Understanding DataEase
Forms

Graham Smith
PLM Consulting, Inc.
gsmith@plmconsulting.com
October 2005

Preface

DataEase forms are unlike DFD forms in most significant ways. At least once you get beyond the fact that forms are used to create and modify tables. For that reason, a new approach to the creation of forms needs to be considered. Beginning with their names.

The Naming of Forms

While this should be a fairly simple issue, it is not. And the reason it is not is because DataEase uses Forms both as a way of defining Tables and as a way for the users to interact with the data. In DFD, the two things were the same, since it was pretty much a form=table kind of thing. In DataEase, however, there can be multiple forms over tables and that changes everything.

In the earlier days of DataEase (including Express), there were often cases where a form would get messed up while it was being tinkered with and as a result, if it defined the table, the table would get messed up as well. For that reason, many people decided that they needed to have separate Table Owning Forms (TOF) and Table Using Forms (TUF). The TOF would only be used for managing the table, and the TUF would be used by the users.

This was a good idea, but it resulted in a dilemma. The TUF had to have the same name as the Table or you could not use F10 (F10 will always take you to the form that has the same name as the table). While at the same time, the TOF had to have the same name as the table or you could not use Install and Replace.

There is no one correct answer to this dilemma and any solutions I have seen proposed always ended up with some compromise and extra work. Here are the options as I see them.

1. Let the TOF and the TUF be the same thing. If the forms and tables are not too complicated, it doesn't really matter that much. As a simple compromise, the ownership can be taken away from the form except when it is required to change something in the table. The only down side is that you have to include all the fields from the table in the form – just hide the ones that the user doesn't need to see.

2. Give the TUF the same name as the table. This allows F10 to work. Forget about TOFs unless you have to change the table. When you do, just create a form over the table for as long as it is needed then get rid of it (put a _ or something in front of the name to distinguish it). The only down side is that you cannot install or replace a TOF if it does

not have the same name as the table. Not really that big a deal in most cases.

3. Give the TUF a different name than the table. This allows the TOF to have the same name as the table and install and replace work. The downside is that F10 won't work and you will have to give the user a replacement. In a system where a good deal of control is required, this is not such a problem. (This is my preferred method)

Standard Toolbars

For small applications, there are two different ways you can use the built in toolbars. One is to just use the default menu, the other is to create a custom toolbar.

Custom toolbars are a very simple way to give an application a easier to use interface, without a lot of extra work. Just open Preferences, click on the Toolbar button, and check Custom toolbar. You will then be prompted to pick which icons you want to add and in what order.

Pick the functions that are used most, either in general or for this application. In all truth, you don't really need to have a toolbar at all, since every function on the toolbar is also available on the menus. Many of the functions also have Function keys assigned as well. But toolbars are traditional and users may complain if they are not there.

The best thing about the custom toolbar is that once you have created it, it will automatically be used in all forms you create. The one downside to them is that you only have access to the functions listed and you cannot modify them to use scripts or CDFs.

Regardless of whether you have setup a custom menu or not, it is possible to turn the toolbar off or define a custom toolbar for each form.

Simulated Toolbars

If you turn off the toolbar completely, it is possible to use graphic objects to create your own simulated toolbars. The primary advantages are that you can choose what graphics you like, place them where you wish, and control what they do. Following is a sample of a simulated toolbar.

Pull-Down Menus

While the pull-down menus that appear across the top of the page can be heavily customized, they can never be turned off completely. There is very little limit to what you can do with these, excepting that you cannot execute OML script from them.

Like the toolbars, there are a lot of items of dubious value. However, unlike the toolbar, there is no way to create a "master menu" that will populate forms as they are created. Menus do have the option to Add, Delete, Copy, and Paste items, so you can create a form with a suitable menu then just copy it to other forms. Following is a list of a revised menu that has what I consider to be the most important items:

```
&File
    Designer &View       F4
    ------------
    &Print...            Ctrl+P
    P&rint Setup...
    ------------
    &Export...           Ctrl+E
&Edit
    <Undo>               Ctrl+Z
    ------------
    Cu&t                 Ctrl+X
    &Copy                Ctrl+C
    Copy &Record
    Copy Spec&ial...
    &Paste               Ctrl+V
    C&lear Field         F6
    Clear &Form          F5
    ------------
    &Save                F8
    Save &As New Record  F2
    &Delete Record...    F7
    R&estore Record      Ctrl+F5
&View
    &Form
    &Table
    ------------
    &All Records
    &Queried Records
&Goto
    &First Record
    &Previous Record     Shift+F3
    &Next Record         F3
    &Last Record
    ------------
    &Related Form...     F10
    ------------
    Loo&kup...           Ctrl+F10
    <Return Data to>     Ctrl+D
    ------------
    All Forms            Ctrl+O
&Query
    &Select Records...   Alt+F5
    So&rt Records...
    ------------
    Selection &Filter...
    ------------
    &QBM - New Report... F9
&Help
    &Contents
    ------------
    &User Help   Alt+F1
    ------------
    &About DataEase...
```

You can also add a couple custom entries: Reports and Procedures. These give you a place to put Reports and Procedures that act on the existing record using ExecCDF and a global to get the unique value of the record.

It is also possible to add entries in the GoTo section to take the user to specific forms. This is of particular consequence if you decide to remove the F10 functionality.

Form Layout

Unless you can be assured that everyone using the application has the same monitor resolution and font size, you are going to have to design for the lowest common denominator. In most cases, this is going to be 800x600. If you want to design a form that will fit on a single screen, with no scroll bars, you will have to keep all screen object in a area no more than 8 1/8" wide and 5 1/8" tall (less if you use the standard toolbars).

Pick a standardized layout that is consistent throughout the application and is easy on the eye. The choice to distinguish Required fields is up to you, but it can be helpful to the user. Same goes for distinguishing Prevent Data-Entry fields so the user doesn't get confused when they cannot put the cursor into the field.

Decide in advance if you are going to use choice lists for yes/no or check boxes. I prefer the check box and most users I have spoken with seem to prefer that as well.

There is an option on subforms to never hide the scroll bar. Unless you have a special case, you should always check this button. This makes it clear that this is a subform and not just a collection of fields. I also find it helpful to use a different looking background to further help emphasize this fact.

Using Forms as Menus

There is nothing fundamentally wrong withDataEase Menus, but they are limiting since they cannot be data-connected. A better choice is to create a one record table that menus can be placed on. Here's why.

You can use this one record table as a place to hold constant values. Place a virtual field on the table and create a derivation that will assign global variables to the constant values. Here is a sample derivation:

```
SetGlobal(1,ActivePeriod) + SetGlobal(2,PriorPeriod)
```

Make sure that the Menu forms are prevent data-entry which will automatically cause them to display the first record. When the user opens a menu, the values will be automatically loaded into memory. Place and hide the virtual field on the form and make sure that it is the first thing in the tab order so that it will parse first.

Naming Conventions

In DataEase, there can be no two documents, regardless of type, that have the same name. This makes naming things a bit more of a challenge. One way to deal with this is to use prefixes that not only increase the variability of names, but also sort them in the catalog and allow them to be grouped into functional units. Here are some prefixes that can be used and how I have used them in the past.

$ System Forms, Reports, Procedures

> Menus

^ Data-Entry Forms for Reports

_ Table Using Forms (when the TOF has the same name as the table)

- Reports and List Record Procedures

get Reports used for lookup tables

% Control and Processing Procedures

z Template Procedures

zz Temporary procedures used during migration or development

zzzz Very temporary procedures for testing things.

Form not= Table

One of the most unfortunate things about most DataEase applications is that the developer fails to take advantage of the the one feature that most distinguishes a DataEase application from a DFD application – the fact that the Form and the Table are NOT the same thing!

I say that this is unfortunate because it puts severe limits on how users are able to view and interact with the data. The purpose of a database is to store data and produce information. Anything that gets in the way of this primary purpose is, at best, counter-productive.

Not every user needs to be able to enter and modify data – some just need to be able to look things up. This can be done using reports, but can also be done by giving the users a form that is prevent-data-entry and specifically designed for doing searches. If you then teach the users a bit about using QBM and Export, you will find that they may be able to do a lot of things you never thought of.