[9.10] Group fraction digits for easier reading

If you often work with numbers with many digits, it is tedious and tiresome to read the numbers and copy them without making a mistake. These two routines help somewhat, by converting the number to a string with spaces between the fractional digits. I show two routines:

grpfrc(x,f)	Is a function that returns *x* as a string, with groups of *f* digits. This routine is best used in a program, for displaying results.
grpfrcan()	Shows the latest answer in the history display as a string with grouped fraction digits. This routine is best used from the command line, to display the last answer. The number of group fraction digits is fixed, but can be easily changed in the code.

Some examples:

grpfrc(1.23456789012,3)	returns	"1.234 567 890 12 E0"
grpfrc(1.23456789012,4)	returns	"1.2345 6789 012 E0"
grpfrcan()	returns	"1.414 213 562 37 E0"
	when 1.41421356237 is the first item in the history display	

My examples show the exponent 'E0' because I had the Exponent mode set to Engineering, but these routines work for any Exponent mode setting.

The code for *grpfrc():*

```
grpfrc(x,g)
func
©grpfrc() - Group fraction digits
©12 May 99/dab
©dburkett@infinet.com
©x: number to format
©g: number of fraction digits in group

local n,dp,el,ex,mn,fn,f1

©Convert argument to string
string(x)→n

©Initialize exponent string
""→ex

©Find decimal point and ᴇ, if any
instring(n,".")→dp
instring(n,"ᴇ")→El

©Get exponent or set end of string
if El≠0 then
right(n,dim(n)-El+1)→ex
else
dim(n)+1→El
endif

©Get mantissa & fraction strings
left(n,dp)→mn
mid(n,dp+1,El-dp-1)→fn

©Separate fraction digits with space
""→f1
```

```
while dim(fn)>g
 f1&left(fn,g)&" "→f1
 right(fn,dim(fn)-g)→fn
endwhile

©Build & return final result
mn&f1&fn&" "&ex

Endfunc
```

And the code for *grpfrcan(),* which just calls *grpfrc()* with the latest history area result:

```
grpfrcan()
func
©Group fraction digits of ans(1)
©2 nov 99/dab
©dburkett@infinet.com
grpfrc(expr("ans(1)"),3)
Endfunc
```

Note that the first argument to *grpfrc()* in *grpfrcan()* must be *expr("ans(1)"),* not just *ans(1),* as you might expect. Otherwise, *grpfrcan()* just grabs the first history result the first time you run it, then rudely *modifies* your code with that answer, and always returns the same result.

Change the second argument, 3, to a different number of fraction digits if needed.