National Patient Information Reporting System: National Data Warehouse

NDW Production Database

Technical Guide

Version 2.0

February 2008

Department of Health and Human Services

Indian Health Service

Office of Information Technology (OIT)

Contents

Version Control

Version	Date	Notes
1.0	September 2007	FY07 Contract Deliverable (D1.12.3) Accepted October 15, 2007 Appendix C - NDW Column Detail is a separate document, in PDF format.
2.0 Review	February 2008	Review, update; add/expand: add DB2 V9, Indexes, MQTs, Tablespaces (including Large), Sequences, Nicknames, Compression sections; updated figures, added Topology, Current and Planned Server Configuration figures; Appendix A, B now separate documents; Appendix C now available at NDW Metadata web site, as noted in text. FY08 Bridge Contract D1.7.3 COTR approved April 10, 2008

Overview

The National Data Warehouse (NDW) environment is comprised of three main databases:

- Sandia
- Wildhrse
- Temecula

Sandia is the transactional database, the source for most data mart extracts. Wildhrse is a mirror (with a few exceptions) of Sandia, which allows query capability without affecting transactions, and also serves as the computational database for user population information. Temecula is the sole source of references tables for Sandia, Wildhrse, and all data marts through federation.

The NDW utilizes tables, views, sequences, and temporary tables to maintain data received from remote sites and areas in an orderly fashion for use in marts, queries, special requests, and data mining. The design of the NDW maximizes performance for transaction processing. Each logically grouped set of tables shares both a common tablespace (storage area) as well as a common buffer pool (memory area) to minimize disk I/O.

Data is sent to the NDW, typically via FTP, and is received by the Integration Engine (IE). The IE recognizes the file format and sends the file to the NDW directories for loading.

The Acknowledgement (ACK) process recognizes the file, logs the file into the NDW database and sends an Email to the sending site that the file has been received.

The Import Engine (IMP) process subsequently loads the files to the database, where the data is available for transmission and loading to the various marts.

Process descriptions included in this document refer to the Sandia database, unless noted.

Design Parameters

- The NDW environment contains all data needed for the data marts. The NDW can be modified to accommodate future business needs if necessary.
- The NDW environment exists on a separate server than all other environments, except the federated reference table environment. The environment encompasses Enterprise Information Integration (EII), Extract Transform Load (ETL), and Enterprise Application Integration (EAI) at various levels.
- All data within the NDW is updated continuously from files received from remote sites.
- Availability for transactions is maximized. The NDW is kept online 24/7, except for maintenance, as described later in the Import Engine (IMP) process.

System Environment

This is the physical environment of the NDW main database:

Server	FTP Address	AIX Level	Database	Partition Level	DB2 Level	Comments
Gollum	198.45.1.20	5.3+	Sandia	4	V9.1.x	Primary NDW database.
Gimli	198.45.1.9	5.2	Temecula	2	V8.2.8	Reference table database; used for federation to other databases.
			Wildhrse	2	V8.2.8	Calculation and research database; Userpop calculation and archive database. Future source of UP/WL mart but will be moved to Smeagol Server.
Bilbo	198.45.1.8	5.3	Hollywd	2	V9.1.x	General Data Mart DB; volatile structure based on User demands.
			Rte66	2	V9.1.x	Test or “sample” data mart used to determine structure and design of future data marts.
Smeagol	198.45.1.21	5.2	Tatonka	2	V8.2.8	User Population / Workload Data Mart; contains Workload archive data.

Server	FTP Address	AIX Level	Database	Partition Level	DB2 Level	Comments
Arwen	198.45.1.8	5.2	Isleta	2	V8.2.8	Primary development database. Small mirror image of NDW primary; can be refreshed or restored to baseline. Used for unit testing.
			Caesar	2	V8.2.8	Secondary development database. Small mirror image of NDW primary; can be refreshed or restored to baseline. Used for unit testing.
			Laguna	2	V8.2.8	Primary QA database. Small - medium mirror image of NDW primary; can be refreshed or restored to baseline. Used for integration testing prior to release to production

For a detailed list of NDW related schemas and tables, see the following documents:

- *NDW Schemas and Tables/Views/Nicknames*
- *Reference Tables*

See the *NDW Physical Models* document for corresponding models.

Detailed descriptions of data elements are available at the IHS Meta Data internet web site: http://www.ihs.gov/CIO/scb/metadata/

System Access

All databases are enterprise compliant to allow various environments to access the database, including ODBC, JDBC, OLE, and CLI.

Native access languages supported are SQL, SQLJ, SQL-Proc, and XML (under UTF-8) and direct.

Note: ANSI92 SQL processes may not be supported under V9 and are supported not under UTF-8.

Security

The NDW is restricted from outside access. Access is limited to NDW personnel for maintenance, data refreshing, and report production.

- Only authorized users are allowed access to the NDW.
- Security controls commensurate with those for a transactional, non query based database and adhering to IHS standards, as outlined in separate security documents, are enforced (Sandia). Security controls associated with Wildhrse are enforced as read-only and accessible by authorized users only.
- Temecula is updatable for tables associated with the SCB (Standard Code Book) only by users authorized to make such changes. Other reference tables are updatable only by authorized users. All changes in the Temecula database are tracked in a plain English audit table.
- Security controls associated with individual data marts are specified in their associated technical documents.
- All authorized users may access all data.
- Additional security will be added throughout the enterprise under V9, including LDAP and role based.

NDW Design

The NDW environment includes the following schemas.

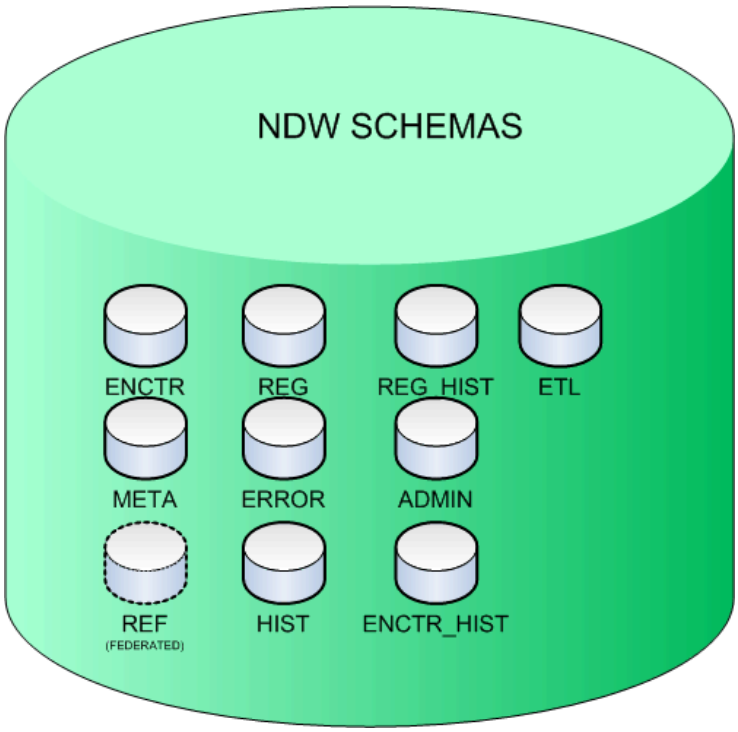

Figure 1. NDW Schemas

The NDW is a multi-partitioned, multi node environment designed to maximize performance. A node is logical and physical partition within the database that enhances performance by promoting CPU parallelism and opportunistic behavior and improves I/O throughput. The NDW is supported by a hybrid SQL/XML engine under UTF8 to support future enhancement and capabilities in anticipation of the growing capability of XML/XQuery.

DB2 V9

DB2 V9.x provides strong and comprehensive capabilities to enhance the current EII, ETL, and EAI capabilities of the NDW. A Meta data layer is currently being expanded to enhance the ETL layers, and a number of tools are being used to enhance the EAI paradigm. Under V9, multiple applications can be more tightly linked, using a single repository such as Workbench, and reduce the number of stovepipe applications. For more information, see the “Plans” section on page 22.

Structure

These are the main structure types associated with the NDW.

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 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 another 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 is utilized in the NDW. Column compression is not used for dates, timestamps, or characters under dimensional size of char(4), for obvious reasons.

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 to non homogenous databases (non DB2) and even to data sources, such as Microsoft's Excel© or SAS.

Additional information on federation available at this website:

http://www.redbooks.ibm.com/abstracts/sg244249.html

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.

NDW Process Flow

The following figure illustrates the NDW process flow.

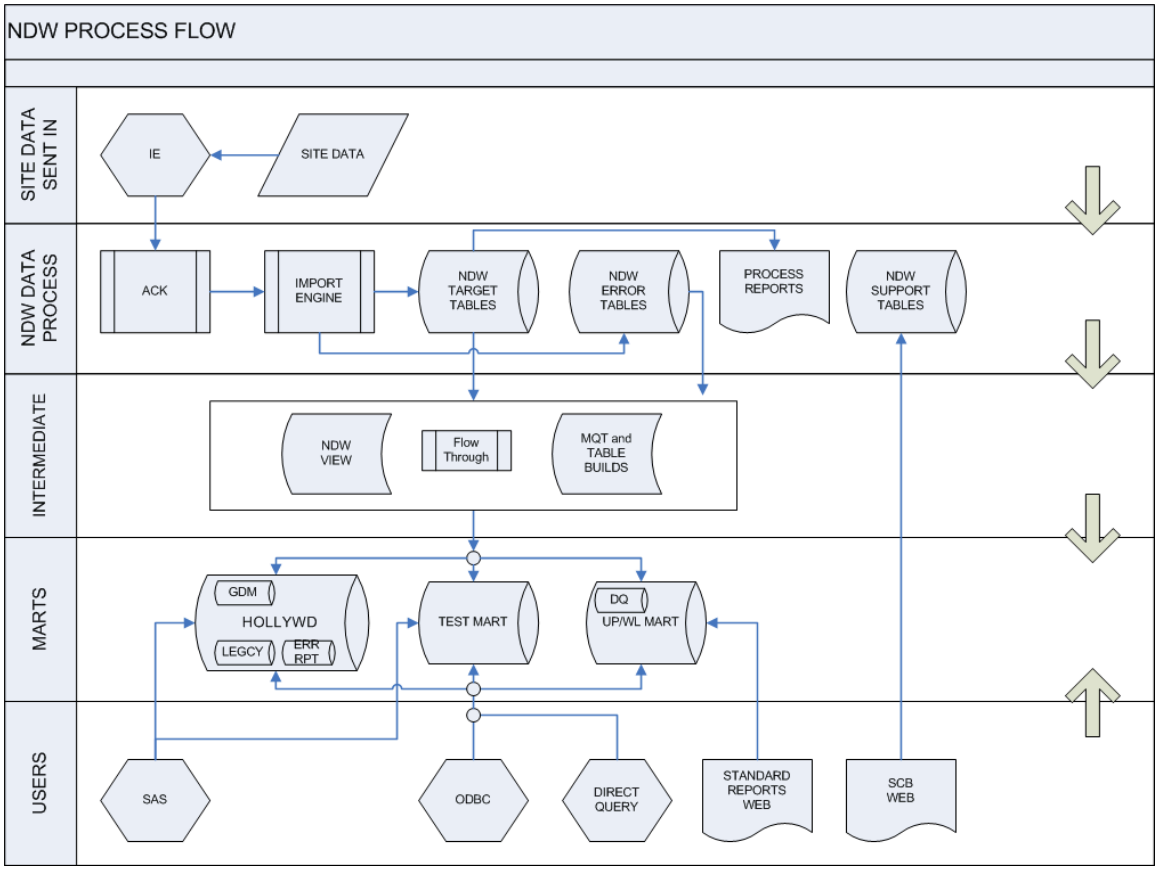

Figure 2. NDW Process Flow

NDW Processes

These are 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.
PROMOTER	Loads "orphan" Enctr records into the NDW after a valid REG record has been received.
ETL/TEL	Extracts information for use by various marts. Some extracts utilize views and MQTs. Some extracts are straight data extracts. Subsequent 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.
BACKUPS	Performed daily to enable a full database restore in the event of a disaster or other requirement.
SWEEPER	Removes Non-current and records marked for deletion into the ENCTR_HIST schema tables that are compressed to save storage.
Post Load Reports	Report the status of files sent to the NDW. These reports are emailed to the sending site.
Error Tracking Reports	Report errors encountered while loading the data from the files that are sent to the NDW. These reports are available on the intranet NDW web site.

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 are notational information associated with HL7 processing.

- 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 Public drive at this location:

\NPIRS\7. NPIRS Documents\21. NDW Implementation Documents\Documatron Implementation\Certified Documentation\ACK

IMP Process

IMP is an automated process that contains a number 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.
- Rejected encounter records are written to the ERROR schema tables.
- The ADMIN.EXPORT_INFO table is updated with file statistics as each file is processed.

The NDW is automatically updated by IMP 24/7 Sunday through Thursday and on demand Friday through Sunday. This allows a maintenance 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 Public drive at this location:

\NPIRS\7. NPIRS Documents\21. 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.

Marts directly supported through an ETL/TEL process include the Test Data Mart (RTE66), General Data Mart (HOLLYWD), Data Quality Mart (HOLLYWD), Error Reporting Mart (HOLLYWD), NDW Computation Mart (WILDHRSE), and the User Population/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, 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, 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.

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

Sweeper and Compression Processes

The NDW environment uses a Sweeper process (Figure 3), which runs automatically, to reduce storage requirements for both less used data and duplicate data. As the IMP process enters data in 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 can not be utilized.

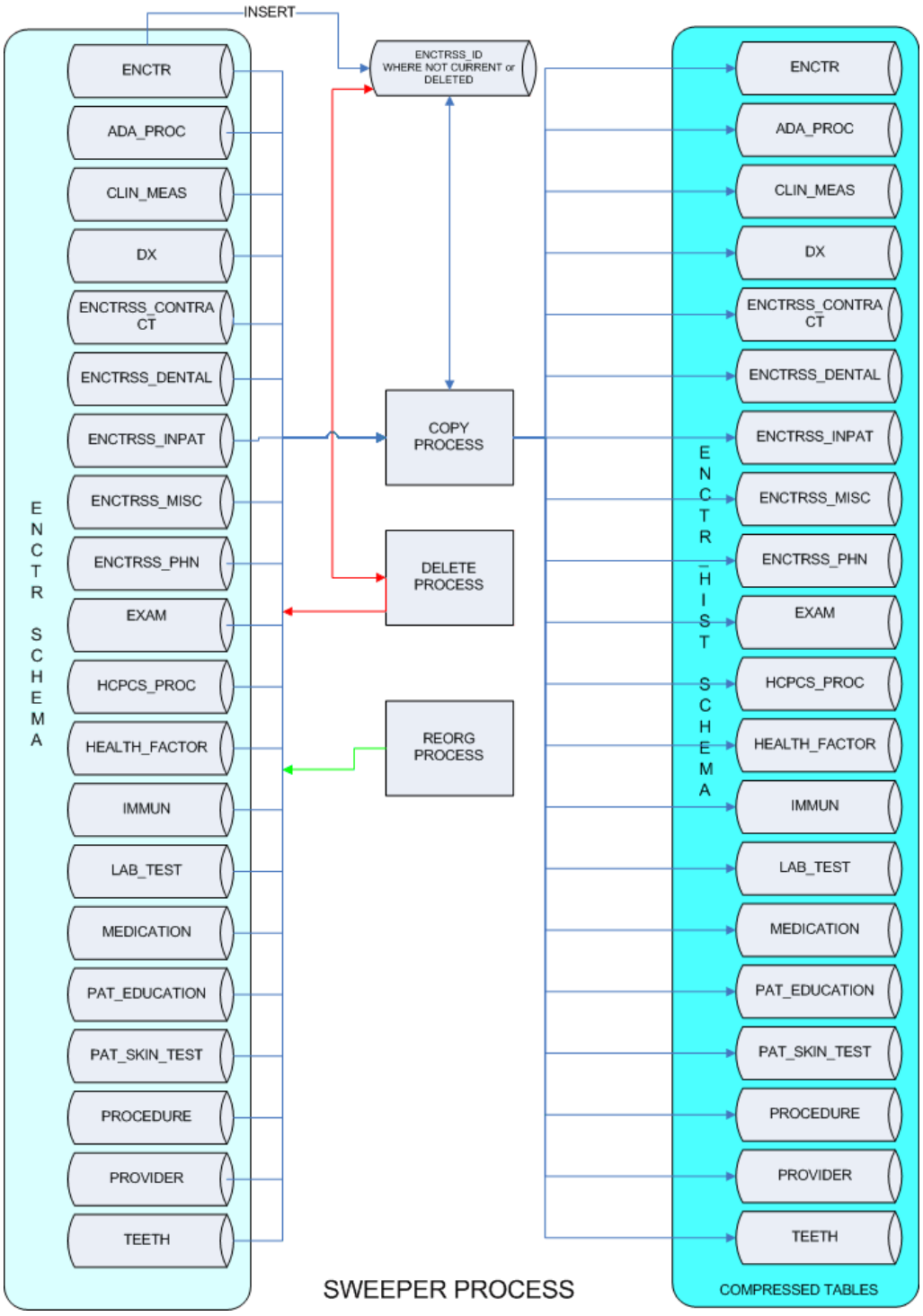

Figure 3. Sweeper Process

Backups

Backups of the NDW database are performed daily. Several types of backups are available: full offline, full online, incremental, delta, Mobility on Demand (MOD), and tablespace. The type of backup is parameter driven in the backup script, depending on the day of the week.

NDW has multiple nodes, each backed up individually. In the event of a restore, backup of each node is required.

NDW is identified as a mission critical database in the Emergency Management Plan (EMP). A copy of the latest version of the *Emergency Management Plan (EMP) for the National Patient Information Reporting System (NPIRS)* document is available on the Public drive at this location:

P:\NPIRS\11. Security\Emergency Management Plan\Current\

NDW Data Flow

The NDW receives and processes data from multiple sources. Although the data load process into the NDW is designed for RPMS data, other data formats are also processed, and all data loads are based on the same process.

Facilities transmit data export files to the NDW. On arrival, the data export file goes first to the IE, where it is reformatted and output as a post IE data export file. After the post-IE data export moves the file to the incoming NDW directory, the ACK process assigns an internally generated unique export_id to the data export and then moves the file to the appropriate directory. The data export is then processed and loaded into the NDW (SANDIA database).

Once loaded, the data is copied to WILDHRSE, which is synchronized with SANDIA on a periodic basis and is designed to allow maximum throughput transaction processing for the NDW environment. WILDHRSE is used for special data requests, data investigation, data archive, and user population calculations. For more information, see the *User Population/Workload Data Mart Technical Guide* (Version 1.2).

Current plans will move the Tatonka calculations and data mart to be self contained on WILDHRSE. The WILDHRSE physical location will also be moved to the Smeagol server. The purposes of these two moves is to continue to maximize the transactional functions of Sandia with little or no impact to data loads, to minimize ETL and calculation time and impact for Userpop/Workload processes, and to improve data research capability that is co-located with the reporting data.

The following figure illustrates the general data flow through the NDW environment.

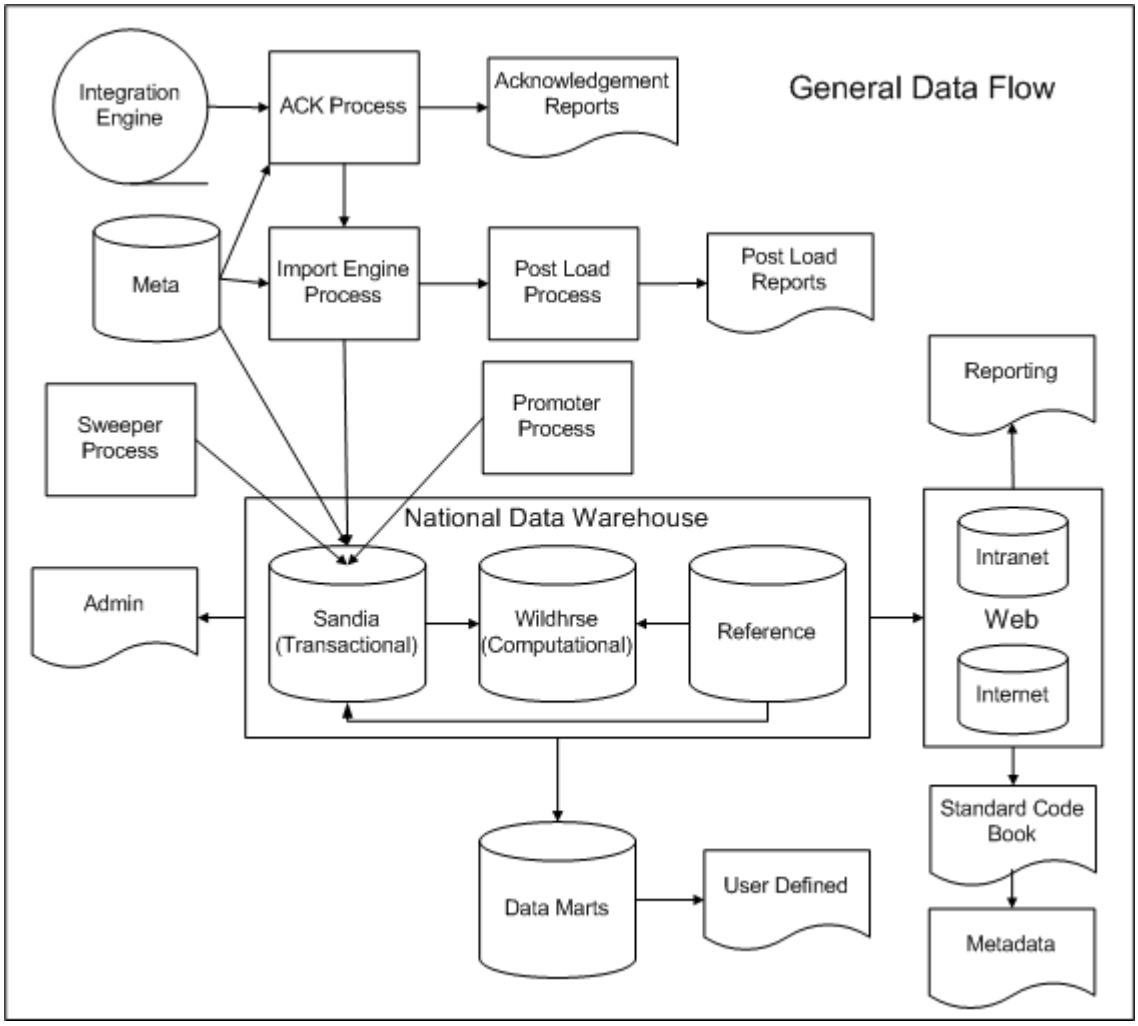

Figure 4. NDW Data Flow

Related Environments

Overall System Environment

The following figure illustrates the role of the NDW environment within the overall system environment.

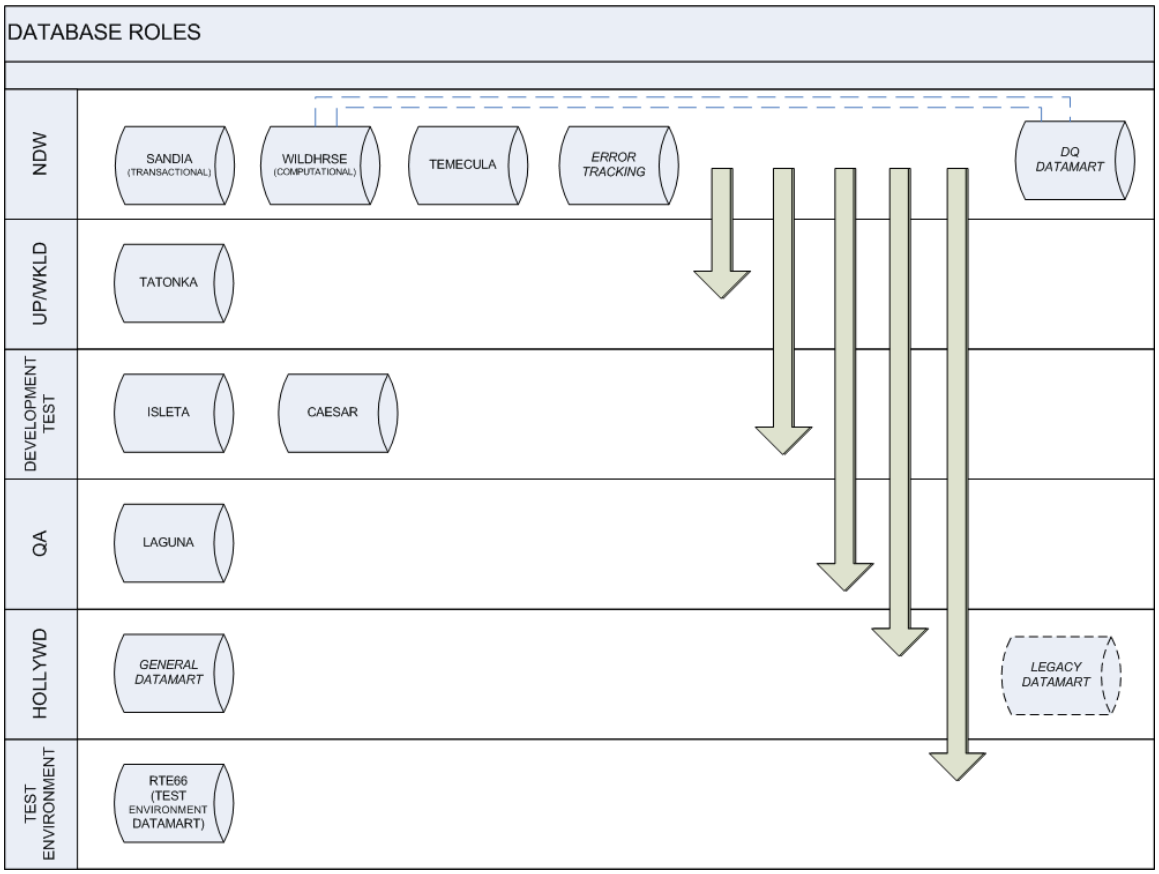

Figure 5. Database Roles

Note that the Legacy Data Mart was imported from the Legacy NPIRS database and contains pre-2006 data. For more information, see the *Legacy Data Mart Getting Started Guide* (Version 1.0).

The following diagram illustrates the interrelationship of the NDW, related Marts and associated hardware (current server diagram)

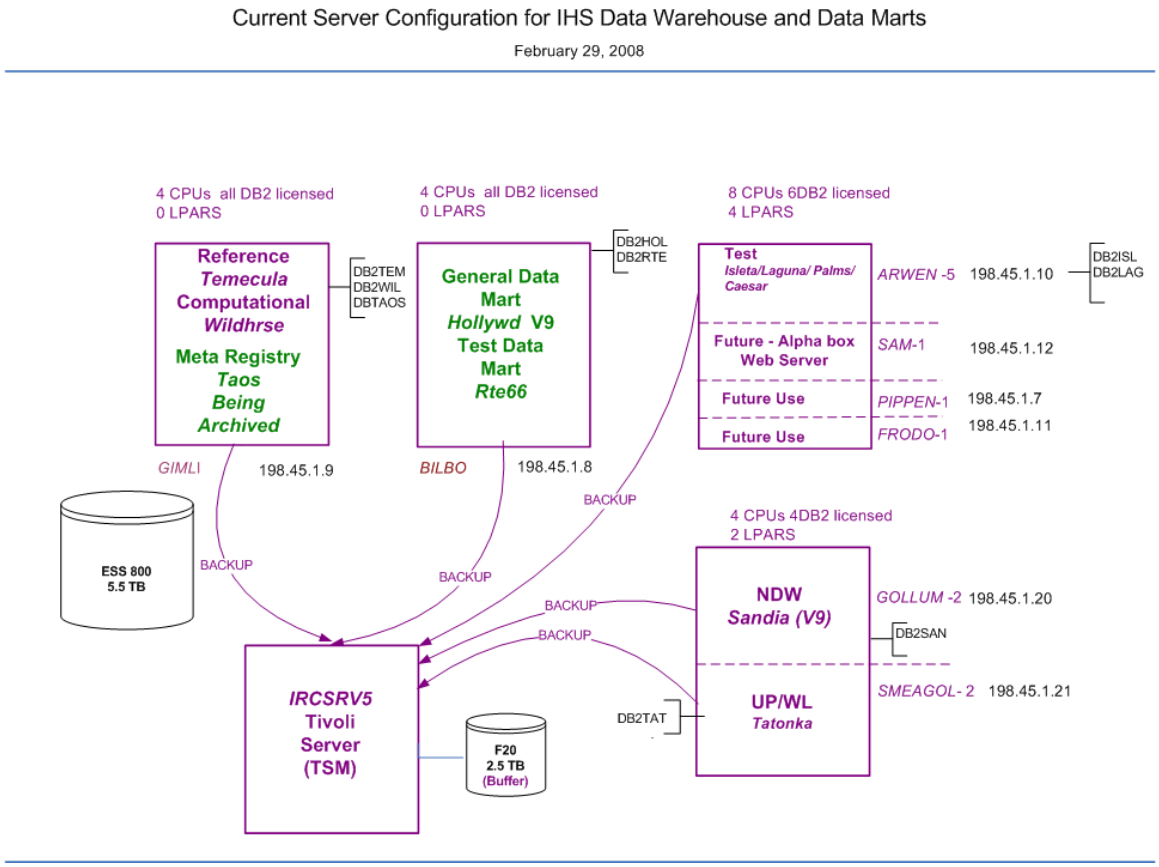

Figure 6. Current IHS Data Warehouse/Data Marts Server Configuration

The following diagram illustrates the high level topography of the NDW and related Marts.

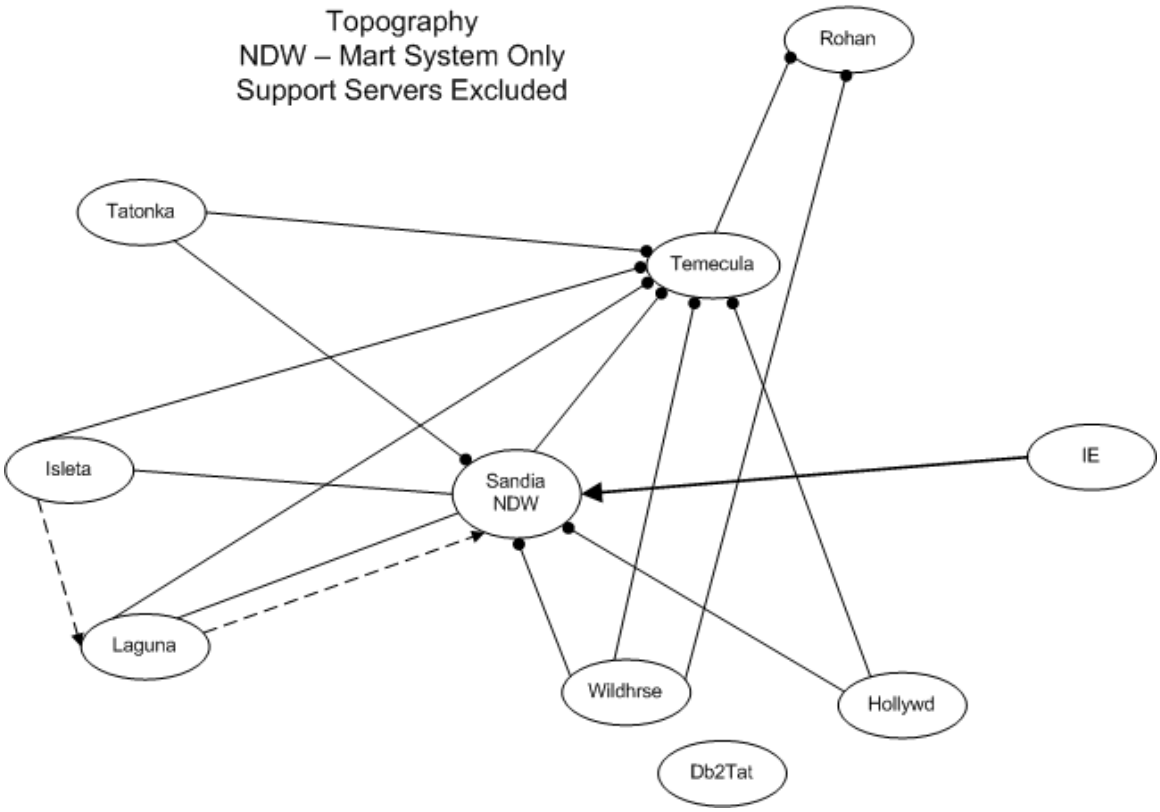

Figure 7. NDW/Data Mart System Topography

Test and Quality Assurance Environments

Two additional database environments, Isleta/Caesar and Laguna, support changes to the NDW database or associated applications. These are used for development (ISLETA and CAESAR database) and QA (LAGUNA database). Both environments reside on a separate server (ARWEN) from the NDW environment and are occasionally refreshed from NDW or from a predetermined baseline on an as needed basis.

These environments are typically kept in synchronization with NDW in regards to structure, except when testing structure changes, as illustrated in the following figure.

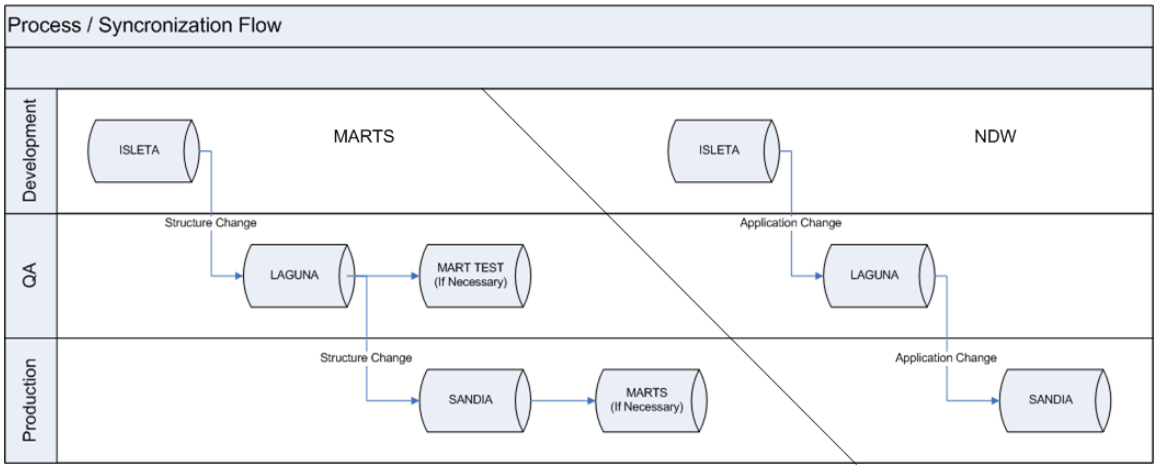

Figure 8. Process/Synchronization Flow

All changes, including application changes, are tested first in the development (Isleta) database environment and then in the QA (Laguna) database environment prior to production release and movement (promotion) to production in NDW.

Plans

At the time of the release of this document several changes are under way. Among these are:

1) Re-tasking of servers and movement of databases. The following diagram shows the proposed plan, which will maximize the Government's investment in current storage and servers while improving risk mitigation.

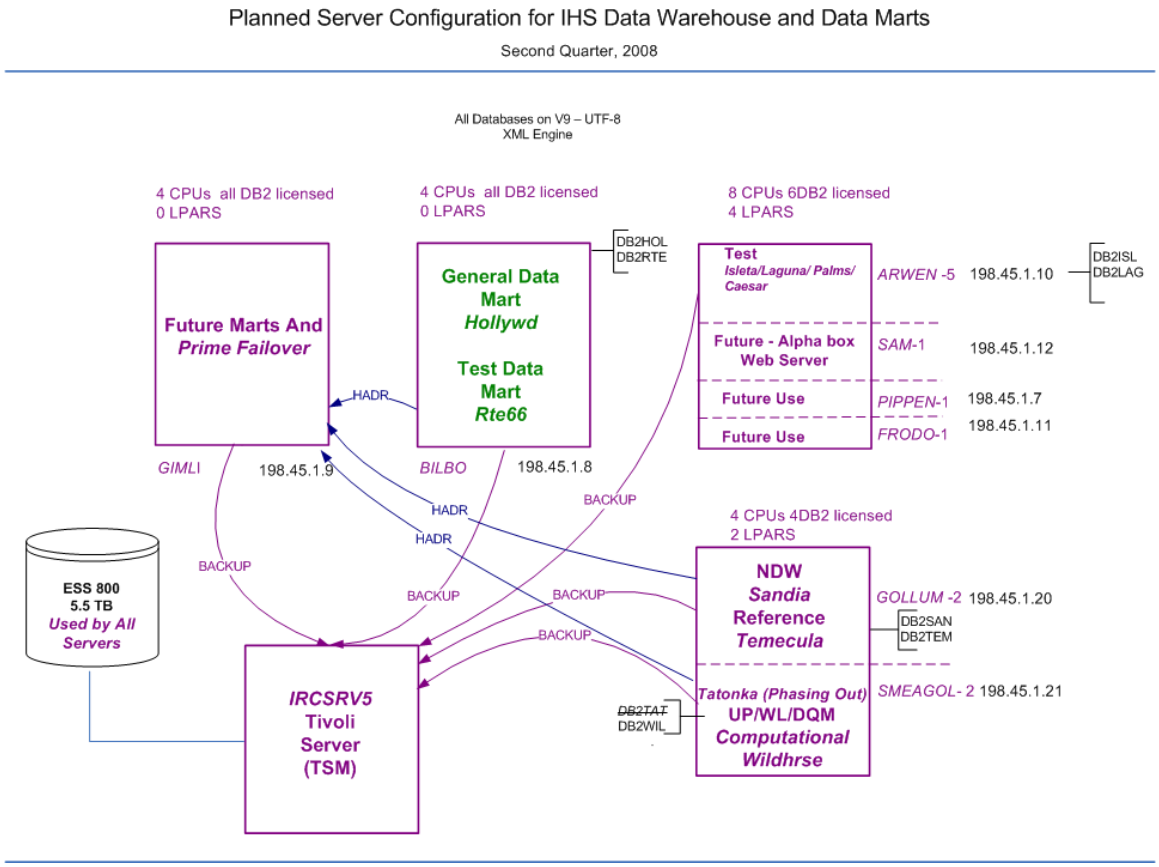

Figure 9. Proposed IHS Data Warehouse/Data Marts Server Configuration

2) More tightly managed applications using both Rational and Workbench. This will improve the EAI index and the manageability and flexibility of the system to satisfy the Government's current and future needs.

3) Under DB2 V9, a robust Meta Data layer that is largely automated will manage more of the ETL processes, eliminating some scripting and also improving the flexibility of the NDW.

4) There will be a complete change in the Change Management paradigm to a top-down model for future paradigm changes.

   Current process:

   a) Changes in applications or DB structures are defined.

   b) Requirements are derived.

   c) Changes are implemented.

   d) Documentation is updated.

   e) Models are updated.

   Future process:

   a) Requirements are derived.

   b) Models are tested and evaluated against business processes and rules.

   c) Changes in applications or DB structures are defined.

   d) Documentation is updated.

   e) Models are updated.

   f) Changes are implemented.

5) All DB2 V8.x databases will be upgraded to V9.x contemporaneously with an upgrade to AIX 5.3+.

6) All operating systems will be upgraded to AIX6

   The AIX6 upgrade will further improve performance. This will have minimal performance improvements on the P650 servers, but should provide a 5-15% performance improvement on the P570 server.