

KIDS Distribution saved on Dec 22, 2007@10:34:45
ALPHA VERSION OF LISTER USING ANY SORT VALUES AND COMPUTED OUTPUT FIELDS
KIDS:GFTDICLGFT*0.1*1^

INSTALL NAME

GFTDICLGFT*0.1*1

"BLD",4742,0)

GFTDICLGFT*0.1*1^^0^3071222^n

"BLD",4742,1,0)

^^195^195^3071220^

"BLD",4742,1,1,0)

THIS WORK IS THE INTELLECTUAL PROPERTY OF GEORGE TIMSON.

"BLD",4742,1,2,0)

HE MAKES IT AVAILABLE UNDER THE TERMS OF "GPL". NO ONE MAY COPYRIGHT

"BLD",4742,1,3,0)

THESE ENHANCEMENTS.

"BLD",4742,1,4,0)

"BLD",4742,1,5,0)

Two enhancements are offered here to the FileMan "Lister". The Lister

"BLD",4742,1,6,0)

is an API call to LIST^DIC. It is documented at

"BLD",4742,1,7,0)

http://www.hardhats.org/fileman/pm/db_dic_1.htm.

"BLD",4742,1,8,0)

These new enhancements are designed to be completely backwards-compatible,

"BLD",4742,1,9,0)

which is to say that they should not change any existing expected behavior of the Filer.

"BLD",4742,1,10,0)

"BLD",4742,1,11,0)

"BLD",4742,1,12,0)

The first enhancement to LIST^DIC allows the caller to retrieve the values of COMPUTED EXPRESSIONS from the lister, instead of or along with the

"BLD",4742,1,13,0)

FIELD values that the Lister has always allowed to be specified in the third parameter of the call. This third parameter ("FIELDS") is a string

"BLD",4742,1,14,0)

separated by semicolons (;). Any "semicolon-piece" may now consist of a COMPUTED EXPRESSION that makes sense in the context of the File named in

"BLD",4742,1,15,0)

the first parameter.

"BLD",4742,1,16,0)

"BLD",4742,1,17,0)

Examples:

"BLD",4742,1,18,0)

"BLD",4742,1,19,0)

|NOWRAP|

"BLD",4742,1,20,0)

"BLD",4742,1,21,0)

1) D LIST^DIC(2,,"PHARMACY PATIENT:DIALYSIS PATIENT;.03","P",4,,,,,"GFT","ERR")

"BLD",4742,1,22,0)

|WRAP|

"BLD",4742,1,23,0)

This example shows how a computed expression like "PHARMACY PATIENT:DIALYSIS PATIENT"
"BLD",4742,1,24,0)
can be used to navigate over ("join") to a file different from the primary file,
File 2.

"BLD",4742,1,25,0)
The "GFT" array returned by this sample call might look like:

"BLD",4742,1,26,0)
|NOWRAP|
"BLD",4742,1,27,0)
GFT("DILIST",0)="4^4^1^"
"BLD",4742,1,28,0)
GFT("DILIST",0,"MAP")="IEN^IX(1)^C1^.03^WID(WRITE)"
"BLD",4742,1,29,0)
GFT("DILIST",1,0)="2255^ABARCA,SILVESTRE^YES^JUN 7,1988^
6-7-88 320340 Male "
"BLD",4742,1,30,0)
GFT("DILIST",2,0)="2432^ABELLARD,LINDA^^MAR 4,1955^
3-4-55 316005 Female"
"BLD",4742,1,31,0)
GFT("DILIST",3,0)="309^ABRAHAM,KARLA^^AUG 9,2000^
9-00 465217 Female"
"BLD",4742,1,32,0)
GFT("DILIST",4,0)="823^ABRAHIM,FAROUK^^DEC 4,1944^
12-4-44 579322 "
"BLD",4742,1,33,0)
|WRAP|
"BLD",4742,1,34,0)

8-

"BLD",4742,1,35,0)
In this example, the first of the 4 patients returned has a value for the "DIALYSIS PATIENT" field in File 55, and the other three do not.

"BLD",4742,1,36,0)
The computed field is named "C1" in
"BLD",4742,1,37,0)
the "map" because it is the first "field" in the string of "FIELDS" and is Computed on-the-fly. The same call without the "P" in the 4th ("FLAGS")
"BLD",4742,1,38,0)
parameter would return:
"BLD",4742,1,39,0)
|NOWRAP|
"BLD",4742,1,40,0)

"BLD",4742,1,41,0)
GFT("DILIST",0)="4^4^1^"
"BLD",4742,1,42,0)
GFT("DILIST",0,"MAP")=".03^C1"
"BLD",4742,1,43,0)
GFT("DILIST",1,1)="ABARCA,SILVESTRE"
"BLD",4742,1,44,0)
GFT("DILIST",1,2)="ABELLARD,LINDA"
"BLD",4742,1,45,0)
GFT("DILIST",1,3)="ABRAHAM,KARLA"
"BLD",4742,1,46,0)
GFT("DILIST",1,4)="ABRAHIM,FAROUK"
"BLD",4742,1,47,0)
GFT("DILIST",2,1)=2255
"BLD",4742,1,48,0)
GFT("DILIST",2,2)=2432

```

"BLD",4742,1,49,0)
GFT("DILIST",2,3)=309
"BLD",4742,1,50,0)
GFT("DILIST",2,4)=823
"BLD",4742,1,51,0)
GFT("DILIST","ID",1,.03)="JUN 7,1988"
"BLD",4742,1,52,0)
GFT("DILIST","ID",1,"C1",1)="YES"
"BLD",4742,1,53,0)
GFT("DILIST","ID",2,.03)="MAR 4,1955"
"BLD",4742,1,54,0)
GFT("DILIST","ID",2,"C1",1)=""
"BLD",4742,1,55,0)
GFT("DILIST","ID",3,.03)="AUG 9,2000"
"BLD",4742,1,56,0)
GFT("DILIST","ID",3,"C1",1)=""
"BLD",4742,1,57,0)
GFT("DILIST","ID",4,.03)="DEC 4,1944"
"BLD",4742,1,58,0)
GFT("DILIST","ID",4,"C1",1)=""
"BLD",4742,1,59,0)
GFT("DILIST","ID","WRITE",1,1)="           6-7-88      320340
Male "
"BLD",4742,1,60,0)
GFT("DILIST","ID","WRITE",2,1)="           3-4-55      316005
Female"
"BLD",4742,1,61,0)
GFT("DILIST","ID","WRITE",3,1)="           8-9-00      465217
Female"
"BLD",4742,1,62,0)
GFT("DILIST","ID","WRITE",4,1)="           12-4-44     579322 "
"BLD",4742,1,63,0)

```

```

"BLD",4742,1,64,0)
Note the "C1" subscripts in the "unpacked" array.
"BLD",4742,1,65,0)

```

```

"BLD",4742,1,66,0)
|WRAP|
"BLD",4742,1,67,0)
2) A Computed Expression can be MULTIPLE-VALUED, which is why there is an extra "1"
subscript just above depending from the "C1" subscripts.
"BLD",4742,1,68,0)
Any FIELD LABEL of a multiple-valued Field is a multiple-valued Computed
Expression. For example, in the PATIENT File, the string "ALIAS"
"BLD",4742,1,69,0)
is a multiple-valued expression that return all the Patient's ALIASes:
"BLD",4742,1,70,0)
|NOWRAP|
"BLD",4742,1,71,0)
    D LIST^DIC(2,,"ALIAS",,,,4) ZW ^TMP("DILIST",,$J)
"BLD",4742,1,72,0)
^TMP("DILIST",3348,0)="4^4^1^"
"BLD",4742,1,73,0)
^TMP("DILIST",3348,0,"MAP")="C1"
"BLD",4742,1,74,0)
^TMP("DILIST",3348,1,1)="ABARCA,SAM"
"BLD",4742,1,75,0)
^TMP("DILIST",3348,1,2)="ABARCA,SILVESTRE"

```

```

"BLD",4742,1,76,0)
^TMP("DILIST",3348,1,3)="ABELLARD,LINDA"
"BLD",4742,1,77,0)
^TMP("DILIST",3348,1,4)="ABRAHAM,KARLA"
"BLD",4742,1,78,0)
^TMP("DILIST",3348,2,1)=2255
"BLD",4742,1,79,0)
^TMP("DILIST",3348,2,2)=2255
"BLD",4742,1,80,0)
^TMP("DILIST",3348,2,3)=2432
"BLD",4742,1,81,0)
^TMP("DILIST",3348,2,4)=309
"BLD",4742,1,82,0)
^TMP("DILIST",3348,"ID",1,"C1",1)="ABARCA,SAM"
"BLD",4742,1,83,0)
^TMP("DILIST",3348,"ID",1,"C1",2)="CONSTATINE,SILVESTRE"
"BLD",4742,1,84,0)
^TMP("DILIST",3348,"ID",2,"C1",1)="ABARCA,SAM"
"BLD",4742,1,85,0)
^TMP("DILIST",3348,"ID",2,"C1",2)="CONSTATINE,SILVESTRE"
"BLD",4742,1,86,0)
^TMP("DILIST",3348,"ID",3,"C1",1)="FOX,LINDA"
"BLD",4742,1,87,0)
^TMP("DILIST",3348,"ID",4,"C1",1)="
"BLD",4742,1,88,0)
^TMP("DILIST",3348,"ID","WRITE",1,1)=" 6-7-88
320340 Male "
"BLD",4742,1,89,0)
^TMP("DILIST",3348,"ID","WRITE",2,1)=" 6-7-88
320340 Male "
"BLD",4742,1,90,0)
^TMP("DILIST",3348,"ID","WRITE",3,1)=" 3-4-55
316005 Female"
"BLD",4742,1,91,0)
^TMP("DILIST",3348,"ID","WRITE",4,1)=" 8-9-00
465217 Female"
"BLD",4742,1,92,0)
|WRAP|
"BLD",4742,1,93,0)
This shows patient `2255 with two ALIASes, and patient `2432 with one ALIAS.
"BLD",4742,1,94,0)

"BLD",4742,1,95,0)

"BLD",4742,1,96,0)
3) "ALIAS:ALIAS SSN" is a slightly more complicated multiple-valued
"BLD",4742,1,97,0)
Computed Expression that retrieves the "ASIAS SSN" fields
"BLD",4742,1,98,0)
from each of the "ALIAS" multiples:
"BLD",4742,1,99,0)
|NOWRAP|
"BLD",4742,1,100,0)
D LIST^DIC(2,, "ALIAS;ALIAS:ALIAS SSN;.03;@", "P", 4) ZW ^TMP("DILIST", $J)
"BLD",4742,1,101,0)
|NOWRAP|
"BLD",4742,1,102,0)
^TMP("DILIST",3348,0)="4^4^1^"
"BLD",4742,1,103,0)

```

```

^TMP("DILIST",3348,0,"MAP")="IEN^C1^C2^.03"
"BLD",4742,1,104,0)
^TMP("DILIST",3348,1,0)="2255^ABARCA,SAM~CONSTATINE,SILVESTRE^399399333~499499499^J
UN 7,1988"
"BLD",4742,1,105,0)
^TMP("DILIST",3348,2,0)="2255^ABARCA,SAM~CONSTATINE,SILVESTRE^399399333~499499499^J
UN 7,1988"
"BLD",4742,1,106,0)
^TMP("DILIST",3348,3,0)="2432^FOX,LINDA^999222222^MAR 4,1955"
"BLD",4742,1,107,0)
^TMP("DILIST",3348,4,0)="309^^^AUG 9,2000"
"BLD",4742,1,108,0)

"BLD",4742,1,109,0)

"BLD",4742,1,110,0)

"BLD",4742,1,111,0)

"BLD",4742,1,112,0)

"BLD",4742,1,113,0)

"BLD",4742,1,114,0)
The second enhancement to LIST^DIC allows for sorting
"BLD",4742,1,115,0)
output from any file in virtually any order. The enhancement opens up
"BLD",4742,1,116,0)
the use of the eighth parameter of the LIST^DIC call.
"BLD",4742,1,117,0)
The eighth parameter ("INDEX") can now be the [bracketed] name of a SORT TEMPLATE,
"BLD",4742,1,118,0)
or the (name or) number of a field, or a COMPUTED EXPRESSION. See the
"BLD",4742,1,119,0)
following examples.
"BLD",4742,1,120,0)

"BLD",4742,1,121,0)
To make this new functionality work, you will have to add an "X" to the
"BLD",4742,1,122,0)
FLAGS parameter, the fourth parameter in the call. Also, the new functionality
only works at the File level;
"BLD",4742,1,123,0)
the first parameter must be a file number and the second parameter must be empty.
"BLD",4742,1,124,0)

"BLD",4742,1,125,0)

"BLD",4742,1,126,0)

"BLD",4742,1,127,0)
Examples:
"BLD",4742,1,128,0)

"BLD",4742,1,129,0)
1) Caller no longer needs to know whether the field he wants to sort by
"BLD",4742,1,130,0)
has a cross-reference on it. If it has an index, that index will be used.
Otherwise

```

"BLD",4742,1,131,0)
the sort will be done on-the-fly. Here is how to sort users (File 200)
"BLD",4742,1,132,0)
by LAST OPTION ACCESSED
"BLD",4742,1,133,0)
(Field 202.1):
"BLD",4742,1,134,0)
D LIST^DIC(200,,.01,"X",,,,202.1)
"BLD",4742,1,135,0)

"BLD",4742,1,136,0)
2) Boolean-valued Computed Expressions are very useful for FileMan
"BLD",4742,1,137,0)
sorts. They can simulate
"BLD",4742,1,138,0)
the functionality of FileMan's "SEARCH" Option.
"BLD",4742,1,139,0)
Here is how to retrieve only DRUGs (File 50) with a MESSAGE (Field 101)
"BLD",4742,1,140,0)
containing
"BLD",4742,1,141,0)
"HOME MED", and a GENERIC NAME field containing "VITAMIN":
"BLD",4742,1,142,0)

"BLD",4742,1,143,0)
D LIST^DIC(50,, ".01;101", "XP",,,, "MESSAGE[\"HOME MED\"&(#.01[\"VITAMIN\"])\")
"BLD",4742,1,144,0)

"BLD",4742,1,145,0)

"BLD",4742,1,146,0)
3) "INDEX" can also be the bracketed name of a SORT TEMPLATE. The
"BLD",4742,1,147,0)
Template can
"BLD",4742,1,148,0)
be of either type, sort or search. The SORT algorithm, however, must not
"BLD",4742,1,149,0)
specify
"BLD",4742,1,150,0)
"asking the user" for "From" or "To" fields. Here is how to retrieve the
"BLD",4742,1,151,0)
first
"BLD",4742,1,152,0)
nine LAB TEST values (File 60), sorted by the "LR TEST DICTIONARY"
"BLD",4742,1,153,0)
Template:
"BLD",4742,1,154,0)

"BLD",4742,1,155,0)
D LIST^DIC(60,,.01,"XP",9,, "[LR TEST DICTIONARY]")
"BLD",4742,1,156,0)

"BLD",4742,1,157,0)

"BLD",4742,1,158,0)
4) The eighth parameter is in fact evaluated as the "BY" parameter of
"BLD",4742,1,159,0)
a call to EN1^DIP. Thus, it can contain a string of "answers", separated by
commas.

"BLD",4742,1,160,0)

Here is how to sort patients (File 2) by year of birth, and within that

"BLD",4742,1,161,0)

alphabetically:

"BLD",4742,1,162,0)

"BLD",4742,1,163,0)

D LIST^DIC(2,,.01,"XP",,,,,,"YEAR(DOB),NAME")

"BLD",4742,1,164,0)

"BLD",4742,1,165,0)

"BLD",4742,1,166,0)

5) The "BY" answers can navigate downward into multiples. Here is how to

"BLD",4742,1,167,0)

sort OPTIONS (File 19) by the Menu Options they call:

"BLD",4742,1,168,0)

"BLD",4742,1,169,0)

"BLD",4742,1,170,0)

D LIST^DIC(19,,.01,"XP",,,,,,"10,.01")

"BLD",4742,1,171,0)

"BLD",4742,1,172,0)

"BLD",4742,1,173,0)

"BLD",4742,1,174,0)

"BLD",4742,1,175,0)

So, combining the two functionalities introduced here, we write a call that

"BLD",4742,1,176,0)

retrieves from the OPTION File all menu options, and for each the number of options

"BLD",4742,1,177,0)

depending from them in the MENU multiple:

"BLD",4742,1,178,0)

"BLD",4742,1,179,0)

D LIST^DIC(19,,"COUNT(MENU)",,"XP",,,,,,"TYPE=""menu"")

"BLD",4742,1,180,0)

"BLD",4742,1,181,0)

The output from this call would be:

"BLD",4742,1,182,0)

|NOWRAP|

"BLD",4742,1,183,0)

^TMP("DILIST",1584,0)=1251^*^0^

"BLD",4742,1,184,0)

^TMP("DILIST",1584,0,"MAP")=IEN^IX(1)^.01^1C

"BLD",4742,1,185,0)

^TMP("DILIST",1584,1,0)=2^1^XUCOMMAND^8

"BLD",4742,1,186,0)

^TMP("DILIST",1584,2,0)=8^1^XUMAIN^14

"BLD",4742,1,187,0)

^TMP("DILIST",1584,3,0)=9^1^EVE^12

"BLD",4742,1,188,0)

^TMP("DILIST",1584,4,0)=10^1^XUPROG^6

"BLD",4742,1,189,0)

```

^TMP("DILIST",1584,5,0)=11^1^XUTIO^17
"BLD",4742,1,190,0)
^TMP("DILIST",1584,6,0)=18^1^XUSER^14
"BLD",4742,1,191,0)
^TMP("DILIST",1584,7,0)=19^1^XUCORE^3
"BLD",4742,1,192,0)
^TMP("DILIST",1584,8,0)=36^1^XU-SPL-MENU^6
"BLD",4742,1,193,0)
^TMP("DILIST",1584,9,0)=43^1^XQHELP-MENU^7
"BLD",4742,1,194,0)
    . . . . .
"BLD",4742,1,195,0)
    and so on
"BLD",4742,4,0)
^9.64PA^^
"BLD",4742,6.3)
32
"BLD",4742,"ABNS",0)
^9.66A
"BLD",4742,"KRN",0)
^9.67PA^8989.52^19
"BLD",4742,"KRN",.4,0)
.4
"BLD",4742,"KRN",.401,0)
.401
"BLD",4742,"KRN",.402,0)
.402
"BLD",4742,"KRN",.403,0)
.403
"BLD",4742,"KRN",.5,0)
.5
"BLD",4742,"KRN",.84,0)
.84
"BLD",4742,"KRN",3.6,0)
3.6
"BLD",4742,"KRN",3.8,0)
3.8
"BLD",4742,"KRN",9.2,0)
9.2
"BLD",4742,"KRN",9.8,0)
9.8
"BLD",4742,"KRN",9.8,"NM",0)
^9.68A^6^5
"BLD",4742,"KRN",9.8,"NM",2,0)
DICL^^0^B13720609
"BLD",4742,"KRN",9.8,"NM",3,0)
DICLGFT^^0^B13130854
"BLD",4742,"KRN",9.8,"NM",4,0)
DIP0^^0^B17561996
"BLD",4742,"KRN",9.8,"NM",5,0)
DICU1^^0^B37653223
"BLD",4742,"KRN",9.8,"NM",6,0)
DICU2^^0^B30429119
"BLD",4742,"KRN",9.8,"NM","B","DICL",2)

"BLD",4742,"KRN",9.8,"NM","B","DICLGFT",3)

"BLD",4742,"KRN",9.8,"NM","B","DICU1",5)

```


"BLD",4742,"KRN",9.8,"NM","B","DICU2",6)
"BLD",4742,"KRN",9.8,"NM","B","DIP0",4)
"BLD",4742,"KRN",19,0)
19
"BLD",4742,"KRN",19.1,0)
19.1
"BLD",4742,"KRN",101,0)
101
"BLD",4742,"KRN",409.61,0)
409.61
"BLD",4742,"KRN",771,0)
771
"BLD",4742,"KRN",870,0)
870
"BLD",4742,"KRN",8989.51,0)
8989.51
"BLD",4742,"KRN",8989.52,0)
8989.52
"BLD",4742,"KRN",8994,0)
8994
"BLD",4742,"KRN","B",.4,.4)
"BLD",4742,"KRN","B",.401,.401)
"BLD",4742,"KRN","B",.402,.402)
"BLD",4742,"KRN","B",.403,.403)
"BLD",4742,"KRN","B",.5,.5)
"BLD",4742,"KRN","B",.84,.84)
"BLD",4742,"KRN","B",3.6,3.6)
"BLD",4742,"KRN","B",3.8,3.8)
"BLD",4742,"KRN","B",9.2,9.2)
"BLD",4742,"KRN","B",9.8,9.8)
"BLD",4742,"KRN","B",19,19)
"BLD",4742,"KRN","B",19.1,19.1)
"BLD",4742,"KRN","B",101,101)
"BLD",4742,"KRN","B",409.61,409.61)
"BLD",4742,"KRN","B",771,771)
"BLD",4742,"KRN","B",870,870)
"BLD",4742,"KRN","B",8989.51,8989.51)
"BLD",4742,"KRN","B",8989.52,8989.52)
"BLD",4742,"KRN","B",8994,8994)

```
"BLD",4742,"MSC")
D:\R\GFTDICLGFT.KID
"BLD",4742,"MSCOM")
ALPHA VERSION OF LISTER USING ANY SORT VALUES AND COMPUTED OUTPUT FIELDS
"MBREQ")
0
"QUES","XPF1",0)
Y
"QUES","XPF1","??")
^D REP^XPDH
"QUES","XPF1","A")
Shall I write over your |FLAG| File
"QUES","XPF1","B")
YES
"QUES","XPF1","M")
D XPF1^XPDIQ
"QUES","XPF2",0)
Y
"QUES","XPF2","??")
^D DTA^XPDH
"QUES","XPF2","A")
Want my data |FLAG| yours
"QUES","XPF2","B")
YES
"QUES","XPF2","M")
D XPF2^XPDIQ
"QUES","XPI1",0)
YO
"QUES","XPI1","??")
^D INHIBIT^XPDH
"QUES","XPI1","A")
Want KIDS to INHIBIT LOGONS during the install
"QUES","XPI1","B")
NO
"QUES","XPI1","M")
D XPI1^XPDIQ
"QUES","XPM1",0)
PO^VA(200,:EM
"QUES","XPM1","??")
^D MG^XPDH
"QUES","XPM1","A")
Enter the Coordinator for Mail Group '|FLAG|'
"QUES","XPM1","B")

"QUES","XPM1","M")
D XPM1^XPDIQ
"QUES","XPO1",0)
Y
"QUES","XPO1","??")
^D MENU^XPDH
"QUES","XPO1","A")
Want KIDS to Rebuild Menu Trees Upon Completion of Install
"QUES","XPO1","B")
NO
"QUES","XPO1","M")
D XPO1^XPDIQ
"QUES","XPZ1",0)
Y
```

```

"QUES", "XPZ1", "??")
^D OPT^XPDH
"QUES", "XPZ1", "A")
Want to DISABLE Scheduled Options, Menu Options, and Protocols
"QUES", "XPZ1", "B")
NO
"QUES", "XPZ1", "M")
D XPZ1^XPDIQ
"QUES", "XPZ2", 0)
Y
"QUES", "XPZ2", "??")
^D RTN^XPDH
"QUES", "XPZ2", "A")
Want to MOVE routines to other CPUs
"QUES", "XPZ2", "B")
NO
"QUES", "XPZ2", "M")
D XPZ2^XPDIQ
"RTN")
5
"RTN", "DICL")
0^2^B13720609
"RTN", "DICL", 1, 0)
DICL ;SEA/TOAD,SF/TKW-VA FileMan: Lookup: Lister ;22DEC2007
"RTN", "DICL", 2, 0)
; ;22.0;VA FileMan;**GFT**;Mar 30, 1999
"RTN", "DICL", 3, 0)
;
"RTN", "DICL", 4, 0)
;
"RTN", "DICL", 5, 0)
LIST (DIFILE, DIFIEN, DIFIELDS, DIFLAGS, DINUMBER, DIFROM, DIPART, DINDEX, DISCREEN, DIWRITE,
DILIST, DIMSGA, DIC) ;
"RTN", "DICL", 6, 0)
; ENTRY POINT--return a list of entries from a file
"RTN", "DICL", 7, 0)
; (Note: DIC parameter only passed if called from ^DICQ)
"RTN", "DICL", 8, 0)
;
"RTN", "DICL", 9, 0)
IN ; Entry point from LIST^DIC
"RTN", "DICL", 10, 0)
I '$D(DIQUIET) N DIQUIET S DIQUIET=1
"RTN", "DICL", 11, 0)
I '$D(DIFM) N DIFM S DIFM=1 D INIZE^DIEFU
"RTN", "DICL", 12, 0)
N DICLERR S DICLERR=$G(DIERR) K DIERR
"RTN", "DICL", 13, 0)
;
"RTN", "DICL", 14, 0)
INPUT ; Validate input parameters
"RTN", "DICL", 15, 0)
N DIERN, DIPE, DIDENT
"RTN", "DICL", 16, 0)
S DIFLAGS=$G(DIFLAGS)
"RTN", "DICL", 17, 0)
I DIFLAGS["I", DIFLAGS["Q" S DIFLAGS=DIFLAGS_"Q"
"RTN", "DICL", 18, 0)
S DIFIELDS=$G(DIFIELDS)

```

```

"RTN", "DICL", 19, 0)
  I DIFIELDS'["-IX" D
"RTN", "DICL", 20, 0)
  . N DID S DID=";"_DIFIELDS_" ;"
"RTN", "DICL", 21, 0)
  . I (DID["@!(DIFLAGS["I")), DID'[";IX;" , DID'[";IXE", DID'[";IXIE" Q
"RTN", "DICL", 22, 0)
  . S DIDENT(-5)=1 Q
"RTN", "DICL", 23, 0)
  S DINUMBER=$G(DINUMBER) I DINUMBER="" S DINUMBER="*"
"RTN", "DICL", 24, 0)
  I '$D(DIPART(1)) S DIPART(1)=$G(DIPART)
"RTN", "DICL", 25, 0)
  I '$D(DIFROM(1)) S DIFROM(1)=$G(DIFROM)
"RTN", "DICL", 26, 0)
  I $O(DIFROM(1)) D
"RTN", "DICL", 27, 0)
  . N E S E=9999 F S E=$O(DIFROM(E),-1) Q:'E Q:DIFROM(E)]""
"RTN", "DICL", 28, 0)
  . I E N I F I=1:1:E I DIFROM(I)="" D BLD^DIALOG(202,"FROM values"),OUT Q
"RTN", "DICL", 29, 0)
  . Q
"RTN", "DICL", 30, 0)
  S DIFROM("IEN")=$G(DIFROM("IEN"))
"RTN", "DICL", 31, 0)
  S DINDEX("WAY")=1 I DIFLAGS["B" S DINDEX("WAY")=-1
"RTN", "DICL", 32, 0)
  S DINDEX=$G(DINDEX)
"RTN", "DICL", 33, 0)
  I '$D(DISCREEN("S")) S DISCREEN("S")=$G(DISCREEN) D:DISCREEN("S")]""
"RTN", "DICL", 34, 0)
  .N X S X=DISCREEN D ^DIM I '$D(X) D BLD^DIALOG(202,"SCREEN") ;**GFT CHECK FOR
GOOD MUMPS CODE
"RTN", "DICL", 35, 0)
  S DIWRITE=$G(DIWRITE)
"RTN", "DICL", 36, 0)
  ;
"RTN", "DICL", 37, 0)
OUTPUT ; Establish output file name, starting output subscript no.
"RTN", "DICL", 38, 0)
  I $G(DILIST)="" S DILIST="^TMP("DILIST", $J) "
"RTN", "DICL", 39, 0)
  E I DIFLAGS'["h" D I $G(DIERR) D OUT Q
"RTN", "DICL", 40, 0)
  . I DILIST'?1"1U.7UN.ANP,DILIST'?1"1"1U.7UN.ANP D Q
"RTN", "DICL", 41, 0)
  . . D BLD^DIALOG(202,"target array")
"RTN", "DICL", 42, 0)
  . S DILIST=$NA(@DILIST@("DILIST"))
"RTN", "DICL", 43, 0)
  . Q
"RTN", "DICL", 44, 0)
  K @DILIST
"RTN", "DICL", 45, 0)
  S DILIST("ORDER")=$S(DINDEX("WAY")=1:0,1:DINUMBER+1)
"RTN", "DICL", 46, 0)
  I DINUMBER="*",DINDEX("WAY")=-1 D
"RTN", "DICL", 47, 0)
  . S DINDEX("WAY")=1,DINDEX("WAY","REVERSE")=1

```

```

"RTN", "DICL", 48, 0)
. S DILIST("ORDER")=0
"RTN", "DICL", 49, 0)
. Q
"RTN", "DICL", 50, 0)
;
"RTN", "DICL", 51, 0)
FILE ; Validate file number and IENS.
"RTN", "DICL", 52, 0)
I DIFLAGS["h" D FILE^DICUF(.DIFILE, .DIFIEN, DIFLAGS)
"RTN", "DICL", 53, 0)
I $G(DIERR) S DIFROM="", DIFROM("IEN")="" D OUT Q
"RTN", "DICL", 54, 0)
D SCREEN^DICUF(DIFLAGS, .DIFILE, .DISCREEN)
"RTN", "DICL", 55, 0)
;
"RTN", "DICL", 56, 0)
CHECKS ;
"RTN", "DICL", 57, 0)
I $TR(DIFLAGS, "BIKMPQSufhuX")="" S DIERN=301, DIPE(1)=DIFLAGS D ERROUT Q ;GFT:
"X" added
"RTN", "DICL", 58, 0)
S DIFLAGS=DIFLAGS_3
"RTN", "DICL", 59, 0)
I DINUMBER!=""*, DINUMBER<1!(DINUMBER\1'=DINUMBER) D Q
"RTN", "DICL", 60, 0)
. S DIERN=202, DIPE(1)="Number" D ERROUT
"RTN", "DICL", 61, 0)
;
"RTN", "DICL", 62, 0)
IXANDID ; Gather information about index and field data to be returned.
"RTN", "DICL", 63, 0)
N DIOUT S DIOUT=0
"RTN", "DICL", 64, 0)
IXNAME ; Set default index name if null.
"RTN", "DICL", 65, 0)
N DIGFT, DIGFTEMP
"RTN", "DICL", 66, 0)
I DIFLAGS["X" D DICL^DICLGFT G BADQ ;NOTE: A CROSS-REF MUST BE 1U.UN (IX^DICE); AN
INDEX MUST BE 1A.AN
"RTN", "DICL", 67, 0)
I DINDEX!="#", DINDEX'?1U.UNP S DINDEX=$$DINDEX(DIFILE, DIFLAGS)
"RTN", "DICL", 68, 0)
D
INDEX^DICUIX(.DIFILE, DIFLAGS, .DINDEX, .DIFROM, .DIPART, DINUMBER, .DISCREEN, DILIST, .DIO
UT)
"RTN", "DICL", 69, 0)
BADQ I DIOUT!($G(DIERR)) D KTMPIX^DICL1 Q
"RTN", "DICL", 70, 0)
I $D(DISCREEN("V")) D VPDATA^DICUF(.DINDEX, .DISCREEN)
"RTN", "DICL", 71, 0)
I $O(DIFROM(DINDEX("#")+1))!(DINDEX!="#"&($O(DIPART(DINDEX("#"))))) D
BLD^DIALOG(202, "Index"), KTMPIX^DICL1 Q
"RTN", "DICL", 72, 0)
D IDENT^DICU1(DIFLAGS, .DIFILE, DIFIELDS, DIWRITE, .DIDENT, .DINDEX)
"RTN", "DICL", 73, 0)
I $G(DIERR) D KTMPIX^DICL1 Q
"RTN", "DICL", 74, 0)
;

```

```

"RTN","DICL",75,0)
BRANCH ; Continue on to actual search.
"RTN","DICL",76,0)
  D PREP^DICL1
"RTN","DICL",77,0)
  I $G(DIGFTEMP) ["^" K @DIGFTEMP ;**
"RTN","DICL",78,0)
  Q
"RTN","DICL",79,0)
  ;
"RTN","DICL",80,0)
DINDEX(DIFILE,DIFLAGS) ; Set DINDEX to index name for KEY.
"RTN","DICL",81,0)
  N I,X S X=""
"RTN","DICL",82,0)
  I $G(DIFLAGS) ["K" D
"RTN","DICL",83,0)
  . S I=$O(^DD("KEY","AP",DIFILE,"P",0)) Q:'I
"RTN","DICL",84,0)
  . S X=$P($G(^DD("IX",+$P($G(^DD("KEY",I,0)),U,4),0)),U,2) Q
"RTN","DICL",85,0)
  Q:X?1U.UNP X
"RTN","DICL",86,0)
  Q "B"
"RTN","DICL",87,0)
  ;
"RTN","DICL",88,0)
ERROUT D BLD^DIALOG(DIERN,.DIPE,.DIPE),OUT Q
"RTN","DICL",89,0)
  ;
"RTN","DICL",90,0)
OUT I DICLERR=""!$G(DIERR) D
"RTN","DICL",91,0)
  . S DIERR=$G(DIERR)+DICLERR_U_($P($G(DIERR),U,2)+$P(DICLERR,U,2))
"RTN","DICL",92,0)
  I $G(DIMSGA)'="" D CALLOUT^DIEFU(DIMSGA)
"RTN","DICL",93,0)
  Q
"RTN","DICL",94,0)
  ;
"RTN","DICL",95,0)
  ; Possible messages returned
"RTN","DICL",96,0)
  ; 202 The input parameter that identifies the |1
"RTN","DICL",97,0)
  ; 301 The passed flag(s) '|1|' are unknown or in
"RTN","DICL",98,0)
  ;
"RTN","DICLGFT")
0^3^B13130854
"RTN","DICLGFT",1,0)
DICLGFT ;GFT-- USE ANY SORT VALUES FOR LISTER;22DEC2007
"RTN","DICLGFT",2,0)
  ;;22.0;VA FileMan;**GFT**;Mar 30, 1999
"RTN","DICLGFT",3,0)
  ;
"RTN","DICLGFT",4,0)
DICL ;FROM ^DICL RETURN TO BADQ^DICL WITH DIERR DEFINED OR ELSE WE HAVE DINDEX SET
UP CORRECTLY TO GET SORTED OUTPUT

```

```

"RTN", "DICLGFT", 5, 0)
  N X, I, DITEMP, DICLGFT
"RTN", "DICLGFT", 6, 0)
  D TMPB^DICUIX1(.DITEMP, DIFILE) ;SETS DITEMP=something like "^TMP("DICLB", 2, 3188"
"RTN", "DICLGFT", 7, 0)
  S DIGFTEMP=DITEMP_" " ;so we can remember to KILL the temporary array
"RTN", "DICLGFT", 8, 0)
  S X=$$SORT(DIFILE, DINDEX, DIGFTEMP) I X D BLD^DIALOG(-X, $P(X, U, 2)) K @DIGFTEMP Q
"RTN", "DICLGFT", 9, 0)
  ;now we have the answers in @DITEMP
"RTN", "DICLGFT", 10, 0)
  ;D COMMON1^DICUIX2 probably need some of this
"RTN", "DICLGFT", 11, 0)
  S DICLGFT=$P(X, U, 2), DINDEX("#")=DICLGFT ;NUMBER OF LEVELS IN OUR SORT
"RTN", "DICLGFT", 12, 0)
  S DINDEX("IXTYPE")="["
"RTN", "DICLGFT", 13, 0)
  F I=1:1:DICLGFT+1 S DINDEX(I, "WAY")=DINDEX("WAY")
"RTN", "DICLGFT", 14, 0)
  S DINDEX(1, "ROOT")=DITEMP_" ", X=DITEMP
"RTN", "DICLGFT", 15, 0)
  F I=1:1:DICLGFT S X=X_"", DINDEX("_ (I) _")", DINDEX(I+1, "ROOT")=X_" "
"RTN", "DICLGFT", 16, 0)
  F I=1:1:DICLGFT S DINDEX(I, "FILE")=DIFILE
"RTN", "DICLGFT", 17, 0)
  ;S DINDEX(1, "GET")="DIVAL=ZZZ" ;?????????
"RTN", "DICLGFT", 18, 0)
  S DINDEX(1, "TYPE")="[" , DINDEX("AT")=1
"RTN", "DICLGFT", 19, 0)
  F I=1:1:DICLGFT S DINDEX(I)=$G(DIFROM(I)) ;FROM VALUES
"RTN", "DICLGFT", 20, 0)
  S DINDEX(DICLGFT+1)=0
"RTN", "DICLGFT", 21, 0)
  Q
"RTN", "DICLGFT", 22, 0)
  ;
"RTN", "DICLGFT", 23, 0)
  ;
"RTN", "DICLGFT", 24, 0)
  ;
"RTN", "DICLGFT", 25, 0)
  ;
"RTN", "DICLGFT", 26, 0)
SORT(DIFILE, BY, DICLARAY) ;SORT FILE BY TEMPLATE OR FIELD(S), AND PUT RESULTS IN
'DICLARAY' ARRAY
"RTN", "DICLGFT", 27, 0)
  ;EXTRINSIC FUNCTION RETURNS
"RTN", "DICLGFT", 28, 0)
  ;"OK^n" IF SUCCESSFUL, where 'n' is number of levels
"RTN", "DICLGFT", 29, 0)
  ;
"RTN", "DICLGFT", 30, 0)
  N
L, DIC, FLDS, FR, TO, DHD, DIASKHD, DIPCRIT, PG, DHIT, DIOEND, DIOBEG, DCOPIES, IOP, DQTIME, DIS, D
ISTOP, DISPAR, DIFIXPTH, DISH, DISO
"RTN", "DICLGFT", 31, 0)
  N D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13
"RTN", "DICLGFT", 32, 0)
  N DIQUIET, DISUPNO S DIQUIET=1, DISUPNO=1

```

```

"RTN", "DICLGFT", 33, 0)
N X ;, I F L=1:1:$L(BY, "", "") S X=$P(BY, "", L) D S $P(BY, "", L)=X
"RTN", "DICLGFT", 34, 0)
; .F I="+", "#", "!" S:$E(X)=I X=$E(X, 2, 999) S:$E(X, $L(X))=I X=$E(X, 1, $L(X)-1)
"RTN", "DICLGFT", 35, 0)
N DIOSL S DIOSL=99999999
"RTN", "DICLGFT", 36, 0)
N DIFIXPT S DIFIXPT=1, DHD="@@" ;TRICK TO AVOID DEVICE SELECTION!
"RTN", "DICLGFT", 37, 0)
S DIOBEG="K ^UTILITY($J, "H") S DISH=1, IOT="",$X=0,$Y=0" ;TRICK TO SUPPRESS
SUBHEADERS IN SORT TEMPLATE, WHETHER OR NOT THERE IS A PRE-SORT
"RTN", "DICLGFT", 38, 0)
I '$D(^DIC(DIFILE, 0, "GL")) Q "401^"_DIFILE
"RTN", "DICLGFT", 39, 0)
S DIC=^("GL")
"RTN", "DICLGFT", 40, 0)
;
"RTN", "DICLGFT", 41, 0)
N DICLGFT S DICLGFT=1
"RTN", "DICLGFT", 42, 0)
;
"RTN", "DICLGFT", 43, 0)
I $G(BY)=" Q "-202^SORT VALUE"
"RTN", "DICLGFT", 44, 0)
DIBT S X=0 I $G(BY)?1[".E1"] S X=$O(^DIBT("F"_DIFILE, $TR(BY, "[ ]"), 0)) I
X&$O(^X))!'X Q "-202^SORT TEMPLATE '"_BY_"'" ;MUST HAVE EXACTLY ONE TEMPLATE OF
THAT NAME
"RTN", "DICLGFT", 45, 0)
I X S L=$O(^DIBT(X, 2, 999), -1) I L S DICLGFT=L D G A:$D(X) Q "-202^SORT TEMPLATE
'"_BY_"'" ;NUMBER OF LEVELS
"RTN", "DICLGFT", 46, 0)
.F L=1:1:L I $G(^DIBT(X, 2, L, "ASK")) K X Q
"RTN", "DICLGFT", 47, 0)
; I X, 'L S DICLGFT0=1
"RTN", "DICLGFT", 48, 0)
;
"RTN", "DICLGFT", 49, 0)
FIELD N DICLGFTX, DD S
DICLGFTX=$G(BY), DICLGFT=$L(DICLGFTX, "", ), FR=",,,,,,", TO=",,,,,,", ;SORT BY
FIELD
"RTN", "DICLGFT", 50, 0)
S DD=DIFILE F S FLDS=$P(DICLGFTX, "", ), DICLGFTX=$P(DICLGFTX, "", 2) Q:FLDS="" D
"RTN", "DICLGFT", 51, 0)
.S FLDS=$P(FLDS, ";") I $D(^DD(DD, FLDS, 0))
"RTN", "DICLGFT", 52, 0)
.E S FLDS=$O(^DD(DD, "B", FLDS, 0)) Q:'FLDS
"RTN", "DICLGFT", 53, 0)
.S L=+$P(^DD(DD, FLDS, 0), U, 2) I L S DD=L, DICLGFT=DICLGFT-1 ;GOING DOWN INTO A
MULTIPLE, SO LEVEL OF SORT IS 1 LESS THAN WE THOT
"RTN", "DICLGFT", 54, 0)
;
"RTN", "DICLGFT", 55, 0)
A I DICLARAY["^", DICLARAY["(" Q "-202^BAD ARRAY "_DICLARAY
"RTN", "DICLGFT", 56, 0)
K ^UTILITY("DICLGFT", $J), @DICLARAY
"RTN", "DICLGFT", 57, 0)
;
"RTN", "DICLGFT", 58, 0)
DHIT S DHIT="" ;I $G(DICLGFT0) S DHIT="1," ;IF IT IS JUST A LIST

```



```

"RTN", "DICLGFT", 59, 0)
  F L=1:1:DICLGFT S X="DIOO" _L, DHIT="$S ($G (" _X_ " ) "" "" : " _X_ ", 1:1) , " _DHIT
"RTN", "DICLGFT", 60, 0)
  S DHIT=" (" _DHIT_ "D0)", DHIT="S @DICLARAY@ " _DHIT_ "="" "" "" ; CREATES SOMETHING LIKE
DHIT = S @DICLARAY@ ($S ($G (DIOO2) ) "" : DIOO2, 1:1) , $S ($G (DIOO1) ) "" : DIOO1, 1:1) , D0)=""
"RTN", "DICLGFT", 61, 0)
;
"RTN", "DICLGFT", 62, 0)
  S L=0, FLDS="X "" "QUIT"" ; X"
"RTN", "DICLGFT", 63, 0)
  S $X=0, $Y=0 ; , IOP="NULL"
"RTN", "DICLGFT", 64, 0)
DIP D EN1^DIP ; HERE IS THE BIG CALL TO FILEMAN'S PRINT MODULE!
"RTN", "DICLGFT", 65, 0)
  Q "OK^" _DICLGFT ; EXIT WITH 'DICLGFT' DEFINED AS THE NUMBER OF LEVELS
"RTN", "DICLGFT", 66, 0)
;
"RTN", "DICLGFT", 67, 0)
;
"RTN", "DICLGFT", 68, 0)
;
"RTN", "DICU1")
0^5^B37653223
"RTN", "DICU1", 1, 0)
DICU1 ; SEA/TOAD, SF/TKW-VA FileMan: Lookup Tools, Get IDs & Index ; 19DEC2007
"RTN", "DICU1", 2, 0)
; ; 22.0; VA FileMan; **GFT** ; Mar 30, 1999
"RTN", "DICU1", 3, 0)
;
"RTN", "DICU1", 4, 0)
IDENTS (DIFLAGS, DIFILE, DIDS, DIWRITE, DIDENT, DINDEX) ;
"RTN", "DICU1", 5, 0)
; get definition of fields to return with each entry
"RTN", "DICU1", 6, 0)
;
"RTN", "DICU1", 7, 0)
ID1 ; prepare to build output processor:
"RTN", "DICU1", 8, 0)
;
"RTN", "DICU1", 9, 0)
  S DIDS=" ; " _DIDS_ " ; "
"RTN", "DICU1", 10, 0)
  I DIDS[" ; @ ; " S DIDS ("@" )=""
"RTN", "DICU1", 11, 0)
  E S : DIDS' [" ; -WID ; " DIDS ("WID")="" S : DIDS="" ; ; " DIDS ("FID")=""
"RTN", "DICU1", 12, 0)
  N DICRSR, DICOUNT S (DICRSR, DICOUNT)=0
"RTN", "DICU1", 13, 0)
  I DIFLAGS ["P" S DICRSR=1, DIDENT (-3)="" IEN"
"RTN", "DICU1", 14, 0)
  N DIFORMAT, DIDEFALT S DIDEFALT=$S (DIFLAGS ["I" : "I", 1 : "E")
"RTN", "DICU1", 15, 0)
;
"RTN", "DICU1", 16, 0)
ID1A ; for Lister: add indexed fields to DIDENT array (to build 1 nodes)
"RTN", "DICU1", 17, 0)
;
"RTN", "DICU1", 18, 0)
  I DIFLAGS [3, DIFLAGS' ["S", DIDS' [" ; -IX", '$D (DIDS ("@" ) ) D

```

```

"RTN", "DICU1", 19, 0)
. S DIDENT=-2, DIDENT(-2)=1
"RTN", "DICU1", 20, 0)
. D THROW^DICU11 (DIFLAGS, .DIDENT, .DIDS, .DICRSR, .DICOOUNT, DIDEFALT, .DINDEX)
"RTN", "DICU1", 21, 0)
. S DIDENT=0
"RTN", "DICU1", 22, 0)
;
"RTN", "DICU1", 23, 0)
ID2 ; decide whether to auto-include the .01 in the field list
"RTN", "DICU1", 24, 0)
; will come out in 1 node for Lister, in "ID" nodes for Finder
"RTN", "DICU1", 25, 0)
;
"RTN", "DICU1", 26, 0)
N DIUSEKEY S (DIUSEKEY, DIDENT)=0
"RTN", "DICU1", 27, 0)
I '$D(DIDS("@")), DIDS'[";- .01; ", DIFLAGS'["S" D
"RTN", "DICU1", 28, 0)
. I DIFLAGS[4 S DIUSEKEY="1F" Q
"RTN", "DICU1", 29, 0)
. I DIDS["; .01; "!(DIDS["; .01E") Q
"RTN", "DICU1", 30, 0)
. S DIUSEKEY=1 N DISUB F DISUB=1:1:DINDEX("#") D Q:'DIUSEKEY
"RTN", "DICU1", 31, 0)
. . Q:$G(DINDEX(DISUB, "FIELD"))'=.01 ;**GFT
"RTN", "DICU1", 32, 0)
. . S DIUSEKEY=DINDEX(DISUB, "FILE")'=DIFILE
"RTN", "DICU1", 33, 0)
. Q
"RTN", "DICU1", 34, 0)
I DIUSEKEY S DIDENT(-2)=1, DIDENT=.01
"RTN", "DICU1", 35, 0)
N DICODE, DIDEF, DIEFROM, DIETO, DINODE, DIPIECE, DISTORE, DITYPE, DIFRMT2
"RTN", "DICU1", 36, 0)
N DILENGTH, DIOUTI S DILENGTH=$L(DIDS, ";"), DIOUTI=0
"RTN", "DICU1", 37, 0)
;
"RTN", "DICU1", 38, 0)
ID3 ; Process auto-included .01 field (if included) on first pass,
"RTN", "DICU1", 39, 0)
; Start loop to process each field from DIFIELDS parameter
"RTN", "DICU1", 40, 0)
; and Identifiers.
"RTN", "DICU1", 41, 0)
;
"RTN", "DICU1", 42, 0)
F D Q:$G(DIERR)!DIOUTI
"RTN", "DICU1", 43, 0)
. S DIFORMAT=""
"RTN", "DICU1", 44, 0)
. I DIUSEKEY D Q
"RTN", "DICU1", 45, 0)
. . D BLD S DIUSEKEY=$S(DIUSEKEY="1F": "F", 1:0)
"RTN", "DICU1", 46, 0)
. . S:DIDENT=-2 DIDENT=.01 Q
"RTN", "DICU1", 47, 0)
. D Q:'DIDENT
"RTN", "DICU1", 48, 0)

```

```

. . S DIUSEKEY=0
"RTN", "DICU1", 49, 0)
. . ; Find next Identifier
"RTN", "DICU1", 50, 0)
. . I $D(DIDS("FID")) D Q
"RTN", "DICU1", 51, 0)
. . . S DIDENT=$O(^DD(DIFILE, 0, "ID", DIDENT))
"RTN", "DICU1", 52, 0)
. . . I 'DIDENT K DIFRMT2
"RTN", "DICU1", 53, 0)
. . . I DIDENT="" S:DIDS=";" DIOUTI=1 K DIDS("FID")
"RTN", "DICU1", 54, 0)
. .
"RTN", "DICU1", 55, 0)
ID4 . . ; Find next field in DIFIELDS input parameter.
"RTN", "DICU1", 56, 0)
. .
"RTN", "DICU1", 57, 0)
. . S DICOUNT=DICOUNT+1
"RTN", "DICU1", 58, 0)
. . S DIDENT=$P(DIDS, ";", DICOUNT)
"RTN", "DICU1", 59, 0)
. . I DIDENT="", DICOUNT'<DILENGTH S DIOUTI=1
"RTN", "DICU1", 60, 0)
ID4X ..I DIDENT]"" N X S X=DIDENT I +X'=X D Q:$D(X) ;***GFT
"RTN", "DICU1", 61, 0)
...N DISVFILE
"RTN", "DICU1", 62, 0)
...S DISVFILE=DIFILE N DIFILE S DIFILE=DISVFILE ;Q^DIC2 KILLS DIFILE
"RTN", "DICU1", 63, 0)
...D EXPR^DICOMP(DIFILE, "m", X) Q: '$D(X)
"RTN", "DICU1", 64, 0)
...S DICRSR=DICRSR+1 S:$G(Y) ["m" DIGFT(DICRSR, "MULTIPLE")=1 S:$G(Y) ["D"
DIGFT(DICRSR, "DATE")=1
"RTN", "DICU1", 65, 0)
...S Y="C"_(DICOUNT-1) ;COMPUTED
"RTN", "DICU1", 66, 0)
...S:DIFLAGS["P" $P(DIDENT(-3), U, DICRSR)=Y ;THIS WILL BECOME THE PACKED "MAP"
"RTN", "DICU1", 67, 0)
...S:DIFLAGS["P" DIDENT(-3, $O(^DD(DISVFILE, " "), -1)+1, Y)="" ;THIS IS THE UNPACKED
MAP
"RTN", "DICU1", 68, 0)
...S DIDENT(DICRSR, Y, 0)="D COMP^DICU1("_DICRSR_)"
"RTN", "DICU1", 69, 0)
...M DIGFT(DICRSR)=X S DIDENT=""
"RTN", "DICU1", 70, 0)
. .
"RTN", "DICU1", 71, 0)
ID4A . . ; process IX specifier
"RTN", "DICU1", 72, 0)
. .
"RTN", "DICU1", 73, 0)
. . I DIDENT["IX" D Q
"RTN", "DICU1", 74, 0)
. . . I $$BADIX(DIDENT) D ERR202 Q
"RTN", "DICU1", 75, 0)
. . . Q:DIDS[";-IX;"
"RTN", "DICU1", 76, 0)
. . . D THROW^DICU11(DIFLAGS, .DIDENT, .DIDS, .DICRSR, .DICOUNT, DIDEFALT, .DINDEX)

```

```

"RTN", "DICU1", 77, 0)
. .
"RTN", "DICU1", 78, 0)
ID4B . . ; process FID, WID, and @ specifiers
"RTN", "DICU1", 79, 0)
. .
"RTN", "DICU1", 80, 0)
. . I DIDENT["FID" D S DIDENT="" Q
"RTN", "DICU1", 81, 0)
. . . Q:DIDENT="-FID"!(DIDS[";-FID;")
"RTN", "DICU1", 82, 0)
. . . D GETFORM^DICU11(.DIDENT,.DIFRMT2,.DIDS,.DICOUNT)
"RTN", "DICU1", 83, 0)
. . . S DIDS("FID")=1 Q
"RTN", "DICU1", 84, 0)
. . I DIDENT["WID" D S DIDENT="" Q
"RTN", "DICU1", 85, 0)
. . . I DIDENT'="WID",DIDENT'="-WID" D ERR202 Q
"RTN", "DICU1", 86, 0)
. . . Q:DIDENT="-WID"!(DIDS[";-WID;")
"RTN", "DICU1", 87, 0)
. . . D WRITEID^DICU11(DIFILE,.DIDENT,.DICRSR) K DIDS("WID") Q
"RTN", "DICU1", 88, 0)
. . I DIDENT["@" D:DIDENT'="@ " ERR202 Q
"RTN", "DICU1", 89, 0)
. . I 'DIDENT D:DIDENT'="" ERR202 Q
"RTN", "DICU1", 90, 0)
. .
"RTN", "DICU1", 91, 0)
ID4C . . ; process field # specifiers from DIFIELDS parameter
"RTN", "DICU1", 92, 0)
. .
"RTN", "DICU1", 93, 0)
. . D GETFORM^DICU11(.DIDENT,.DIFORMAT,.DIDS,.DICOUNT)
"RTN", "DICU1", 94, 0)
.
"RTN", "DICU1", 95, 0)
. ; Here we quit if field is already in the DIDENT array.
"RTN", "DICU1", 96, 0)
. I DIDS=";;",DIFLAGS[4,DIUSEKEY'="F",DIDENT=.01 Q
"RTN", "DICU1", 97, 0)
. I DIDS=";;",DIFLAGS[3,DINDEX("FLIST") [ ("^"_DIDENT_"^") Q
"RTN", "DICU1", 98, 0)
.
"RTN", "DICU1", 99, 0)
ID5 . . ; for file IDs, we skip non-display IDs
"RTN", "DICU1", 100, 0)
.
"RTN", "DICU1", 101, 0)
. N DIPLUS S DIPLUS=+DIDENT
"RTN", "DICU1", 102, 0)
. N DILAST S DILAST=$P(DIDENT,DIPLUS,2,999)
"RTN", "DICU1", 103, 0)
. I DIDENT["-" D Q
"RTN", "DICU1", 104, 0)
. . I DILAST'="" D ERR202 Q
"RTN", "DICU1", 105, 0)
. . I '$D(^DD(DIFILE,-DIPLUS)) D ERR(501,DIFILE,"",",-DIPLUS) Q
"RTN", "DICU1", 106, 0)

```

```

. E I (DILAST'?..1"E".1"I") & (DILAST'?..1"I".1"E") D ERR202 Q
"RTN", "DICU1", 107, 0)
. Q: DIDS (";-_DIDENT_";")
"RTN", "DICU1", 108, 0)
. I $D(DIDS("FID")) D I DINODE="W """""" Q
"RTN", "DICU1", 109, 0)
. . S DINODE=$G(^DD(DIFILE, 0, "ID", DIDENT))
"RTN", "DICU1", 110, 0)
. I $G(DIFRMT2)]"" S DIFORMAT=DIFRMT2
"RTN", "DICU1", 111, 0)
. D BLD Q
"RTN", "DICU1", 112, 0)
;
"RTN", "DICU1", 113, 0)
ID6 ; Write Identifiers: add to output processor
"RTN", "DICU1", 114, 0)
; ID Parameter: add ID parameter to output processor
"RTN", "DICU1", 115, 0)
;
"RTN", "DICU1", 116, 0)
Q:$G(DIERR)
"RTN", "DICU1", 117, 0)
I $D(DIDS("WID")) D WRITEID^DICU11(DIFILE, .DIDENT, .DICRSR)
"RTN", "DICU1", 118, 0)
I DIWRITE="" D
"RTN", "DICU1", 119, 0)
. S DIDENT="ZZZ ID" I DIFLAGS["P" S DICRSR=DICRSR+1
"RTN", "DICU1", 120, 0)
. S DIDENT(DICRSR, DIDENT, 0)="N DIMSG "_DIWRITE
"RTN", "DICU1", 121, 0)
. S:DIFLAGS["P" $P(DIDENT(-3), U, DICRSR)="IDP" Q
"RTN", "DICU1", 122, 0)
Q
"RTN", "DICU1", 123, 0)
;
"RTN", "DICU1", 124, 0)
BLD ; get fetch code for value
"RTN", "DICU1", 125, 0)
D GET^DICUIX1(DIFILE, DIFILE, DIDENT, .DIDEF, .DICODE) Q:DIDEF=""!$G(DIERR)
"RTN", "DICU1", 126, 0)
I DIFORMAT="" S DIFORMAT=$S(DIUSEKEY="1F": "I", 1:DIDEFALT)
"RTN", "DICU1", 127, 0)
D
"RTN", "DICU1", 128, 0)
. N DIVALUE S DIVALUE=DIDENT
"RTN", "DICU1", 129, 0)
. I DIUSEKEY["F", $D(DIDS("FID")), DIDENT]=.01 S DIVALUE="FID("_DIVALUE_)"
"RTN", "DICU1", 130, 0)
. S:DIFORMAT="I" DIVALUE=DIVALUE_DIFORMAT
"RTN", "DICU1", 131, 0)
. I DIFLAGS["P" S $P(DIDENT(-3), U, (DICRSR+1))=DIVALUE Q
"RTN", "DICU1", 132, 0)
. Q:DIUSEKEY="1F"
"RTN", "DICU1", 133, 0)
. S DIDENT(-3, +DIDENT, DIVALUE)="" Q
"RTN", "DICU1", 134, 0)
BLD1 ; set up format code and load with fetch code into DIDENT
"RTN", "DICU1", 135, 0)
N DIVALUE, DISUB S DIVALUE=DICODE, DISUB=0

```

```

"RTN", "DICU1", 136, 0)
  S DITYPE=$P(DIDEF,U,2) I DITYPE["C" D
"RTN", "DICU1", 137, 0)
  . S DIVALUE=$$FORMAT^DICU11(DIDENT,DICODE,DIUSEKEY,DIFORMAT,DIDEFALT,DIFLAGS)
"RTN", "DICU1", 138, 0)
  I DIUSEKEY="1F",DIDENT=.01 S DIDENT=-2,DISUB=.01
"RTN", "DICU1", 139, 0)
  I DIFLAGS["P" S DICRSR=DICRSR+1
"RTN", "DICU1", 140, 0)
  I DITYPE["C" S DIDENT(DICRSR,DIDENT,DISUB,DIFORMAT)=DIVALUE Q
"RTN", "DICU1", 141, 0)
  S DIDENT(DICRSR,DIDENT,0)=DIVALUE
"RTN", "DICU1", 142, 0)
  S DIDENT(DICRSR,DIDENT,0,"TYPE")="C"
"RTN", "DICU1", 143, 0)
  Q
"RTN", "DICU1", 144, 0)
  ;
"RTN", "DICU1", 145, 0)
  ;
"RTN", "DICU1", 146, 0)
COMP(DIGFTI) ;EXECUTE A COMPUTED FIELD!
"RTN", "DICU1", 147, 0)
  N X,Y,J,I
"RTN", "DICU1", 148, 0)
  S J=0 F Y=$L(DIEN,""):-1:1 S X=$P(DIEN,"",Y) I X]"" N @("D"_J) S
@("D"_J)=X,J=J+1 ;Temporarily set D0,D1,etc
"RTN", "DICU1", 149, 0)
  M X=DIGFT(DIGFTI)
"RTN", "DICU1", 150, 0)
  I '$D(DIGFT(DIGFTI,"MULTIPLE")) X X D:$D(DIGFT(DIGFTI,"DATE")) S
^TMP("DIMSG",$J,1)=X Q ;SINGLE-VALUED COMPUTED EXPRESSION
"RTN", "DICU1", 151, 0)
  .N Y S Y=X X:Y ^DD("DD") S X=Y
"RTN", "DICU1", 152, 0)
  N DICMX S DICMX="S ^TMP("DIMSG",$J,D)=X" X X ;MULTIPLE-VALUED COMPUTED
EXPRESSION
"RTN", "DICU1", 153, 0)
  Q
"RTN", "DICU1", 154, 0)
  ;
"RTN", "DICU1", 155, 0)
  ;
"RTN", "DICU1", 156, 0)
ERR(DIERN,DIFILE,DIENS,DIFIELD,DI1) ;
"RTN", "DICU1", 157, 0)
  ;
"RTN", "DICU1", 158, 0)
  ; add an error to the message array
"RTN", "DICU1", 159, 0)
  ; GET
"RTN", "DICU1", 160, 0)
  ;
"RTN", "DICU1", 161, 0)
  N DIPE
"RTN", "DICU1", 162, 0)
  S DIPE("FILE")=$G(DIFILE)
"RTN", "DICU1", 163, 0)
  S DIPE("IEN")=$G(DIENS)

```

```

"RTN", "DICU1", 164, 0)
  S DIPE ("FIELD") = $G (DIFIELD)
"RTN", "DICU1", 165, 0)
  S DIPE (1) = $G (DI1)
"RTN", "DICU1", 166, 0)
  D BLD ^ DIALOG (DIERN, .DIPE, .DIPE)
"RTN", "DICU1", 167, 0)
  Q
"RTN", "DICU1", 168, 0)
  ;
"RTN", "DICU1", 169, 0)
ERR202 D ERR (202, "", "", "", "FIELDS") Q
"RTN", "DICU1", 170, 0)
  ;
"RTN", "DICU1", 171, 0)
BADIX (DIDENT) ;
"RTN", "DICU1", 172, 0)
  ;
"RTN", "DICU1", 173, 0)
  N DIBAD S DIBAD = DIDENT' = "IX" & (DIDENT' = "-IX") & (DIDENT' ? 1 "IX" 1 "E" . 1 "I")
"RTN", "DICU1", 174, 0)
  S DIBAD = DIDENT' ? 1 "IX" 1 "I" . 1 "E" & DIBAD
"RTN", "DICU1", 175, 0)
  Q DIBAD
"RTN", "DICU1", 176, 0)
  ;
"RTN", "DICU1", 177, 0)
  ; 202 The input parameter that identifies the |1
"RTN", "DICU1", 178, 0)
  ;
"RTN", "DICU2")
0^6^B30429119
"RTN", "DICU2", 1, 0)
DICU2 ; SEA/TOAD, SF/TKW-VA FileMan: Lookup Tools, Return IDs ; 19DEC2007
"RTN", "DICU2", 2, 0)
  ; ; 22.0; VA FileMan; **126,999,GFT** ; Mar 30, 1999
"RTN", "DICU2", 3, 0)
  ;
"RTN", "DICU2", 4, 0)
IDS (DIFILE, DIEN, DIFLAGS, DINDEX, DICOUNT, DIDENT, DILIST, DIONODE) ;
"RTN", "DICU2", 5, 0)
  ;
"RTN", "DICU2", 6, 0)
  ; ENTRY POINT--add an entry's identifiers to output
"RTN", "DICU2", 7, 0)
  ;
"RTN", "DICU2", 8, 0)
I1 ; setup 0-node and ID array interface, and output IEN
"RTN", "DICU2", 9, 0)
  ;
"RTN", "DICU2", 10, 0)
  I DIFLAGS ["h" N F, N, I M F = DIFILE S N = $G (DIONODE), I = +$G (DIEN) N DIFILE, DIONODE, DIEN
M DIFILE = F S DIEN = I S : N] "" DIONODE = N K F, N, I
"RTN", "DICU2", 11, 0)
  I '$D (DIONODE) S DIONODE = $G (@DIFILE (DIFILE) @ (+DIEN, 0))
"RTN", "DICU2", 12, 0)
  N DID, DIDVAL
"RTN", "DICU2", 13, 0)
  I DIFLAGS ["P" N DINODE S DINODE = +DIEN

```

```

"RTN", "DICU2", 14, 0)
E S @DILIST@ (2, DICOUNT) = +DIEN
"RTN", "DICU2", 15, 0)
;
"RTN", "DICU2", 16, 0)
I1A ; output primary value (index for Lister, .01 for Finder)
"RTN", "DICU2", 17, 0)
;
"RTN", "DICU2", 18, 0)
I DIFLAGS ["P", $D(DIDENT(-2)) D
"RTN", "DICU2", 19, 0)
. N DIOUT S DIOUT=$NA(@DILIST@(1, DICOUNT))
"RTN", "DICU2", 20, 0)
. I DIFLAGS[3 N DISUB D Q
"RTN", "DICU2", 21, 0)
. . F DISUB=0:0 S DISUB=$O(DIDENT(0, -2, DISUB)) Q:'DISUB D
"RTN", "DICU2", 22, 0)
. . . I DINDEX("#") '>1 D SET(0, -2, DISUB, DIOUT, .DINDEX, .DIFILE) Q
"RTN", "DICU2", 23, 0)
. . . N I S I=$NA(@DIOUT@(DISUB)) D SET(0, -2, DISUB, I, .DINDEX, .DIFILE)
"RTN", "DICU2", 24, 0)
. I $D(DIDENT(0, -2, .01)) D SET(0, -2, .01, DIOUT, "", .DIFILE)
"RTN", "DICU2", 25, 0)
. Q
"RTN", "DICU2", 26, 0)
;
"RTN", "DICU2", 27, 0)
I2 ; start loop: loop through output values
"RTN", "DICU2", 28, 0)
;
"RTN", "DICU2", 29, 0)
I DIFLAGS ["P" N DILENGTH S DILENGTH=$L(DINODE)
"RTN", "DICU2", 30, 0)
N DICODE, DICRSR, DIOUT, DISUB S DICRSR=-1
"RTN", "DICU2", 31, 0)
F S DICRSR=$O(DIDENT(DICRSR)) Q:DICRSR=""!($G(DIERR)) S DID="" F S
DID=$O(DIDENT(DICRSR, DID)) Q:DID=""!($G(DIERR)) S DISUB="" F D
Q:DISUB=""!$G(DIERR)
"RTN", "DICU2", 32, 0)
. I DIFLAGS ["P", DID=-2 Q
"RTN", "DICU2", 33, 0)
. S DISUB=$O(DIDENT(DICRSR, DID, DISUB)) Q:DISUB=""
"RTN", "DICU2", 34, 0)
. K DIDVAL
"RTN", "DICU2", 35, 0)
I20 . ; output indexed field if "IX" was in FIELDS parameter
"RTN", "DICU2", 36, 0)
. I DID=0 D Q
"RTN", "DICU2", 37, 0)
. . D SET(DICRSR, DID, DISUB, "DIDVAL", .DINDEX, .DIFILE)
"RTN", "DICU2", 38, 0)
. . I DIFLAGS ["P" D ADD(.DIFLAGS, .DINODE, .DILENGTH, DIDVAL, DIEN, DILIST) Q
"RTN", "DICU2", 39, 0)
. . M @DILIST@ ("ID", DICOUNT, 0, DISUB) = DIDVAL Q
"RTN", "DICU2", 40, 0)
.
"RTN", "DICU2", 41, 0)
I3 . ; output field
"RTN", "DICU2", 42, 0)

```



```

. ; distinguish between computed and value fields
"RTN", "DICU2", 43, 0)
.
"RTN", "DICU2", 44, 0)
. I DID D Q:$G(DIERR)
"RTN", "DICU2", 45, 0)
. . ; process fields that are not computed.
"RTN", "DICU2", 46, 0)
. . I $G(DIDENT(DICRSR, DID, 0, "TYPE")) ='C" D
"RTN", "DICU2", 47, 0)
. . . D SET(DICRSR, DID, DISUB, "DIDVAL", .DINDEX, .DIFILE) Q
"RTN", "DICU2", 48, 0)
. .
"RTN", "DICU2", 49, 0)
I4 . . ; computed fields
"RTN", "DICU2", 50, 0)
. . E D
"RTN", "DICU2", 51, 0)
. . . N %, %H, %T, A, B, C, D, DFN, I, X, X1, X2, Y, Z, Z0, Z1
"RTN", "DICU2", 52, 0)
. . . N DA M DA=DIEN S DA=$P(DIEN, ", ")
"RTN", "DICU2", 53, 0)
. . . N DIARG S DIARG="D0"
"RTN", "DICU2", 54, 0)
. . . N DIMAX S DIMAX=+$O(DA(""), -1)
"RTN", "DICU2", 55, 0)
. . . N DIDVAR F DIDVAR=1:1:DIMAX S DIARG=DIARG_", D"_DIDVAR
"RTN", "DICU2", 56, 0)
. . . N @DIARG F DIDVAR=0:1:DIMAX-1 S @("D"_DIDVAR)=DA(DIMAX-DIDVAR)
"RTN", "DICU2", 57, 0)
. . . S @("D"_DIMAX)=DA
"RTN", "DICU2", 58, 0)
. . . X DIDENT(DICRSR, DID, 0) S DIDVAL=$G(X)
"RTN", "DICU2", 59, 0)
COMPDT . . . I $P($G(^DD(DIFILE, DID, 0)), U, 2) ["D" N Y S Y=DIDVAL X:Y ^DD("DD") S
DIDVAL=Y
"RTN", "DICU2", 60, 0)
. .
"RTN", "DICU2", 61, 0)
I5 . . ; set field into array or pack node
"RTN", "DICU2", 62, 0)
. .
"RTN", "DICU2", 63, 0)
. . I DIFLAGS'["P" M @DILIST@("ID", DICOUNT, DID)=DIDVAL
"RTN", "DICU2", 64, 0)
. . E D ADD(.DIFLAGS, .DINODE, .DILENGTH, DIDVAL, DIEN, DILIST)
"RTN", "DICU2", 65, 0)
.
"RTN", "DICU2", 66, 0)
I6 . ; output display-only identifier
"RTN", "DICU2", 67, 0)
.
"RTN", "DICU2", 68, 0)
. E D
"RTN", "DICU2", 69, 0)
. . N %, D, DIC, X, Y, Y1
"RTN", "DICU2", 70, 0)
. . S D=DINDEX
"RTN", "DICU2", 71, 0)

```

```

. . S DIC=DIFILE(DIFILE,"O")
"RTN","DICU2",72,0)
. . S DIC(0)=$TR(DIFLAGS,"2^fglpqtuv104")
"RTN","DICU2",73,0)
. . M Y=DIEN S Y=$P(DIEN,"")
"RTN","DICU2",74,0)
. . S Y1=$G(@DIFILE(DIFILE)@( +DIEN,0)),Y1=DIEN
"RTN","DICU2",75,0)
. .
"RTN","DICU2",76,0)
I7 . . ; execute the identifier's code
"RTN","DICU2",77,0)
. .
"RTN","DICU2",78,0)
. . N DIX S DIX=DIDENT(DICRSR,DID,0)
"RTN","DICU2",79,0)
. . X DIX
"RTN","DICU2",80,0)
. . I $G(DIERR) D Q
"RTN","DICU2",81,0)
. . . N DICONTXT I DID="ZZZ ID" S DICONTXT="Identifier parameter"
"RTN","DICU2",82,0)
. . . E S DICONTXT="MUMPS Identifier"
"RTN","DICU2",83,0)
. . . D ERR^DICF4(120,DIFILE,DIEN,"",DICONTXT)
"RTN","DICU2",84,0)
. .
"RTN","DICU2",85,0)
I8 . . ; set output from identifier into output array or pack node
"RTN","DICU2",86,0)
. . N DIGFT S DIGFT=$NA(@DILIST@("ID","WRITE",DICOOUNT)) I DID?1"C"1.2N S
DIGFT=$NA(@DILIST@("ID",DICOOUNT,DID)) ;**GFT
"RTN","DICU2",87,0)
. . N DI,DILINE,DIEND S DI="" S:DIFLAGS["P" DIEND=$O(@DIGFT@("z"),-1)
"RTN","DICU2",88,0)
. . I $O(^TMP("DIMSG",$J,""))="" S ^TMP("DIMSG",$J,1)=""
"RTN","DICU2",89,0)
. . F D Q:DI=""!$G(DIERR)
"RTN","DICU2",90,0)
. . . S DI=$O(^TMP("DIMSG",$J,DI)) Q:DI=""
"RTN","DICU2",91,0)
. . . S DILINE=$G(^TMP("DIMSG",$J,DI))
"RTN","DICU2",92,0)
. . . I DIFLAGS["P" D ADD(.DIFLAGS,.DINODE,.DILENGTH,DILINE,DIEN,DILIST,DI) Q
"RTN","DICU2",93,0)
. . . S DIEND=DIEND+1,@DIGFT@(DIEND)=DILINE
"RTN","DICU2",94,0)
. . . Q
"RTN","DICU2",95,0)
. . K DIMSG,^TMP("DIMSG",$J)
"RTN","DICU2",96,0)
;
"RTN","DICU2",97,0)
I9 ; for packed output, set pack node into output array
"RTN","DICU2",98,0)
;
"RTN","DICU2",99,0)
I '$G(DIERR),DIFLAGS["P" S @DILIST@(DICOOUNT,0)=DINODE
"RTN","DICU2",100,0)

```

```

Q
"RTN", "DICU2", 101, 0)
;
"RTN", "DICU2", 102, 0)
;
"RTN", "DICU2", 103, 0)
SET (DICRSR, DIFID, DISUB, DIOUT, DINDEX, DIFILE) ; Move data to DIOUT.
"RTN", "DICU2", 104, 0)
  N F1, F2 M F1=DIFILE N DIFILE M DIFILE=F1
"RTN", "DICU2", 105, 0)
  S F1=$O (DIDENT (DICRSR, DIFID, DISUB, "")) , F2=$O (DIDENT (DICRSR, DIFID, DISUB, F1))
"RTN", "DICU2", 106, 0)
  F F1=F1, F2 D:F1]""
"RTN", "DICU2", 107, 0)
  . I DIDENT (DICRSR, DIFID, DISUB, F1) ["DIVAL" N DIVAL S @DINDEX (DISUB, "GET")
"RTN", "DICU2", 108, 0)
  . N X S @("X="_DIDENT (DICRSR, DIFID, DISUB, F1))
"RTN", "DICU2", 109, 0)
  . I $G (DIERR), DIFLAGS["h" K DIERR, ^TMP ("DIERR", $J) S X=DINDEX (DISUB)
"RTN", "DICU2", 110, 0)
  . I X[""" S X=$$CONVQQ^DILIBF (X)
"RTN", "DICU2", 111, 0)
  . I +$P (X, "E") '=X S X="" _X ""
"RTN", "DICU2", 112, 0)
  . I F2="" S @ (DIOUT_ "=" _X) Q
"RTN", "DICU2", 113, 0)
  . S O=$NA (@DIOUT@ (F1)), @ (O_ "=" _X) Q
"RTN", "DICU2", 114, 0)
Q
"RTN", "DICU2", 115, 0)
;
"RTN", "DICU2", 116, 0)
TRANOUT (DISUB, DIVL) ; Execute TRANSFORM FOR DISPLAY on index value
"RTN", "DICU2", 117, 0)
  N X S X=DIVL
"RTN", "DICU2", 118, 0)
  N DICODE S DICODE=$G (DINDEX (DISUB, "TRANOUT"))
"RTN", "DICU2", 119, 0)
  I DICODE]"" X DICODE
"RTN", "DICU2", 120, 0)
Q X
"RTN", "DICU2", 121, 0)
;
"RTN", "DICU2", 122, 0)
ADD (DIFLAGS, DINODE, DILENGTH, DINEW, DIEN, DILIST, DILCNT) ;
"RTN", "DICU2", 123, 0)
;
"RTN", "DICU2", 124, 0)
; for Packed output, add DINEW to DINODE, erroring if overflow
"RTN", "DICU2", 125, 0)
; xform if it contains ^
"RTN", "DICU2", 126, 0)
;
"RTN", "DICU2", 127, 0)
A1 N DINEWLEN, DELIM S DINEWLEN=$L (DINEW), DELIM=$S ($G (DILCNT) '>1:"^", 1:"~")
"RTN", "DICU2", 128, 0)
  S DILENGTH=DILENGTH+1+DINEWLEN
"RTN", "DICU2", 129, 0)
  I DILENGTH>255 D ERR^DICF4 (206, "", "", "", +DIEN) Q ;**HERE IS WHERE A PACKED STRING

```

```

IS FORCED TO BE ONLY 255 CHARACTERS LONG
"RTN", "DICU2", 130, 0)
  I DIFLAGS' [2, DINEW[U S DIFLAGS="2^" _DIFLAGS D ENCODE (DILIST, .DINODE)
"RTN", "DICU2", 131, 0)
  I DIFLAGS [2, DINEW[U! (DINEW["&") S DINEW=$$HTML^DILF (DINEW) Q:$G (DIERR)
"RTN", "DICU2", 132, 0)
  S DINODE=DINODE_DELIM_DINEW
"RTN", "DICU2", 133, 0)
  Q
"RTN", "DICU2", 134, 0)
  ;
"RTN", "DICU2", 135, 0)
ENCODE (DILIST, DINODE) ;
"RTN", "DICU2", 136, 0)
  ;
"RTN", "DICU2", 137, 0)
  ; ADD: HTML encode records already output (we found an embedded ^)
"RTN", "DICU2", 138, 0)
  ; procedure: loop through list encoding &s
"RTN", "DICU2", 139, 0)
  ;
"RTN", "DICU2", 140, 0)
E1 N DILINE, DIRULE S DIRULE (1, "&")="&";"
"RTN", "DICU2", 141, 0)
  N DIREC S DIREC=0 F S DIREC=$O (@DILIST@ (DIREC)) Q:'DIREC D
"RTN", "DICU2", 142, 0)
  . S DILINE=@DILIST@ (DIREC, 0) Q:DILINE' ["&
"RTN", "DICU2", 143, 0)
  . S @DILIST@ (DIREC, 0)=$$TRANSL8^DILF (DILINE, .DIRULE)
"RTN", "DICU2", 144, 0)
  I DINODE["& S DINODE=$$TRANSL8^DILF (DINODE, .DIRULE)
"RTN", "DICU2", 145, 0)
  Q
"RTN", "DICU2", 146, 0)
  ;
"RTN", "DIP0")
0^4^B17561996
"RTN", "DIP0", 1, 0)
DIP0 ;SFISC/XAK-COMPUTED FIELD ON A SORT, EDITING A SORT TEMPLATE ;9DEC2007
"RTN", "DIP0", 2, 0)
  ;;22.0;VA FileMan;**2,999,1003,GFT**;Mar 30, 1999
"RTN", "DIP0", 3, 0)
  S P=P_Q, DPP=$P (X, U, 1)
"RTN", "DIP0", 4, 0)
C ;
"RTN", "DIP0", 5, 0)
  S DICOMP=N $E ("?", 'L), DM=X, DQI="Y (" , DA="DPP (" _DJ_", ""OVF" _N """, ", DICMX="D
M^DIO2" G COLON:X?.E1": " D EN^DICOMP K DUOUT G X: '$D (X), X:Y ["m"
"RTN", "DIP0", 6, 0)
  D XA, BB^DIP:Y ["B" S:Y ["D" R=R_ ""^D" S Y=U_DPP, DPP (DJ, "CM")=X_ " I D" _ (N#100) _ ">0 S
DISX (" _DJ_")=X" G S^DIP
"RTN", "DIP0", 7, 0)
  ;
"RTN", "DIP0", 8, 0)
XA F %=0:0 S %=$O (X (%)) Q:%="" S @ (DA_ "%)=X (%)"
"RTN", "DIP0", 9, 0)
  Q
"RTN", "DIP0", 10, 0)
  ;

```

```

"RTN", "DIP0", 11, 0)
;
"RTN", "DIP0", 12, 0)
COLON D ^DICOMPW K DUOUT
"RTN", "DIP0", 13, 0)
  I $D(X), $S($D(DIL(+DP)):DIL(+DP)=DL, 1:1) S
DPP(DJ, DL, +Y)=DP_U_(Y["m"])_U_X, DIL(+DP)=DL, N=+Y, DL=+DP, DV=$J("", DJ*2-
2)_SO(^DD(DL, 0, "NM", 0))_"FIELD" S:$D(DIPP(DIJ, +DP))#2
$P(DIPP(DIJ), U, 3)=DIPP(DIJ, +DP) D XA, L G Y^DIP
"RTN", "DIP0", 14, 0)
X I $G(BY)]"" S X=DM_", "_BY, BY="" G C ;TRY TACKING ON THE REST OF THE "BY", AFTER
THE FIRST COMMA!
"RTN", "DIP0", 15, 0)
  I $G(DIQUIET) G Q^DIP
"RTN", "DIP0", 16, 0)
  G B^DIP
"RTN", "DIP0", 17, 0)
;
"RTN", "DIP0", 18, 0)
EDT ;
"RTN", "DIP0", 19, 0)
  S DIE="^DIBT(", DR=".01;3;6", DA=X, DIPP=DI, DIOVRD=1 D ^DIE S
DI=DIPP, DE=$S(L=0!L:"SORT", 1:L) K DR, DIE, DIPP, DIOVRD I '$D(DA)!($D(Y)) S
(X, DJ)=+$G(DPP(0)) Q
"RTN", "DIP0", 20, 0)
  S %=$G(DPP(0, "IX")) I $P(%U, 2)]"", $P(%U, 4) D I $G(DPP(0))']"" S (X, DJ)=0 Q
"RTN", "DIP0", 21, 0)
  . N X, I, Y, F, T, O, Q, DIEDITBY S DIEDITBY=1 K FR(0), TO(0), DISPAR(0), DIPP
"RTN", "DIP0", 22, 0)
  . S BY(0)="^"_$P(%U, 2), L(0)=$P(%U, 4)
"RTN", "DIP0", 23, 0)
  . F I=L(0):1 Q:'$D(DPP(I)) M DIPP(I)=DPP(I) K DPP(I)
"RTN", "DIP0", 24, 0)
  . F I=1:1:(L(0)-1) D Q:'$G(L(0))
"RTN", "DIP0", 25, 0)
  .. S F=$P($G(DPP(I, "F")), U, 2), T=$P($G(DPP(I, "T")), U, 2) S:F]"" FR(0, I)=F S:T]""
TO(0, I)=T
"RTN", "DIP0", 26, 0)
  .. S O=$P($G(DPP(I)), U, 4), Q=" S:O["!" Q=Q_"!" S:O["#" Q=Q_"#"
S:$P($G(DPP(I)), U, 5)]"" Q=Q_"^"_$P(DPP(I), U, 5) S:Q]"" DISPAR(0, I)=Q
"RTN", "DIP0", 27, 0)
  .. I $G(DISPAR(0, I))] "", $G(DPP(I, "OUT"))]"" S DISPAR(0, I, "OUT")=DPP(I, "OUT")
"RTN", "DIP0", 28, 0)
  .. K DPP(I) Q
"RTN", "DIP0", 29, 0)
  . D BYOK^DIP100
"RTN", "DIP0", 30, 0)
  . I $G(DPP(0))] "" S X=DPP(0) F I=0:0 S I=$O(DIPP(I)) Q:'I S X=X+1 M
DPP(X)=DIPP(I)
"RTN", "DIP0", 31, 0)
  . K DIPP Q
"RTN", "DIP0", 32, 0)
  S DIPP="", DIJ=0 F DJ=$G(DPP(0)):0 S DJ=$O(DPP(DJ)) Q:'DJ S
DIJ=DIJ+1, %X="DPP(DJ, ", %Y="DIPP(DIJ, " D %XY^%RCR
"RTN", "DIP0", 33, 0)
  S DIJ=0 F DJ=$G(DPP(0)):0 S DJ=$O(DPP(DJ)) Q:DJ="" D
"RTN", "DIP0", 34, 0)
  . S DIJ=DIJ+1 N X S X=$P(DPP(DJ), U, 4), X=$TR(X, "B", ""), X=$S(X[Q:$P(X, Q, ($L(X, Q)-
1)), 1:X)

```

```

"RTN", "DIP0", 35, 0)
. S $P (DIPP (DIJ), U, 3) = X_ $P (DPP (DJ), U, 3) _ $P (DPP (DJ), U, 5)
"RTN", "DIP0", 36, 0)
. S % = + DPP (DJ) D E1 S %X = 0 D E2 K DPP (DJ)
"RTN", "DIP0", 37, 0)
. Q
"RTN", "DIP0", 38, 0)
S DJ = $G (DPP (0)), DIJ = 0 F S DIJ = + $O (DIPP (DIJ)) Q: 'DIJ S DJ = DJ + 1 D DJ ^ DIP
Q: $D (DTOUT) ! ($D (DIRUT)) ! (' $D (DJ)) W: X = "@" " Deleted."
"RTN", "DIP0", 39, 0)
K DIPP, DIJJ S: X' = U X = 1 S: ' $D (DXS) DXS = 1 S DIEDT = 1 Q
"RTN", "DIP0", 40, 0)
E1 ;
"RTN", "DIP0", 41, 0)
F DIJJ = 0: 1 Q: ' $D (^ DD (% , 0, "UP")) S
DIPP (DIJ, %) = $P (DIPP (DIJ), U, 3), % = + ^ ("UP"), $P (DIPP (DIJ), U, 3) = $O (^ ("NM", 0)), $P (DIPP (DIJ), U, 1) = %
"RTN", "DIP0", 42, 0)
Q
"RTN", "DIP0", 43, 0)
E2 S %X = $O (DPP (DJ, %X)) I %X' > 0 K %X Q
"RTN", "DIP0", 44, 0)
G E2: ' $D (DPP (DJ, %X, 100)) S % = %X D E1 S % = DPP (DJ, %X, 100)
"RTN", "DIP0", 45, 0)
I $P (% , U, 3) S
DIPP (DIJ, +%) = $P (DIPP (DIJ), U, 3), $P (DIPP (DIJ), U, 3) = $P (^ DIC (+%, 0), U) _ ": ", $P (DIPP (DIJ), U) = +% G E2
"RTN", "DIP0", 46, 0)
I %' ["Y (1) " S % = $F (% , "OVF0") Q: ' % S
% = + $E (DPP (DJ, %X, 100), % + 2, 99), % = $P (DPP (DJ, %X, 100), U) _ U _ DPP (DJ, "OVF0", %) Q: %' ["Y (1) "
"RTN", "DIP0", 47, 0)
WHO S
G = $TR ($P ($P ($P (% , "Y (1) ", 2), ")): ^ (" , 2), " ) , " " " " ) , P = $P (% , "Y (1) ", 3) , P = $P ($P (P, "U", 2), " ) ) I G] " " , P] " " S P = + $O (^ DD (%X, "GL", G, P, 0))
"RTN", "DIP0", 48, 0)
I P, $D (^ DD (%X, P, 0)) S: DIJJ DIPP (DIJ, +% ) = DIPP (DIJ, %X) , DIPP (DIJ, %X) = $P (^ (0), U) _ ": "
S: ' DIJJ DIPP (DIJ, +% ) = $P (DIPP (DIJ), U, 3) , $P (DIPP (DIJ), U, 3) = $P (^ (0), U) _ ": "
"RTN", "DIP0", 49, 0)
G E2
"RTN", "DIP0", 50, 0)
;
"RTN", "DIP0", 51, 0)
;
"RTN", "DIP0", 52, 0)
;
"RTN", "DIP0", 53, 0)
L ; FROM DIP: READ SORT-BY VALUE
"RTN", "DIP0", 54, 0)
I $D (BY) #2 K DIC S DIC = " ^ DD (DL, " , DIC (0) = "Z", X = $P (BY, " , " ) , BY = $P (BY, " , " , 2, 99) I
X' = "@" K DV Q
"RTN", "DIP0", 55, 0)
K DIR D
"RTN", "DIP0", 56, 0)
. N X S DIR (0) = "FOU", DIR ("A") = DV
"RTN", "DIP0", 57, 0)
. S X = $P ($G (DIPP (DIJ)), U, 3) I X] " " S DIR ("B") = X
"RTN", "DIP0", 58, 0)
. I X = " " S X = $G (DV (1)) I X] " " S DIR (0) = "FOAU", DIR ("A") = DV _ ": " _ X _ " // "
"RTN", "DIP0", 59, 0)

```

```

. S DIR(0)=DIR(0)_"^1:255",DIR("?")="^D DIC^DIP0"
"RTN", "DIP0", 60, 0)
D ^DIR K DIR, DV, DIRUT, DIROUT S:$D(DTOUT) X="^"
"RTN", "DIP0", 61, 0)
K:X?1"^^1.E DUOUT
"RTN", "DIP0", 62, 0)
I X="@ " K DPP(DJ) S DJ=DJ-1
"RTN", "DIP0", 63, 0)
D SETDIC Q
"RTN", "DIP0", 64, 0)
;
"RTN", "DIP0", 65, 0)
SETDIC K DIC S DIC="^DD(DL, "
"RTN", "DIP0", 66, 0)
S DIC("S")="S %=$P(^ (0), U, 2) I
%[""m"", $S(' %:1, 1:$P(^DD(+%, .01, 0), U, 2) '[""W""&$S($D(DIL(+%)):DIL(+%)=DL, 1:1))" _$S
($D(DICS):" " _DICS, 1:"")
"RTN", "DIP0", 67, 0)
S DIC("W")="W:$P(^ (0), U, 2) "" multiple)"" I $T(DICW^DIALOGZ)]"" D
DICW^DIALOGZ(DL)
"RTN", "DIP0", 68, 0)
S DIC(0)="ZE" _$E("O", $D(DIPP)#10) Q
"RTN", "DIP0", 69, 0)
;
"RTN", "DIP0", 70, 0)
DIC D SETDIC, ^DIC, DIP^DIQQ Q
"VER")
8.0^22.0
**END**
**END**

```