MUMPS Development Committee

Extension to the MDC Standard
Type A Release of the MUMPS Development Committee

Sign of Zero in $FN

March 23, 1996

Produced by the MDC Subcommittee #13
Data Management and Manipulation

Ed de Moel, Chairman
MUMPS Development Committee

Dan Bormann, Chairman
Subcommittee #13

1. Identification of the Proposed Change

1.1 Title

Sign of Zero in $FN

1.2 MDC Proposer and Sponsor

Proposer
SC13 Data Management and Manipulation
Chair: Dan Bormann

Sponsor
Frederick L. Hiltz
Brigham and Women's Hospital
10 Vining Street
Boston, MA 02115-6198
617-732-7028
fhiltz@bics.bwh.harvard.edu

1.3 Motion

None. Final version approved by MDC on March 23, 1996. This document supersedes X11/SC13/95–27.

1.4 History

23 Mar 96	X11/SC13/95–27	Accepted by MDC as type A, 19:2:6
25 Oct 95	X11/SC13/95–8	SC13 accepted as type A, 20:0:4.
03 Jun 95	X11/SC13/95–8	Proposed by Frederick Hiltz. SC13 accepted as type B, 20:0:3.
08 Feb 95	X11/95–20	Interpretation published.
29 Jan 95	X11/TG17/94–8	Interpretation accepted by MDC, 29:0:2.
13 Dec 94	X11/TG17/94–8	Accepted by the Interpretations Task Group without dissent.

1.5 Dependencies

This proposal modifies ANSI/MDC X11.1–1995.

Proposals that depend on this proposal: none.

2. Justification of the Proposed Change

2.1 Needs

National Institute of Standards and Technology (NIST) has questioned whether $FN(numexpr,+) should contain a plus sign when the value of numexpr is zero. (X11/94–44, problem statement ID#45, "Ambiguity in Whether Zero is Positive or Negative in $FNUMBER.")

2.2 Existing Practice in Area of the Proposed Change

7.1.5.6 specifies how $FN edits the plus or minus sign:

+   Force a plus sign ("+") on positive values of numexpr. Position of the "+" (leading or trailing) is dependent on whether or not fncodatom of "T" is specified.

–   Suppress the negative sign "–" on negative values of numexpr.

Other clauses refer to the sign of zero:

7.1.1 Variables – "Subscripts can be positive, negative, or zero; they can be integer or noninteger numbers as well as nonnumeric strings (other than empty strings)."

7.1.4.3 Numeric Data Values –

    d. The number zero is represented by the one-character string "0".

    e. The representation of each positive number contains no "–".

    f. The representation of each negative number contains the character "–" ...

    g. The representation of each positive integer contains only digits and no leading zero.

7.1.4.4 Meaning of numlit – "Note that numlit denotes only nonnegative values."

8.1.5 Timeout – "If the optional timeout is present, the value of numexpr nust be nonnegative."

8.2.5 FOR – "b. ... and $numexpr_2$ is nonnegative." "c. ... and $numexpr_2$ is negative."

An informal survey of some products in December 1994 showed:

Product	$FN(0,"+")
InterSystems	"+0"
Datatree	"+0"
Digital	"0"
Micronetics	"+0"
Greystone	"+0"
PFCS	"+0"

The MDC has interpreted the standard as described in 3.

2.3 Justification

The standard consistently applies the common arithmetic convention that zero is neither positive nor negative.

Any other convention for the sign of zero would clearly contradict 7.1.4.3.

Absent any mention of zero, then, the "+" editing action of $FNUMBER does not apply when the value of numexpr is zero.

3. Description of the Proposed Change

3.1 General Description of the Proposed Change

The "+" editing action of $FN does not apply when the value of numexpr is zero. "Positive" and "negative" are given explicit definitions.

3.2 Annotated Examples of Use

```
W $FN(0,"+")
0
```

3.3 Formalization

Amend ANSI/MDC X11.1–1995. Insert the following definitions into clause 4:

    4.xx negative: A numeric value less than zero. Zero is not negative.

    4.yy positive: A numeric value greater than zero. Zero is not positive.

Insert the following sentence at the end of the first text paragraph in clause 7.1.5.6:

    Note: Zero is neither positive nor negative.

4. Implementation Effects

4.1 Effect on Existing User Practices and Investments

Users should verify the behavior of their implementation. Routines that are intended to be portable and that depend on the functions specified herein may need to be changed.

4.2 Effect on Existing Vendor Practices and Investments

Vendors should verify their implementations and notify their customers of nonconformance or of changes made to achieve conformance.

4.3 Techniques and Costs for Compliance Verification

This interpretation agrees with the conformance test MVTS V.8.2 from MUMPS Systems Laboratory as quoted in the NIST question (see 2.1) and its accompanying letter to the MDC.

4.4 Legal Considerations

None.

5. Closely Related Standards Activities

5.1 Other X11 Proposals Under Consideration

X11/SC13/96–4 Leading Zero in $FN also amends the text of 7.1.5.6.

5.2 Other Related Standards Efforts

None.

5.3 Recommendations for Coordinating Liaison

None.

6. Associated Documents

None.

7. Issues, Pros and Cons, and Discussion

The Interpretations Task Group interpreted the standard as written. This proposal makes that interpretation an explicit change to the standard; however it also presents an opportunity to amend that standard.

7.1 January 1995 MDC meeting

No pros or cons offered.

7.2 June 1995 MDC meeting

Pro	Con
1 Answers NIST question (3)	1 Makes most implementors look like idiots? (1)
2 Disambiguates standard (3)	2 May break existing code (3)
3 Makes standard idiot-proof (2)	
4 May fix broken code (1)	

7.3 October 1995 MDC meeting

Pro	Con
1 Mathematically correct (4)	1 Redundant (0)
2 Clarifies NIST issue (6)	

7.4 March 1996 MDC meeting

Pro	Con
1 Mathematically correct (3)	1 May break existing code (3)
2 Clarifies NIST issue (3)	2 Redundant (0)
3 May fix broken code(1)	

(Number of citations in the vote.)