

John Finianos, JF Information Consultancy S.a.r.l, Beirut, Lebanon  
Jugdish Mistry, Professional Solution Providers Ltd, London, UK

## ABSTRACT

Ok! A full fledged Data Warehouse has been implemented at your organization for a while now by using the right tools to model, map and load the DW. Supposedly, the warehouse should have covered a major part of the business. Data has been used extensively to produce all kind of reports and run various types of applications including customer profitability and Product scoring. Everybody in the organization is very happy. Everything was going right until:

- 1) users start asking for more and more information,
- 2) new products are introduced in the business and should be covered by the DW,
- 3) major fixes are needed to rectify some of the implemented business rules and tie up some inconsistency in field naming conventions.

In all cases, a major upgrade to the DW has to take place on time, with minimal effect on the present usability of the DW.

This paper discusses a practical approach to upgrade an existing DW starting from logical modeling, to mapping changes and implementation using a the SAS DW administrator, to generation of SAS load processes, to entity structure changes, and to data loading. This approach is used at the National Bank of Kuwait.

Products used are SAS/BASE, SAS/WH Administrator, SAS/FSP, SAS/CONNECT SAS/ACCESS to IMS, SAS/ACCESS to DB2 on MS Windows and IBM OS/390.

## INTRODUCTION

In versioning a Data Warehouse (DW), the main challenge is how to do it in a timely manor with minimum disruption to the Data Warehouse users and applications. It is a tricky process that requires a careful and realistic planning and a fast implementation strategy

The Data Warehouse has been at its present version for a period of two years. It has almost 3 years of data loaded within it. In order to introduce new products and improve the overall quality of the data, it is being upgraded from version 7 to version 8.

Initial plans were to implement a full update to next version. This means an all out effort that is not trivial and could potentially take a long time, specially if approached as a one off convert and implement exercise.

With deadline rapidly approaching, an alternative approach was undertaken. Basically a new version of the warehouse was required to be fully built and operational within a time span of 4-5 months.

To meet the business deadlines imposed it was decided to scope down the currently intended implementation; to a version called V7.8. The approach taken was to slightly change the new model and undo modifications made to the model. Rendering it more normalized when going from version 7 to 8. Attributes that had moved to the super entity were moved back the subclass entity but also left in the super entity. Hence we were keeping the model as close to Version 8 as we could; as well as keeping the data in version 7 form. Any attributes moved would not initially be populated in version 7.8.

The implications of this were that existing mapping rules would not require changing. New mapping rules would be introduced for new entities and essential changes would also be implemented for the newly defined model/version.

The actual implementation steps were then identified and grouped into three main categories:

- Business rules related - to include the modification of existing source to target field maps and the addition of any new ones.
- Data Warehouse related - to include the removal and/or renaming of existing entities and fields and the addition of new entities and their fields.
- Code and application related - to include any SAS program and application interfacing with the physical entities and fields of the data warehouse.

Implementing the business rule changes is a core step in the implementation process. The first step is to reflect the changes in the logical and physical model. A DDL file is then generated and used as a building stone for the rest of the implementation steps. The next step is to communicate all the business rule changes via the source to target field maps. So, it is important that these maps are written in a very simple, clear, and consistent manner for the implementer to understand.

The DW related changes were implemented after processing the newly generated DDL, base lining the existing DW, and running comparison reports between the existing DW and the new one. The identified changes were then implemented in steps in both the SAS/DWA PC and the mainframe environments. These steps include:

- The copying of the DW Administrator environment to an environment parallel to the existing one,
- The modification of the metadata in the new environment to reflect the identified changes. API programs were used to carry on repeated and tedious changes.
- The copying of the mainframe Data Warehouse environment to an environment parallel to the existing one. This environment includes both code PDSs and Data Warehouse physical entities.

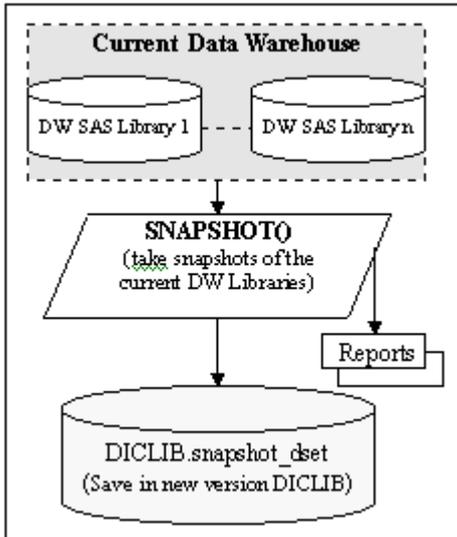
The code and application related changes were implemented on the various user exits, stand alone programs, post process applications and other code affected by the changes implemented in the previous two steps.

The actual method of implementation is discussed in detail via the upcoming sections. Topics such as versioning, baselining the data warehouse, Warehouse Administrator changes, designing the development and test environments, the best approach for data migration, and testing the implementation as well as rollout will be discussed.

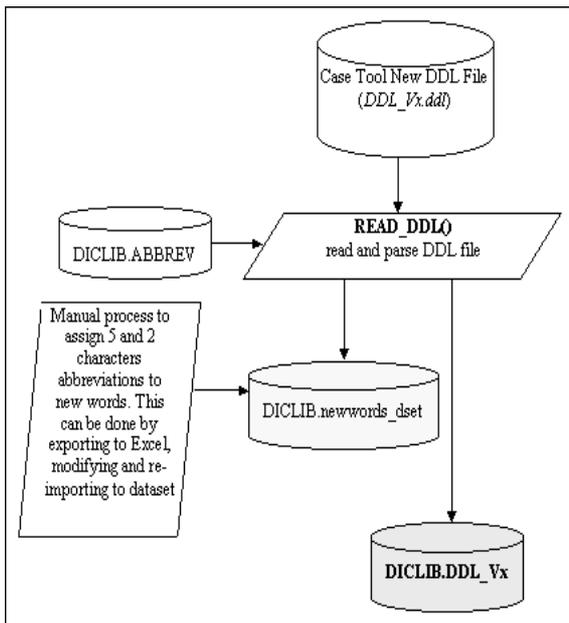
## BASELINING THE DATA WAREHOUSE

The base line process is used to compare the existing DW with the proposed one using the DDL file as the building block.

The first step in the process is to take a snapshot of the existing of the existing DW and save it in a dataset (see next Flowchart). Macro %SNAPSHOT() is used for this purpose.

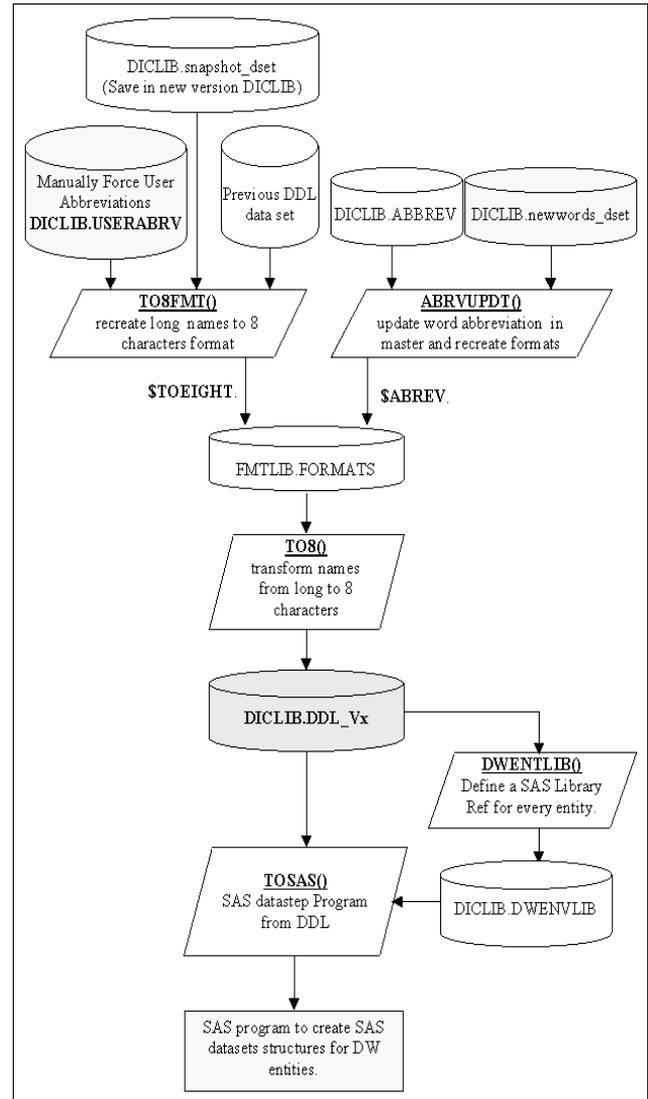


The next step is to process the generated DDL for the new DW and generate a SAS/Base program to build the new DW entity structures (see next Flowchart). Macro %READ\_DDL() is used to read the DDL, parse it and save it in a SAS dataset. It also identifies any new word used in the naming convention and save in a SAS data set to be manually assigned an abbreviation for the long to eight characters conversion algorithm.

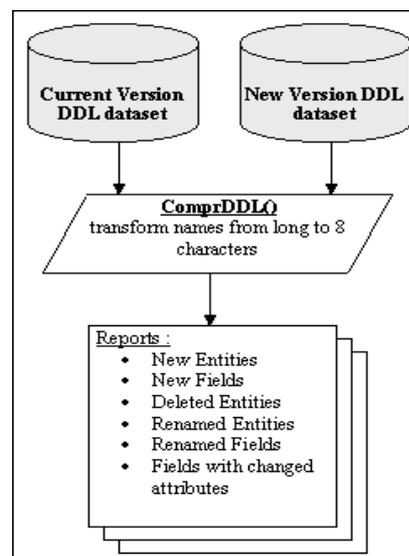


This conversion algorithm checks if an assigned naming convention already used in the previous DDL. If it does, the already assigned short name is used. If it does not, it checks the manually forced user abbreviation dataset for a possible entry. If entry exists, it uses the user assigned short naming convention. Otherwise, it uses the word abbreviation formats and assign a new short naming convention. The short name is then saved back in the new DDL dataset (see next Flowchart).

The final step in the process is to convert the parsed and converted DDL to a SAS program used to build the structures of the new version for the DW entities (see next flowchart). But first every entity must be assigned to a SAS library and this is done manually in the DW environment SAS library dataset (DWENVLIB).



After the new version DDL dataset is generated, it is compared to the previous DDL dataset using macro %ComprDDL() to produce various type of reports highlighting the exact differences between the two DDLs (see next Flowchart).



## SAS/WAREHOUSE ADMINISTRATOR METADATA

A copy of the current DW administrator environment, including the metadata, is created into another root directory. This is to preserve the previous version and continue working with it until the upgraded version is fully functional. For a short period and until all required upgrades are done, the two environments are kept in sync by implementing any new changes in parallel.

Metadata updates are done based on the results of the %ComprDDL() macro. This macro was run as part of the baseline process to produce a set of reports about the nature of changes entity by entity. For trivial changes, API programs were written to delete and rename existing metadata. All other changes, including mapping and process exit changes, were done manually. Load processes were then regenerated, tested thoroughly, and uploaded to the mainframe migration test environment.

## DEVELOPMENT AND TESTING ENVIRONMENTS

Two environments are maintained for V7 of the warehouse production and test. It was decided that for V7.8 to follow the same architecture. The two environments would be known as Q and I. The Q environment would be a full production environment for V7.8 and I the test/development environment for V7.8.

The production (Q) environment would contain the new definitions for V7.8 and be populated with migrated data. All code that exists in the V7, P environment migrated and moved to the Q environment with the same naming conventions as P.

The rationale for the two environments was to firstly create a V7.8 warehouse in Q and then prove that existing V7 code could be used to load in parallel with V7. This would provide a base from which V7.8 development could take place. It also provided a good feeling and moral booster as it would be the first mile stone in the implementation of V7.8.

Having a Q and I environments provided the luxury of being able to fully test all development changes made for V7.8 in an environment that still protected "production" data moved to the Q environment.

Additionally when V7.8 is fully implemented the Q and I environments could replace P and T environments respectively allowing V7.8 to replace V7. The transition for users would be transparent.

Only one month's worth of data (September 1999) was moved into the Q environment. It was not necessary to move all historical data at present as the Q environment was still being developed.

Moving only one month meant that the P and Q environments could be loaded simultaneously, or shortly after, that of V7 DW load. Hence, subsequent month loads can be compared to ensure no errors had arisen with the move and confirm the code converted to load V7.8 is correct.

Additionally the DASD requirements for the implementation could be optimized and kept to a minimum.

Once the Q and I environments had been defined and data/code moved, one had to be careful to ensure that any changes made to code in the P environment were propagated to Q, via I. To minimize the workload, changes to the P environment were reduced to those that had a high priority and were very essential.

## TESTING THE IMPLEMENTATION

Testing has been prevalent throughout the implementation of V7.8. Where possible, tests were written in such a way that they were repeatable and highlighted any errors found via the form of a report.

A test plan was developed in the very early stages of implementation. As development time was very short it was important to make sure problems were discovered and rectified as soon as possible.

The test plan incorporated the following:

- 1) Every step of the way, tests were put in place that were repeatable and provided simple reports on the testing.
- 2) Checking and comparing all data moved from P to Q. PROC COMPARE was used extensively to ensure that all data sets and variables matched exactly.
- 3) PROC COMPARE used with each month's load.
- 4) Record counts were compared with each month's load.
- 5) Statistics and frequency counts were produced and compared with each load to ensure that discrepancies were identified as soon as possible.
- 6) Records of all tests run were kept up to date and strictly monitored.

All new mapping rules were tested thoroughly by proofing the mapping rules before implementation. Changes were made during the documentation phase for writing mapping rules. The forms were revisited and standardized and improved allowing for easier implementation into the SAS/DWA.

More tests were introduced at loading of the operational data (ODD). This means that anomalies in the data sources were detected before the data was loaded into the DW.

Business users were involved at the definition and implementation of mapping rules.

As the Q environment was to grow monthly similarly to the P environment testing was much simpler as there was a fixed base to compare against.

## ROLLOUT

The rollout plan was as follows

- 1) Complete testing V7.8 by December 1999 load (i.e. after load in Jan 2000)
- 2) Begin loading 2000 data into V7.8 (i.e. by February 2000)
- 3) Get end user signoff on new entities and data.
- 4) Educate users on new data and functionality provided in V7.8
- 5) Allow users sufficient time to change any programs that might require changing.
- 6) Move all historical data into V7.8
- 7) Move V7.8 into production.

The plan was to begin loading V7.8 with the start of the millennium and move into production shortly after. V7.8 had been implemented and had been fully tested by December 1999. However, it was decided to load V7.8 in parallel with V7 for a further couple of months so that the user community could migrate their code to use the new warehouse and also become familiar with the new sources of data.

A presentation was made to users as it was important to keep them informed of all changes and amendments to V7.8 and gain their confidence and approval of the new warehouse.

The code used to migrate one month from V7 to V7.8 was used again to move the remaining historical months. The same tests as before performed to prove all data was moved completely and correctly.

## CONCLUSION

Given tight deadlines it is important to scale down an implementation such that it becomes an achievable target. This paper outlines an

approach that has been proven to work. It is not a simple task and does require substantial planning and teamwork to succeed.

## REFERENCES

SAS is a registered trademark or trademarks of SAS Institute Inc, in the USA and other countries.

Other brand and product names are registered trademarks or trademarks of their respective companies.

## ACKNOWLEDGMENTS

The authors wish to thank the entire National Bank of Kuwait DW team for their professionalism, dedication to the project, and their assistance in the process of writing this paper.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

John Finianos  
JF Information Consultancy Sarl  
St. Joseph Center  
Building B, Suite B10  
Awkar, Beirut, Lebanon  
Tel: +961 (0) 3 713430  
Fax +961 (0) 4 542248  
Email: [JFIC@cyberia.net.lb](mailto:JFIC@cyberia.net.lb)

Jugdish Mistry  
Professional Solution Providers Ltd  
62 Stanhope Rd  
Cippenham, Slough  
Berkshire SL1 6JS, UK  
Tel: +44 (0) 1628 603 428  
Fax: +44 (0) 1628 541 561  
Email: [jugdish@pspltd.demon.co.uk](mailto:jugdish@pspltd.demon.co.uk)