

## DataEase 6.5 - Opening up DataEase to the World

### Introduction

The rapid adoption of new information technologies and their integration with existing systems poses new challenges to DBMS developers and DataEase users in particular. With the new release of DataEase (6.5) Sapphire fully meets the demand for seamless connectivity with other systems opening up a variety of new opportunities to DataEase users.

The purpose of this article is to give you an overview of the new aspects and improvements of DataEase 6.5 with an insight into some of the technical details to show the benefits you will derive from upgrading to 6.5.

The article covers the following topics

- Data access mechanisms employed in the previous versions of DataEase (versions prior to 6.5) and their limitations
- OLE DB
- The benefits OLE DB brings up to DataEase users
- The new scripting features and practical examples of how customers will be using these features in real situations
- WebPublisher 2 as a means of opening DataEase databases to the Internet world
- Summary

### Data Access Mechanisms

What is DataEase from a data management viewpoint?

DataEase is a consumer of data on the one hand and a provider of data from the other. DataEase is a file server system meaning that all information is stored in files and is accessible from all workstations connected to a LAN through a client program. It has its own proprietary database, i.e. information is stored in a DataEase native format. In order to communicate with other programs this information must be easily accessible and exposable to any type of applications – both DataEase and external. So, it is of fundamental importance what mechanisms are employed to provide connectivity between different databases.

There are several ways to access data in the DataEase development environment:

- Through one of the native database interfaces that provide direct connections to data in DataEase applications, and
- Through one of the standard database interfaces such as ODBC (Open Database Connectivity) and OLE DB (Object Linking and Embedding Database).

#### Native database interfaces

Data stored within a database can be accessed through special data access mechanisms or application programming interfaces (APIs) that are used in application development. Implementations of programming interfaces are libraries that contain packages of functions providing all the necessary functionality for handling data – ability to connect to a database, create and process queries, etc.

Each DBMS vendor implements its own sets of functions called 'native database interfaces' to provide native connections to data stored in its databases. The data processing functionality is provided by the core of each DBMS called a database engine. Different native database interfaces can be implemented in different languages (even proprietary to a specific vendor). In fact, functions from different vendors' libraries can be called differently and have different number of parameters through in most cases they perform the same actions.

The appropriate DataEase-specific interface library is called PRISM and provides the most efficient way for handling data in the DataEase native format.

However, this mechanism impedes communication between DBMSs from different vendors because application developers might not wish to learn the technical details of each

database they work with. As a rule, they prefer to use one of the standard database interfaces described below.

#### Standard database interfaces (ODBC)

ODBC is an industry standard to access data from a variety of relational data stores.

Each DBMS vendor stores databases in the proprietary format (for example, DataEase stores data, tables, interfaces to data, etc. in files, whilst Access stores everything including data in one file) and has specific interfaces to databases. The data stored within a specific DBMS can be exposed to another DBMS through an ODBC driver. An ODBC driver translates standard queries and data handling functions (ODBC standard-compliant) into commands that can be understood by a specific DBMS, for which this ODBC driver is written. Each DBMS requires a different driver.

As a rule, a driver that ships with a particular DBMS and intended to accept and process SQL commands via an ODBC connection, consists of a standard set of ODBC APIs responsible for data access, data retrieval and manipulation. In case if the DBMS' data processing power surpasses the capability of the ODBC driver, an ODBC-enabled application will need a more powerful than ODBC access: a driver that implements the whole range of DBMS-provider' proprietary interfaces and provides all the necessary data handling functionality.

A couple of third-party ODBC drivers were created to enable applications from other vendors to access data in DataEase databases using SQL. To communicate with other vendors' applications, DataEase has special non-ODBC drivers that ensure good interoperability.

#### Limitations of ODBC

ODBC drivers can be rather slow or not powerful enough.

ODBC is bound to the SQL language and can access data stored in relational databases only.

The transition from the use of ODBC to OLE DB helps completely remove these limitations. Why?

#### **Switching to OLE DB**

Whereas ODBC was created to access relational databases, OLE DB is designed to access data from multiple heterogeneous data stores. Relational sources can be DBMSs like DataEase, databases on mainframes, back-end databases (ORACLE and Microsoft SQL server, etc.), desktop DBMSs like Access, FoxPro, etc., and more. Non-relational information sources include Windows NT and UNIX, mail servers, hierarchical DBMSs, graphical data, geographical data, and more.

From the technical standpoint, OLE DB is a standard set of OLE interfaces enabling access to any type of data from any data store. These interfaces support all the data managing functionality of any DBMS, making its data exposable to the outside. The form in which data is retrieved from any information source needn't be SQL-compliant, as in the case of ODBC. At a minimum, an access provider must implement the interfaces to databases necessary to expose data in tabular form. The only requirement to any data source is that it must provide sufficient inherent functionality to process its own data.

OLE DB consists of a data Provider and data Consumer.

#### OLE DB Provider

The Provider is an application that allows other products (e.g. Excel, Crystal, Access) to read and modify DataEase tables. Also, the Provider permits use of SQL against DataEase native database.

### OLE DB Consumer

An OLE DB Consumer is responsible for retrieving data from an external OLE DB- compliant data source and formatting it to use as the basis of a DataEase table or tables. It can be any kind of data - relational or hierarchical, structured or flat, written in a standard or a proprietary format, and accessible or inaccessible through ODBC.

In conjunction with the Provider, it permits DataEase to access native DataEase tables in a different application. As with existing drivers, OLE DB provider allows for seamless interchange of data between DataEase and other systems, and for DataEase to be used as functional extension of existing systems written in other products. However it should be noted that the Consumer is not intended for creating complete applications over external data sources - for such purposes NetPlus will be used.

Thus, DataEase 6.5 applications will be able to seamlessly interchange data with other OLE DB compliant systems - both ways! - because the OLE DB database technology outperforms ODBC in most ways.

### Example

Any DataEase application is comprised of GUI and data stored within the DataEase system. Like most DBMSs, DataEase has specific types of data recognizable by DataEase applications only - choice-lists. Thus, data contained within a choice-list cannot be directly transferred into external applications even through an ODBC driver because ODBC specification does not support the choice-list data type.

OLE DB lifts this constraint: the choice-list data can be read into external non-DataEase applications, the GUI of which can be implemented in any programming language, for example, Visual Basic, C++, etc.

Fig.1 below shows the Members form part of the Club ParaDease sample application reproduced in VB.

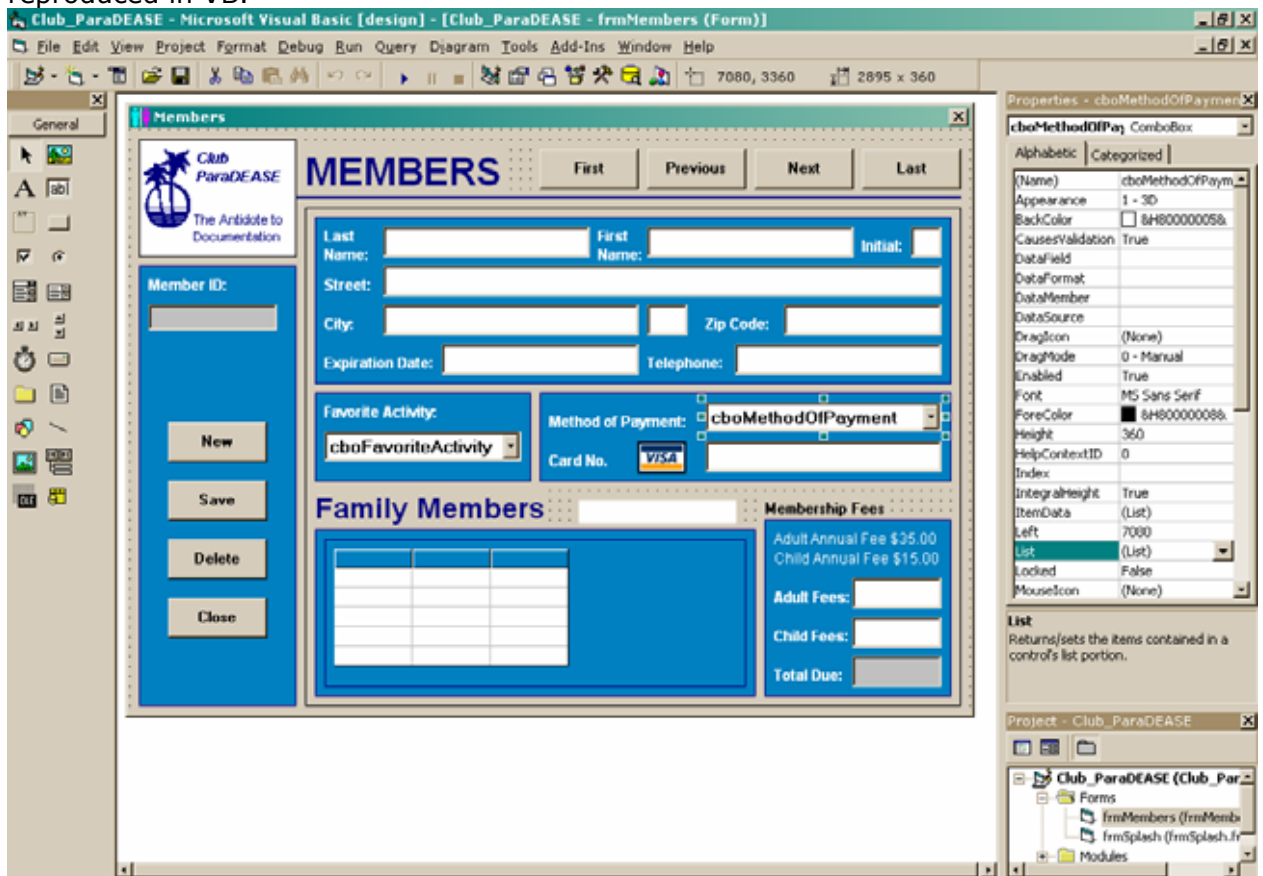


Fig.1 The Members form from the Club Paradise application.

Without a doubt, the GUI can be created in any other language and designed in compliance with the requirements of a corporate style.

With the DataEase OLE DB Provider, the table underlying this form will be populated with data retrieved from the DataEase's Club ParaDease.

Another advantage of using OLE DB is that DataEase data is accessible any time, with the source DataEase application running or not.

### **The benefits of using OLE DB**

OLE DB gives you higher performance because:

1. Via DataEase 6.5 you are able to view/modify/report on DataEase for DOS data either via the DataEase 6.5 interface or using Crystal Reports, Excel, Access etc. Therefore, it is possible to extend the life of a DOS application whilst planning the conversion.
2. DataEase 6.5 can be used to report on or integrate with data held in a variety of sources e.g. SQL Server, Access.
3. Performance limitations of ODBC no longer apply.

This technology will also work with DE DOS 5.16/7 but 4.53 users will not be able to use this technology...

The implication is not interoperability (and its associated problems) but extending the life of these databases by utilising DE6.5 "in conjunction with" i.e. separate applications are required.

### **The new scripting features in DataEase 6.5**

DataEase 6.5 now supports a wider scripting feature set described below and introduces a number of improvements to the existing versions of DataEase.

1. DataEase 6.5 preloads CDF functions and starts treating them as permanent part of the product. Technically, this means that the CDF-related files - .dll and .ini - are automatically registered into the system during setup and then captured and executed whenever a CDF is called. In the previous versions *each* CDF had to be registered manually: a developer had to manually type in the function name, its parameters and enter the library name where the function resides. Among other things, CDF preload permits easy execution of actions from scripts.
2. Ability to process objects not only in the current document in scripts (not DQL) or increased/full use of the .(dot) notation in scripts.  
Starting from version 6.0, DataEase supports script processing for all types of controls on forms. The DataEase v. 6.0 and 6.1 users have been able to effect controls and values they display only within the current document, staying within the bounds of one record.  
DataEase 6.5 breaks up this limitation and implements a more sophisticated event triggering mechanism thus making each element an object with which objects from other documents can communicate. (Please see Fig.2)  
For example, when we move the cursor to a field on form, the field gets the Got Focus event that can be processed – we can change the value/colour of the field on another form, etc. After some text has been typed in the field and Enter pressed, another event - Lost Focus - occurs that also can be processed in some way.
3. New in DataEase 6.5 is the ability to assign optional names to field objects and summary fields.  
Any object is referred to by its name. A data-related control (or field object) can be used in scripts only if it has its own unique name.  
In previous versions both a table field and a field object (edit box, or list box, etc.) to display data in a table have had the same names. Also, all summary fields were

named 'Summary', which posed particular difficulties: it was impossible to change the background colour of summary fields.

In DataEse 6.5 it is possible to give field objects and summary fields optional names that will be unique, the default names coinciding with the original names.

4. Another improvement applies to the appearance of the Script Editor: any object that has a script is marked in pick-lists, QBM Data Model, etc., in **bold**. Please see Fig.2.

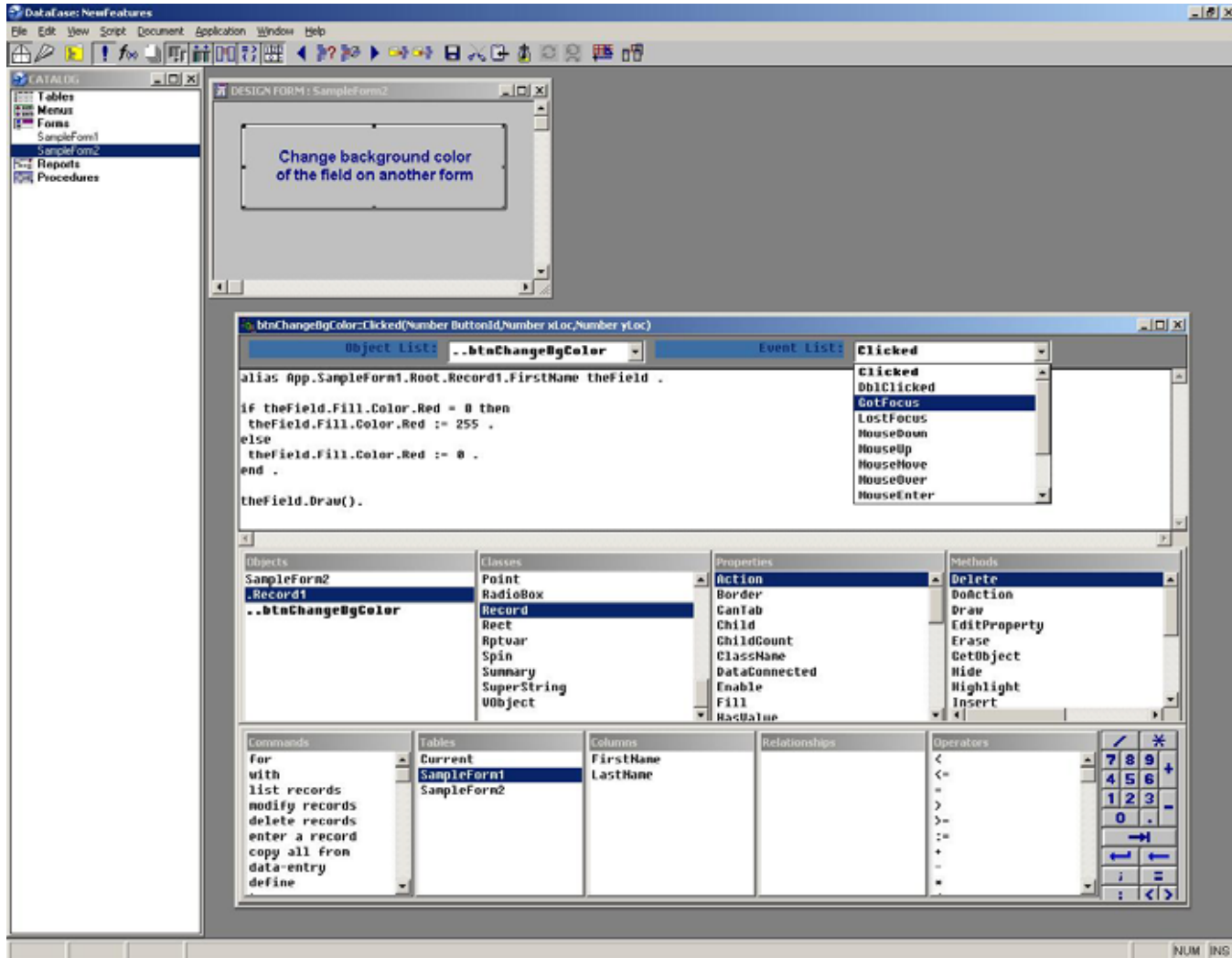


Fig.2. Scripted objects are now marked in bold.

5. New is the ability to print the text of scripts assigned to form objects from the Form Definition window: in Designer View select File >> Print >> Include Definition ON >> print scripts!
6. Kevin Glossop? Enhanced script navigation in Script Editor (previous, next, etc.)
7. ? Script Editor closable/ minimisable if not needed. There are data-related pick-lists (i.e., table names, the names of built-in functions and operations, etc.) that could be closed if not needed but the rest of the pick-lists remained visible all the time.
8. ? Columns selectable in Script Editor (use existing icons).
9. Another improvement is the implementation of the Carriage Return event on buttons. Earlier, you were unable to run the script on a particular button by just moving the cursor to it using the keyboard and then clicking Enter. The script could be run only if you clicked on the button with the mouse.
10. ? Now DataEse 6.5 supports the Click on button event – параметры (какая кнопка мыши нажата, X и Y координаты). У поля есть событие KeyInput

(код нажатой кнопки). Раньше все параметры были =0, а теперь реализовано.

## **WebPublisher 2 as a means of opening DataEase to the world of database interoperability**

Integration of some of the web technologies into the DataEase environment makes DataEase applications accessible through a common Web browser. WebPublisher (1.0, now 2.0) is a proprietary software solution that comes with DataEase 6.5 and allows the user to create Web-forms from normal DataEase forms. Without a doubt, making a normal DataEase application web-enabled gives users more flexibility because users gain access to information stored within their DataEase applications – desktop or networked versions - any time from any web-connected computer.

The second version of WebPublisher is an enhanced version of WebPublisher 1.0 with some restrictions and limitations removed and several new features added. In a nutshell, WebPublisher 2.0 is now capable of doing the following things:

- Publishes DE6.5 forms quickly and simply to the web.
- Different users can see different parts of the same application.
- Different users can see subsets of the database – i.e. just their orders.
- Screens can be custom designed and company logo added.
- Data can now be published from DOS5, DE6, SQL, and Access – at the same time!  
No need for complex Cold Fusion type technologies...

Anne Williams explains the strengths of WebPublisher 2.0 in greater detail in her [article](#) published in the spring Dialogue issue.

Some of the You have already had a chance to adopt WebPublisher 2.0. To see what has come out of it, please read the HBC case study [http://www.dataease.com/p\\_case\\_hbc.php](http://www.dataease.com/p_case_hbc.php) .

### **Summary**

#### Was

- ODBC data access technology that required LinkEase and special Sapphire –written drivers created over other vendors API
- WebPublisher 1.0
- Not sufficient Scripting support

#### Now

DataEase has evolved into a powerful, feature-rich product that gives its end-users greater flexibility and provides more functionality.

- Both OLE DB provider and consumer have been implemented
- WebPublisher 2.0
- Advanced Scripting support