Data Warehouse Documentation

for

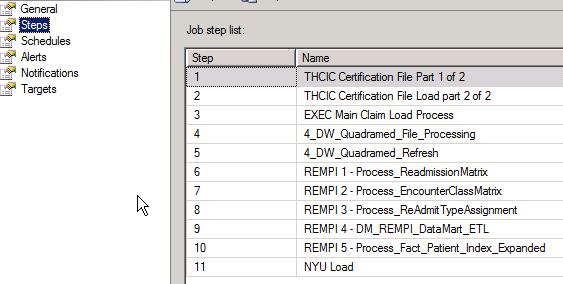
DFW Hospital Council Foundation

Prepared by:



# Primary Nightly Processing

The bulk of Claim processing is handled by a nightly SQL Server Agent Job:



The Job processing any Historical THCIC Cert File loads to put in back claims for a new hospital joinging the IQSC Data Initiative

# THCIC Certification File Processing

**Packages:**   
9\_LoadTHCICCertificationFile.dtsx  
9\_LoadTHCICCertificationFileToWarehouse.dtsx

**Description:**

This package handles the loading of THCIC Certification files.

This package is controlled by entries in the following table: IQSC\_DataWarehouse.CertFile\_Staging.CertificationFileQueue

When a new set of certification files are received and are ready to be processed an entry will have to be added to this table with the following information:

DataYear (ex: 2009)  
DataQuarter (ex: 4)  
Hospid (IQSC\_Hopsital\_ID) Ex: 870044  
CertificationFileName (ex: D:\Projects\Legacy\th859900\_4q09\_inpat\_enc.txt)  
ChargeFileName (ex: D:\Projects\Legacy\th859900\_4q09\_inpat\_rev.txt)  
SubmissionSrc (For THCIC Certification Files, this will always be 11)

**Tables Impacted:**IQSC\_DataWarehouse.ETL.CurrentQuarterList  
IQSC\_DataWarehouse.dbo.DimQueueStatus  
IQSC\_DataWarehouse.CertFile\_Staging.CertificationFile\_09q2  
IQSC\_DataWarehouse.CertFile\_Staging.ChargeFile\_09q2  
IQSC\_DataWarehouse.dbo.DimBatch  
IQSC\_DataWarehouse.CertFile\_Staging.ClaimBuildLog  
IQSC\_DataWarehouse.dbo.DimClaimPatient

IQSC\_DataWarehouse.dbo.FACT\_Claim

IQSC\_DataWarehouse.dbo.FACT\_ClaimCharge

IQSC\_DataWarehouse.dbo.FACT\_ClaimValue

IQSC\_DataWarehouse.dbo.FACT\_IQSC\_PUDF\_Computed

IQSC\_DataWarehouse.dbo.FACT\_IQSC\_PUDF\_MEDPAR

IQSC\_DataWarehouse.dbo.Map\_Claim\_Condition

IQSC\_DataWarehouse.dbo.Map\_Claim\_ECodes

IQSC\_DataWarehouse.dbo.Map\_Claim\_ICD\_Diagnosis

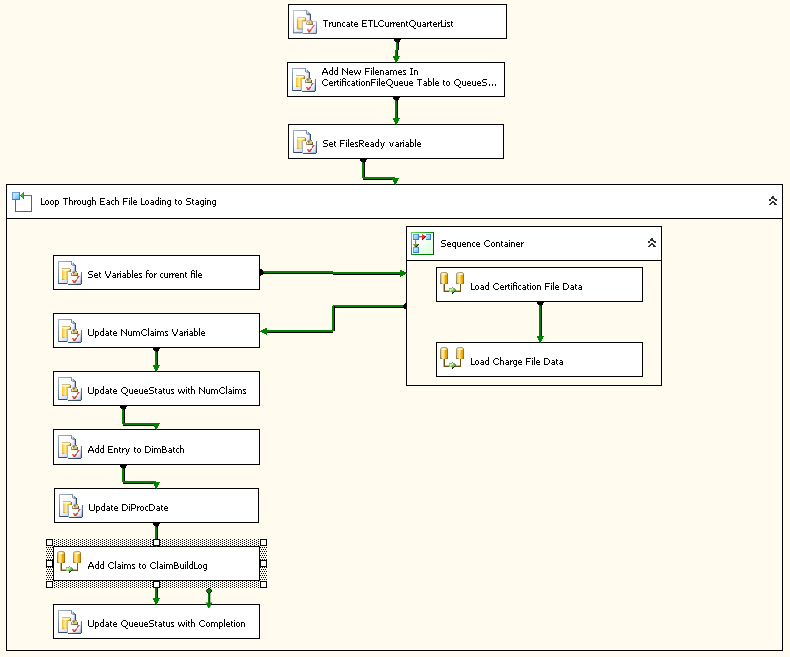
IQSC\_DataWarehouse.dbo.Map\_Claim\_ICD\_Procedure

IQSC\_DataWarehouse.dbo.Map\_Claim\_Occurrence

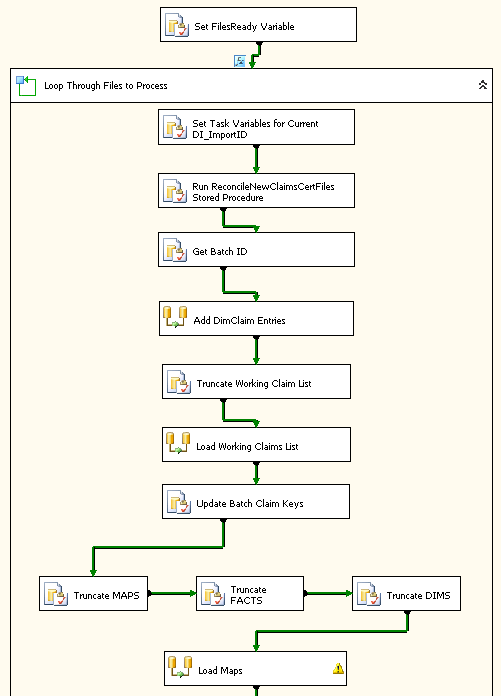
IQSC\_DataWarehouse.dbo.Map\_Claim\_OccurrenceSpan

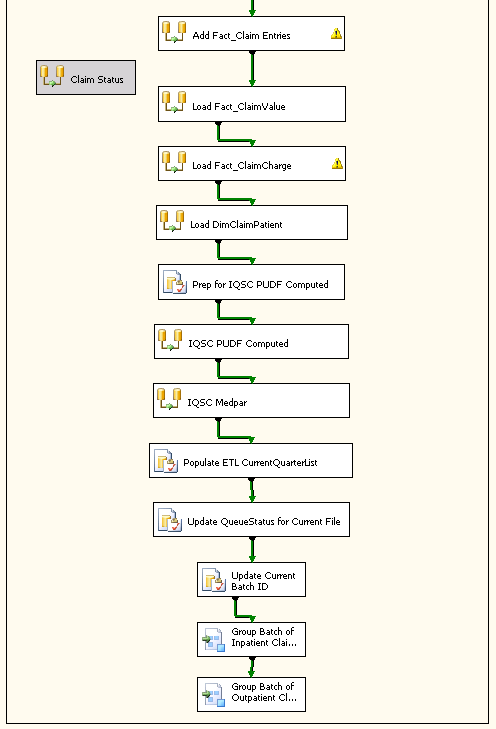
IQSC\_DataWarehouse.dbo.Map\_Claim\_Physician

**9\_LoadTHCICCertificationFile.dtsx  
Full Package:**

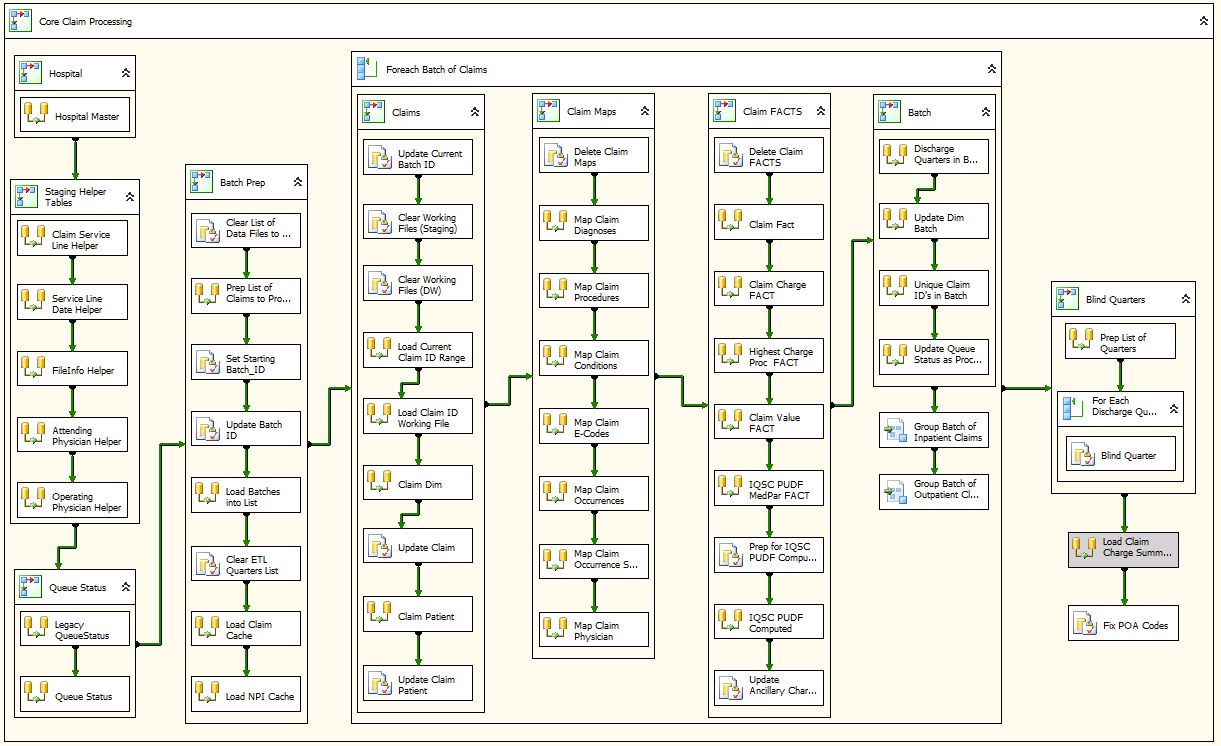


**9\_LoadTHCICCertificationFileToWarehouse.dtsx  
Partially Zoomed Package:**

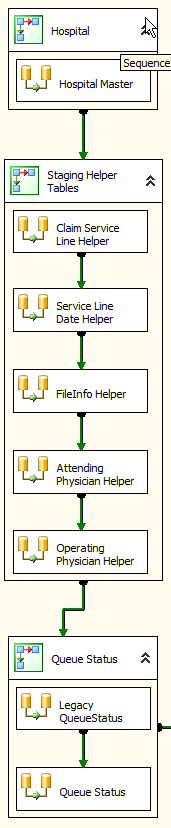




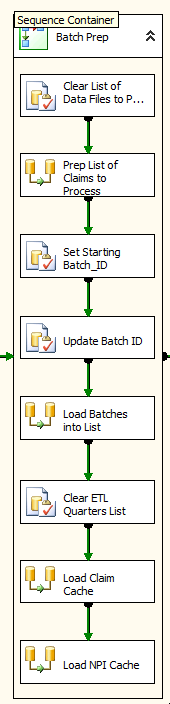
# Main Claim Load Processing



This is the primary Claims load package. It pulls all related Claim data over from DFWHC\_Staging (populated by DQA/DSS application)



1. “Hospital” This first step is needed until the Legacy system is phased out. The Dim Hospital table in the Data Warehouse is kept in sync with the Hospital Master on DISQLWAREHOUSE
2. Second Block of activities creates several “helper” tables in the DFWHC\_Staging.DataWarehouse schema. These are necessary due to the poor indexing and query performance of several tables in the DFWHC\_Staging database.
   1. Claim Service Line Helper table makes pulling Claim Charge detail over significantly faster
   2. Service Line Date Helper pulls “Service Date” which is a new attribute populated in FACT\_ClaimCharge.
   3. FileInfo Helper: This pulls the POA\_Code string out of Staging to be used to re-populate POA\_Code in each Claim Diagnosis record (due to a bug in DQA/DSS).
   4. Attending & Operating Physician Helper tables – this was another area of very poor performance. By doing this pre-processing step the nighly pull of Claim physician information is significantly improved.
3. Legacy QueueStatus – This pulls over a copy of the QueueStatus Claim File driver tables for Inpatient / Outpatient claims on DISQLWAREHOUSE. This is used to keep track of what has been processed over on the Legacy side in contrast to what has is processed in the new Data Warehouse. This will go away once the Legacy claims processing system is shut down.
4. Queue Status step – This keeps the Dim Queue Status table loaded with the list of Hospital Claim Files waiting to process.

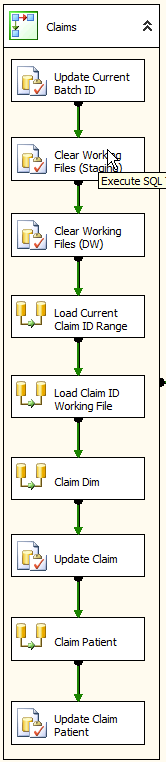


“Batch Prep” is a series of steps that prepare the current set of Claims to be processed into “Batches”. A “Batch” accomplishes two things:

1. Batches break the claims down into smaller datasets, which improves processing throughput.
2. Batches make sure that Hospital Claim files are processed in the right order if more than one file had dropped in a day or since the last time the Claim processing had run

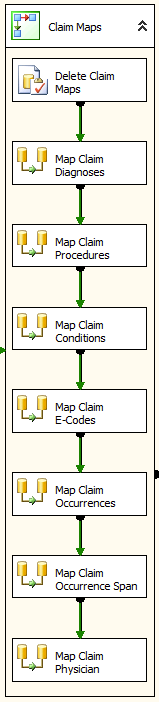
The “Batch Prep” tasks are as follows:

1. First the “helper” tables are truncated to be clear and ready for use
2. A list of Hospital Claim Files are retrieved from the DimQueueStatus table. Currently this is capped at 15 files to limit the amount of claims processed in any given night but this will probably be tweaked up or down over time based on observation.
3. The “Starting” Batch ID is set from Dim Batch (one greater than the last Batch ID).
4. Batch ID’s are set in the list of Claim Data Files
5. This Batch Claim File list is loaded into an in-memory “Recordset” object which drives the “For Each” sequence container that follows.
6. The ETL Quarters List table is cleared in the Data Warehouse. This is populated with a list of Discharge Quarters from this particular night’s run which is used for Blinding, among other things.
7. The In-Memory Claim Lookup cache is loaded. This makes the Claim Lookup step(s) faster later.
8. The In-Memory NPI Lookup Cache is loaded, again it makes the SSIS lookup transform for that dataset faster overall in later steps.
9. The “Batch Prep” Sequence Container leads into a “For each” loop that runs claims processing for each batch of claim records coming from staging.



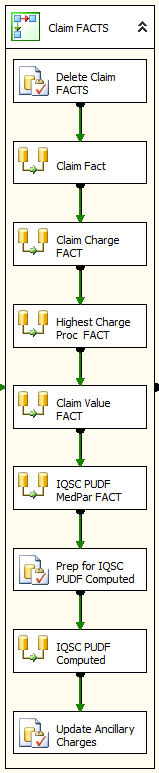
The “Claims” Sequence Container is the crucial first step to getting the Batch setup and the core Claim table updated. The steps are as follows:

1. The Current Batch ID is set based on the last Batch ID used in DimBatch
2. Several “working” tables are cleared in the Data Warehouse schema of DFWHC\_Staging database.
3. “Working” tables are cleared in the Data Warehouse as well.
4. The current DQA/DSS Claim Id range is set for the batch
5. A narrow “Claim ID” working file is loaded based on the Claim ID range previously set in order to improve performance of the load
6. The Primary Claim Dimension is loaded. This happens in two steps, first any new Claims are loaded and secondly Claim updates are processed separately.
7. Similarly, Claim Patient is processed. Inserts are handled in one step, Updates are handled in another.



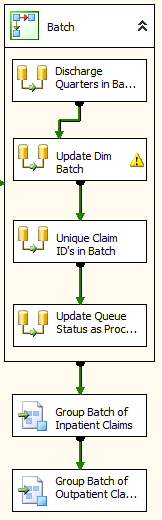
The “Claims Maps” Sequence Container pulls in all the “many-to-many” tables surrounding claims. Specifically:

1. First step clears out the records in all the Map tables (via a stored procedure) for the current Batch being processed
2. Map Claim Diagnosis contains all the ICDx Diagnosis records for a Claim
3. Map Claim Procedure contains all the ICDx Procedures for a Claim
4. Similarly, Conditions, E-Codes, Occurrences, Occurrence Span and Claim Physician.



The “Claims FACTS” Sequence Container pulls in all the FACT tables surrounding claims. Specifically:

1. First step clears out the records in all the FACT tables (via a stored procedure) for the current Batch being processed
2. Claim FACT is loaded – this is at the same grain as Dim Claim – contains the bulk of the Dim connections for “slicing” Claims
3. Claim Charge FACT is loaded next – this is ultimately the largest FACT table in the Data Warehouse
4. Highest Charge Procedure FACT is a new table introduced to assist in some reporting that Greg wanted that is not performant otherwise
5. Claim Value is another 1-to-many FACT table tied to Claim
6. Next are the loads for two of the three related tables that are used to constitute the IQSC “PUDF” dataset:
   1. IQSC PUDF Computed
   2. IQSC PUDF MEDPAR
7. Next Ancillary & Accomodation charges are updated in the Claim FACT based on the revenue codes associated to the detail Claim Charges.



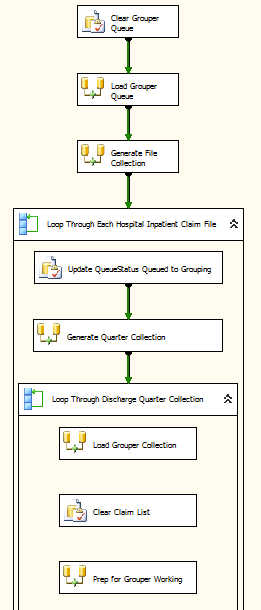
The final steps of the Batch “LOOP” wrap things up. First, it is determined what “Discharge Quarters” there are in the batch.

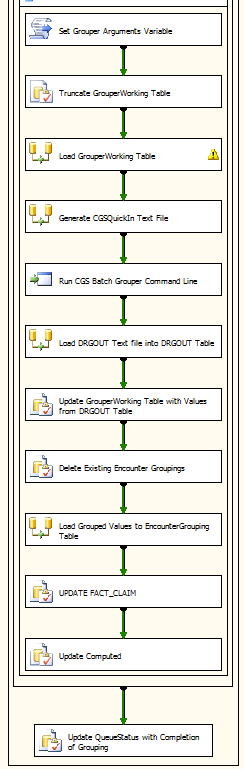
Second, the Dim Batch is Updated, and all the Hospital Claim Data Files that were processed are updated in the Dim Queue Status table

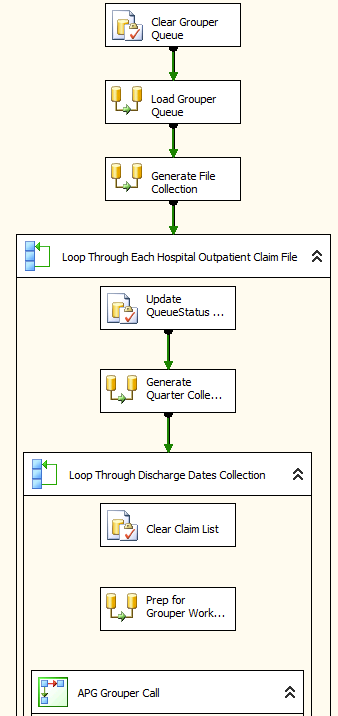
Finally, this batch of claims is sent to the Inpatient & Outpatient groupers.

# Inpatient / Outpatient Grouper Processing

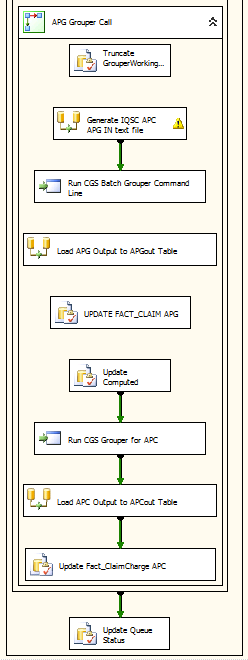
* 1. Inpatient Grouper (Top Half)



* 1. Inpatient Grouper (Bottom Half)  
       
     
  2. Outpatient Group (Top Half)



* 1. Outpatient Group (Bottom Half)



# QuadraMed Processing

Packages:   
4\_DW\_Quadramed\_File\_Processing.dtsx  
QuadraMed\_32bit\4\_DW\_Quadramed\_Refresh3.dtsx

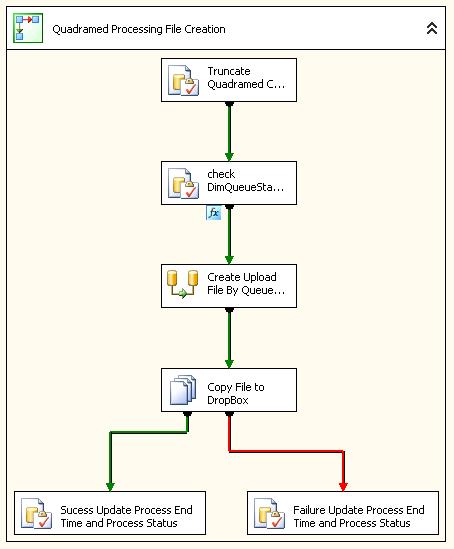
**Description:**

This package generates files to be processed by QuadraMed and updates the data based on the results which QuadraMed returns. If a file is generated successfully, it will be moved to the following directory: \\house\QuadraMed\_DI\_Extract\LoadSupp.

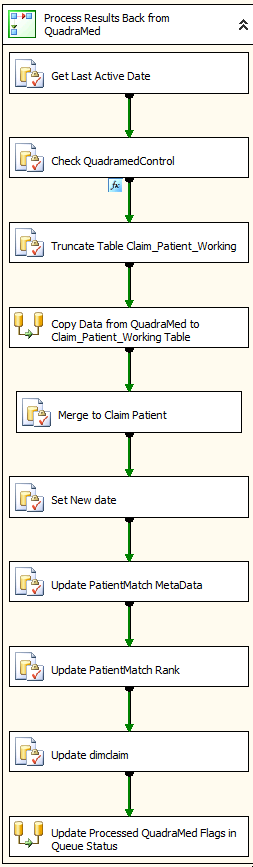
**Tables Impacted:**

IQSC\_DataWarehouse.ETL.QuadramedControl  
IQSC\_DataWarehouse.ETL.Quadramed\_QueueStatusList  
IQSC\_DataWarehouse.dbo.DimQueueStatus  
IQSC\_DataWarehouse.QuadraMed.Claim\_Patient\_Archive  
IQSC\_DataWarehouse.QuadraMed.Claim\_Patient

**4\_DW\_Quadramed\_File\_Processing.dtsx  
Full Package:**

****

**5\_DW\_Quadramed\_Refresh.dtsx  
Full Package:**



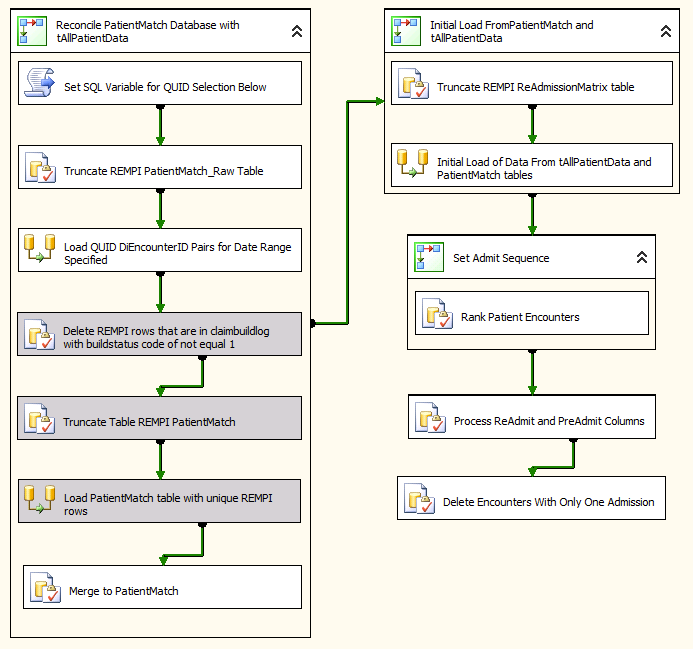
# REMPI Processing

**REMPI Process**

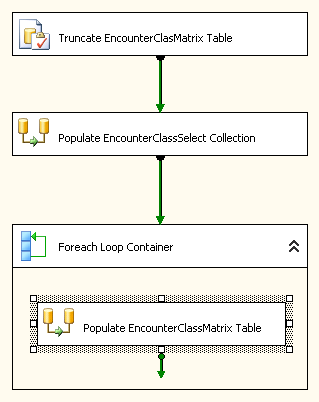
Packages:   
8\_REMPI\_1\_Process\_ReadmissionMatrix.dtsx  
8\_REMPI\_2\_Process\_EncounterClassMatrix.dtsx  
8\_REMPI\_3\_Process\_ReAdmitTypeAssignment.dtsx  
8\_REMPI\_4\_DM\_REMPI\_DataMart\_ETL.dtsx  
8\_REMPI\_5\_Process\_Fact\_Patient\_Index\_Expanded.dtsx  
 **Description:**

**Tables Impacted:**

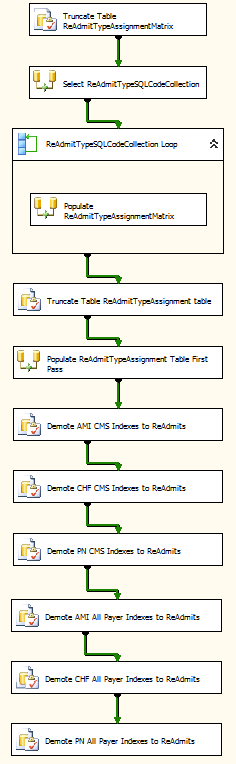
IQSC\_DataWarehouse.REMPI.PatientMatch\_Raw   
IQSC\_DataWarehouse.REMPI.PatientMatch  
IQSC\_DataWarehouse.REMPI.ReAdmissionMatrix  
IQSC\_DataWarehouse.REMPI.EncounterClassMatrix  
IQSC\_DataWarehouse.REMPI.ReAdmitTypeAssignment  
IQSC\_DataWarehouse.REMPI.ReAdmitTypeAssignmentMatrix  
IQSC\_DataWarehouse.REMPI.FactEncounter  
IQSC\_DataWarehouse.REMPI.FactComorbid  
IQSC\_DataWarehouse.REMPI.FactReAdmit  
IQSC\_DataWarehouse.REMPI.REMPI\_Rates\_Data\_Cube  
IQSC\_DataWarehouse.REMPI.REMPI\_Reimbursement  
IQSC\_DataWarehouse.REMPI.Fact\_Patient\_Index\_Expanded

**8\_REMPI\_1\_Process\_ReadmissionMatrix.dtsx  
  
**

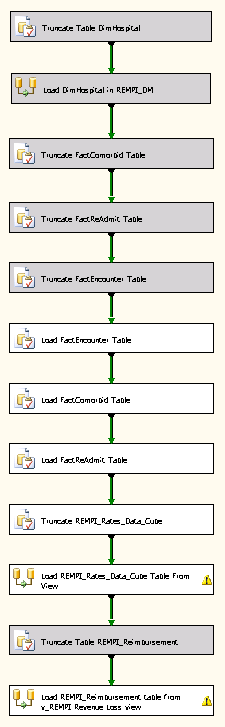
**8\_REMPI\_2\_Process\_EncounterClassMatrix.dtsx**

****

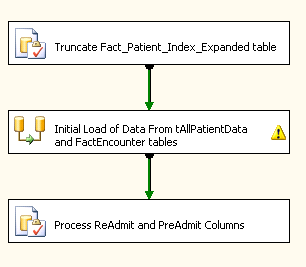
**8\_REMPI\_3\_Process\_ReAdmitTypeAssignment.dtsx**

****

**8\_REMPI\_4\_DM\_REMPI\_DataMart\_ETL.dtsx**

****

**8\_REMPI\_5\_Process\_Fact\_Patient\_Index\_Expanded.dtsx**

****

# THCIC PUDF File Load Processing

Package: Load\_THCIC\_PUDF\_2004\_Format.dtsx

**Description:**

This package handles the loading of the data files received from the state.

It relies on two files – Base\_Tab.txt and Charges\_Tab.txt which should be loaded in the DropBox location currently mapped to D:\DropBox\THCIC\_PUDF\_Files\

This package will verify that both files exist prior to processing.

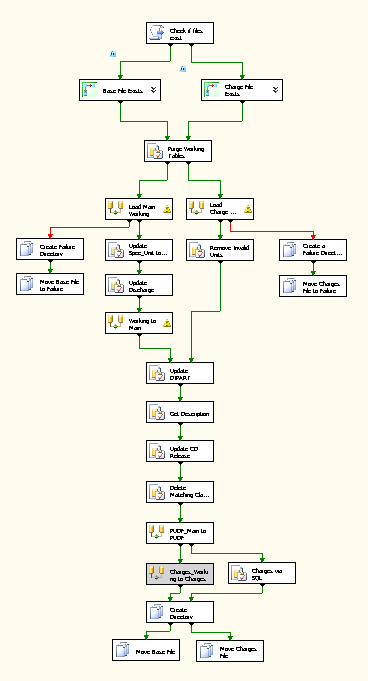
If a file exists, but encounters an error during processing, it will create a folder in the Failure directory with the Date of the run and move the failed file. Ex: D:\DropBox\THCIC\_PUDF\_Files\Failure\11302011

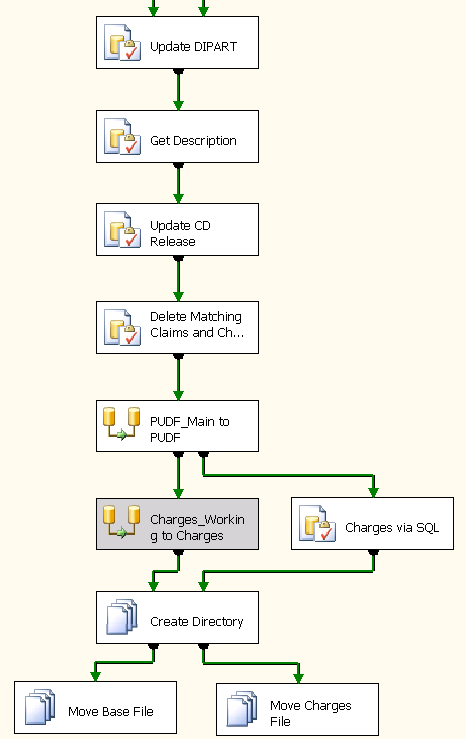
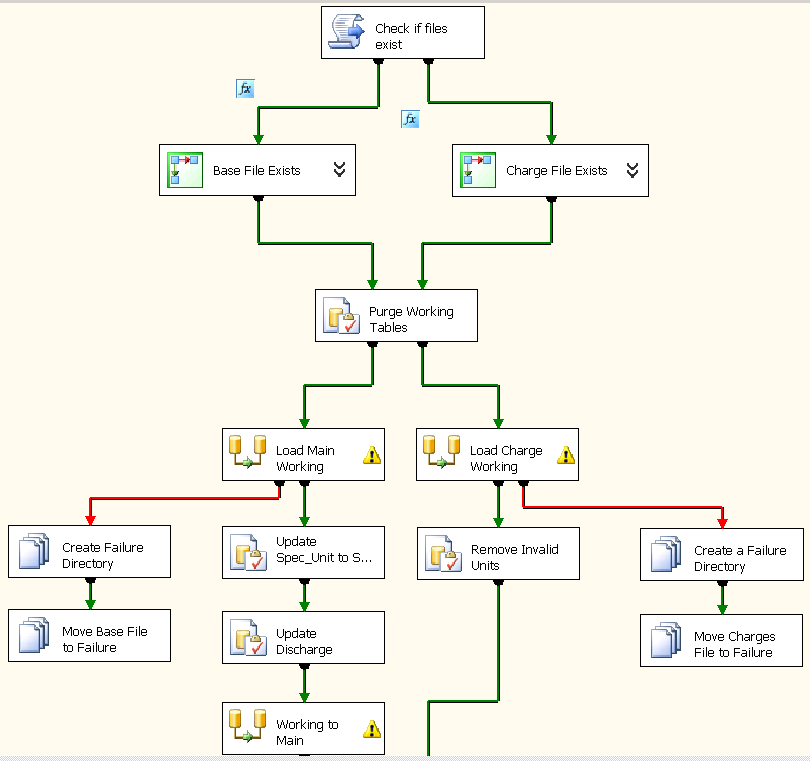
If the files successfully process it will create a folder in the Success directory with the Date of the run and move the completed files. Ex: D:\DropBox\THCIC\_PUDF\_Files\Success\11302011

**Tables Impacted:**

THCIC.dbo.THCIC\_PUDF\_Main\_Working  
THCIC.dbo.THCIC\_PUDF\_Main  
THCIC.dbo.THCIC\_PUDF\_Charges\_Working  
THCIC.dbo.THCIC\_PUDF\_Charges

**Full Package:**



**Zoomed Package:**

# | Master Death Index Load Process

Package: 7\_DW\_MasterDeathIndexLoad.dtsx

**Description:**

This package will process any of the .txt files located in the following DropBox Location: D:\DropBox\MasterDeathIndex\_Files

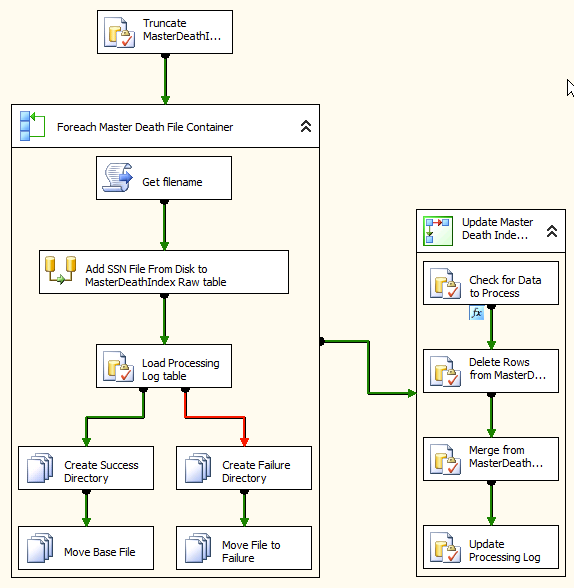
If a file exists, but encounters an error during processing, it will create a folder in the Failure directory with the Date of the run and move the failed file.   
Ex: D:\DropBox\MasterDeathIndex\_Files\Failure\11302011

If the files successfully process it will create a folder in the Success directory with the Date of the run and move the completed files.   
Ex: D:\DropBox\MasterDeathIndex\_Files\Success\11302011

**Tables Impacted:**

IQSC\_DataWarehouse.MasterDeathIndex.MasterDeathIndex\_Raw  
IQSC\_DataWarehouse.MasterDeathIndex.MasterDeathIndex\_ProcessingLog  
IQSC\_DataWarehouse.MasterDeathIndex.MasterDeathIndex

**Full Package:**

****

# | AHRQ File Load Process

**Packages:**   
1\_DW\_AHRQ\_THCIC\_File\_Loader.dtsx  
1\_DW\_AHRQ\_DI\_File\_Loader.dtsx  
 **Description:**

This package handles the loading of the AHRQ data files.

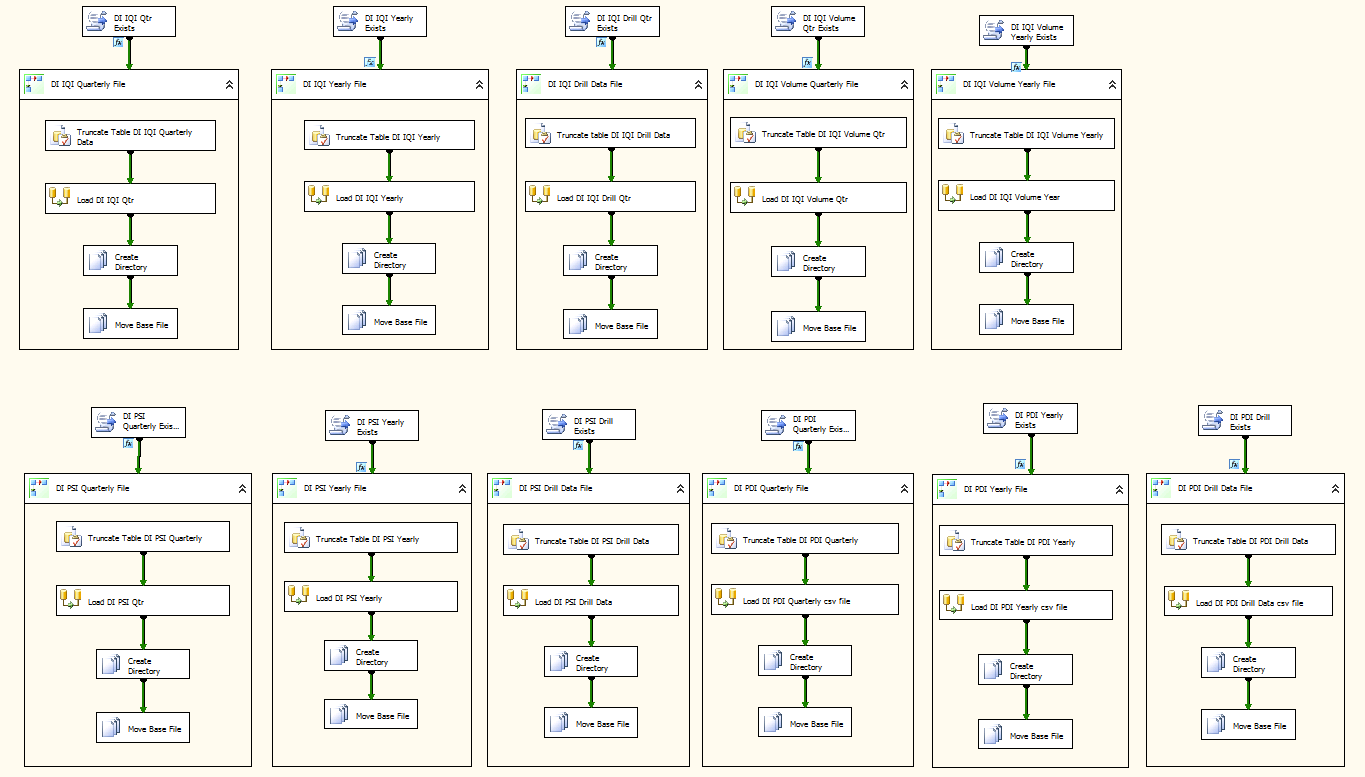
It loads the following files:  
iqi\_qtr.csv  
iqi\_vol\_qtr.csv  
iqi\_vol\_yr.csv  
iqi\_yr.csv  
pdi\_qtr.csv  
pdi\_yr.csv  
pqi\_qtr.csv  
pqi\_yr.csv  
psi\_qtr.csv  
pis\_yr.csv

The Files for THCIC should be placed in D:\DropBox\AHRQ\_Files\THCIC\_Data\  
The Files for DI should be placed in D:\DropBox\AHRQ\_Files\DI\_Data\

This package will load the files independently if they exist.

Once the files successfully process they will create a folder in the Success directory with the Date of the run and move the completed files. Ex: D:\Dropbox\AHRQ\_Files\THCIC\_Data\Success\11302011 for THCIC files and D:\Dropbox\AHRQ\_Files\DI\_Data\Success\11302011 for DI Files.

# | 1\_DW\_AHRQ\_DI\_File\_Loader.dtsx



Note that the various AHRQ tables, being independent, load in parallel.

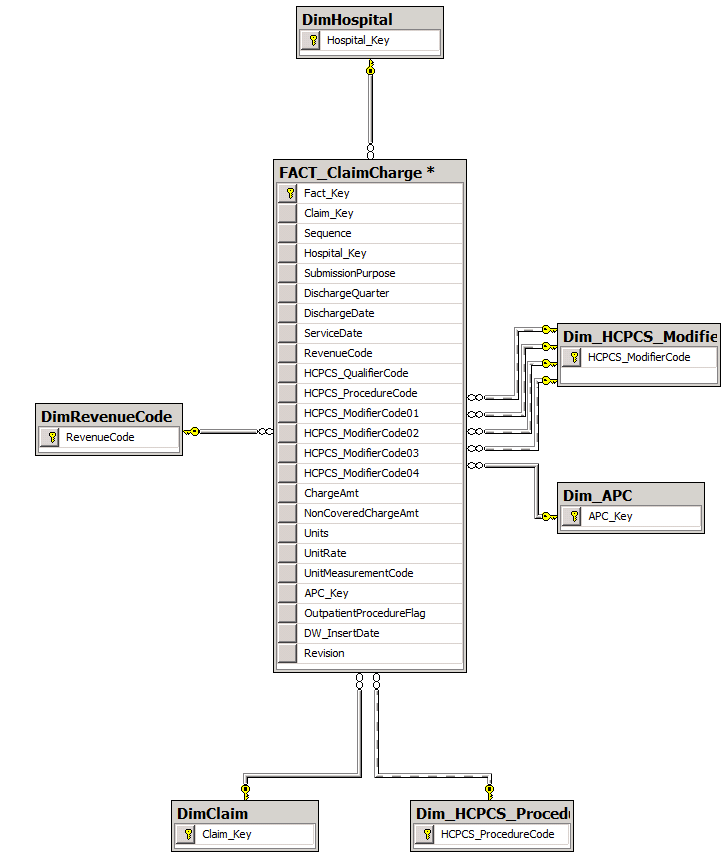
# | 1\_DW\_AHRQ\_THCIC\_File\_Loader.dtsx

# | Database Schema

# | Primary Claim Star Schema

# 

# Primary Claim Star Schema



FACT\_ClaimCharge Star Schema – There is one row per Claim Charge record and this FACT table relates to it’s key dimensions as shown.