

DH ETL Technical Overview

A review of the core technologies and functionality available in DH ETL™

Overview

The data extraction, transformation, and loading (ETL) process usually represents 60% to 80% of the time and costs involved in developing a business intelligence application.

DH ETL dramatically reduces ETL costs by creating data warehouses without programming or code generation.

DH ETL has a metadata driven, visual interface, that eliminates the need for programming scripts and stored procedures when building snowflake and star schemas, as well as denormalized relational models for reporting. The product encapsulates the core Microsoft SQL Server® technologies, such as DTS, to produce SQL Server data warehouses, without the need for programming or programming knowledge, reducing development time by up to 80 – 90%.

At the core of the DH ETL technology is the metadata engine, which manipulates objects such as tables, fields, sorts, relationships, filters, summaries and formulas. This approach creates reusable objects, which further reduce development cycles. The formula engine is able to create complex key performance indicators, and automatically manages formula dependencies.

DH ETL offers two main benefits:

- the reduction of development and maintenance costs
- the enablement of less technical people to build data warehouses

Designed for the needs and budgets of mid-size companies, the product sets new standards for ease of use and price.

Table of Contents

Overview	1
Encapsulating Microsoft Data Transformation Services (DTS)	2
From Operational Data to Reporting Platform to Cubes.....	2
Predefined Schemas for Popular Applications	2
The Data Dictionary and the Metadata Engine.....	2
Reusability - Automatic Propagation of Changes in Business Rules	2
Data Transformation and Metrics - The Formula Engine and Formula Dependency Management	2
The Relationship Engine.....	3
Defining New Tables in the Data Warehouse.....	3
Importing Data	3
Table and Field Properties.....	4
The Learning Curve	4

Encapsulating Microsoft Data Transformation Services (DTS)

DTS is based on an OLE DB architecture that allows you to copy and transform data from a variety of data sources. DTS tasks include¹:

- Importing, exporting, and transforming data
- Copying database objects
- Sending and receiving messages to and from other users and packages
- Executing a set of Transact-SQL statements or Microsoft ActiveX® scripts against a data source

DH ETL encapsulates DTS tasks, hiding from the user the complexities of programming DTS packages.

From Operational Data to Reporting Platform to Cubes

DH ETL is a set of integrated tools that map source data, select data to be imported to the warehouse, and create optimized data models and metrics for reporting and OLAP tools.

Predefined Schemas for Popular Applications

DataHabitat offers out-of-the-box data warehouse schemas for dozens of applications including:

- Great Plains Dynamics, Great Plains eEnterprise
- Best Software MAS 90, MAS 200, MAS 500, and Platinum for Windows
- SalesLogix
- ACCPAC Advantage Series and Pro Series
- Visual AccountMate LAN, SQL, and iSeries

The Data Dictionary and the Metadata Engine

Metadata is stored in a dictionary that supports objects such as tables, fields, sorts, relationships, summaries, filters and formulas. Objects are visually developed and their definitions are stored as metadata. This approach, in addition to providing an advanced level of reusability, enables superior documentation and much faster maintenance cycles. Application logic does not reside in lines of code, but is stored as reusable objects.

Reusability - Automatic Propagation of Changes in Business Rules

When application logic resides in lines of code inside a DTS package, it is impossible to reuse it. As an example, let's suppose we need to create the formula ***net profit = revenues minus cost of sales***. This formula is used in several DTS Packages to calculate Net Profit by Customer, Net Profit by Product Line, Net Profit by Territory, and so on. Now the formula has changed to ***net profit = revenues minus cost of sales minus taxes***. The DTS developer will have to edit all the DTS Packages that include this calculation. With DH ETL, the ***net profit*** formula is a single object that can be used in as many situations as desired. Changes need to be made in one place only.

Data Transformation and Metrics - The Formula Engine and Formula Dependency Management

¹ Source: www.microsoft.com

Formulas are used to create new key fields for data modeling, perform data transformations, and to create business metrics (according to business rules). Formulas have 3 components: Formula Steps, Formula Functions and Formula Conditions.

To create a star schema, we simply copy the key fields from the dimension tables to the fact tables. As an example, an application may require a direct relationship between the Customer Table and the Invoice Detail Table, which is not supported by the operational database.

Formula Conditions replace “if-then-else” and “case” logic commonly found in data transformation tasks and calculation of metrics. Formula Functions include useful algorithms such as Ranking.

Metrics can be created using fields from any table in the data model, as well as reusable filters, summaries and relationships.

Reusable filters (state = “CA” and invoice amount > 1,000) provide an easy way to extract data from fact tables. When defining a formula, a filter can be associated to each formula step. So, a single formula can use several relationships to different tables, several fields from each table, and several filters.

The Formula Engine understands the formula dependencies and drives DTS to recalculate them in the proper sequence. This may be extremely helpful if your data warehouse has thousands of formulas to perform transformations and build metrics.

The Relationship Engine

Highly complex data transformation tasks and metrics can be developed without programming because of the powerful relationship engine. All native relationships among tables are automatically managed.

New relationships can be created to support data modeling and integration. The formula engine uses relationships to build complex metrics involving fields from several tables.

As an example, a key performance indicator can be built into the Customer table using fields from the Accounts Receivable Invoice, Sales Order Invoice, and Opportunity (CRM) table.

Defining New Tables in the Data Warehouse

Many times it is necessary to create new tables in the warehouse to support entities not designed in the operational database. The operational database may not have the Territory table, for example. A new table can be created to formalize the Territory entity, and new fields and metrics can be built into it.

Importing Data

The Import Wizard provides access to any ODBC or OLE DB compatible data sources, and includes the following functions:

- Select database tables and fields to be imported
- Create aggregate tables and summaries
- Select filters
- Import the same table multiple times, with various filters, groupings and/or different selection of fields
- Select data refresh options
 - Disable automatic refresh
 - Define periodic refresh

- Replace
- Append based on control attributes
- Perform union of two or more tables

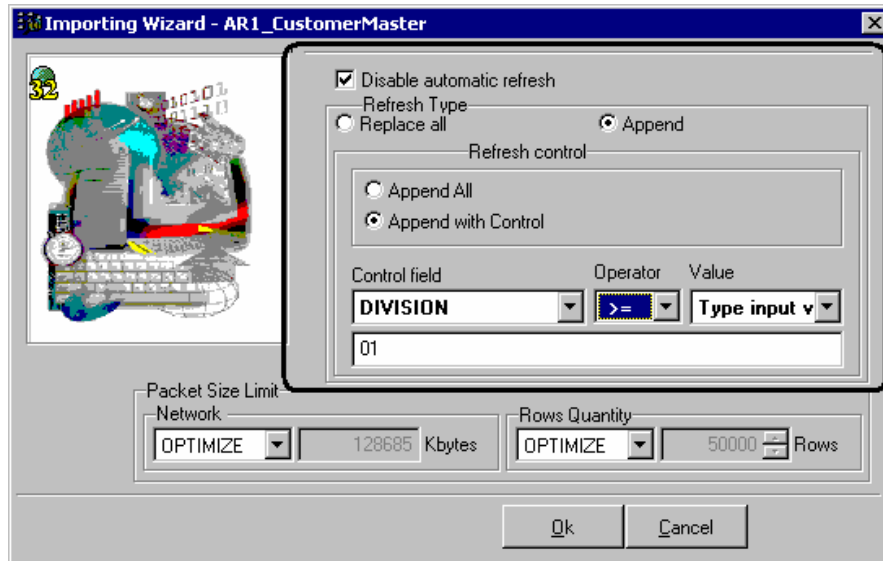


Figure 1. Import Wizard enables different types of data refresh

Table and Field Properties

DH ETL allows the user to modify table and field properties in the data warehouse. The user can change field type, format, length, and decimal positions. In addition, Table and Field Models can be created and used to modify source field properties. The metadata repository, integrated to SQL Server Metadata Services, allows the user to provide long, descriptive table and field names.

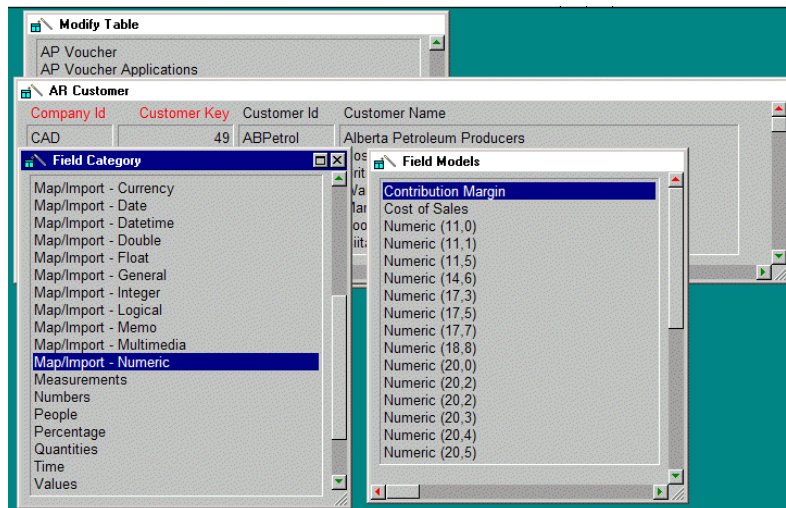


Figure 2. Field categories and models allow reuse of definitions

The Learning Curve

The DH ETL training course takes 8 hours, and covers the following topics:

- Importing Data
- Creating Data Models to support reporting and OLAP
- Creating Metrics (Key Performance Indicators)
- Security Manager
- Scheduling Data and Formula Refresh
- Scheduling Data Exports
- Optimization Tips & Tricks