

Users' Guide to
PROGRAMMER-SPECIFIED FORMATTING

This document describes the *RUNTIME* function

$\forall RESULT \leftarrow [FORMAT [FORMAT] SOURCE$

Elements in the *SOURCE* variable are converted, one-by-one, in the manner specified by the *FORMAT* character vector, into the *RESULT* character array.

If the *SOURCE* variable is a scalar, vector or matrix, then *RESULT* will be a scalar, vector or matrix, respectively. The non-optional *FORMAT* vector consists of one or more format-specifications separated by a comma or a semicolon. Each format-specification contains a field width. Thus, when *SOURCE* is a matrix the fields of *RESULT* will line up in tabular fashion.

If the elements in *SOURCE* (or a row of *SOURCE*) are exhausted before the end of *FORMAT* is reached, $[FORMAT$ continues to process format-specifications until one of the following conditions occur:

- 1) an *I*, *F*, *E* or *A* format-specification is encountered, or
- 2) a semicolon separator is encountered, or
- 3) the end of *FORMAT* is reached.

If *SOURCE* is not exhausted when the end of *FORMAT* is reached, the *FORMAT*-scanner "wraps around" to the beginning of *FORMAT* .

The *FORMAT*-scanner is repositioned to the beginning of *FORMAT* whenever *SOURCE* is a matrix and a new row is encountered.

Format-specifications

NOTATION	TYPE	DESCRIPTION
rqFw.d	fixed-point	field width = w decimal places = d
rqIw	integer (no-point)	field width = w
rEw.s	floating-point	field width = w sig. digits = s
rAw	right justified character	field width = w
rXw	blanks	field width = w
r"text"	literal text	field width = # characters between quotation marks

PREFIXES

r	optional repetition factor
q	optional qualifier(s)

QUALIFIERS

B	blankout all but background, if zero
C	comma insertion
L	left justify
Z	leading zeros
M"text"	floating left appendage, if negative (defaults to minus sign)
N"text"	floating right appendage, if negative
P"text"	floating left appendage, if zero or positive
Q"text"	floating right appendage, if zero or positive
R"text"	background (defaults to blanks)
_"c"	oversignificance character (defaults to underbar)

The `⎕FORMAT RUNTIME` function is patterned after the `ΔFMT APL*PLUS` primitive. The main deviations from `ΔFMT` are:

- *SOURCE* may be a mixed-type array, but may not be a list of arrays.
- literals in *FORMAT* must be surrounded by umlauts or quotation marks, not quote-quads.
- the negative numbers representation in *RESULT* usually contains `(-)` instead of `(-)`.
- *RESULT* is a vector whenever *SOURCE* is a scalar or vector.
- no more than six significant digits will appear in *RESULT*.
- *F*-formats that overflow field are not necessarily '*' d out.
(see *F*-format Note, below)
- `⎕FORMAT` generates descriptive error messages, which are sent to the `⎕IO_BOXES[2]` mailbox.

F-format Note

When a number cannot fit in the specified *F*-format, `⎕FORMAT` attempts to fit the number in a *F*-format field with the same width, w, but fewer decimal places. If even `Fw.0` will not fit, then `Iw` is attempted. If `Iw` overflows, `⎕FORMAT` gives up, and '*'s out the entire field.

Here is an APL program that accepts *FORMAT* from the subject at his terminal and then types back the *RESULT*.

```

      ∇ FTEST
[1]   ⎕ ← ⎕FORMAT[ ⎕ ] 3 2p-65432.5 -10 1.4 0 100 .0005
[2]   → 1
      ∇

```

F8.1
 -65432.5 -10.0
 -0.4 0.0
 100.0 0.0

F7.1
 -65432. -10.0
 -0.4 0.0
 100.0 0.0

F6.1
 -65432 -10.0
 -0.4 0.0
 100.0 0.0

F5.1
 *****-10.0
 -0.4 0.0
 100.0 0.0

M"-\$"P"\$"F10.2
 -\$65432.5 -\$10.00
 -\$0.40 \$0.00
 \$100.00 \$0.00

"0"CF10.2
 -65,432.50 -10.00
 -0.40 0.00
 100.00 0.00

M"("N")"Q" "F10.1
 (65432.5) (10.0)
 (0.4) 0.0
 100.0 0.0

N" DB"Q" CR"M"BF14.2
 65432.5 DB 10.00 DB
 0.40 DB
 100.00 CR

CL19
 -65,432 -10
 0 0
 100 0

R"= "19
 = -65432= = -10
 = = = 0= = = 0
 = = = 100= = = 0

N" "19,E10.2
 ***** -1.0E1
 0 0.0E0
 100 5.0E-4

N" "120,E10.2
 -65432 -1.0E1
 0 0.0E0
 100 5.0E-4