



Introduction to EDW

(Edw is a **D**ata **W**arehouse)

1 What is EDW

EDW (recursive acronym for "Edw is a Data Warehouse") is an array of tools for building information-oriented applications, with an emphasis on Data Warehouse (DW), Extract/Transform/Load (ETL) and Business Intelligence (BI) features. EDW's goal is to allow developers to build these kinds of applications, while giving them room for customization even if they don't develop in Delphi. Thus, even the final user can customize the application, as required.

2 Features

Applications built with EDW offer features that can be grouped into the following functional categories:

- ◆ **ETL:** extracting, consolidating, cleaning, converting, importing data from a number of heterogeneous sources into the database.
- ◆ **Data browsing:** exploring the tables of the DW (or any database) with advanced features such as customizable searches/filters, lookups and master/detail structures), all based on a model that can be created automatically and quickly from the database schema.
- ◆ **Data editing:** inserting, modifying and deleting data, with customizable (possibly complex) business rules to enforce data integrity.
- ◆ **Reporting:** EDW integrates several different reporting engines (including but not limited to ReportBuilder, OpenOffice.org and XML/XSL:FO technologies) to create simple or complex reports that can be exported in many different file formats and/or sent/published in different ways.
- ◆ **Multi-dimensional analysis:** creating interactive analysis "cubes" with advanced features such as slicing, drill-down, rotation, projection, analysis, charting, ability to extract the data in different formats and produce reports.

The extremely modular nature of EDW allows the developer to compose and tune the different features in a fine way, according to the needs of the particular application.

EDW works with different database engines. The applications we have created with EDW so far work with Firebird and Microsoft SQL Server (as well as any database supported through Microsoft's ADO technology), and others will be added shortly. Adding support for a particular database engine is easy and quick in EDW.

3 Philosophy

Three are the main aspects of the philosophy behind the EDW project.

3.1 Creating a *framework*

Our idea of framework is that of a familiar and extensible environment in which to develop productively. Ethea believes in code reuse, but most of all in reusing best practices and methodologies for maximum productivity, without sacrificing flexibility. That's why we have brought to EDW the very same ideas upon which our InstantSolutions framework (www.ethea.it/eng_instantsolutions.asp) is built, and made them even stronger.



3.2 Extensibility

Every aspect of EDW can be customized; every building block of EDW processes and objects can be substituted. EDW aims to offer the programmer powerful tools that do as much as possible on their own, while at the same time leave room for intervention at all levels for those cases (rather frequent in nowadays' applications) in which the default behaviour is insufficient or inadequate.

3.3 Using *meta-data*

Wherever practical, in EDW you program by creating meta-data, rather than cutting code: structures for menus and tool bars, forms and fields, constraints, SQL instructions for data extraction, transformation rules are all stored in external files (usually scripts or XML documents) so that they are easily modified without the need to rebuild the application.

The use of meta-data brings you **maximum reuse** (of both code and portions of an application), **less bugs** and ability to **easily intervene** on a deployed application in cases of "emergency".

4 Technology

The core of EDW is built in Delphi 7, Turbo Delphi for Win32 or Borland Developer Studio 2006. We're considering porting EDW to .NET or other platforms, depending on customers' requests. You can extend EDW at the source code level to achieve a total degree of customization, but you should also know that most of the things needed to build an application with EDW do not require Delphi nor expert developer skills.

EDW is based on a growing set of high-quality libraries: certain modules of EDW require specific libraries. Some of them are included in Delphi, some are Open Source and some commercial. Plus, some libraries used in EDW are developed by Ethea. These are required in those cases in which you need to recompile the EDW modules themselves; cases that, according to our philosophy, become less and less frequent as time goes by. Here is a short list of libraries currently used in EDW:

- ◆ VCL, MyBase, InterBase Express, dbGo (Borland): used in the EDW core and in the data access layers (IBX for InterBase/Firebird and dbGo for ADO/Microsoft SQL Server).
- ◆ InstantObjects, RemObjects Pascal Script, SAX for Pascal (Open Source): used in the core, for persistence of meta-data objects as XML documents, for scripting (used in many places throughout EDW, from configuration files to the GUI) and for importing data from XML sources.
- ◆ Ethea Foundation, CBLib (Ethea): used in the core, in the user interface and in the reporting module based on OpenOffice.org.
- ◆ ReportBuilder, PivotCube VCL (Commercial): used in the reporting module based on Report Builder and in the multi-dimensional analysis module.

5 Availability

At Ethea, we use EDW together with our own BI-related skills to develop custom applications for data gathering and information analysis. Thus, we have developed and continue developing EDW mostly for our own use.

Nonetheless, as we do with Instant Solutions, we are willing to licence the technology itself to those wanting to develop EDW applications on their own (possibly with support from Ethea).

Contact us (info@ethea.it) to request more information, a demo or a trial of EDW, and keep an eye on our web site (www.ethea.it/eng_edw.asp) for further, news, demos and technical details.