

Splitting A Big 835 ERA File Into Multiple Smaller 835 ERA Files

Posted 11/26/2014

Introduction

From time to time, some Healthcare Managed Care entities struggle with implementation of new business requirements using their existing EDI platforms. Those difficulties inspire us to write EDI technical blog posts as to share with you the knowledge and solutions using Chiapas EDI technology. Because Chiapas EDI integration platform has capabilities of 'EDI as SQL' translation, major ETL challenges can be accomplished much faster compared to other existing EDI products. In this blog post, we will discuss about splitting a big 835 EDI file into multiple single 835 EDI files for multiple trading partners.

For Medicaid Managed Care entities and their medical groups, encounter data are very important as often Medicaid pays capitation rates based on the data submitted. The more quality encounter data, the better capitation rates flow back to those Managed Care entities and their trading partners, stabilizing the business operations. Therefore, for everyone's best interest, it is best to submit good, clean, quality encounter data files to the State, which means the files should not have any integrity issues or syntax errors.



The Challenge

In the following hypothetical scenario, a Managed Care Payer collects and aggregates encounter data from twelve different trading partners into one big encounter file and then submits it to a State's health agency on a monthly or quarterly basis for quality of care audit requirements. Later, this Payer receives one big '835 EDI' Electronic Remittance Advice file back from the State which gives a zero-dollar disposition of all submitted encounters.

The Payer wants to share the information responded from the State with each submitting trading partner so that the submitting trading partners have an opportunity to review their data integrity issues, improve the quality of their future encounter data, and improve the quality of their processes, for example, discontinuing the use of invalid CPT/Diagnosis codes and so on. This is where the payer might feel sensitively challenged with the in-house EDI solution. To avoid an unnecessary PHI disclosure and HIPAA breach, the Payer needs to carefully split the big 835 EDI file and distribute back to each trading partner an individual 835 EDI file for the encounter data that they originally submitted, without accidentally sharing other trading partners' encounter data. This is where Chiapas EDI 2014 can be used to supplements the payer's in-house EDI solution, fulfill this deli-

The Solution

Because Chiapas EDI 2014 is a SQL Server-native integration system with innovative integration pipelines built-in, the payer can just plug it in SQL Server Management Studio and start to manipulate the solution.

For the demo purposes, the following exercise has two parts:

1. Create a mock 835 ERA file from Medicaid that contains data from twelve different providers
2. Load this file into Chiapas EDI's bin system and then do the transformations necessary to split it into twelve 'smaller' 835 EDI files.

Part 1: Create a mock 835 ERA file based on encounters submitted by twelve providers

For the demo purposes, we will keep things simple and use the **SEED_835.txt** file included with the Chiapas EDI distribution as our data source and make twelve copies of the data. We will assume that the Claim ID is the 'unique key' that enables the payer to tie each encounter back to the original submitting provider. Next, we will create the new EDI file to have a **PROV00** prefix before each Claim ID, where the **00** is in a range of **01** to **12**, representing each of the twelve submitting providers.

Here are the steps to create a big 835 EDI file:

- a. Load the **SEED_835.txt** file into the BIN system
- b. Create a temporary table with twelve copies of this data, each set having a different Claim ID prefix.
- c. Tell the Chiapas EDI integration system to create a new file.

Step 1a: Load the SEED_835 to the BIN system

From a DOS Command Shell prompt, enter the following:

```
C:
cd c:\chiapas\pipeline\007_DecodeToBINDB_Merge\01_in_edi
copy c:\chiapas\sample\seed_835.txt .
```

This will put the Chiapas EDI integration system to decode the new file and put it into the BIN_835 database table. To view the results, we will have to open SQL Server Management Studio. Open SSMS, then right click 'New Query' on the ChiapasData database that is installed with Chiapas EDI integration platform. Then, copy/paste the following into the SQL window.

The integration system may take up to *15 seconds* to process the file.

```
-- Step 1a: View the Seed 835 data
SELECT BIN_835.*
FROM      BIN_835
WHERE     BIN_ID =
          (
            SELECT      TOP 1
                       BIN_ID
            FROM          BIN_LOG
            WHERE         BIN_FILENAME =
'c:\chiapas\pipeline\007_DecodeToBINDB_Merge\02_done_edi\seed_835.txt'
```

The SQL Results pane should look like the following:

BIN_PK	BIN_ID	U1_OEISA_S01_ISA02_AUTH_NFO_OVL_NO_AUTH_NFO	OEISA_S01_ISA04_SEC_NFO_OVL_NO_SEC_NFO	OEISA_S01_ISA06_SENDR_ID_OVL_MUTLY
1	1	SAMPL_SENDRID..
2	2	SAMPL_SENDRID..
3	3	SAMPL_SENDRID..
4	4	SAMPL_SENDRID..
5	5	SAMPL_SENDRID..
6	6	SAMPL_SENDRID..
7	7	SAMPL_SENDRID..
8	8	SAMPL_SENDRID..

Step 1b: Create a temporary table with twelve copies of this data, each having a different Claim ID prefix.

Again, copy/paste this into the SQL window, highlight it and press **F5** to execute it.

```
SELECT      BIN_835.*,
            CLMPRE.PREFIX + L2100_S01_CLP07_PAYR_CLM_CONTRL_NR NEWCLAIMID
INTO        BIN_835_12PROV
FROM        BIN_835
INNER JOIN
(
    SELECT 'PROV01' PREFIX UNION SELECT 'PROV02' UNION SELECT 'PROV03'
UNION SELECT 'PROV04' UNION
    SELECT 'PROV05' PREFIX UNION SELECT 'PROV06' UNION SELECT 'PROV07'
UNION SELECT 'PROV08' UNION
    SELECT 'PROV09' PREFIX UNION SELECT 'PROV10' UNION SELECT 'PROV11'
UNION SELECT 'PROV12'
) CLMPRE ON 1=1
WHERE      BIN_ID =
            ( SELECT      TOP 1
              BIN_ID
              FROM        BIN_LOG
              WHERE      BIN_FILENAME =
                'c:\chiapas\pipeline\007_DecodeToBINDB_Merge\02_done_edi\seed_835.txt'
            )

ALTER TABLE BIN_835_12PROV DROP COLUMN BIN_ID
ALTER TABLE BIN_835_12PROV DROP COLUMN BIN_PK
ALTER TABLE BIN_835_12PROV DROP COLUMN L2100_S01_CLP07_PAYR_CLM_CONTRL_NR
EXEC sp_RENAME 'BIN_835_12PROV.NEWCLAIMID' , 'L2100_S01_CLP07_PAYR_CLM_CONTRL_NR',
'COLUMN'
```

This SQL query will take the data from the 835 seed file we've just loaded and duplicate it twelve times with the different prefixes specified within the INNER JOIN clause. The ALTER statements will remove the BIN Identifier and Primary Key columns, and then alter the 'L2100_S01_CLP07_PAYR_CLM_CONTRL_NR' column so that it has the twelve prefixes. This is how the 835 Claim ID field is specified within Chiapas EDI's internal mapping scheme.

Step 1c: Tell the Chiapas EDI integration system to create a new file.

Execute the following SQL:

```
INSERT INTO EVENTS (EVENT_OWNER, TSRC_ID, EVENT_DATA01, EVENT_DATA02)
SELECT      -1,
            12,
            'EXEC USP_ORDEREDI 'BIN_835_12PROV'',
            '835_EDI_12PROV.TXT'
```

After a few seconds, Chiapas EDI will generate a new, 48 KB EDI file within the following folder:

C:\chiapas\pipeline\012_EncodeFromBINDB\01_out_edi

And now we have a mock combo 835 EDI 'remittance advice' file from a State healthcare agency.

Part 2: Load the combo 835 ERA file and split it into twelve smaller 835 EDI files

The steps to accomplish this split are as follows:

- a. Load the mock 835 EDI file back into the BIN_835 bin table
- b. Create another temporary staging table and remove the ID and PK fields
- c. Tell Chiapas EDI to create 12 new outbound 835 EDI files based on the temp table

Step 2a: Load the mock 835 EDI file back into the BIN_835 bin table

Back in the DOS Command Shell, enter the following:

```
C:
cd c:\chiapas\pipeline\007_DecodeToBINDB_Merge\01_in_edi
copy
C:\chiapas\pipeline\012_EncodeFromBINDB\01_out_edi\835_edi_12prov.txt .
```

This will pull the mock 835 ERA file into the BIN system.

Step 2b: Create a temporary staging table and remove the ID and PK fields

```
SELECT      *
INTO        BIN_TMP
FROM        BIN_835
WHERE       BIN_ID =
(
            SELECT      TOP 1
                    CONVERT(VARCHAR, BIN_ID) BINID
            FROM        BIN_LOG
            WHERE       BIN_FILENAME = 'c:\chia-
pas\pipeline\007_DecodeToBINDB_Merge\02_done_edi\835_EDI_12PROV.TXT'
)
ALTER TABLE BIN_TMP DROP COLUMN BIN_PK
ALTER TABLE BIN_TMP DROP COLUMN BIN_ID
```

This will create a new temp table. Because we are bypassing the BIN system for this example, we can't use the stored procedures that normally scrubs the BIN data before it is sent to the Chiapas EDI encoding engine.

Step 2c: Generate 12 smaller 835 EDI files

The BIN Encoder pipeline is quite versatile – it can take data from either a BIN table (via a stored procedure) or, in fact, any database table as long as it only contains CGate FLAT column names. In this scenario, we are using dynamic SQL to tell the automation system to pull twelve different extracts from the temp table we've created earlier, and send the encoded output to a new filename.

Execute the following:

```

-- 2c. Generate the events that will create twelve new EDI files
INSERT INTO EVENTS (EVENT_OWNER, TSRC_ID, EVENT_DATA01, EVENT_DATA02)
SELECT      DISTINCT
            -1,
            12,
            'SELECT * FROM BIN_TMP WHERE L2100_S01_CLP07_PAYR_CLM_CONTRL_NR
LIKE '%' + SUBSTRING(L2100_S01_CLP07_PAYR_CLM_CONTRL_NR,1,6) + '%' ORDER BY
L2100_S01_CLP07_PAYR_CLM_CONTRL_NR',
            '835_EDT_SPLIT_' + SUBSTRING
(L2100_S01_CLP07_PAYR_CLM_CONTRL_NR,1,6) + '.TXT'
FROM        BIN_TMP

```

After a few seconds, Chiapas EDI encoding engine will generate twelve new 835 EDI files for twelve different trading partners.

Conclusion

This tutorial covers the whole process of creating a big 835 EDI file compiled from twelve small 835 EDI files, then splitting it back into twelve new EDI generation jobs, commanding Chiapas EDI encoding engine to generate a new 835 EDI file for each Claim ID prefix present within the file. If you happen to face a challenge of splitting a big 835 EDI file, you've got a tool to handle it— Chiapas EDI 2014. You are invited to download the free 30-day evaluation version of Chiapas EDI 2014 at <http://www.chiapas-edi.org>. Our unlimited next business day e-mail support starts the moment you download the evaluation version.

As Chiapas EDI encoding and decoding engines work natively with SQL tables, they simplify many operations that would be very labor intensive with competing EDI technologies. In short, the capabilities of Chiapas EDI integration platform give SQL developers abilities to work with EDI files without having to learn extensively about the intricacies of HIPAA transactions.

If you have any questions or comments regarding the use of **Chiapas EDI 2014** for any healthcare EDI situations you may encounter, please feel free to email us at support@chiapas-edi.org. We will be glad to be of help.