National Patient Information Reporting System: National Data Warehouse

NDW Data Mart DB2

Glossary

Version 2.0

June 2011

Department of Health and
Human Services

Indian Health Service

Office of Information
Technology (OIT)

Contents

Version Control iii

Overview 1

Constructs 1

Table 1
Indexes 1
Views 1
MQTs 2
Temporary Tables 2
Tablespaces (including Normal and Large) 2
Tablespaces, System 3
Sequences 3
Nicknames 3

Compression 4

Federation 4

HACMP/HA 5

NDW Processes 5

ACK Process 6
IMP Process 7
Extract Process 7
Sweeper Process 9
Backup Process 11
Data Integrity Process 11
Un-Duplication Process 11
Post-Load Reports 12
MatchMaker Process 12
Error Tracking Reports 12

Test Environments 13

NPIRS Internal Test Environment 13
NPIRS Data Mart Developers' Test Environment 13

Version Control

Version	Date	Notes
1.0	June 2009	Initial version. Pulled duplicate sections of information from technical guides and created this glossary as one point of reference. COTR approved June 16, 2009.
1.1	January 2011	Annual Review & Update Added additional DB2 processes and their descriptions; replaced sweeper diagram with new one.
2.0	June 2011	Final (w/minor editorial changes)

Overview

This document defines concepts and terminology related to the National Data Warehouse (NDW).

Constructs

The main construct types associated with the NDW are as follows:

Table

A table is a grouping of data having the same construct; usually, but not always, keyed and indexed. Target tables are considered the factual and/or dimensional data about the NDW. All other tables exist in support of the Target tables (Registration and Encounter schemas). Although no free space is associated with the tables (keys are sequential), periodic reorganizations (reorgs) are performed to create new compression dictionaries and eliminate high water marks.

Indexes

Indexes, a physical construct, allow rapid access to data in tables. All primary keys are defined with hashing and are defined with names. Defaults are not allowed. Other indexes include clustering, partitioning, duplicate, unique, and “include,” or can be hybrids of these depending on the objective desired. Free space may or may not be allocated the index depending on the type of index, the likelihood of page or leaf insertion or appendage.

Indexes are reviewed periodically to determine the depth of the tree and page leafing, to determine if a change to the free space is required or if reorganization (reorg) is required. Reorgs are done periodically to reduce high-water marks.

Views

A calculated and structured way of looking at data, often with calculations and joins, which facilitates extracts or resolves complex operations.

MQTs

Materialized Query Tables are similar to views but have the permanence of Tables and the capability to support statistics and indexes, as long as the underlying constructs survive.

Temporary Tables

Temporary tables are similar to tables, but transient in nature. Typically, temporary tables are updated as part of the Extract, Transform, Load (ETL)/Transform, Extract, Load (TEL) process.

Tablespaces (including Normal and Large)

Normal tablespaces are the actual physical storage area used by tables, and temporary tables. Each tablespace has its own memory buffer pool assigned to minimize contention. A tablespace is associated with a buffer pool (allocated memory) and inherits certain characteristics from the buffer pool, in particular, the page size. Tablespaces can either be system managed or database managed. System managed tablespaces grow automatically; database managed tables must be pre-allocated but are faster. All normal tablespaces are database managed.

Large tablespaces – not to be confused with the ability to support Large Object Binary (LOBs) and Character Large Object Binary (CLOBs) data types – are a unique construct under V9 to efficiently use a large Row ID (6 bytes instead of 4), which eliminates the storage barrier of virtually any tablespace and allows more rows per page. Using this construct, the largest single table in a single tablespace can be 512,000 Petabytes (PB). Almost all tablespaces are database managed, and almost all Database Managed Space (DMS) tablespaces are defined on pre-allocated devices rather than files, which allows more precise management of storage allocation. This decision to precisely manage storage was made, even though DMS/Device requires more intervention than DMS/File.

Tablespaces, System

System tablespaces are the actual physical storage area used by views, sorts, joins, and internal operations. This storage is both pre-allocated and dynamically allocated. A tablespace has its own memory buffer pool assigned to minimize contention. A tablespace is associated with a buffer pool (allocated memory) and inherits certain characteristics from the buffer pool, in particular, the page size. Tablespaces can be either system managed or database managed. System managed tablespaces grow automatically. Database managed must be pre-allocated, but are faster. Almost all tablespaces are database managed, and almost all DMS tablespaces are defined on pre-allocated devices rather than files, allowing more precise management of storage allocation. This decision to precisely manage storage was made even though DMS/Device requires more intervention than DMS/File.

Sequences

Sequences can be used by applications to “grab” a next sequential value for use in a table. Sequence objects are ideal for generating sequential, unique numeric key values. A sequence can be accessed and incremented by multiple applications concurrently without the hot spots and performance degradation associated with other methods of generating sequential values, such as table incrementing, and avoids the transportability issues on identity keys. DB2 does not wait for a transaction to COMMIT before allowing the sequence to be incremented again by another transaction. IMP, for example, will cycle at least hundreds of transactions before a commit.

Sequences are used by the Acknowledgement (ACK) and Import Engine (IMP) processes to assign automatically the next sequential number from cache for use by the process in a manner that is persistent and much faster than table access.

Nicknames

The ability of DB2 to reference tables in an external database is called Federation. The reference is referred to by nicknames. A nickname is a local name for a remote table and can support partial or complete tables, with or without security controls. Nicknames may be in any schema and completely mimic physical tables. Data changes in remote tables are reflected immediately in local nicknames, but structure changes are not.

Compression

Both column and row compression are utilized in the NDW. The compression algorithms maximize utilization of the mass storage paradigm while maintaining, or optimizing, retrieval I/O. Row compression provides a level of data de-duplication in that it stores the data in a separated internal table and uses pointers to the data. For example, if a table contains 968 values of 'Smith' and 312 values of 'Jones', the actual values are stored once and pointers for each are stored in the table.

Federation

A federated process is a DB2 process of connecting other databases or data sources through defined connections called crservers or Distributed Relational Database Architecture (DRDA) on an enterprise system. DRDA is a set of protocols, or rules, that enable a user to access distributed data regardless of where it physically resides. It provides an open, robust heterogeneous distributed database environment. DRDA provides methods of coordinating communication among distributed locations. This allows applications to access multiple remote tables at various locations and to have them appear to the end user as if they were a logical whole.

A distinction should be made, however, between the architecture and the implementation. DRDA describes the architecture for distributed data and nothing more. It defines the rules for accessing the distributed data, but it does not provide the actual application programming interfaces (APIs) to perform the access. So DRDA is not an actual program, but is more like the specifications for a program.

Federation is supported by non-homogenous databases (non-DB2) as well as ODBC data sources such as Microsoft's Excel© or SAS.

Additional information on federation is available at:

http://www.redbooks.ibm.com/redbooks/pdfs/sg247032.pdf

DB2 is a DRDA-compliant Relational Database Management System (RDBMS) product; that is, it follows the DRDA specifications. DRDA is supported and certified by The Open Group only on DB2 6.5 and higher, Informix 11 and higher, and Oracle Gateway for DB2.

Reference Tables are currently federated through a nickname. The Federation, from the Temecula database, is instantaneous.

HACMP/HA

High Availability Cluster Multi-Processing (previously known as Power-HA) is the AIX driven capability of ensuring a fail-over in the event of a server or primary component failure. DB2-High Availability (HA) is the database component of the same feature which allows for the DB2 databases to be set up as shared applications that failover to each server. These components work together to ensure continuous run capability in the event of most hardware failures.

Additional information on HACMP is available at:

http://www.redbooks.ibm.com/redbooks/pdfs/sg247363.pdf

NDW Processes

Several processes operate in the NDW and follow NPIRS Accepted Practices in their implementation. The table below briefly describes the main processes associated with the NDW database:

Process	Description
ACK	Acknowledges receipts of a file and notifies the sending site that the file has been received. Inserts a new record for each export in the Admin tables and assigns a unique export_id to the file. The export_id is assigned from the sequence cache.
IMP	Physically loads the files received and acknowledged by ACK into appropriate tables. Uses Meta data to determine table column and length information and mapping. Redirects errors to error tables. Updates Admin tables with results summary of the loading process.
Extract	Extracts information for use by various marts. Some extracts utilize views and MQTs. Some extracts are straight data extracts. The typical ETL process extracts data into a flat file from a view, table or MQT. These files are the either loaded, or inserted into separate data marts. Some use interim tables and the resultant table is then derived. Loads into various marts/databases may be with or without the ‘checklengths’ parameter enabled. This parameter is based on the precision of the column count desired. There are specific cases where extracts are used for multiple loads and truncation is desired for one or more of other loads.

Process	Description
Sweeper	Process which removes non-current records and records marked for deletion into the ENCTR_HIST schema tables that are compressed to save storage.
Back Ups	Performed daily to enable a full database restore in the event of a disaster or other requirement.
Data Integrity	Compares data physically stored within the NDW with Post-IE data files to ensure data received is stored properly. The process moves the data, untransformed, from the IE into ‘staging’ tables from which it may electronically be compared to the data in the NDW.
Un-Duplication	In order to count a person only once in each IHS Area, a pre-established set of business rules are applied to un-duplicate the registration and encounter records which are used for User Population and Workload reporting purposes.
Post-Load Reports	Report the status of files sent to the NDW. These reports are emailed to the sending site.
MatchMaker	Links “orphan” ENCTR records in the NDW to a valid REG record has been received subsequent to the load of the ENCTR record.
Error Tracking Reports	Information on errors encountered while loading the data from the files that are sent to the NDW.

ACK Process

ACK is an automated process that performs these functions:

- Verifies the basic integrity of each received file.
- Strips and archives NTE (NoTE) records from the file, if found. NTE records contain notational information associated with HL7 processing, but serve no purpose in the NDW Data Mart.
- Generates and assigns a unique export ID, which is a number that remains associated with the data set.
- Creates a database record that describes the data export file and its contents.
- Sends email reports that describe the ACK analysis and handling of each file.

The ACK process rejects files based on specified criteria. ACK is a JAVA application.

Additional information on this process can be found on the NPIRS Public drive at:

P:\NPIRS\7. NPIRS Documents\22. NDW Implementation Documents\Documatron Implementation\Certified Documentation\ACK

IMP Process

The Import Engine process (IMP) is an automated process that consists of a series of methods that validate and transform certain registration and encounter data before loading the data into the NDW database.

- The META tables, specifically META.COLUMNS, control the mapping for the IMP.
- Errors encountered and transformations performed during the load process are logged in the ADMIN.LOAD_ERRORS table.
- Registration and encounter records are loaded to target tables in the REG and ENCTR schemas.
- The ADMIN.EXPORT_INFO table is updated with file statistics as each file is processed.

The NDW is automatically updated by IMP Sunday through Thursday and on demand Friday through Sunday. This allows a maintenance and ETL window on Friday through Saturday. This also allows for faster performance of data mart refreshes.

Additional information on this process can be found on the NPIRS Public drive at:

P:\NPIRS\7. NPIRS Documents\22. NDW Implementation Documents\Documatron Implementation\Certified Documentation\ImportEngine

Extract Process

To ensure that marts are refreshed with a proper snapshot of the database, all extract processes involve at least the following:

- For Encounter (ENCTR) related data, a snapshot of either the desired ENCTRSS_ID or EXPORT_ID is stored in a temporary table. Extracts from all ENCTR schema tables are made from this temporary table. This process allows continued processing of incoming files while eliminating partial file extracts to a mart.
- A similar process is used for Registration (REG) related data.
- All extracts use a “with ur” (dirty read) operation to avoid conflicts with transactional processing.
- Data in the marts is either replaced, inserted, updated or deleted depending on the specific process involved.

Marts directly supported through an ETL/TEL process include the Test Data Mart, General Data Mart, Data Quality Mart, Export Tracking Mart, NDW Computation Mart, User Population Reporting Mart and the Workload Reporting Mart (TATONKA).

Extracts are performed from within the NDW production environment to flat files for loading to the various marts.

The extract processes utilized for the data transfer are of the following types:

1. EL (Extract, Load). The tables inside NDW are extracted into a flat file for subsequent load into another mart.

2. TEL (Transform, Extract, Load)

    a) A variation of the EL process, TEL builds a flat file from a view or MQT (Materialized Query Table) that pre-exists by performing calculations or transformations with or without joins and/or OLAP (On Line Analytical Processing).

    b) Builds a temporary table in SANDIA with calculations and transforms (with or without joins and/or OLAP), and then extracts these to another mart for load.

3) ETL (Extract, Transform, and Load) is an extract process, then a transform at the receiving process, prior to or during the load. This is often done with third party software.

4) TTEL (Transform/Transform, Extract, and Load) builds a temporary table in the database with calculations or transforms (with or without joins and/or OLAP), updates the table, then extracts these to another mart for load; then potentially, there is further updating of the table in the mart.

In addition, NDW supports several triggers to support the various marts in the HOLLYWD and WILDHRSE databases. These triggers identify data that has changed since the last extract of data from NDW environment for these databases. The triggers update audit type tables with pertinent information to be used in the incremental ETL/TEL processes.

Most extract processes are controlled by cron scripts, and most processes are re-entrant processes. Re-entrant processes are processes that can be restarted from a failure point, if they fail to run for some reason such as a conflict, resource, or storage issue. Processes not currently controlled by cron or not re-entrant are being modified to comply with this behavior.

Specific schedules and requirements are referenced by various SLA's.

All extract processes are SQL based processes with simple AIX shell script wrappers.

Sweeper Process

The NDW environment uses a weekly automatic Sweeper process (**Error! Reference source not found.**1), which runs automatically, to reduce storage requirements for both less used data and duplicate data. As the IMP process inserts records into the ENCTR tables,

- IMP sets the CURR_ENCTR_FG to mark older records for the same UNIQ_ENCTR_CODE as non-current.
- For records entered that are marked deleted by the Areas, IMP retains those records but sets ENCTR_DEL_FG to mark the record as deleted.

Data thus flagged have no use in day-to-day reporting, but must be retained for both historical and data quality reporting. Additional coding is required to exclude this data from most reporting. Removing this data periodically improves performance.

The Sweeper process identifies all ENCTRSS_IDs with either flag set. It then moves (Copy process, then Delete process) all records from ENCTR.ENCTRSS and related ENCTR schema tables to the ENCTR_HIST.ENCTR and ENCTR_HIST related tables.

After a successful move, the original target (ENCTR schema) tables are reorganized (Reorg process) to remove empty space left by the deleted records. This is necessary because the primary key, ENCTRSS_ID, is a sequential, unique value, and back space cannot be utilized.

The weekly Sweeper process is a precursor for the ETL process. An additional sweeper process – the History Sweeper – is also utilized. Whereas the weekly Sweeper moves data from the ENCOUNTER schema to the ENCOUNTER_HIST schema as soon as the data becomes non-current, the History Sweeper, which is run quarterly, moves data from ENCOUNTER_HIST into ENCOUNTER_HIST_ARCHIVE when the data is more than 2 years old (as of the most recent September 30th).

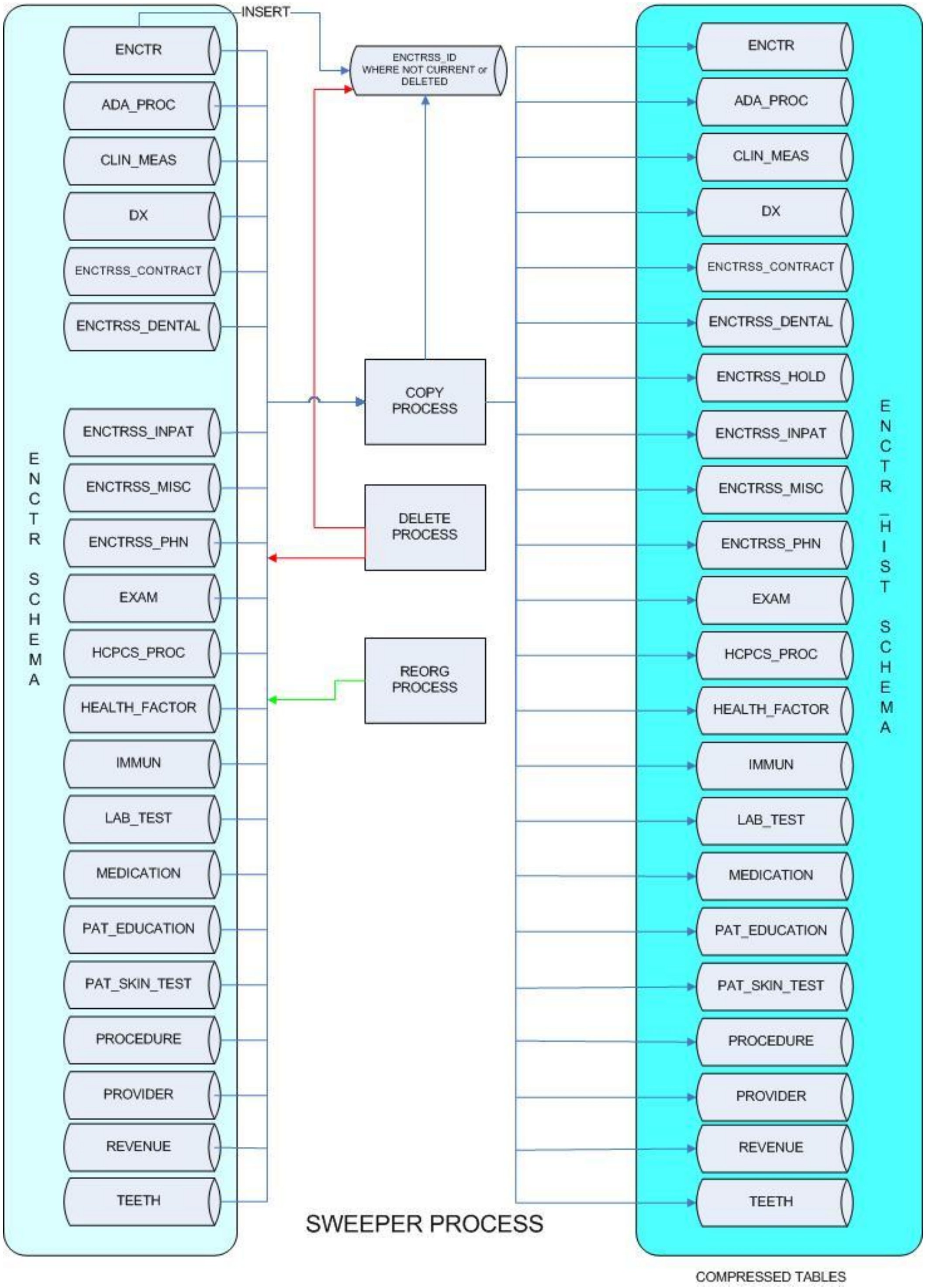

Figure 1 - *Sweeper Process*

Backup Process

All files within the NDW are backed up to the Tivoli Storage Management system (TSM). The TSM stores on tape (LTO-2/4) for storage within the IHS facility as well as offsite. Tivoli uses a ‘forever’ full incremental technology for AIX backup. DB2 (database) backups are run from the DB2 environment. These are either online or offline backups. In addition to these 2 types of backups, various other configurations including full, incremental and delta as well as packaged logs.

The specific type of backup is determined by the risk-mitigation analysis of each database backed up. For example, online backups may be run during production schedules, and offline backups run during non-production periods.

In addition, LAN and LAN-free backups are handled differently based on the specific configuration of the database. Transaction logs are backed up and moved to TSM periodically. DB2 transaction logs are currently managed manually, but will be auto-configured with the change to DB2V9.5. Transaction logging via LAN-free is currently suppressed.

Data Integrity Process

The Data Integrity (DI) process is a mandated process that helps ensure the data received is the data stored. The NDW does no cleaning or verification of data, but stores the data received. There are minor transformations that occur, and they are known. The DI process operates with the files received from the Integration Engine (IE) and compares the data contained in these file to the data stored in the NDW.

The process moves the data, untransformed, from the IE into ‘staging’ tables from which it may electronically be compared to the data in the NDW. The data movement and comparison is done largely by views. The DI process identifies the file type by the directory in which files are stored. External data, such as filename, must be appended to the data stream of the data loads.

Un-Duplication Process

The NDW stores all records received (duplicates or not). The same record can be received from one source multiple times, or the same record received from multiple sources.

The unduplication process is performed in two steps. The first step is an initial unduplication of records that have been sent more than once from the same source due to modifications by the site. The second step takes place after the encounter data Extract, Transform and Load (ETL) process and before report generation. An official unduplication is performed against the entire NDW database and encompasses all export file formats. Using a pre-approved set of business rules combined with the most recent Row Create Date/Timestamp, a Duplicate Flag is created to identify whether a record is a duplicate or not. Details on the business rules that drive the unduplication process can be found in the document entitled *NPIRS Basic Business Rules.*

Post-Load Reports

After the export data has been loaded into the NDW database, a Post-Load report, titled "NDW Post Data Load Report," is generated and emailed automatically to the designated person(s) at the sending site. The information in this report will match the ACK report information for the *same* data export file, with the addition of a Load Date, Load Status, Number of Encounters, Number of Registrations, and details on any Errors, if applicable.

MatchMaker Process

Since it is possible for encounter data to be received before registration data, it is important to match encounter data to registration data when the registration data is received. The purpose of the MatchMaker process is to correlate or link previously unlinked data in the ENCTR tables with data in the REG tables. Data is normally linked during the IMP process; however, there are times that ENCTR data is loaded before there is corresponding REG data. The MatchMaker process typically runs as part of the ETL process tree.

Error Tracking Reports

Error Tracking reports contain detailed information regarding errors encountered while loading the data from the files that are sent to the NDW. These reports are available on the NDW intranet web site at http://rohan.d1.na.ihs.gov, and include titles such as *Registrations Not Included on User Population Reports, Missing Registrations by Facility,* and *Registrations Potentially Countable on User Population Reports (*which includes specific information for each registration record found to be in error).

Test Environments

NPIRS Internal Test Environment

A testing environment which allows for complete testing of all changes to the NDW system prior to implementing those changes in the production system. This environment exists within both the ISLETA and LAGUNA databases.

NPIRS Data Mart Developers' Test Environment

A testing environment which allows NPIRS and others outside of NPIRS to develop and test their data mart and the and the extract, transform, and load (ETL) processes from a sample NDW database to their data mart, prior to their implementation in the production environment. It exists within the RTE66 database, and contains all of the target and reference tables in the National Data Warehouse (NDW). The NPIRS Data Mart Developer's Test Environment allows an authorized user to evaluate the type and structure of data, and to become familiar with the capabilities of the NDW, before accessing the General Data Mart or requesting a new data mart. It further allows a user to become familiar with appropriate and efficient methods of retrieving data from a relational database.