

A user has entered wild card characters (* ? ~) as data into an indexed field.

This has caused a problem with the integrity of the index and the records will have to be modified, but how does one search for a wild card character without displaying all records (DOS or Windows)?

The simplest way is to write a DQL procedure to search for the wild card characters as:

```
for FORM
with textpos ( FieldName , "*" ) > 0 or
textpos ( FieldName , "?" ) > 0 or
textpos ( FieldName , "~" ) > 0 ;
list records
  KeyField.
```

How to subtract a number of working days from a date field (DOS or Windows).

The first step is to deduct the number of days to the original date.

```
OrigDate - WorkingDays
```

Next the number of weekends need to be derived, simply by dividing the number of WorkingDays by the number of working days in a week, rounded down to get rid of decimal place and then multiplying that by the number of days in weekend.

```
floor ( WorkingDays / 5 ) * 2
```

Finally, we need to ensure that the resulting date is not a weekend. Do this by getting the remainder of the WorkingDays divided by the number of days in a week, deducting that from the original date and if that is a weekend, include the number of days in a weekend into the calculation.

```
if ( weekday ( OrigDate - mod (
WorkingDays , 5 ) ) > 5 , 2 , 0 )
```

The completed field derivation is:

```
OrigDate - WorkingDays -
( floor ( WorkingDays / 5 ) * 2 ) -
if ( weekday ( OrigDate - mod (
WorkingDays , 5 ) ) > 5 , 2 , 0 )
```

Counting the number of sub groups

Objective:

To print/count the number of sub-groups within a group. The following DataEase for DOS query will group by the field SALESPERSON, sub-group by the field CUSTOMER TYPE from the SALES form and count the number of sub-groups per group.

Explanation of temporary variables:

PREV_PERSON

Will hold the value of the last SALESPERSON group. An 'if' statement will be used to determine whether a new group has started by comparing the temporary PREV_PERSON variable to the SALESPERSON field.

If a new group has started, this temporary variable will be set equal to the actual group value (SALESPERSON).

PREV_CTYPE

Will hold the value of the last CUSTOMER TYPE group. An 'if' statement will be used to determine whether a new sub-group has started.

If a new sub-group has started, this temporary variable will be set equal to the actual sub-group value (CUSTOMER TYPE).

COUNTER

Will be incremented each time there is a sub-group change to accumulate the number of sub-groups within a SALESPERSON group. When the SALESPERSON group changes, this will be re-set to 1.

Query:

```
define temp "PREV_PERSON" text .
define temp "PREV_CTYPE" text .
define temp "COUNTER" number .
for SALES;
if temp PREV_PERSON not = SALESPERSON then
  temp PREV_PERSON :=
  SALESPERSON .
  temp PREV_CTYPE := CUSTOMER TYPE .
  temp COUNTER := 1 .
end
if temp PREV_CTYPE not = CUSTOMER TYPE then
  temp PREV_CTYPE := CUSTOMER TYPE .
  temp COUNTER := temp COUNTER + 1 .
end
list records
  SALESPERSON in groups ;
  CUSTOMER TYPE in groups ;
  CUSTOMER NAME ;
  SALES ;
  temp COUNTER .
end
```

Format:

```
.Group Header
SALESPERSON: (Salesperson)
.Group Header
  CUSTOMER TYPE: (Customer Type)
.items
  (Customer Name) (Sales)
.Group Trailer
.Group Trailer
  NUMBER OF CUSTOMER TYPES:
  (temp Counter)
.end
```

See Figure 1. for example output.

It is important to list SALESPERSON in groups and CUSTOMER TYPE in groups within list records. This sorts all records in order before any conditional logic is applied.

Figure 1.

SALESPERSON:	Jane		
CUSTOMERTYPE:	RETAIL		
		HALLMARK	7,000.00
		STOP AND SHOP	900.00
		NUMBER OF CUSTOMER TYPES:	1
SALESPERSON:	John		
CUSTOMERTYPE:	MANUFACTURING		
		FABRIC WORLD	200.00
		PLASTICS CORP	600.00
CUSTOMERTYPE:	RETAIL		
		JANES CLOTHING OUTLET	8,000.00
		NUMBER OF CUSTOMERTYPES:	2

Is it possible to have DataEase for DOS reports run to screen in colour?

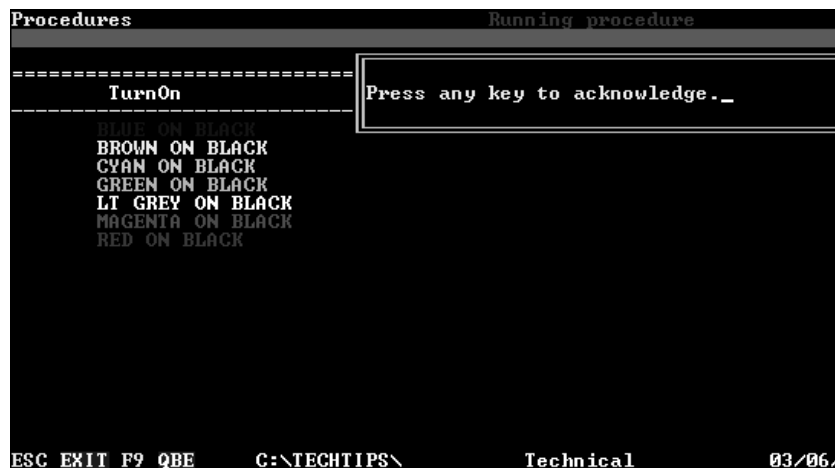
Only by embedding ansi code either side of the text or fields and sending the report to c:con.

The relevant codes are as follows:

Turn on string	Color	Turn off string
@[1B5B33313B316D]	RED ON BLACK	@[1B5B303B316D]
@[1B5B33323B316D]	GREEN ON BLACK	@[1B5B303B316D]
@[1B5B33333B316D]	BROWN ON BLACK	@[1B5B303B316D]
@[1B5B33343B316D]	BLUE ON BLACK	@[1B5B303B316D]
@[1B5B33353B316D]	MAGENTA ON BLACK	@[1B5B303B316D]
@[1B5B33363B316D]	CYAN ON BLACK	@[1B5B303B316D]
@[1B5B33373B316D]	LT GREY ON BLACK	@[1B5B303B316D]

Include a message statement to ensure the display remains on screen.

Ensure the report destination is disk, with the disk output file name "c:con".



```
Procedures Running procedure
=====
TurnOn Press any key to acknowledge._
-----
BLUE ON BLACK
BROWN ON BLACK
CYAN ON BLACK
GREEN ON BLACK
LT GREY ON BLACK
MAGENTA ON BLACK
RED ON BLACK
=====
ESC EXIT F9 QBE C:\TECHTIPS\ Technical 03/06/
```

Give DOS screen reports a colourful face-lift.

Calculating the difference in hours between two periods

Objective:

To calculate the elapsed time between a start date/time and an end date/time and the fractional number of days for the elapsed time in both DOS and Windows.

Field definition:

```
Field name: START_DT
Field type: Date
Field name: START_TIME
Field type: Time
Field name: END_DT
Field type: Date
Field name: END_TIME
Field type: Time
Field name: TOT_ELAPSED
Field type: Number
Prevent data-entry
Derivation: if ( START_TIME <=
END_TIME ,
( END_DT - START_DT )
* 24 +
( END_TIME - START
_TIME ) / 3600 ,
( END_DT - START_DT )
* 24 +
( -1 * (
START_TIME
- END_TIME ) ) /
3600 )
```

Field name: FRACT_DYS

Field type: Number

Prevent data-entry

Derivation: (END_DT - START_DT)
+ (((END_TIME -
00:00:00) -
(START_TIME -
00:00:00)) / 86400)

The basic assumption is that END_DT/END_TIME is >= START_DT/START_TIME.

START_DT/TIME

A start date and time is entered into these fields.

END_DT/TIME

An end date and time is entered into these fields.

TOT_ELAPSED

Total elapsed time between the start/end dates and times entered above.

FRACT_DAYS

The fractional number of days between the start/end dates and times entered.

Posting group summaries to another file

Objective:

This DataEase for DOS query will summarise sales for each SALESPERSON in the SALES file and post the results in the SUMMARY SALES file.

Since an Enter Record statement is used, the transactions will be added to the SUMMARY SALES file.

Explanation of temporary variables:

PREV_PERSON

Will hold the value of the last SALESPERSON group. An 'if' statement will be used to determine whether a new group has started by comparing the temporary PREV_PERSON variable to the SALESPERSON field. If a new group has started, this temporary variable will be set equal to the actual group value (SALESPERSON) and a record will be entered into the SUMMARY SALES file.

Explanation of files/fields:

The following files and corresponding fields are used:

SALES	SUMMARY SALES
Salesperson	Sum_Salesperson
Dollar_Sales	Sum_Sales

Query:

```
define temp "PREV_PERSON" text .
for SALES ;
if PREV_PERSON not = SALESPERSON then
  assign temp PREV_PERSON :=
  SALESPERSON .
  enter a record in SUMMARY SALES
    SUM_SALESPERSON := SALES
    SALESPERSON ;
    SUM_SALES := sum of SALES
    named "match"
    with (SALESPERSON =
```

```

        SUMMARY SALES SUM_
        SALESPERSON ) DOLLAR_
        SALES .
end
list records
SALESPERSON in groups ;
DOLLAR_SALES : item sum .
end

```

Format:

```

.Group Header
.items
.Group Trailer
SALESPERSON: (Salesperson) TOTAL
SALES: (Dollar_Sales)
.end

```

```

- REPORT TOTAL SALES: (
Dollar_Sales)

```

See Figure 2. for example output.

It is important to list SALESPERSON in groups. This sorts all records in order before any conditional logic is applied.

The DOLLAR_SALES field (with statistical value = sum) is used for the report total. A format (report output) is not necessary in order to post group summaries.

Figure 2.			
SALESPERSON:	JANE	TOTAL SALES:	7,900.00
SALESPERSON:	JOHN	TOTAL SALES:	12,800.00

--		REPORT TOTAL SALES:	20,700.00

Writing records to another form without massive output

In DataEase for Windows to allow records to be written from one form to another, in an order specified by a "list records", without listing all the records that are in the source form, use the following steps:

Create a new field on the source form, 1 character text, virtual called "Dummy". Save, switch to user mode.

Select "QUERY" - "QBM New report". Group by "Dummy",

do not display any other fields in that grouping. In "Group:Table Name" select and order by the desired field, do not display any other fields. "OK" and choose default layout, "OK"

There is no need to print, so "Cancel".

In Design report, highlight the field that is being ordered, **not the group field**, go to display and tick "Do not print".

In "Document", select "Query by Model" and convert to DQL. Click "Yes". In the DQL script, after the full stop and before the "end", include your "enter a record" in TARGETFORM. COPY ALL FORM sourceform. F2 to compile.

At new DQL Layout, choose merge. Save. Run.

Novell Networking

We've had many calls about installing DataEase on a network server, rather than local workstations. This increases flexibility and installation speed, as well as making maintenance and upgrading easier.

Server considerations

Install DataEase as Novell Network. (Do Not Display Conflicting User Name).

Make sure every user has read/write access.

Flag all the files : Flag *.* +R +S.

Flag PRINTERS.DAT, DENETWRK.OVL, CONFIGUR.DAT
+RWS

All users must have full rights to these files for locking abilities.

Workstation considerations

AUTOEXEC.BAT

SET DENAME=UNIQUENAME

DENAME must be unique for each workstation.

If the DENAME you're trying to sign on with is being used then DataEase will tell you.

SET DEPATH=DRIVE:DIRECTORY

DEPATH must specify the drive/directory where the

DataEase file DENETWRK.OVL

and PRINTERS.DAT are located on the server.

If this is incorrect DataEase locking strategies will not work properly.

CONFIG.SYS

Files=100

This is the minimum recommended for running DataEase.

Buffers=20

Each buffer requires 532 bytes of memory, Buffers will use High memory if DOS is loaded High.

SHELL.CFG/NET.CFG

File Handles=100

This is the minimum recommended for running DataEase

User rights

The user rights for the DataEase Program directory and Database Data Directory should be :RWCEMF: If the user has read/write access but not modify this will result in the DBM file being deleted after a form reorganisation.

Locking strategies

DataEase can run out of locks on a file server for a number of reasons. A user could be performing a search on all records thus increasing the amount of lock requests to the server. A workaround for this is to increase the locks available on the server and the locks available to each workstation. The maximum amount of locks available on the server are 200000 locks. The maximum amount of locks per workstation should never be more than (Max locks for Server/Max Locks Per Workstation).

To change the lock settings on Novell, type from the network console: SET <RETURN>. Choose option 6 for Lock Settings. This will give you a brief description of the syntax to change the settings. To change the settings type SET Maximum Record Locks Per Workstation = Maximum Record Locks/Number of Work Stations. SET Maximum Record Locks = Highest available.

Novell troubleshooting check list

- 1 Files in CONFIG.SYS should be > 60.
- 2 Files in SHELL.CFG/NET.CFG should be > 60.
- 3 DEPATH must be set correctly to the directory where the DENETWRK.OVL file and PRINTERS.DAT is located, and DENAME must be unique for each user.
- 4 VMC files are set for 16M with 4.5x. (See section 5 of the Installation & New Features Guide)
- 5 Network Installation of DataEase says, Novell - Do Not Display Conflicting User Name.
- 6 Rights of DataEase System Files are Read Only and Shareable. Except DENETWRK.OVL and PRINTERS.DAT (Read/Write and Shareable). Ensure that the DataEase system files are **not** in the data directory.
- 7 No search maps to DataEase files (System & Data).
- 8 The original network creator

- of the data/system directories has not been deleted.
- 9 Page Locking not being used on Database if 4.5x.
Use DBCONVRT.EXE if < 4.53, or BUFSIZE.EXE if >=4.53.
Usage is : DBCONVRT Drive:Directory Database-Name Buffer-Size
DBCONVRT f:\data Personnel 0
BUFSIZE Drive:Directory Database-Name Buffer-Size
BUFSIZE f:\data Personnel 0
 - 10 Ensure TTS is not being used. (If it is make sure the DataEase System and Data Files are flagged Non-Transactional).
 - 11 If Locking conflicts occur increase the locks available to its maximum under Novell by typing SET at the console. Choosing Option 6 for locking strategies. Type Set {Parameter to Value} to increase the locks available to the server.
 - 12 Ensure no search mappings exist between the system and data directories.

Running DataEase under Windows 95

DataEase 4.53 won't run happily under Novell if you use the standard Windows 95 Microsoft Novell drivers and their IPX/SPX protocol. You get many "Form Inconsistent" messages. Compuserv has many experts verifying that this is a problem with Microsoft's WIN95 Novell Shells. If you install Novell ODI drivers instead, then DataEase 4.53 runs perfectly. Using these drivers however you must log into Novell using the autoexec.bat file before WIN95 gets under way.

We have successfully logged a WIN95 machine onto Novell (external to WIN95) and Lan Manager (inside WIN95) and then connected 4.53 and DataEase for Windows to ORACLE (Named Pipes), SQL Server (Named Pipes) and DB2/2 (netbios).

In groups with group totals in DataEase for Windows

If you are experiencing problems with the operation of "in groups with group-totals" this is because the "-" is missing from the object on screen.

To remedy this problem it is possible to edit the file DEMESSAG>MSG file found in the DataEase for Windows directory. The line number for this is 2781, or you could perform a search on the whole string. The line will read "in groups with group totals", just add the missing "-" between "group" and "totals", save and the problem is history.

Error on drive C

You may have received an error message from DataEase for Windows saying that there is an error on drive C:.

This message can occur when you are attempting to create, delete or build an application. DataEase for Windows uses the set path in the autoexec.bat and if you have a directory in this path that no longer exists, this error will happen. The solution is to check the path in the autoexec.bat and remove references to any directories no longer in use. For example, if you have SET PATH = C:\WINDOWS;C:\RUBBISH where RUBBISH no longer exists, you will receive this error message.

Exporting in groups in DataEase for Windows

The following tip is for those of you who have asked for help in exporting in groups.

```
for EMPLOYEES ;
list records
  DEPT in groups ;
  NAME ;
  SEX ;
```

```

AGE .
end .
Export to "c:\data.dat" .
.Form Header Group: Dept
@f[1,1]
.Items Group: Dept
  @f[2,1] @f[2,2] @f[2,3]
.Form Trailer Group: Dept
  The department just processed was @f[1,1]
.end

```

The output file is as follows :

Sales

```

Mary Jones Female 47
Sue Brown Female 46
Derek Cooke Male 49
Bill Johnson Male 46
Kim Hill Female 18
David Lancaster Male 40
Winston Holmes Male 20

The department just processed
was Sales

```

Tech

```

Simon Jones Male 25
Sarah Bullen Female 22
Alastair Smith Male 23
Pam Stone Female 24
John Herron Male 34

The department just processed
was Tech

```

To produce the same effect in an exported data file, the use of nested *for* is more appropriate for multiple groups than the *all* statement.

What you want where you want it

It is a feature of DataEase 4.53 that as long as you ask for something *somewhere* in a DQL list records section, then you can position the resulting item *anywhere* on your report format.

For example, you can write a DQL that lists selected staff using a data-entry value and also include that data-entry value in the title of the report by saying:

```

for EMPLOYEES
with DEPT = data-entry VALUE ;
list records
  Name ;
  Salary ;
  data-entry VALUE .

```

the default columnar layout will place the data-entry VALUE in the items area on the format, which isn't where you want it. Cut and paste it to the top of the format and use it in a heading, such as:

```
'List of staff working in XXXXXX'
```

where XXXXXX is the data-entry field.

DataEase 5 for Windows, however, is far more object-oriented so Field objects can only be parented by Record objects and record objects can only be parented by form objects. Liberal moving of fields on a DataEase 5 for Windows procedure format is not permissible, unless they are summarize fields, ie. the result of a statistical line of DQL such as:

```
Salary : item sum
```

You can obtain the same result as 4.53 by writing a DQL to generate a record object on the format where the data-entry Value could be dropped.

This *could* be done by wrapping the whole DQL inside an outer For loop that processes a file with only one record in it eg.

```
for ONERECFILE ;
list records
  data-entry VALUE .

  for EMPLOYEES
  with DEPT = data-entry VALUE
—(note no ';' in a nested for)
  list records
  Name ;
  Salary .
  end
end
```

and this will give a suitable format where the entire EMPLOYEES file is treated as a subform object and the entire 'background' is now a record object rather than a form object as required.

Alternatively you *could* just put a separate For loop above the EMPLOYEES loop, for example:

```
for ONERECFILE ;
list records
  data-entry VALUE .
end

for EMPLOYEES
with DEPT = data-entry VALUE ;
  list records
  Name ;
  Salary .
end
```

and this would work too, the format now has *two* separate form objects with their appropriate records. The second DQL raises an interesting point though, in that *something* has to parent the two parallel form objects, and DataEase 5 for Windows realises this so automatically inserts a parent form object called the 'Dql Scheduler' which is a system form with one record. Realizing this we come to the *best* solution which is very similar to the first DQL above but doesn't require a OneRecFile to be created, but allows DataEase 5 for Windows to use its own.

```
list records
  data-entry VALUE .

for EMPLOYEES
with DEPT = data-entry VALUE ;
  list records
  Name ;
  Salary .
end
```

In a similar way, if you want to use a temporary variable to calculate some value during a For loop and then print the answer once at the end, then you must *ask* for it in the DQL at the point *where* you want it to appear, eg.:

```
define temp "val" Number .

for EMPLOYEES ;
assign temp val := something you want to do .
list records
  Name ;
  Salary .
end

list records
temp val .
```

Moving on to the topic of grouping we come across a similar concept, as DataEase 5 for Windows will treat grouping by creating a Virtual File containing one record per group, and then treat the actual items file as a subform of the new file.

Thus, whereas in 4.53 it didn't matter *where* in DQL you asked for a field in groups, in DataEase 5 for Windows anything mentioned above the grouped field will appear in the outer 'group header/trailer' form object, and anything requested below the grouped field will appear in the inner 'items' form object. Thus whereas in DataEase 4.53 you could get away with:

```

for EMPLOYEES ;
  list records
    Name ;
    Dept in groups ;
    Salary .
end

```

in DataEase 5 for Windows this will put the Name field in the 'group header' equivalent, but the Salary in the 'items' equivalent. You should write:

```

for EMPLOYEES ;
  list records
    Dept in groups ;
    Name ;
    Salary .
end

```

Finally, if you want to see some memory variables in a 'group trailer' and see those values at the *end* of the group (DataEase 5 for Windows defaults to the beginning values) then you should put an output statement *after* the list records but *before* the end statement.

In summary, to obtain a value at a certain place on the report format, you must *ask* for it in the corresponding place in the DQL.

Go faster in Windows

Have you ever noticed that DataEase 5 for Windows sometimes seems to respond distinctly quicker than on other occasions? If so, it could be that you had another Windows package open simultaneously, for example Microsoft Word, Microsoft Write.

Well, ridiculous as it may sound, just having one of these packages loaded in the background can speed up some DataEase 5 for Windows operations by a factor of 200-300%!

To test this effect I timed four different DataEase 5 for Windows operations with and without Microsoft Write loaded. These were the results using a 486-DX 50MHz machine on a local application:

	Write not loaded	Write loaded
Run an export procedure	17s	5s
Open a form in user view	10s	4s
Open Document Print Options	8s	1s
Sign on to application	32s	33s

As you can see, the only unaffected operation is sign on. Vary the test conditions by:

- loading Write before and after DataEase 5 for Windows.
- running in standard and enhanced Windows mode.
- using and not using a 16MB swap file, none of which had any significant effect on the results.

In addition each test was run consecutively three times, to eliminate any cacheing effects and only the final timings were taken.

Now, why? Well, to be honest, we're not sure! Our best guess is that these other applications, when loaded, probably expose parts of the Windows API that are not in general use outside of Microsoft products and that some of DataEase 5 for Windows's API calls are then automatically rerouted through this more efficient code.

One easy way to make sure Write is always loaded when using DataEase 5 for Windows is to include Write as a 'run minimised' application in your Program Manager 'Startup' group.