

## Minimal Conversion from DataEase for DOS to DataEase 7.2

### Service Proposition

If you want to convert your DOS based DataEase application to the newest and most stable DataEase for Windows version, DataEase 7.2, you can:

- ▶ **'Go it Alone'** and undertake the conversion internally, without any external assistance;
- ▶ Use our **Minimal Conversion Service** for **£199**, get a kick-start on the process, save many hours of work and then look at what else we can do for you!

### Go It Alone

The best way to migrate your DOS application without external assistance would be to use the built-in migration tool found in DataEase 7.2. **Note:** If your application is in version 4 or earlier, you will need to purchase and install a version of DataEase for DOS 5.53, as the migration tool only ports 5.x applications to DataEase 7.2.

As a result of using the migration tool in DataEase 7.2, you will get the following:

- ▶ relationships;
- ▶ forms;
- ▶ data (at time of performing the conversion);
- ▶ users;
- ▶ imports.

However, there are a number of limitations to using this approach, as it will not provide:

- ▶ identification of possibly unused elements
- ▶ data transfer (export/import) routines;
- ▶ field/relationship renaming;
- ▶ form/field level styling;
- ▶ reports and procedures;
- ▶ menus.

**For £199, our Minimal Conversion Service  
will provide ALL of the above!**

## Minimal Conversion Service

For **£199** plus VAT we can take the first steps to upgrade your application to DataEase 7.2 and advise you on what you need to do to get it functioning in full in a Windows environment.

### Identification of Possibly Unused Elements

- ▶ Does your application hold elements no longer used by your end-users? Then there is no need to convert them! Through the Minimal Conversion, you get a list of forms/procedures/menus that are **possibly** unused. All you have to do is confirm their usage, add any others you wish to exclude and send us back the list – we will do the rest!

### Data Transfer (Export/Import) Routines

- ▶ Following the migration of your DOS application to DataEase 7.2 you may wish to test it using the latest set of data currently held in your DOS production database or finally move the new application into production. Data export and import routines allow for a smooth transition of extracting your data from your DOS application and importing it into your new 7.2 application, at any point during development or in readiness for production.

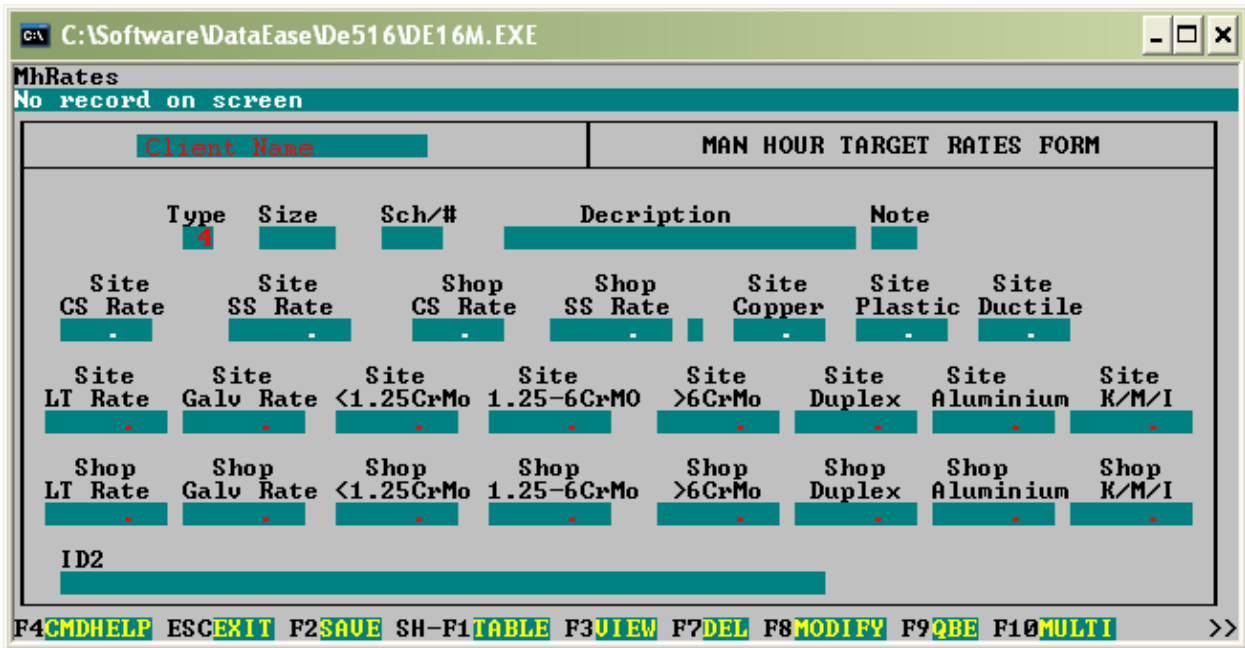
### Field/Relationship Renaming

- ▶ Your application will benefit from a consistent look and feel e.g. all field names should either be in proper, upper or lower case and not a combination of upper and proper case. In addition, fields that contain reserved words e.g. DOS field name "Business Date" (note the space between "Business" and "Date"), means you may have problems creating or running a procedure that checks its field contents, as the word "Date" will be confused with the reserved word of the same name, in DataEase. To combat this problem, the space is removed. Field renaming may have an impact on field derivations, validations and/or DQL syntax in your procedures and these areas will need to be addressed;

### Form/Field Level Styling

- ▶ In addition to providing your application an aesthetic and consistent look & feel to your end-user, field styling allows the end-user to determine which fields are required, not required or prevent-entry/virtuals; e.g. if user X wants to identify and input data only into the required fields of a form which has 50 fields, he knows which fields these are because each of them will have a red border, indicating they are required, around them. The alternative, and of course the painful way of identifying these fields, would be to use a "trial" and "error" approach of entering a partial record, hitting the save icon and acting upon the error message displayed. Very painful on large forms!

Original DOS form



**MhRates**  
No record on screen

Client Name

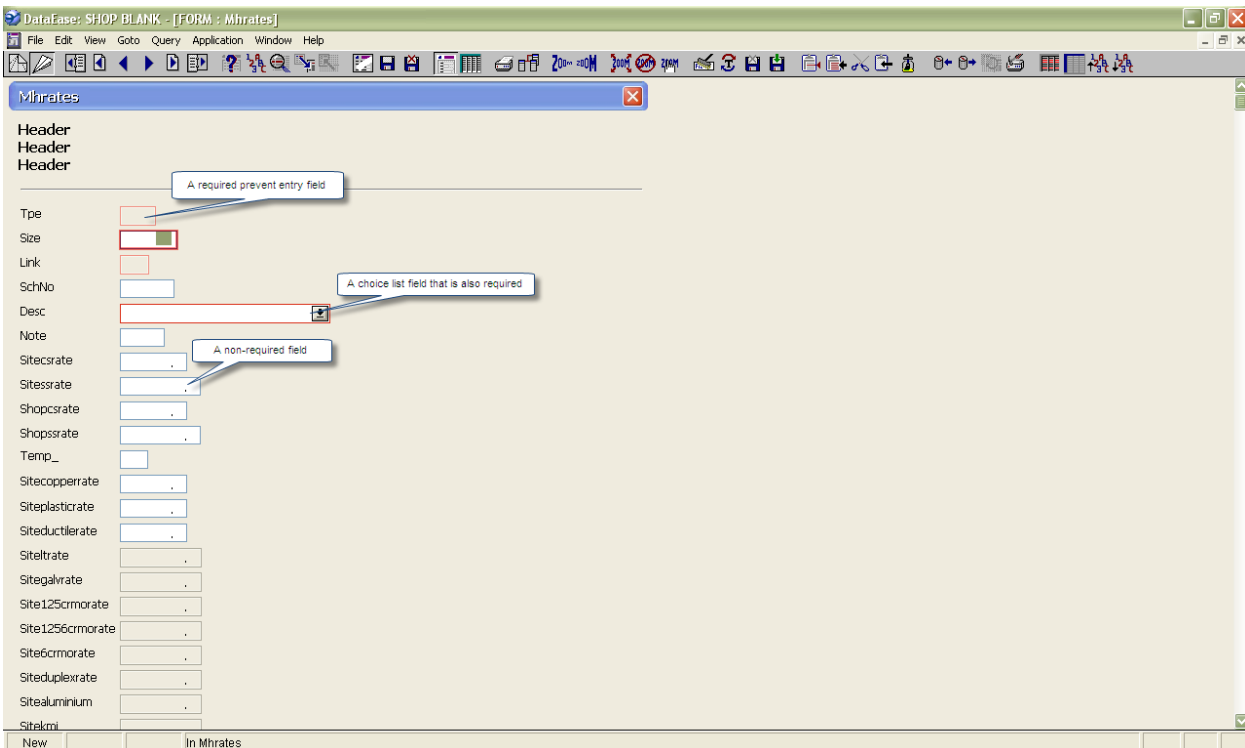
MAN HOUR TARGET RATES FORM

Type	Size	Sch/#	Description			Note	
Site CS Rate	Site SS Rate	Shop CS Rate	Shop SS Rate	Site Copper	Site Plastic	Site Ductile	
Site LT Rate	Site Galv Rate	Site <1.25CrMo	Site 1.25-6CrMO	Site >6CrMo	Site Duplex	Site Aluminium	Site K/M/I
Shop LT Rate	Shop Galv Rate	Shop <1.25CrMo	Shop 1.25-6CrMO	Shop >6CrMo	Shop Duplex	Shop Aluminium	Shop K/M/I

ID2

F4 CMDHELP ESCEXIT F2SAVE SH-F1TABLE F3UIEV F7DEL F8MODIFY F9QBE F10MULTI >>

WINDOWS form after Minimal Conversion



DataEase: SHOP BLANK - [FORM: MhRates]

MhRates

Header  
Header  
Header

A required prevent entry field

Type

Size

Link

SchNo

Desc

Note

A choice list field that is also required

A non-required field

Sitesrate

Sitesrate

Shopcsrate

Shopsrate

Temp\_

Sitecopperrate

Siteplastrate

Siteductlerate

Siteltrate

Sitegalvrate

Site125cmorate

Site1256cmorate

Site6cmorate

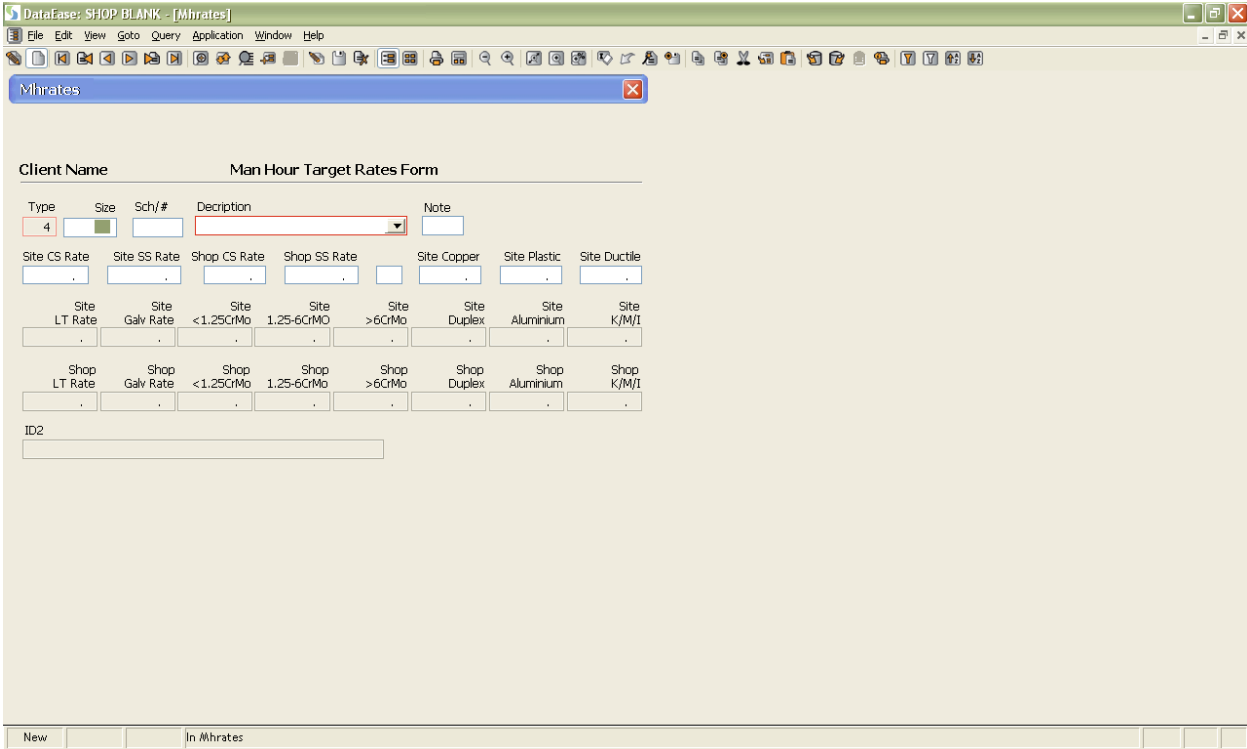
Siteplexrate

Sitealuminium

Sitekmi

New In MhRates

WINDOWS form after the fields have been laid out as in the DOS version.



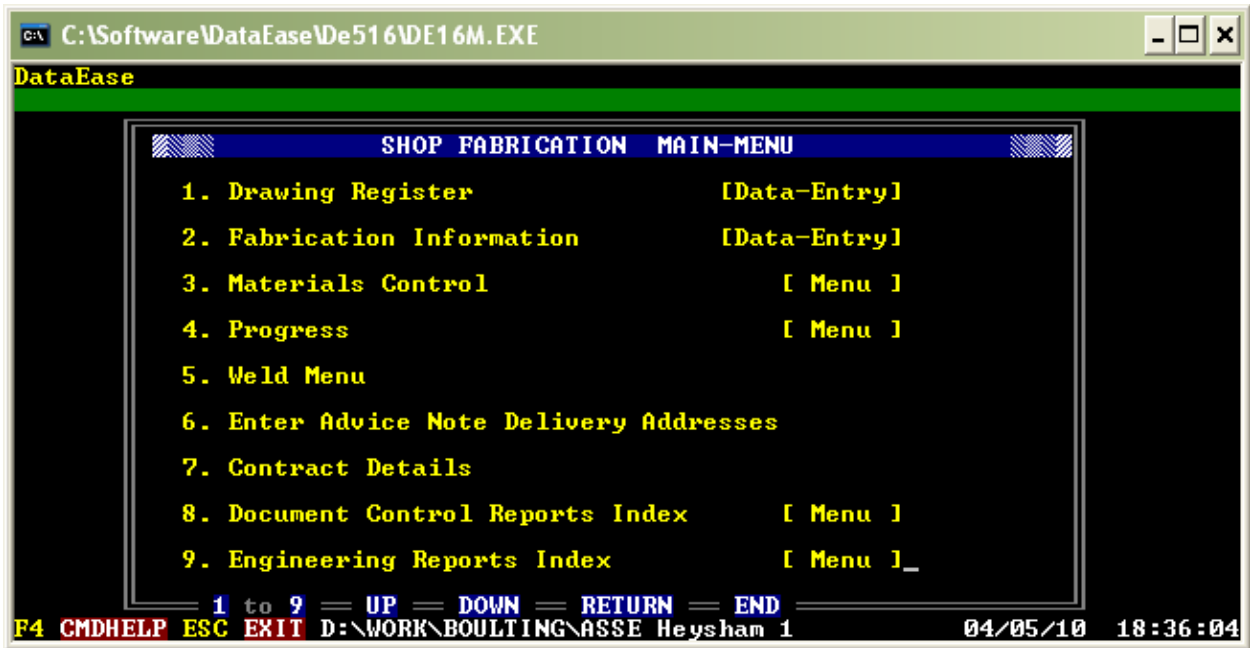
## Reports & Procedures

- ▶ You will of course want your reports and procedures in your new 7.2 application. The minimal conversion will migrate your DOS procedures, to DataEase 7.2, but commented out i.e. the code will reside in your new 7.2 application but will not be functional (QBE's, aka Reports, will be converted to DQL code); in addition, data-entry screens will not be created. The object here is not to do the complete conversion but to simply provide you with a viable starting point for re-developing your routines in DataEase 7.2!

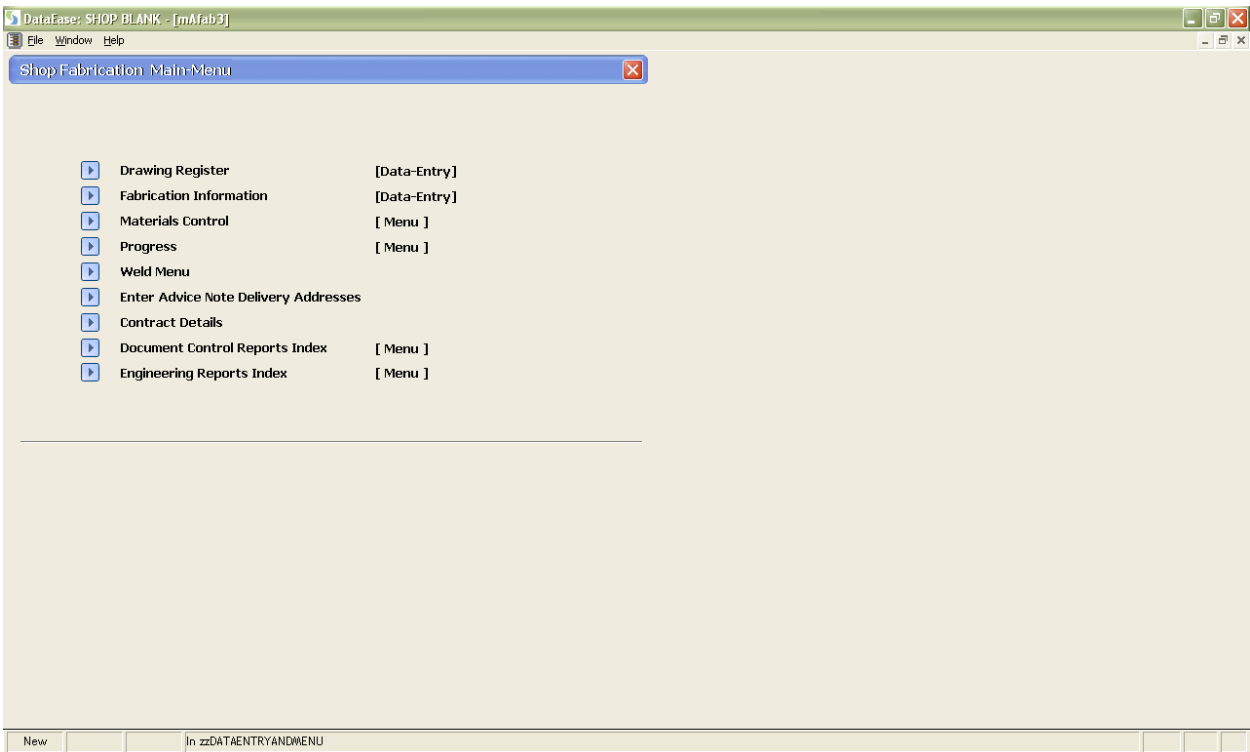
## Menus

- ▶ Naturally, your end-users will expect to run your application from a menu as they did in DOS! Menus in DataEase 7.2 are simply buttons (with an action to run a form or procedure) on a form.

Original DOS menu



WINDOWS menu after minimal conversion



## What you need to do after Minimal Conversion

As we've said before, this is not a full conversion service, so there are a number of tasks that the Minimal Conversions Service will not address:

- ▶ layout of fields on forms;
- ▶ creation of subforms;
- ▶ imports created ad-hoc by users;
- ▶ testing of data to ensure integrity and accuracy;
- ▶ testing of form functionality (e.g. do derivations produce expected result?);
- ▶ procedures - the DQL code in procedures will be commented out and will therefore not be functional nor display fields in the body of the procedure (if it is a listing procedure); QBE's will be converted to DQL;
- ▶ testing of procedures;
- ▶ creation of data-entry screens to procedures;

To address the areas after the minimal conversion is completed you will need to follow the guidelines below:

### Field Layout on Forms

Drag each field around the form using the mouse cursor, align them and set tab stops. Text labels would also have to be renamed where applicable. Use the equivalent DataEase for DOS (DFD) screen (in design and user mode), as a guideline.

### Subforms

Draw your subform on each form, using the default wizard options (which places all fields on the form). When complete, remove unnecessary fields and align the remaining. Again, use the equivalent DFD screen (in design and user mode), as a guideline. Add/amend existing label text.

### Testing Form Functionality

This stage is laborious but essential! What happens if your user is unable to save a record? What happens if a derivation such as in a date field, produces an incorrect value because it relies on a field value elsewhere on the form or on another form? Ensure saving/modifying/deleting records produces the expected/desired result. In addition, EACH derivation/validation will also need to produce the expected/desired result. This can only be done by understanding the purpose of the derivation/validation and if a construct (such as if-else-then) is used, ALL aspects of it are fully tested. For example, if field: "Title" has a derivation formula of: **if (lookup "Customer", "Position" = "Doctor", "Dr", Title)**, then you will need to ensure either text value "Dr" or the value in field "Title" is accepted based on the "Position" specified in form "Customer".

### Testing of Data

This is probably the most time consuming of all areas. For EACH FORM created in DataEase 7.2, export their data out and use a third-party product e.g. **Total Commander** or **Windiff** to compare against the DataEase for DOS equivalent (which was performed during the minimal conversion stage). If using Total Commander, differences will be highlighted in red font. **Total Commander** and **Windiff** are 3<sup>rd</sup> party products and not supported by Sapphire; Sapphire can however advise clients on their use in this context. For each difference, refer back to each of the application's forms, navigate to the appropriate record in user mode, spot the field displaying the differences and address the issue.

## Creating Procedures and their Data-Entry Screens

For EACH procedure, FIRST draw any necessary data-entry fields, ensuring the look and feel are exactly the same as that found in your FORMS. As with creating the field layouts, you will need to make sure all fields are aligned and tab stops set. Text labels would also have to be created where applicable (again use the equivalent DFD screen, in design and user mode, as a guideline). Then UN-comment the DQL code in "scripts" section of the procedure. The code you see will have been taken straight from your DFD application so you may need to make changes such as form/relationship/field names.

The quickest way to do this would be to compile the procedure and if an error message is displayed, address the issue and re-compile till no more error messages are displayed. Please note, it is considered good DataEase programming to spilt groups of your DQL code across several procedures as opposed to having it all in one. When no more error messages are displayed, lay the fields out in the body using the equivalent DFD screen (in design and user mode), as a guideline. Save your procedure at regular intervals.

## Testing Procedures

Once you have created all your procedures, you will of course need to test them! This can be laborious but obviously an integral part of development. There are several ways to achieving this. The most favourable would be to identify a PROCESS or a group of procedures that constitute a process (typically a menu item or group of menu items) and then test their OUTPUTS. Ensure all aspects of the process are fully tested and the same "test" is performed on your DFD application with the outputs matched against the DFW version.

Finally, ensure ALL menu items have the appropriate action (typically to open a form or run a procedure) assigned to them.

---

## Optional Services

Once the Minimally Converted application is delivered, you they have a number of choices on how to proceed, namely:

- ▶ **Option 1** - Undertake the remaining conversion activities in-house using our guidelines listed above and not involve us again in any way
- ▶ **Option 2** - Undertake the remaining conversion activities in-house and involve us selectively, to provide:
  - ◆ Telephone/Email support on a Pay-as-you-Go basis, to help you over the odd few problems you feel you may encounter
  - ◆ Training/Coaching in 7.2 and in planning the conversion activities as well as addressing problems with the conversion
- ▶ **Option 3** - Hand-over core subsets of the remaining conversion activities to us, which we undertake as T&M assignments
- ▶ **Option 4** - Hand over all of the remaining conversion activities to us, which we undertake as a Fixed Price assignment.

Costs for **Options 2 – 4** can be provided on request.