you have chosen a trading partner, the following fields will contain values derived from the trading partner's properties. You can change them manually if you need.
 - **FTP server** – IP Address or URL of the FTP server.
 - **Username**
 - **Password**

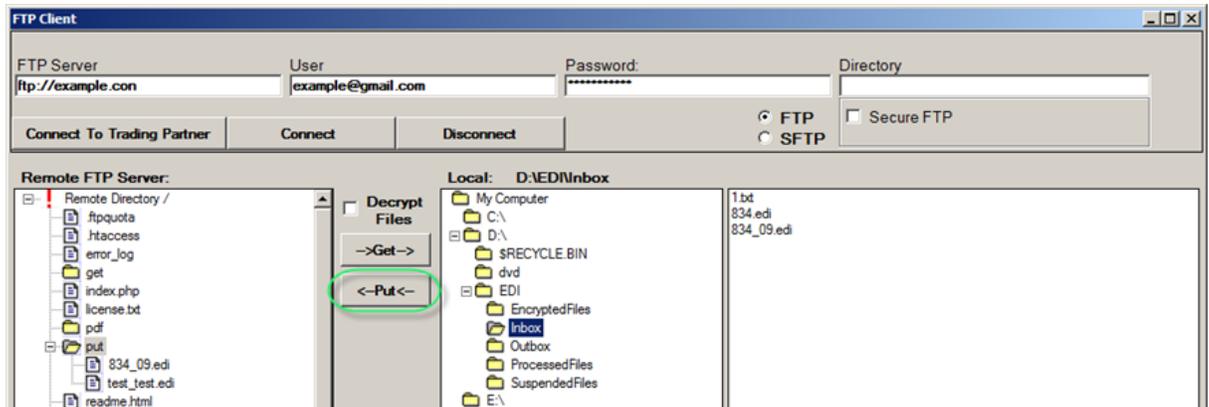
- **Directory** – If you leave this value blank, the FTP root directory will be opened.
- **Secure FTP** – If you enable this option, then you can select between implicit and explicit FTPs. See [Setting up Trading Partners](#) for an explanation of the two secure methods:
 - **Explicit FTPS Connection**
 - **Implicit FTPS Connection**

Note: You can also fill in the connection information manually. In this case, the entered credentials will not be saved once you close the FTP client window. Click on the "Connect" button to establish connection to the FTP server.



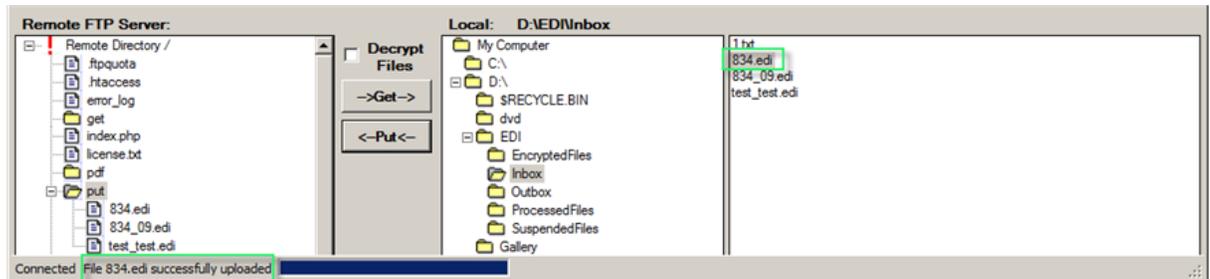
4. Once connected, the content of the folder on the server is displayed. On the right side, you can browse your local PC.

5. To upload a file to the server, select the file on your local PC and click on the "Put" button.

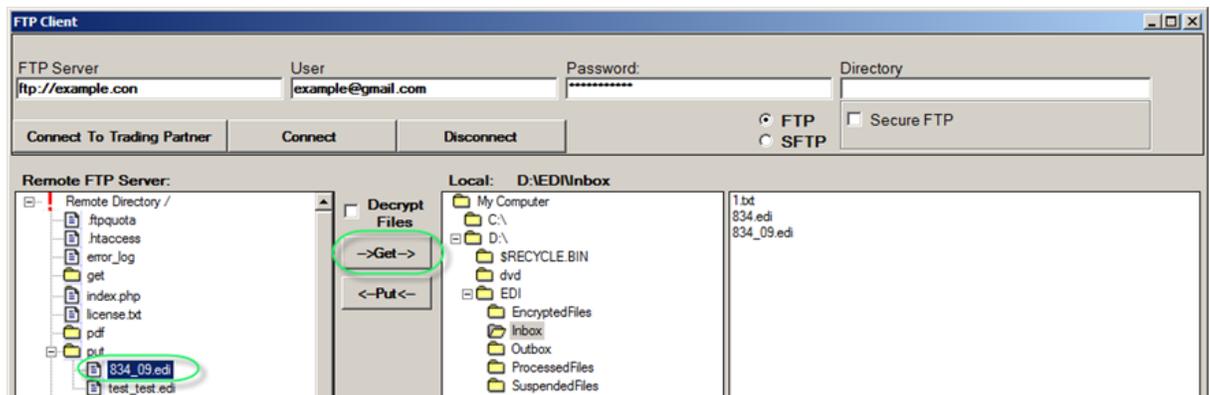


The file will appear in the remote folder. You will receive the "File <filename>

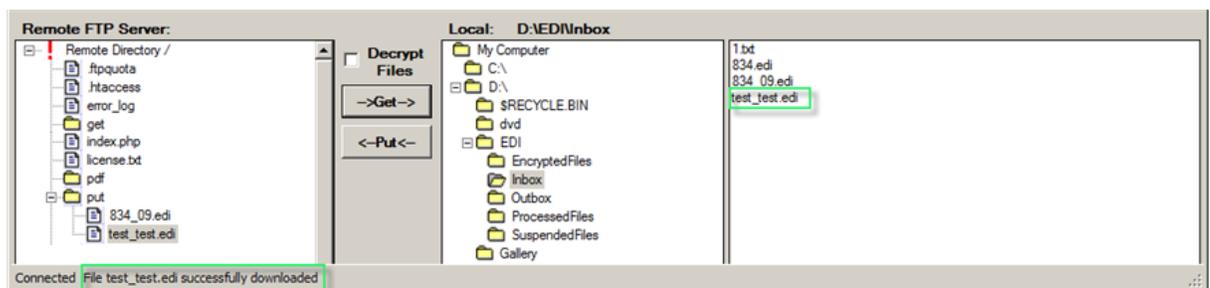
successfully uploaded" message on the bottom status bar of the FTP client.



6. To download a file from the server, select a file in the left side, and then click on the "Get" button.

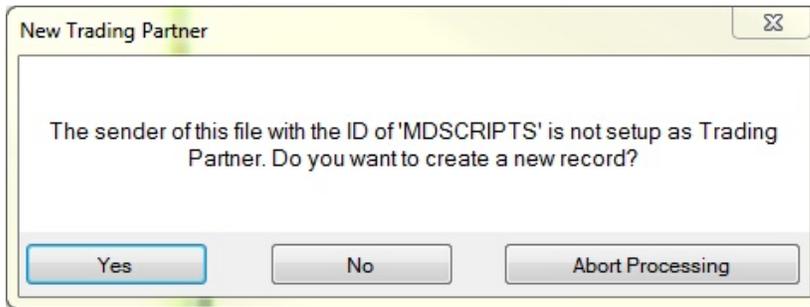


The file will appear in the local folder. You will receive the "File <filename> successfully downloaded" message on the bottom status bar of the FTP client.



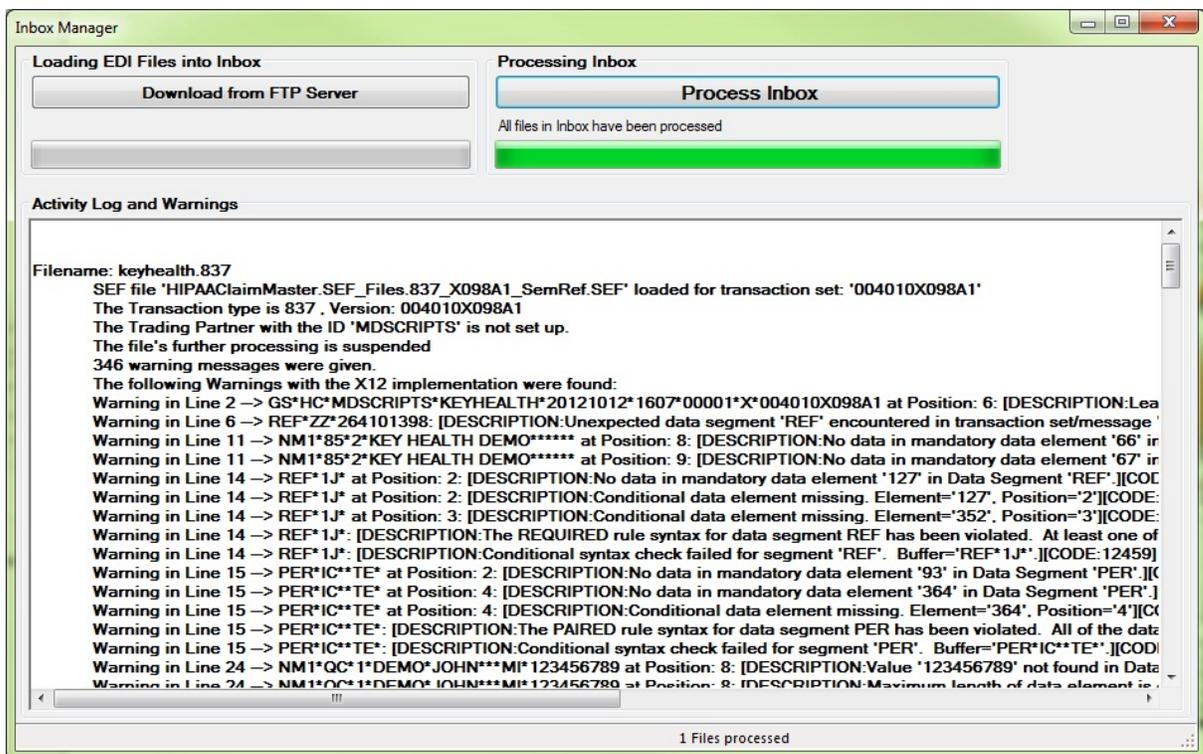
12.4.4 Creating a Trading Partner Automatically

When you process a file with EDI Exchange the sender's ID is compared to the trading partners on file. If the trading partner does not exist you have the opportunity to create a rudimentary new trading partner record. Now you have the choice to either create this new record, process the file without the trading partner record or to abort the operation.



The "New Trading Partner" dialog box

If you abort the process, you still will get the EDI file analysis.



EDI compliance check results after further processing was aborted.

Files without a valid trading partner will be placed into the "suspended files" directory. The final action taken after the analysis is recorded at the end of the results.

```
warning in Line 107 -> LIN*N4**03029120003 at Position: 1: [DESCRIPTION:Data element 350 designated not used contain data
Warning in Line 167 -> LIN*N4**63629126003 at Position: 2: [DESCRIPTION:No data in mandatory data element '235' in Data Segr
Warning in Line 168 -> SE*166*031111112 at Position: 2: [DESCRIPTION:Transaction set/Message header control number '0001' d
Warning in Line 169 -> GE*1*1 at Position: 2: [DESCRIPTION:Functional group header control number '00001' does not match traile

The file has 1 Interchanges, 1 Functional Groups and 1 Transaction Sets
The file contains 4 Claims
The file was moved to: H:\HIPAAsuite\HIPAAsuiteCommDir\Suspense\837\keyhealth.837
on 10/17/2012 at 9:50 AM
```

This file's processing was aborted. The file was moved to the suspended files folder

12.4.5 Certificate based authentication in SFTP

SFTP or FTP over secure shell as it is also known is deemed to be the most secure method of file transport. There are 3 methods of authentication in Sftp.

1. with a user name and password
2. with a user name and a private key cryptographic certificate and
3. with a user name, a certificate and a password

When you select SFTP as the FTP protocol, you will see two check boxes appear.



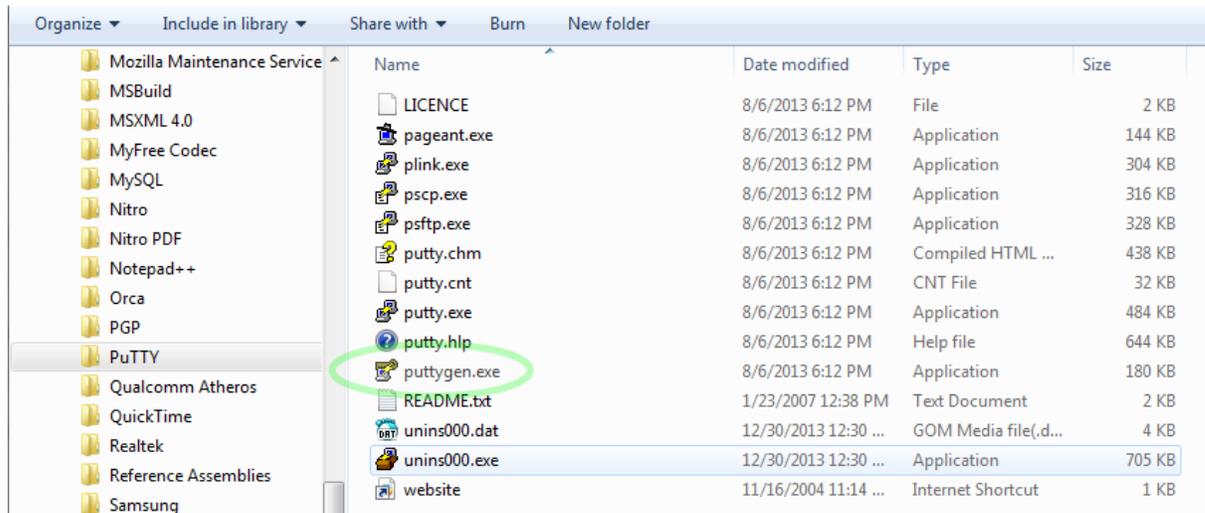
The sub choices when you select SFTP

Leaving both check boxes unchecked will result in the first option: Authentication with user name and password. You can also check just the user name and password with the same result.

In order to use the certificate based authentication you need to create and link to your own certificate. The certification module that HIPAAsuite employs uses a so called private key SSH2 certificate in pem format.

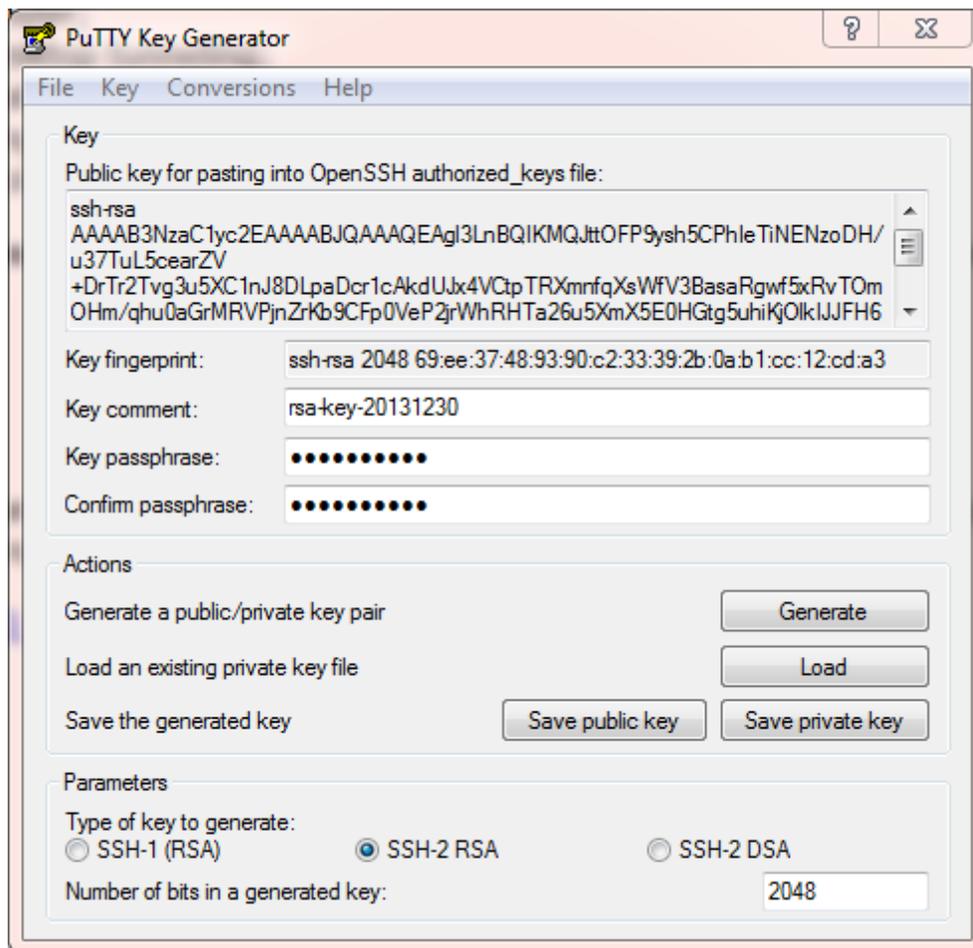
How do you create such a certificate? Here is one way:

The free secure shell program putty has the necessary tools. Download putty from www.putty.org When you install the program you will see several program installed on your computer.



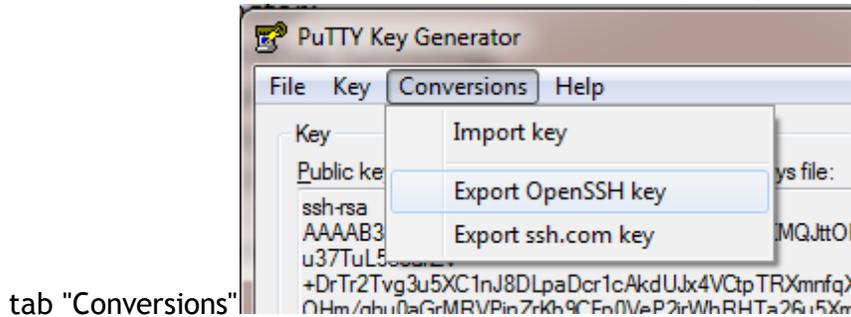
The programs and files that come with putty

One of the programs is puttygen.exe. This program creates the necessary keys. When you start it up, you can generate a key pair.



Creating a public/private key pair with puttygen

You can save the public and private keys separately as .ppk files but that is not what we need. But puttygen has also tool to convert the keys to pem files. The top menu has a



tab "Conversions"

Converting the SSH key into a pem certificate

When we click on "Export OptnSSH key" we can then save the key with an .pem ending and that is it.

12.5 Using Encryption

12.5.1 About Encryption

EDI Exchange supports the **Private Key Infrastructure (PKI)** encryption method. This type of encryption is the most generally accepted method of protecting EDI Files from being pried upon by unauthorized persons. Without going further into the details of PKI, Public-key encryption is a cryptographic technique which enables users to securely communicate on an insecure public network, and reliably verify the identity of a user via digital signatures. Read more in [Private Key Infrastructure](#).

A public-key infrastructure (PKI) is a system for the creation, storage, and distribution of digital certificates which are used to verify that a particular public key belongs to a certain entity. The PKI creates digital certificates which map public keys to entities, securely stores these certificates in a central repository, and revokes them if needed.

A PKI consists of:

- A certificate authority (CA) that both issues and verifies the digital certificates.
- A registration authority which verifies the identity of users requesting information from the CA.
- A central directory is a secure location to store and index keys.

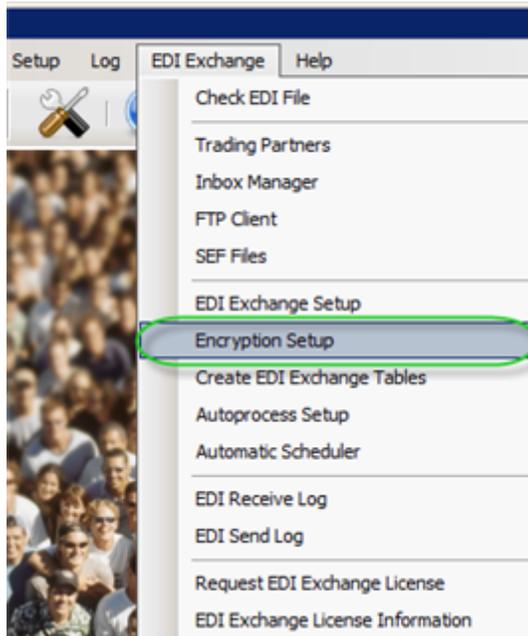
- A certificate management system.

EDI Exchange relies on other software to establish the PKI. It only uses the capabilities of these programs through their Application Programming Interface (API). You need to separately install either [PGP Desktop](#) or the open source GPG4Win programs and set them up with the public keys of your trading partners and your own private key.

12.5.2 Setting up Encryption

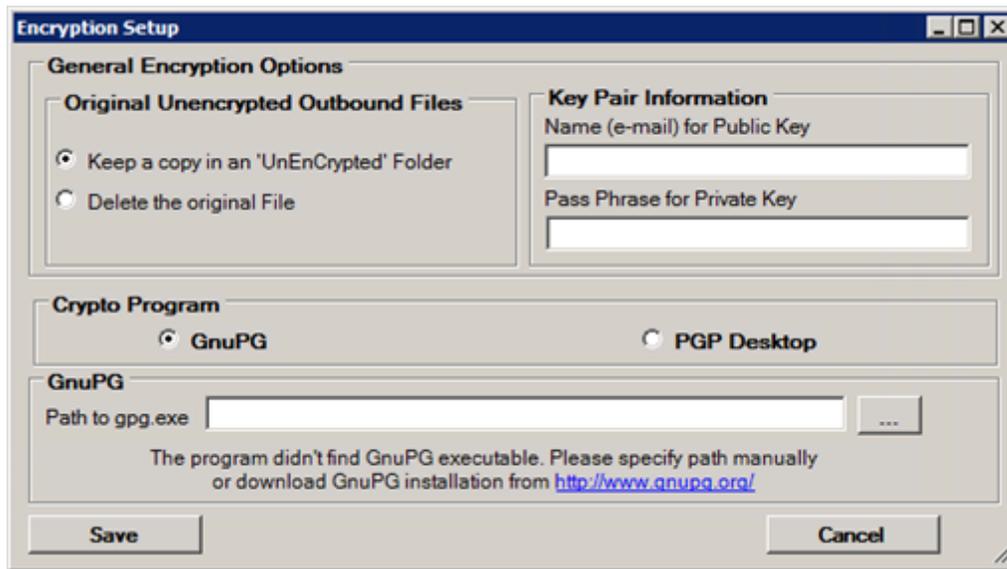
EDI Exchange allows you to set up the encryption for your EDI files. Follow the steps below.

1. Access the "Encryption Setup" window by selecting "Encryption Setup" under the "EDI Exchange" menu.



The encryption setup menu

2. The following window will appear.



The encryption setup screen

3. In this window define the following options:

General Encryption Options

- **Original Unencrypted Outbound Files**

- **Keep a Copy in an 'Unencrypted' Folder** – When the HIPAAsuite program creates an EDI file for a trading partner that has selected encryption, you can keep an unencrypted copy in the "Outbox/[trading partner]/Unencrypted" folder. This is useful when you need to go back to the file and check on problems.

Note: Once you encrypt a file with the public key of your trading partner, you will not be able to open it again. Only the owner of the private key can decrypt it and read it.

- **Delete the Original File** – If you do not want to keep the unencrypted copy, select this option.

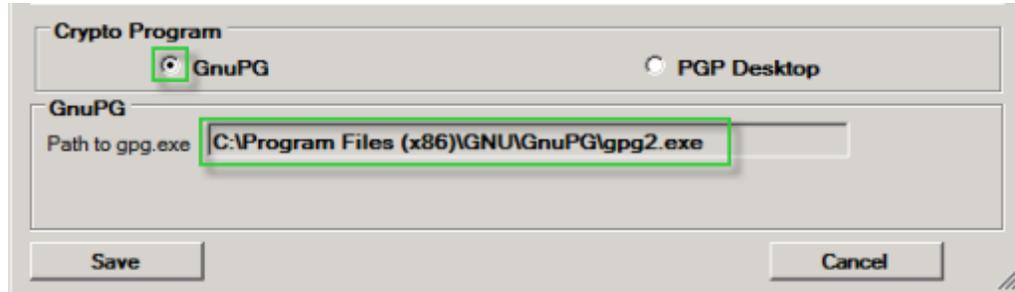
- **Key Pair Information**

- **Name (e-mail) for Public Key** – Enter name or e-mail that will be used to encrypt files for you by your trading partners.
- **Pass Phrase for Private Key** – Enter passphrase here to decrypt files encrypted previously with the pass phrase.

Note: The keys are identified by the email address of their owner.

- **Crypto Program**

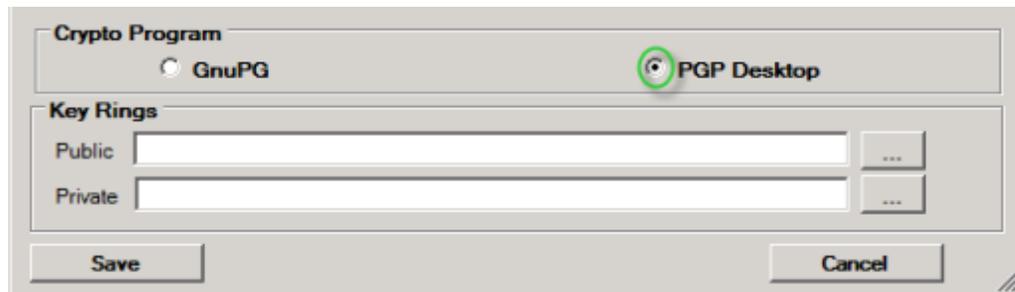
- **GnuPG** – Select this option if you have already installed the [GnuPG](#) software and want to use it.



The configuration using GnuPG

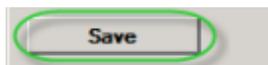
Note: When you select GnuPG, the program checks if the executable `gpg.exe/gpg2.exe` is present on your computer. If it is not found, you will have to specify the path manually or install the program first.

- **PGP Desktop** – Select this option if you have already installed the [PGP Desktop](#) program and want to use it. You will see a different lower half of the screen, where you can indicate the location of the keys. PGP Desktop uses "key rings" – encrypted folders that contain all your keys. The location of these two files is very important for PGP Desktop.



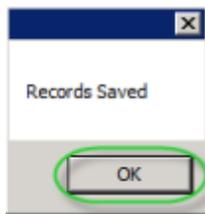
The configuration using PGP Desktop

4. Once the settings are done, click the "Save" button.



The "Save" button

5. The following notification will appear. Click the "OK" button.



Notification window

12.5.3 Using PGP Desktop

Pretty Good Privacy (PGP) is a data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication. PGP is often used for signing, encrypting and decrypting texts, e-mails, files, directories and whole disk partitions to increase the security of e-mail communications. PGP and similar products follow the OpenPGP standard ([RFC 4880](#)) for encrypting and decrypting data. For more information, see [How PGP works](#).

EDI Exchange supports PGP encryption and works seamlessly with [PGP Desktop](#) and open source [Gpg4Win](#) applications.

[PGP Desktop](#) (Symantec's encryption solutions) is a comprehensive suite of encryption applications which provides flexible, multi-layered encryption by bundling Drive Encryption to secure the files stored on local hard drives, and Desktop Email Encryption to secure confidential data in email. For more information, see [PGP Desktop documentation](#).

PGP Desktop key features:

1. Hard drive encryption software locks down the entire contents of a laptop, desktop, external drive, or USB flash drive, including boot sectors, system, and swap files.
2. Enables encrypted email and secure AIM® Instant Messages.
3. Creates storage-independent encrypted containers for transport and sharing of specific files using included utilities; PGP Self-Decrypting Archive, PGP Virtual Disk, and PGP Zip.
4. Includes PGP Shredder which can completely destroy unwanted disk-based files and folders.
5. Drive Encryption can be centrally deployed and managed by Symantec Encryption Management Server.

PGP Desktop key benefits:

1. Secures email without burdening users, to improve compliance with policies and regulations without hindering productivity.
2. Allows users to easily and transparently share encrypted files and folders, improving data security without impacting user productivity.
3. Management by Encryption Management Server centralizes creation, deployment and management of data security policies and reporting.

PGP Desktop bundles the following products:

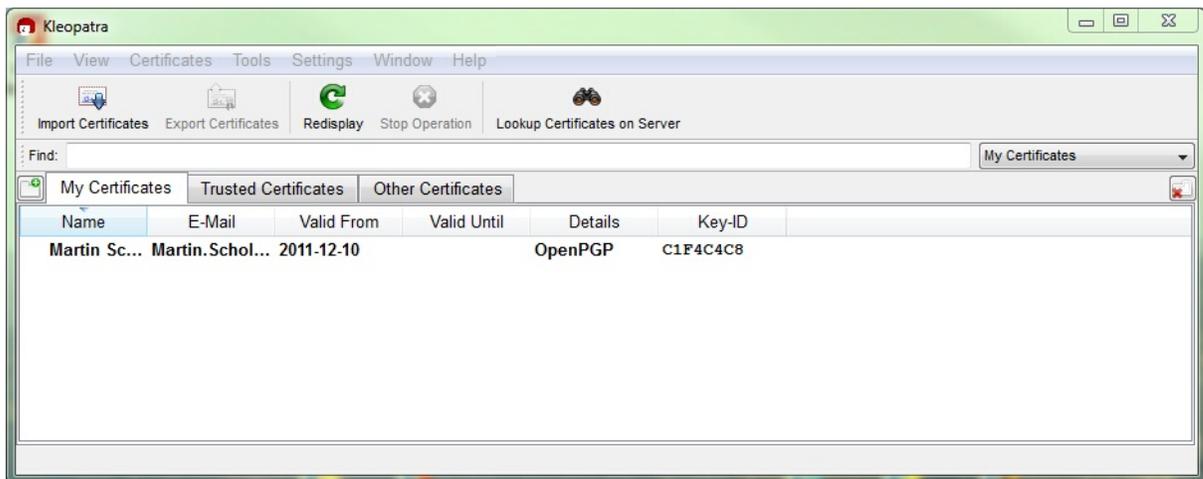
1. Drive Encryption. See [System Requirements](#).
2. Desktop Email Encryption. See [System Requirements](#).
3. Encryption Management Server. See [System Requirements](#).

12.5.4 Using GnuPG

[GnuPG](#) is an Open Source project for the implementation of the OpenPGP (Pretty Good Privacy) protocols of encryption. GnuPG allows to encrypt and sign your data and communication, features a versatile key management system as well as access modules for all kinds of public key directories. GnuPG, also known as GPG, is a command line tool with features for easy integration with other applications. Front-end applications and libraries are also available. Version 2 of GnuPG also provides support for S/MIME.

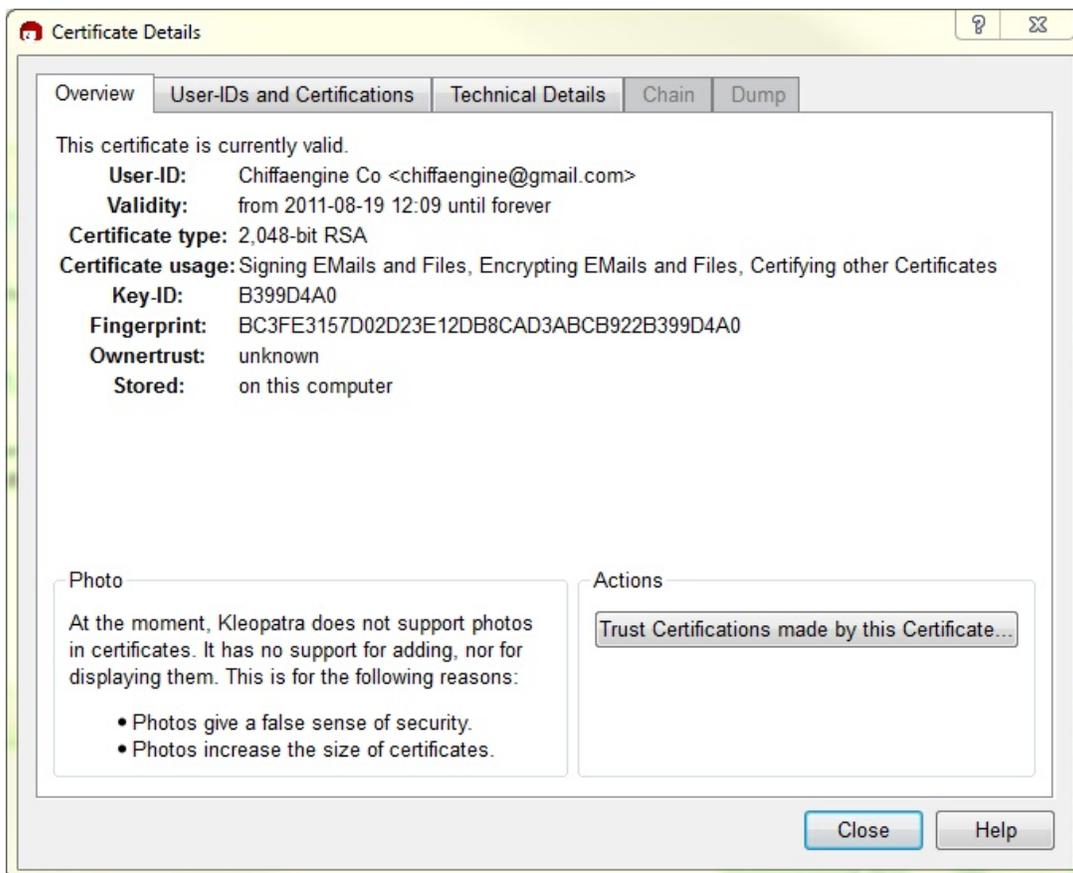
GnuPG is a free software, so it can be freely used, modified and distributed under the terms of the [GNU General Public License](#).

We recommend [Gpg4Win](#) for encrypting of your files and emails. [Gpg4Win](#) supports both relevant cryptography standards, OpenPGP and S/MIME (X.509), and is the official GnuPG distribution for Windows. [Gpg4Win](#) contains Kleopatra as one of its Free Software components. For more information, see [Gpg4Win documentation](#) available both in PDF and HTML versions.



Kleopatra, a certificate manager for OpenPGP and X.509 (S/MIME) and common crypto dialogs

With Kleopatra, it is easy to manage your certificates and create your own ones. It seamlessly integrates with GnuPG. You can manage the key that you receive from your trading partners, because there is a screen to view the details of a key.



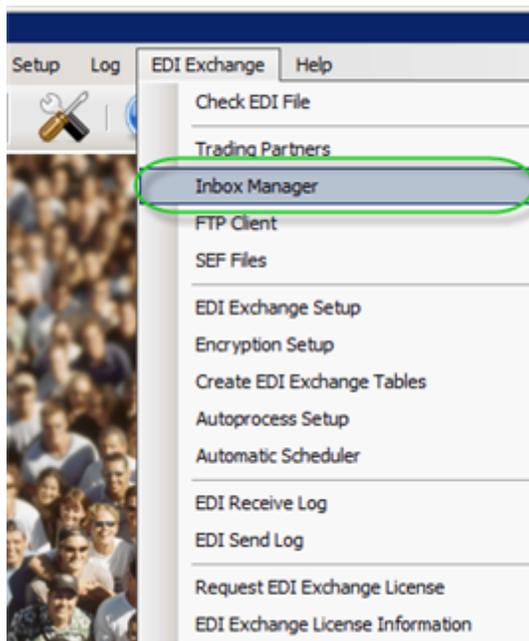
Certificate details with Kleopatra

12.6 Using EDI Exchange Features

12.6.1 Accessing Inbox Manager

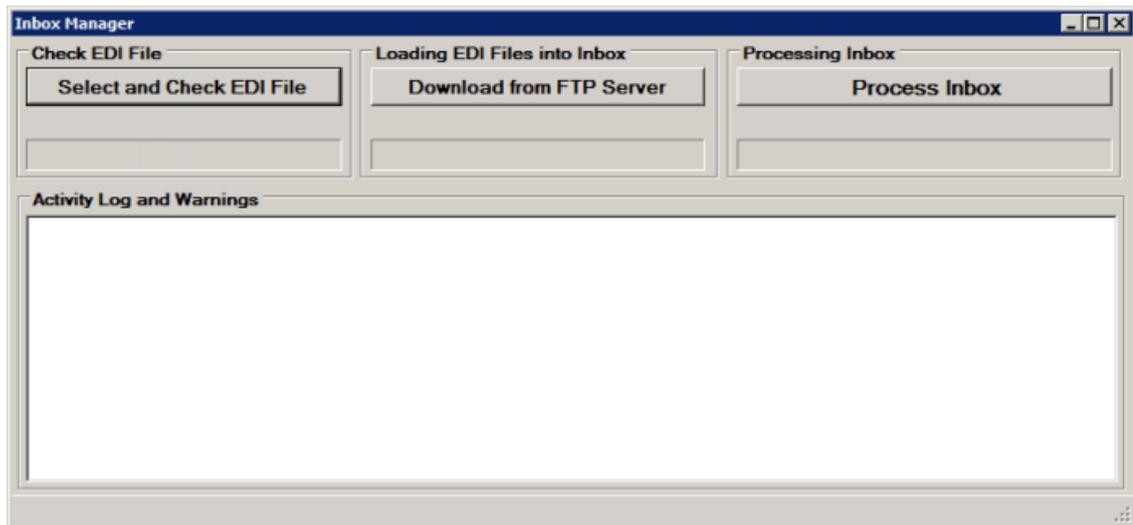
Using the Inbox Manager you can access the most important functions of EDI Exchange. Inbox Manager allows you to download EDI files into the "Inbox" folder and process these files. This screen handles the post-processing of the EDI files, their compliance check and auto-processing options.

1. To access the Inbox Manager, click the "Inbox Manager" under the "EDI Exchange" menu.



The "Inbox Manager" menu item

2. The following screen will come up.



The "Inbox Manager" window

Read more in:

- [Downloading EDI Files From FTP Server](#)
- [Processing EDI Files](#)
- [Selecting and Checking EDI Files](#)

12.6.2 Checking EDI Files

EDI file analysis based on the HIPAA standards. Compliance with HIPAA EDI rules is an essential part of the exchange of EDI documents. The standards are the only agreed upon rules that sender and receiver use to exchange data from completely different backend systems. Strict adherence is therefore necessary to guarantee frictionless operation.

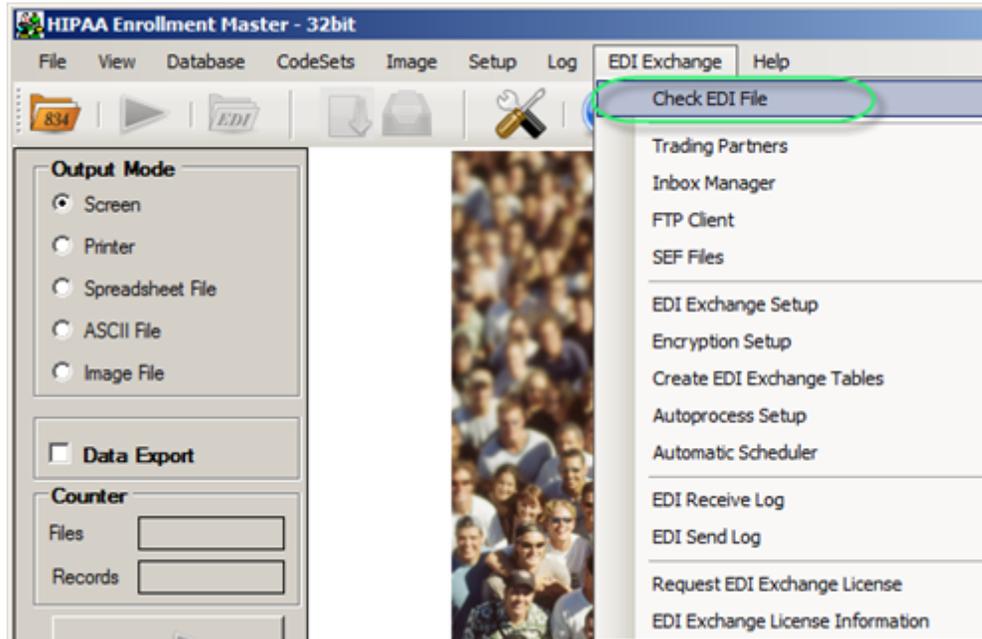
Unfortunately, HIPAA compliance is difficult and the truth is that many HIPAA EDI files are truly bad. Syntax errors, omitted loops, missing elements, wrongly formatted elements – these are the most common EDI errors. Without a true analysis, it is difficult to say what data ends up in your system.

EDI Exchange has a built-in compliance engine that analyzes each incoming and outgoing EDI transaction. Line by line, element by element, error reporting provides a powerful tool to determine the quality of the incoming and outgoing EDI files. All HIPAA file versions are supported. The compliance check creates a detailed report that lists every compliance issue. Outgoing files can also be checked and individual transaction in violation of HIPAA rules can be held back.

Note: You can enable automatic compliance check on outgoing and incoming EDI files. See [Setting up Incoming and Outgoing Files Options](#).

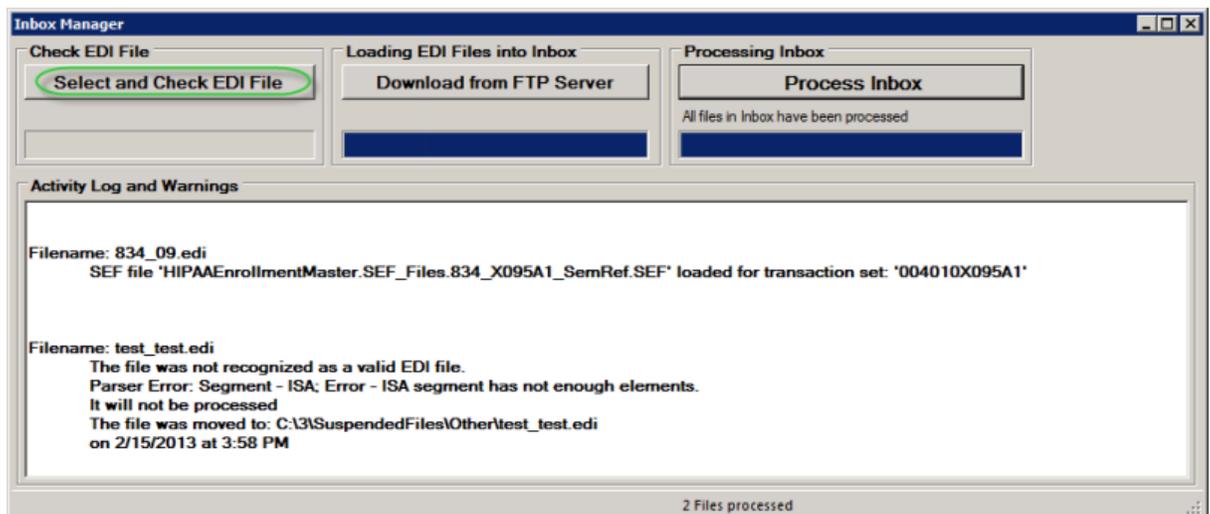
Follow the instructions below to check EDI files for compliance.

1. Select "Check EDI File" under the "EDI Exchange" menu item.



The "Check EDI File" menu item

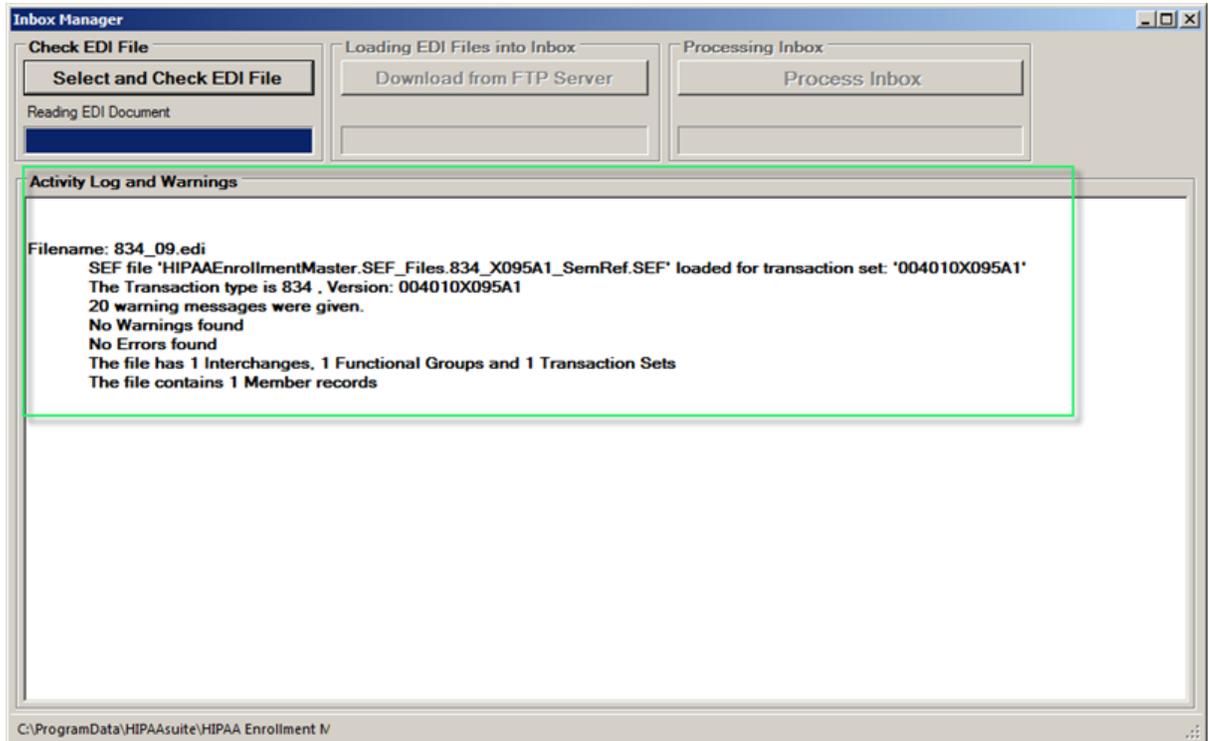
Alternatively, you can click on the "Select and Check EDI File" button in the Inbox Manager window. Read more in [Accessing Inbox Manager](#).



The "Select and Check EDI File" button

2. In the opened file selection dialog, select an EDI file and click "Open."

3. In the Inbox Manager, you can see the result messages for the operation. The details are displayed in the "Activity Log and Warnings" area.



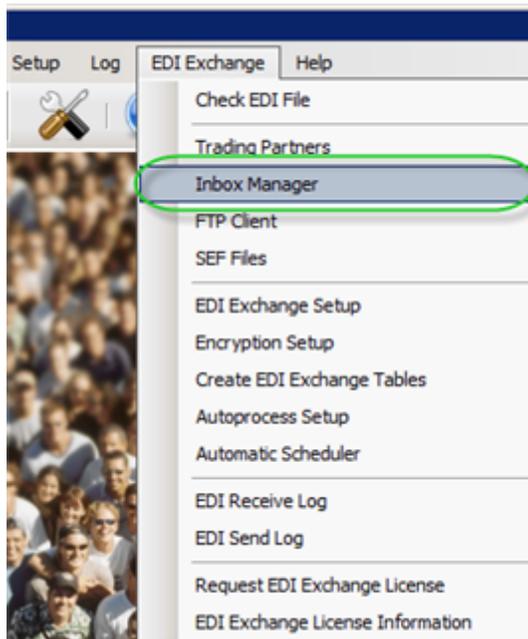
The "Activity Log and Warnings" area displaying log messages

12.6.3 Downloading EDI Files From FTP Server

EDI Exchange Inbox Manager allows you to load EDI files into the "Inbox" folder and process these files. Be sure you have setup FTP settings in the "Remote FTP" tab of the Trading Partner window (see [Setting up Trading Partners](#)).

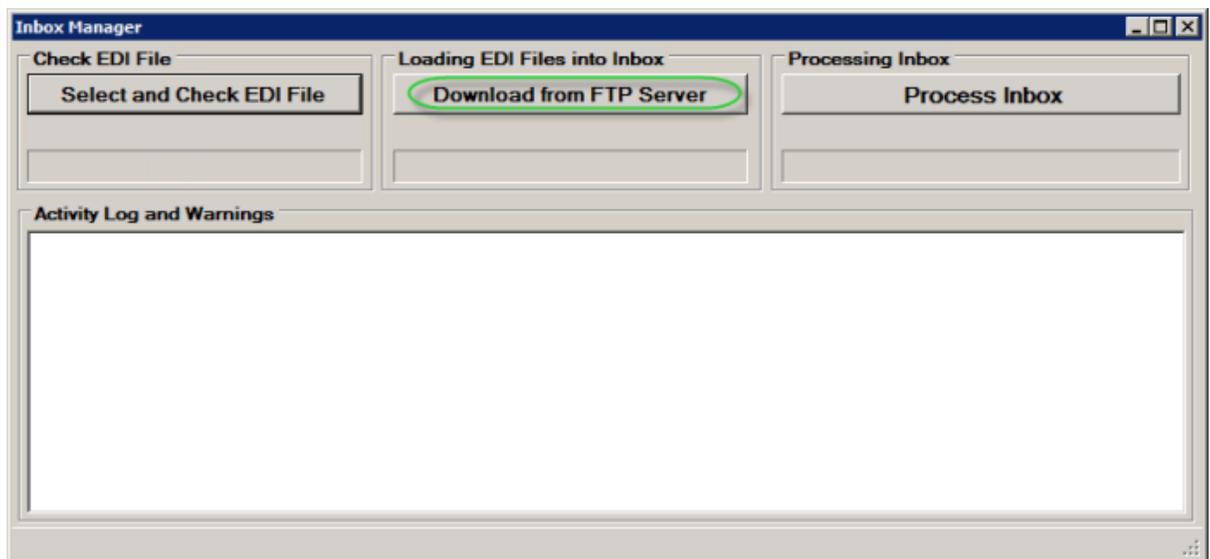
Follow the instructions below to upload EDI files into the "Inbox" folder.

1. Access the Inbox Manager by clicking the "Inbox Manager" under the "EDI Exchange" menu.



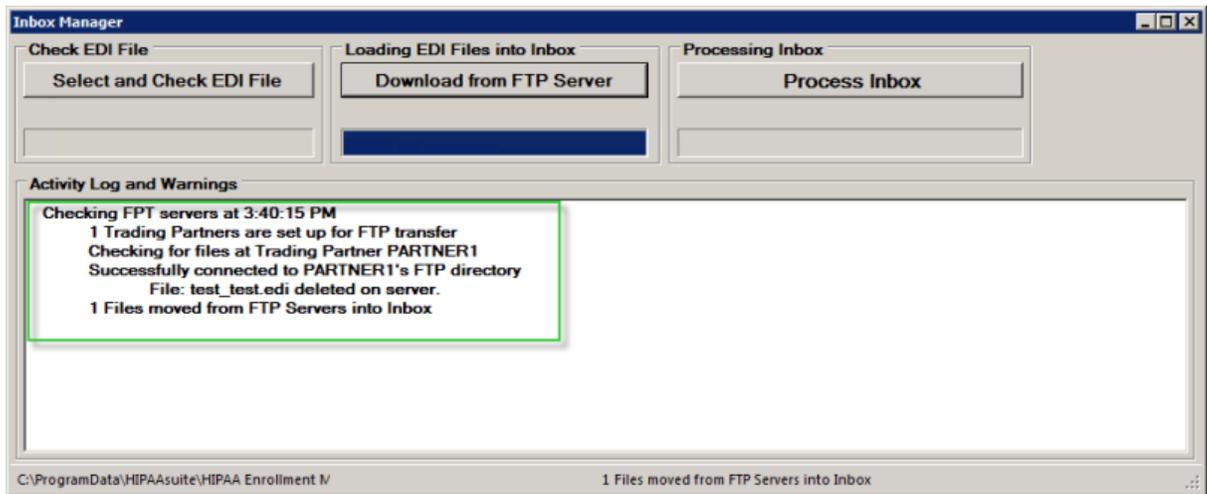
The "Inbox Manager" menu item

2. Click the "Download from FTP Server" button.



The "Download from FTP Server" button

3. When the process has been finished, the "Activity Log and Warnings" area displays the report.



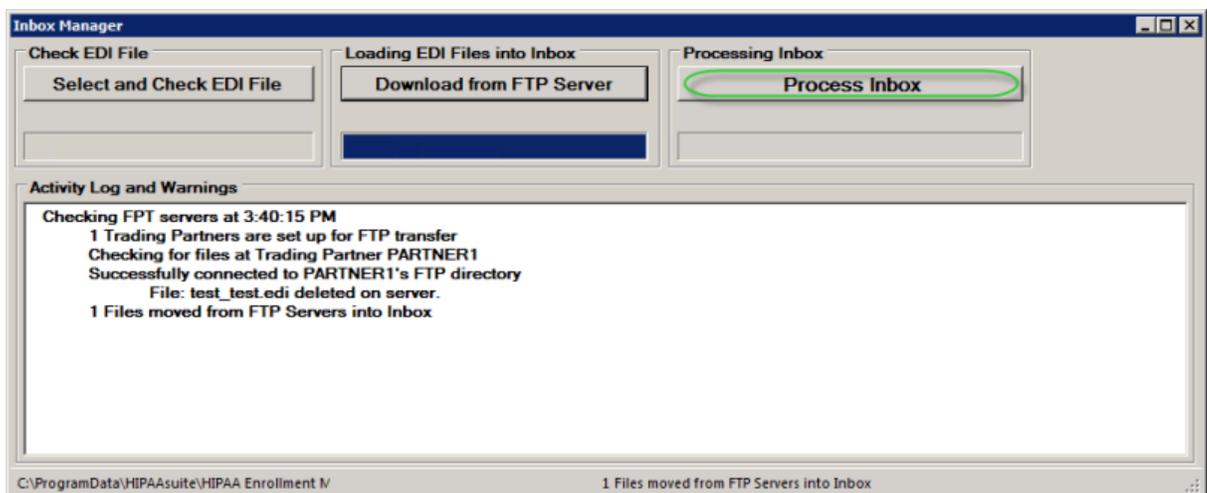
Activity Log and Warnings information

12.6.4 Processing EDI Files

EDI Exchange Inbox Manager allows you to process EDI files downloaded to the "Inbox" folder beforehand.

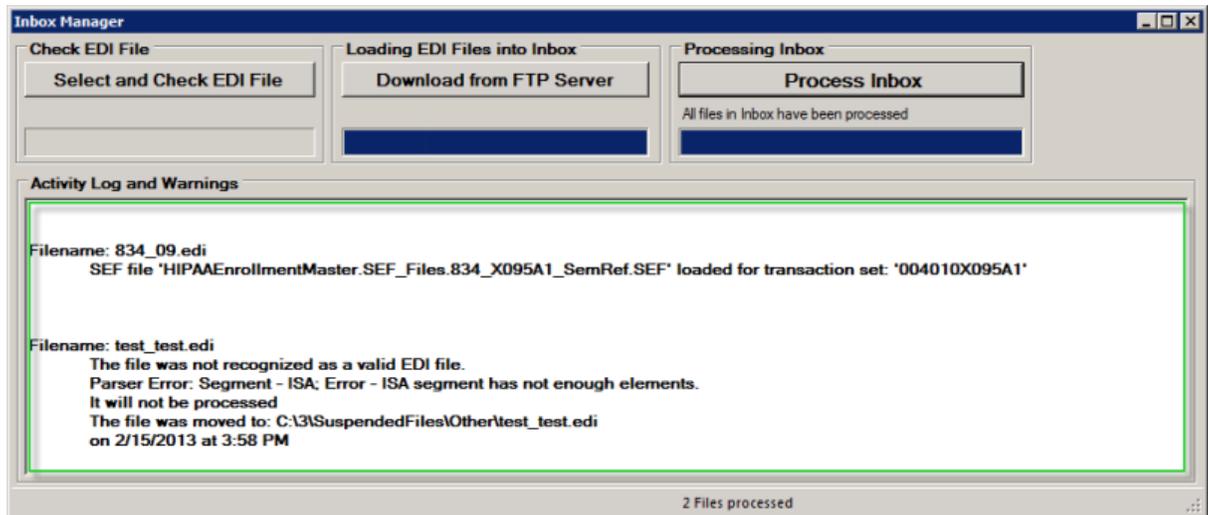
Note: Once you have saved the auto-processing options (see [Defining Auto-Processing Options](#)), the files will not only be analyzed but also processed according to the defined settings. The auto processing enables you to combine and run multiple fulfillment steps together (for example, export, saving, printing.)

1. Once [Downloading EDI Files From FTP Server](#) is completed, and no errors are displayed, click on the "Process Inbox" button. This will autoprocess all files present in the EDI inbox directory.



The "Process Inbox" button

2. Once the processing is completed, review the Activity Log and Warnings information.

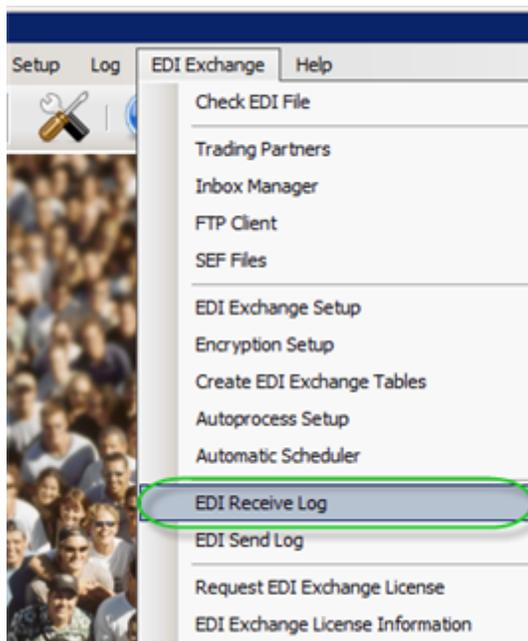


Activity Log and Warnings information

12.6.5 Accessing EDI Receive Log

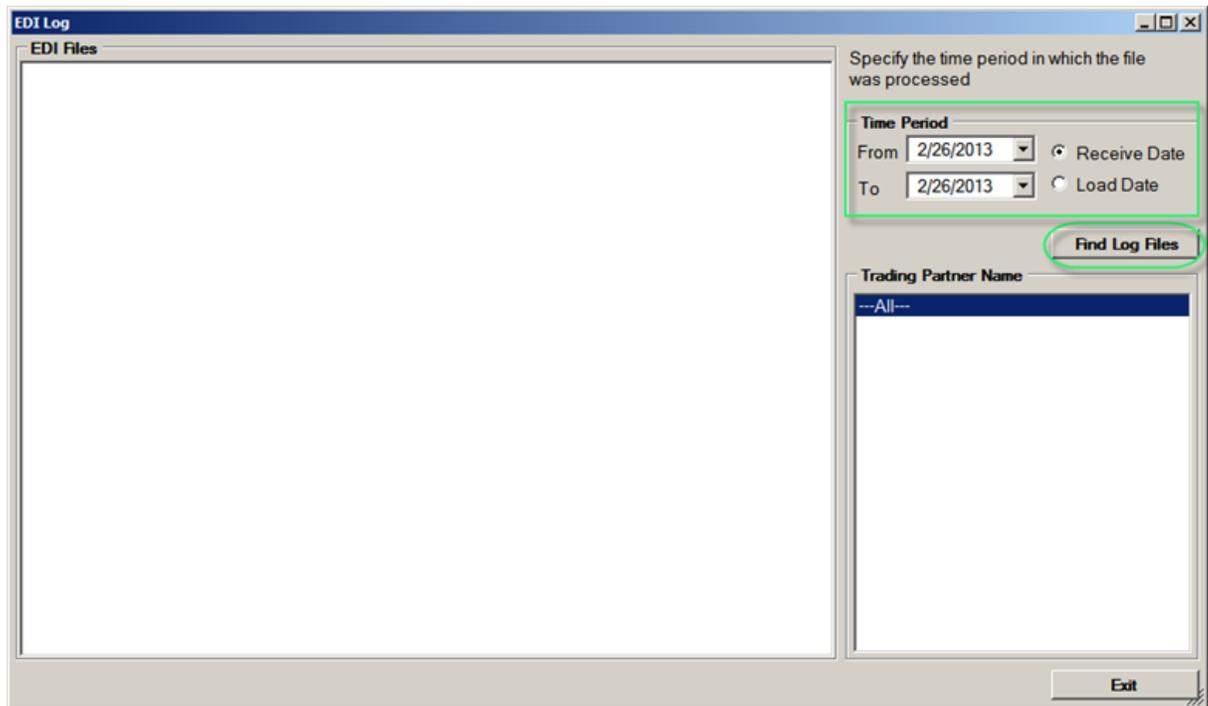
EDI Exchange has a file log. Each processed file creates an entry in the `Trans_Log` table. You can access the "EDI Log" window to query this table and see what files came in and how they were processed. You can access the EDI Receive Log once the application has been initialized. Follow the instructions below.

1. To open the "EDI Log" window, select "EDI Receive Log" under the "EDI Exchange" menu item.



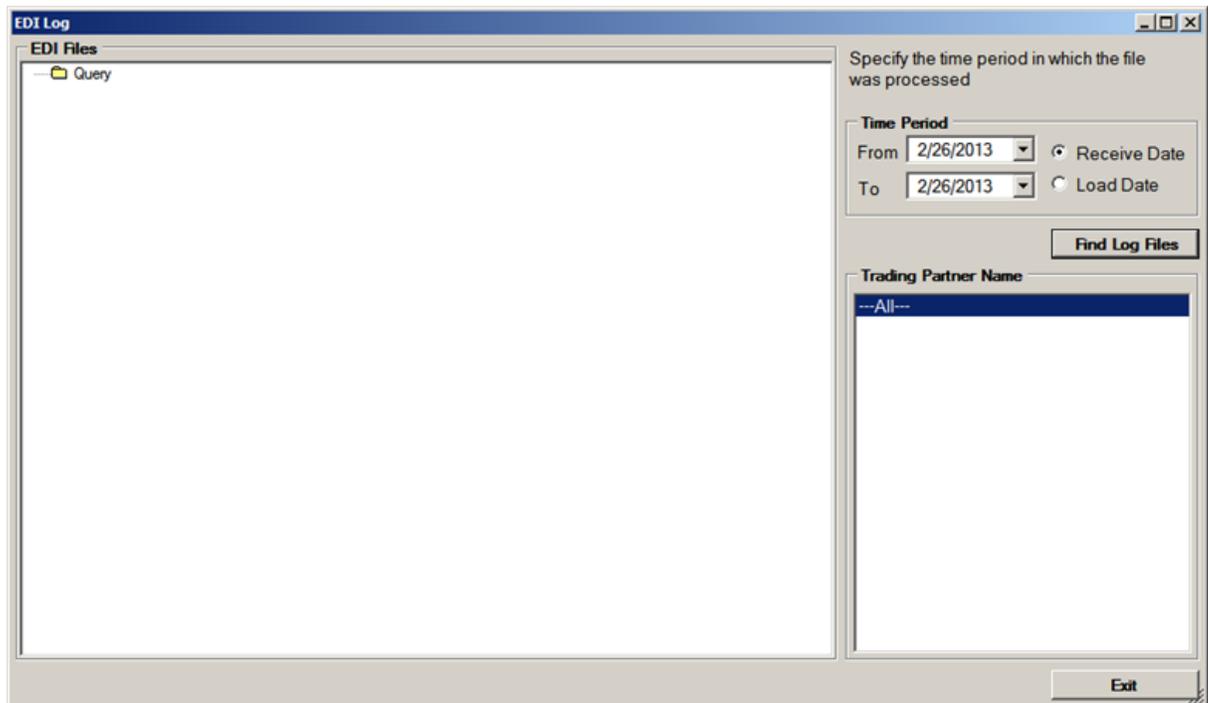
The menu for the EDI Receive Log

2. To display data, specify the time period and select trading partner.
 - **Time Period** – Period of time when the file was processed. Choose one of the available options:
 - **Receive Date**
 - **Load Date**
 - **Trading Partner Name** – You can select your trading partner from the list. If you select "---All--", all your trading partners will be included.
3. Click on the "Find Log Files" button to see the list of log files corresponding to your query.



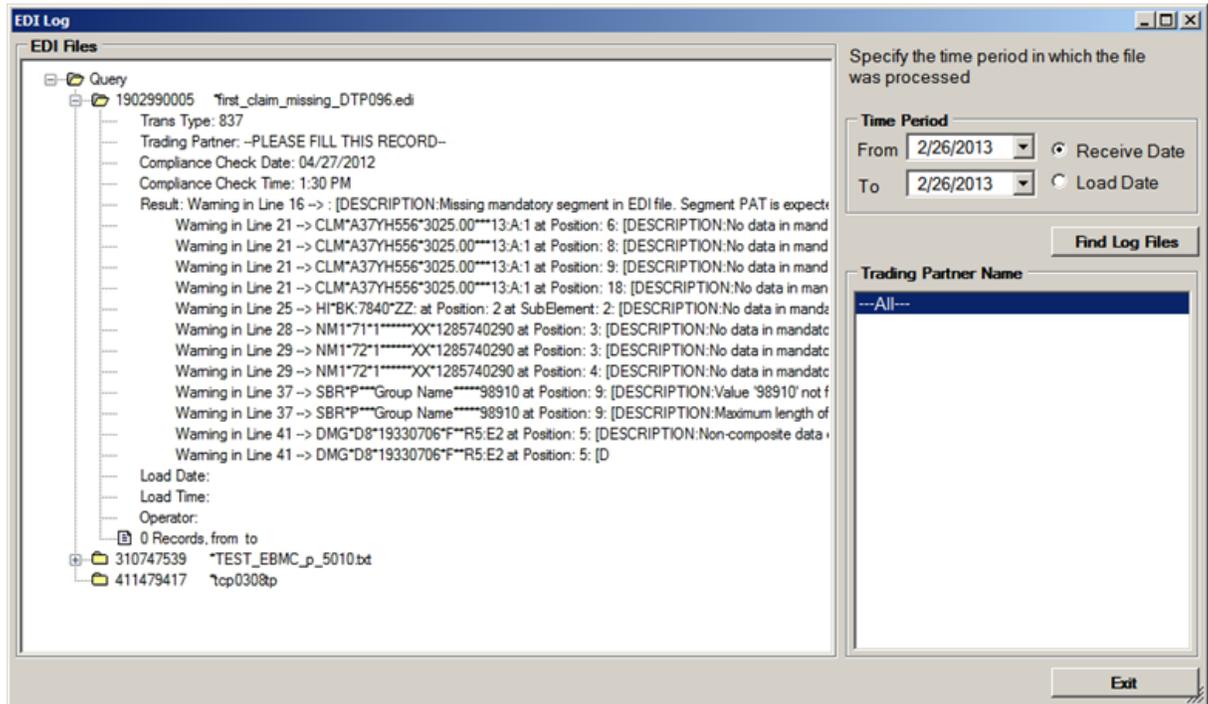
Specifying the time period

4. The log will be displayed in form of a tree.



Displayed log

5. You can open the folder icons and see details related to each file.



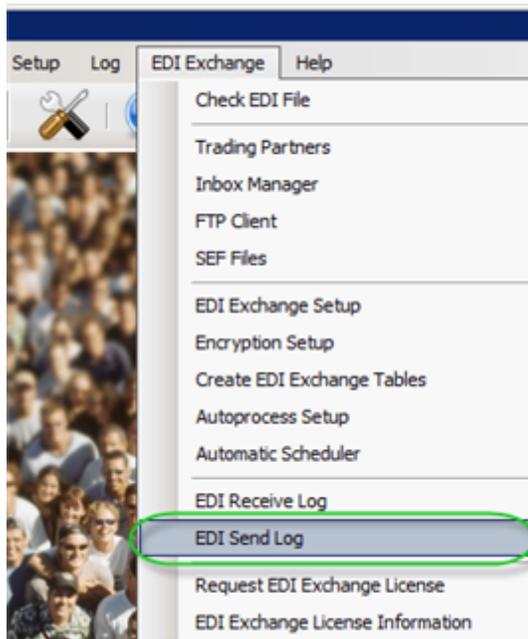
Details of the EDI Receive Log

If the records have been exported to the database, you can see the time, date and record count.

12.6.6 Accessing EDI Send Log

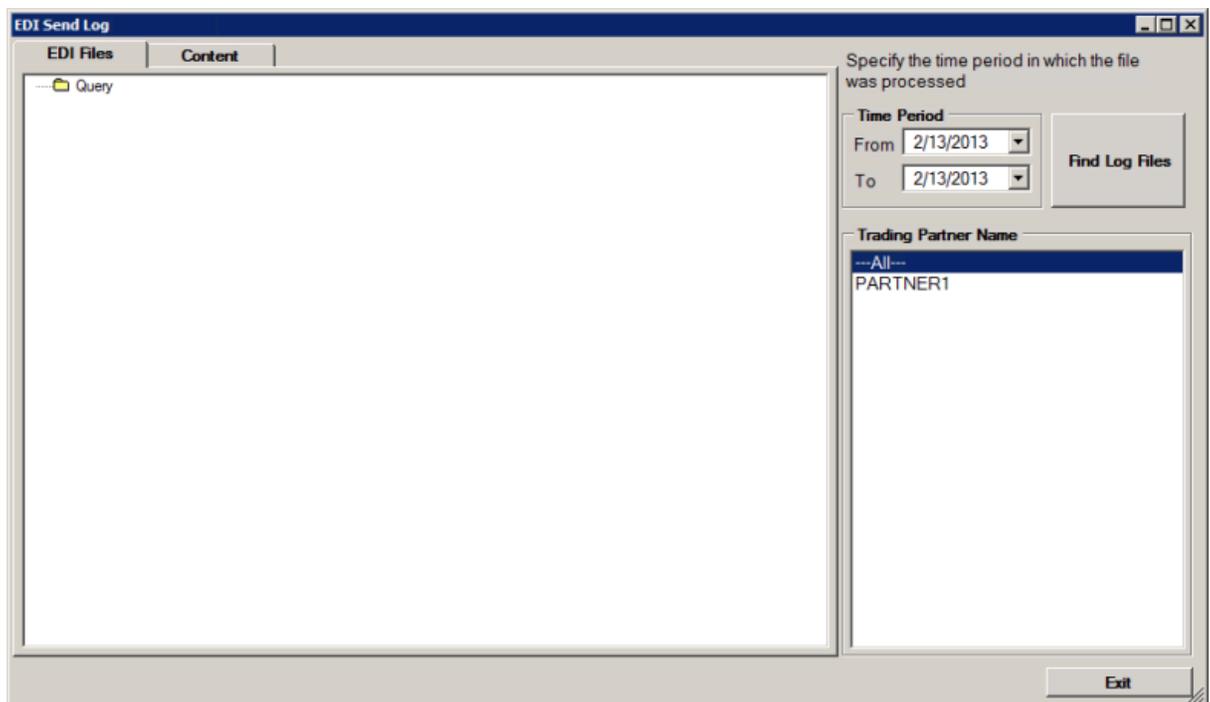
This page contains instructions on how to access the "EDI Send Log" window. It becomes available once EDI Exchange has been initialized.

1. Select "EDI Send Log" under the "EDI Exchange" menu.



The "EDI Send Log" menu

2. The following window will appear:



The "EDI Send Log" screen

3. To display the log data, select the time period and a trading partner.

- **Time Period** – Period of time when the file was processed. Choose one of the

available options:

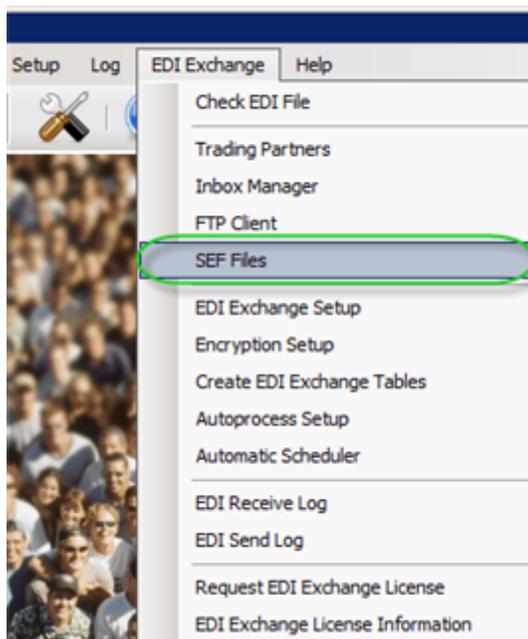
- **Receive Date**
- **Load Date**
- **Trading Partner Name** – Select a trading partner in the list. If you select "--- All---", all trading partners will be included.

4. Click on the "Find Log Files" button to display the list of log files corresponding to your query.

12.6.7 Listing SEF Files

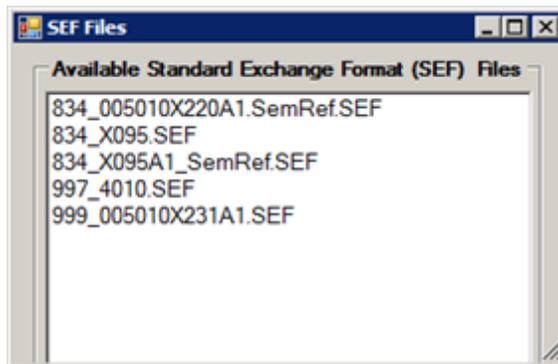
SEF (Standards Exchange Format) files are repositories of standards information that define the format of EDI documents. SEF files are the basis for the compliance check engine that EDI Exchange uses. These files contain all rules of the implementation guide of a transaction. There is a SEF file for each transaction that EDI Exchange is licensed for. For example, for 837 transactions we have SEF files for institutional, professional and dental claims in 4010, 4010A1 and 5010A1 version and the 997 and 999 transactions.

1. To access the list of available SEF Files, select "SEF Files" under the "EDI Exchange" menu.



The "SEF Files" menu item

2. The "SEF Files" window will appear.



A list of SEF files

Chapter

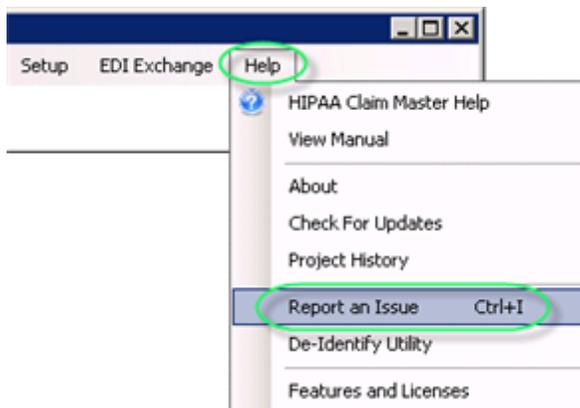
XIII

13 Bug Reporting and Program Update

13.1 Reporting an Issue

HIPAAsuite likes to make bug fixes fast and transparent. For this purpose we include a Bug tracker with the HIPAA Eligibility Responder. [btnet](#), Bugtracker in ASP.Net is an open source project. HIPAAsuite implemented Bugtracker.Net in all its products to track bugs and enhancements. One of the reasons we liked btnet was the screen capture capabilities. We adapted and included this facility.

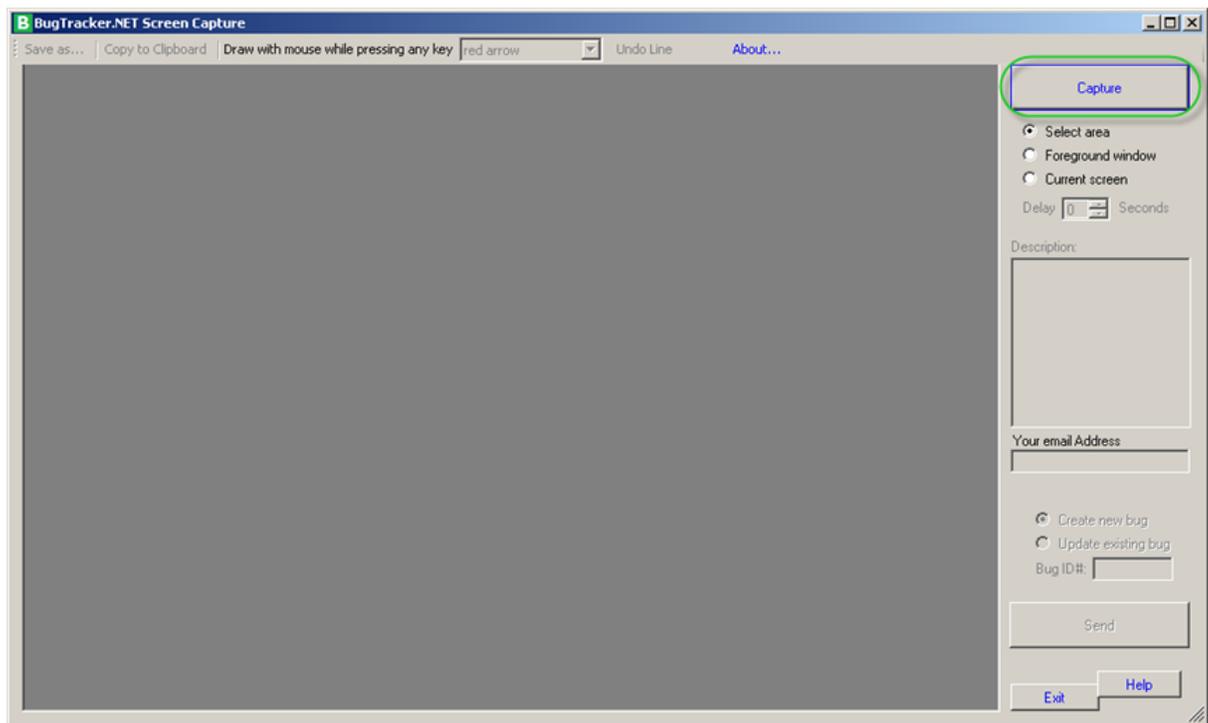
1. You can reach the Issue Tracker under the Help menu. Click on *Help* ▶ *Report an Issue* in the main menu.



The "Report an Issue" menu

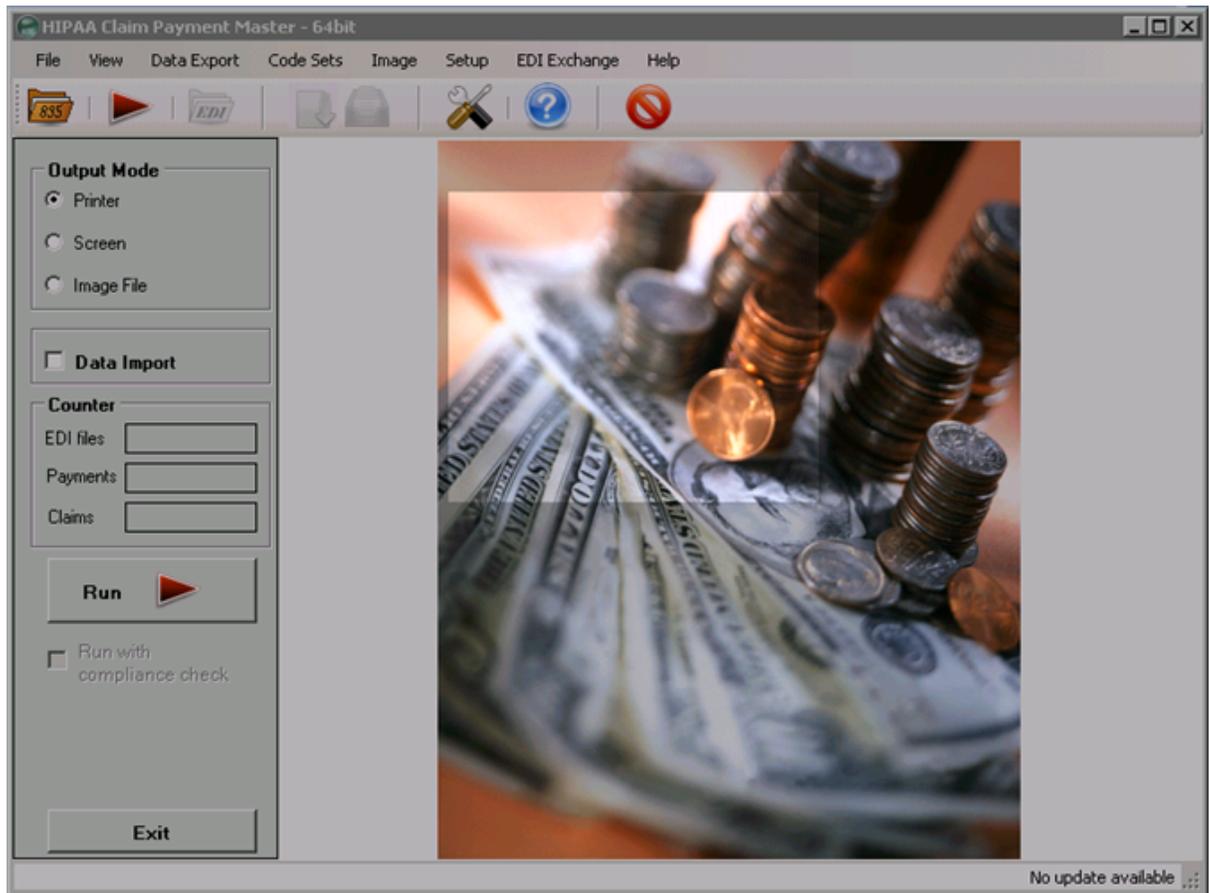
Alternatively, you can use the `ctrl + i` shortcut on your keyboard to call the Issue Tracker. Some screens do not give you access to the menu while they are open, but the key combination `ctrl + i` will call up the issue tracker report screen.

2. Click on "Capture" to snap a picture of your screen.



The "Capture" button

3. Using the mouse, select the area to make a screen-shot of and to attach to the bug report.



Selecting the area

- Once you release the mouse button, the screen capture image appears in the program.

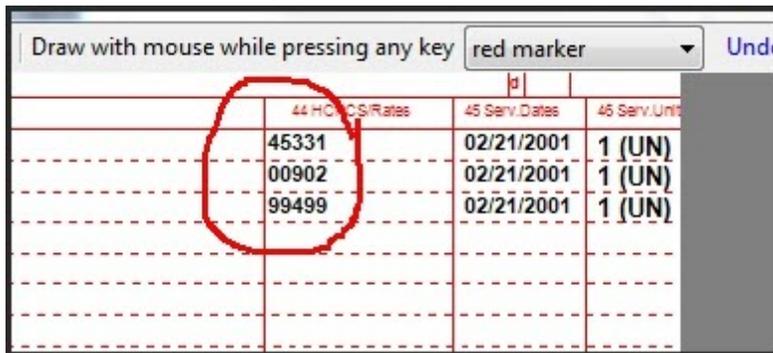
You can enhance your screen shot with drawings. By pressing any key and holding down the mouse while over the canvas, you can highlight in yellow, free draw in red or make red arrows.

You can highlight sections as shown further.

	45331	02/21/2001
	00902	02/21/2001
	99499	02/21/2001

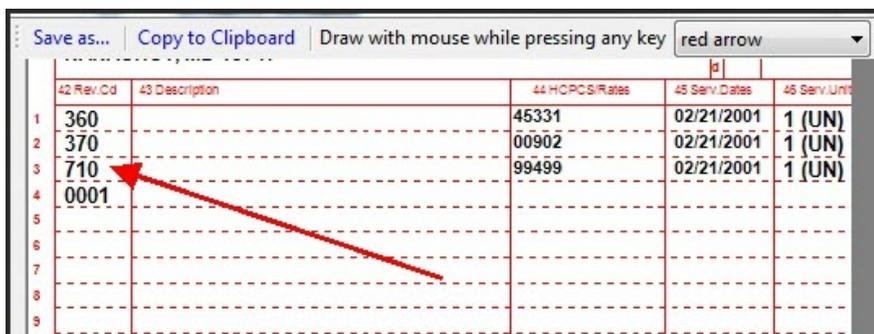
High lighting a section

You can draw circles as displayed on the following screen-shot.



Circling a section

You can point arrows to pinpoint your concerns.



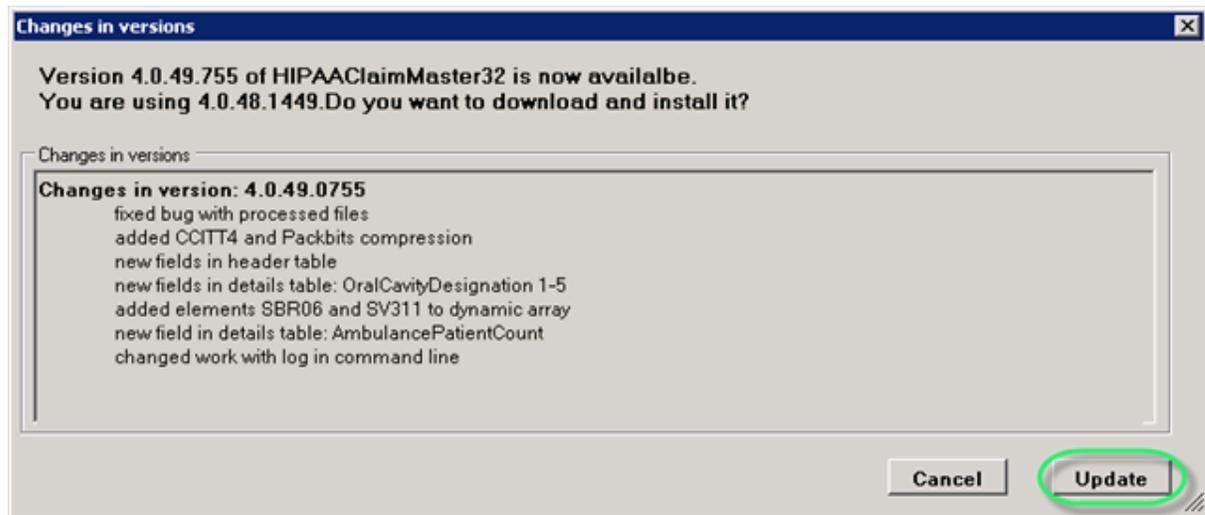
Drawing an arrow

5. Then enter as much information into the Description field as possible.
6. Now enter your email address so that we can get in contact with you.
7. Indicate whether this is a new bug or a follow up on an existing issue.
8. Then click on "Send."
9. Once the bug information has been submitted successfully, you will receive a notification. Click "OK."
10. Then you will get updates on the issue by email.

13.2 Updating the Program

Tip: An alert on version update will be available at the bottom right end section of the application.

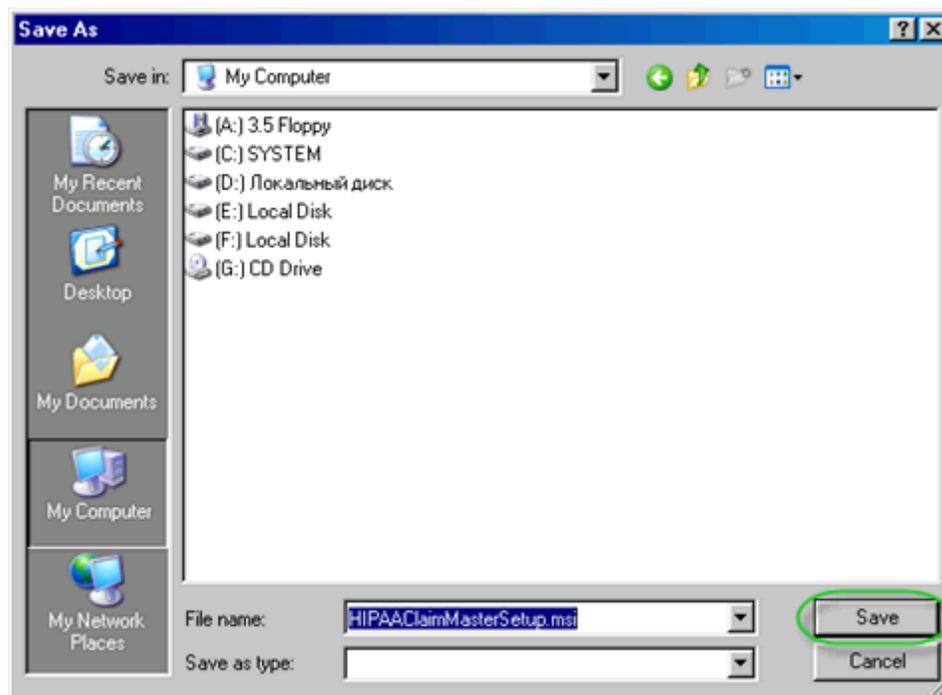
By default, the program checks for updates automatically. When the new version is out, you will see an update report on the "Changes in Versions" window. It shows which changes have been made and displays all differences between your version and the current version. To continue with update, click "Update."



The "Changes in Versions" screen with highlighted "Update" button

Then you will need to re-certify the application.

Specify a location on your local PC where to download the new version.



The "Save as" window with the "Save" button highlighted

Then the program will start downloading.

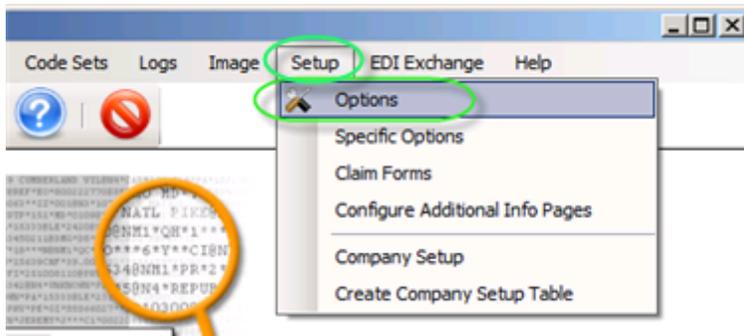


The "Downloading HIPAAClaimMaster32" popup

Then you will need to install the application.

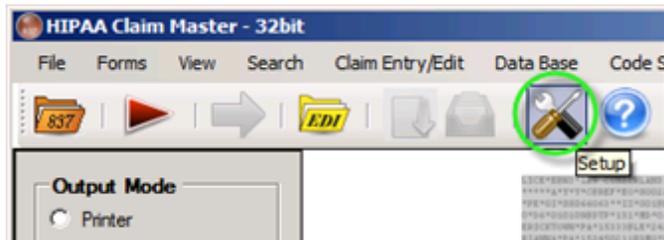
Manual Updates

To switch off automatic updates and stick with the version you have, go select *Setup ▶ Options* in the main menu.



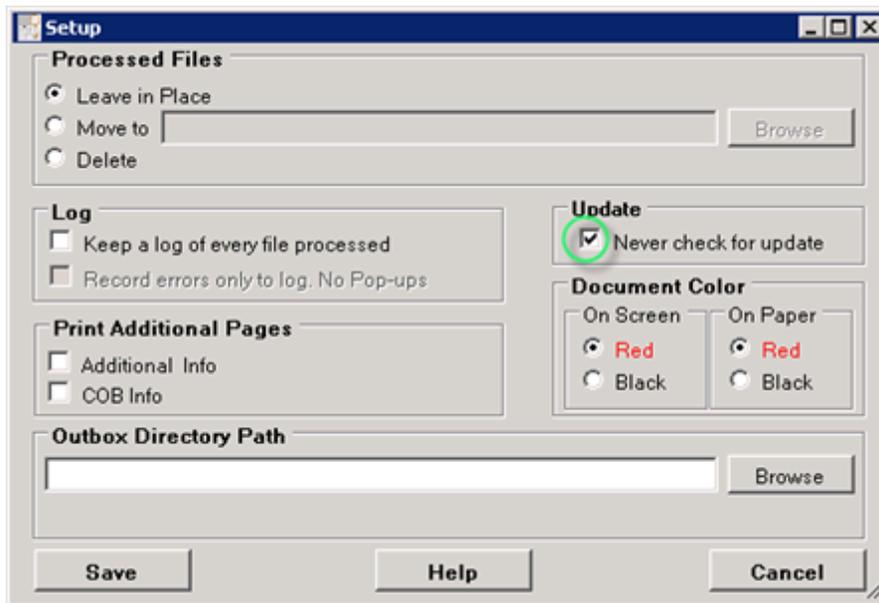
The "Options" menu

Alternatively, you can click on the "Setup" button located on the main toolbar.



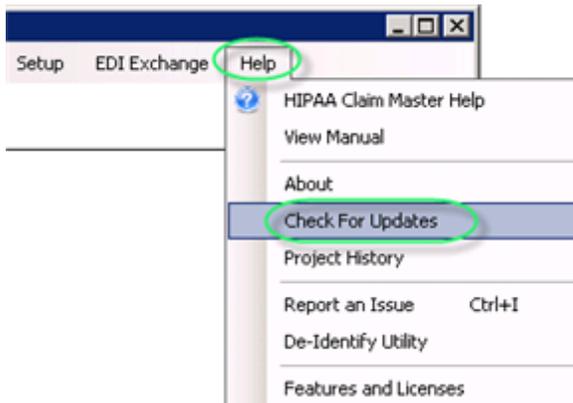
The "Setup" button

The following screen will appear. Check the "Never check for update" option and click "Save."



The "Options" window

To update the application manually, go to *Help* ► *Check for Update* in the main menu.



The "Check for Updates" menu item

Index

- B -

Blank Forms 258

- C -

Choosing the Output Mode 23

- D -

Document Info 256

- E -

Embedding the HIPAA Claim Printer into other applications 328

Extracting and Displaying Claims 17

- G -

Generating Image Files 57, 75

- I -

Introduction 2

- S -

Setting up the HIPAA Claim Printer 32

- T -

The Main Screen 7

Back Cover