X11/1998-30

MUMPS Development Committee

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

NEW $REFERENCE

June 28, 1998

Produced by the MDC Subcommittee #13
Data Management and Manipulation

Art Smith, Chairman
MUMPS Development Committee

Dan Bormann, Chairman
Subcommittee #13

The reader is hereby notified that the following MDC specification has been approved by the MUMPS Development Committee but that it may be a partial specification that relies on information appearing in many parts of the MDC Standard. This specification is dynamic in nature, and the changes reflected by this approved change may not correspond to the latest specification available.

Because of the evolutionary nature of MDC specifications, the reader is further reminded that changes are likely to occur in the specification released, herein, prior to a complete republication of the MDC Standard.

1. Identification

1.1 Title:

NEW $REFERENCE

1.2 MDC Proposer and Sponsor:

Proposer:
Ben Bishop
64 Maolis Road
Nahant, MA 01908
(781) 593-3038
aci@shore.net

Sponsor:
SC13/TG10 Naked Indicator & $REFERENCE
(disolved?)

1.3 Motion:

Final publication version, superseding X11/SC13/1998-4.

1.4 History:

Date	Document	Action	
01 Aug 1998	X11/1998-30	Final publication version	
01 May 1998	X11/SC13/1998-4	Proposed as MDC/Type A	Passed: 14:0:5
01 Aug 1997	X11/SC13/1997-10	Proposed as SC13/Type A	Passed: 11:5:3
01 Jan 1997	X11/SC13/1997-3	Proposed as SC13/Type B	Passed: 12:0:4
01 Sep 1993	X11/SC13/TG10/1993-4	Separated from 'NEW svn Additions'	
		Proposed as SC13/Type B:	Not addressed
20 Oct 1992	X11/SC15/TG9/1992-3	Proposed as SC15/Type B:	Failed: 7-14-5
01 Oct 1992	X11/SC15/TG9/1992-2	Interim document using NEW svn formalism	
01 Sep 1992	X11/SC15/TG9/1992-1	Initial proposal with excessive formalism.	

1.5 Dependencies:

No proposals have been identified which depend on this proposal.
No proposals have been identified upon which this proposal depends.

2. Justification

2.1 Needs

In order to provide true library utilities and functions, there needs to be some means for saving (and restoring) the state of the naked indicator during subroutine and function calls.

2.2 Existing Practice

Programs which make use of library functions cannot make any assumptions about the state of the naked indicator (or $Reference) after the library function is called. One could *stub* a subroutine with code such as the following, but it is not a general solution:

```
Sub(a)  New REF Set REF=$REFERENCE DO
        . New REF Do Sub1(.a)
        I REF]"",$D(@REF) ;is SET $REFERENCE=REF possible?
        E  ;some mechanism to unset $REFERENCE (S $R="" ?)
        Q
Sub1(a) ;original subroutine
```

This example preserves the value of $REFERENCE, but at the cost of STACK-LEVELs and simplicity.

3. Description

3.1 General description

Add $REFERENCE to the list of svns permitted to be NEWed.

3.2 Annotated Examples of Use

```
GO      If $Data(^Test(1234)) ;sets $R
        Do Test1 ;test which uses NEW $R
        W !,"$REFERENCE should equal ^Test(1234), $REFERENCE="_$R
        Do Test2 ;test which does not use NEW $R
        W !,"$REFERENCE should not equal ^Test(1234), $REFERENCE="_$R
        Quit
Test1   NEW $REFERENCE ;save the existing value of $REFERENCE
Test2   ;entry point where NEW $REFERENCE is not performed
        If $Data(^Failure(4321)) ;should set $R
        Quit  ;for Test1 this should restore the saved value of $REFERENCE
```

3.3 Formalization (references are to X11.1-1995 standard)

- **To section 8.2.14 (NEW) add to the list of svns permitted in newsvn:**

```
                    ...
newsvn ::=      $R[EFERENCE]
                    ...
```

- **Add a new paragraph (numbered appropriately) after paragraph '2' of subclause 'd' (NEW svn):**

  *) If the argument specifies $R[EFERENCE], points to a DATA-CELL with a value copied from the prior DATA-CELL (as pointed to by the just-copied NAME-TABLE entry)

4. Implementation Effects

4.1 Effect on Existing User Practices and Investments

None expected; no backward compatibility issues have been identified with this change.

4.2 Effect on Existing Vendor Practices and Investments

None expected.

4.3 Techniques and Costs for Compliance Verification

One could use the example program provided in section 3.2.

4.4 Legal Considerations

None identified.

5. Closely Related Standards Activities

5.1 Other X11 Proposals Under Consideration

None.

5.2 Other Related Standards Efforts

None.

5.3 Recommendations for Coordinating Liaison

None.

6. Associated Documents

X11/1994-47 MDC/A NEW svn Addition: $TEST

7. Issues, Pros and Cons, and Discussion

7.1 September 1992 'NEW svn Additions'
Initial proposal: creation of $TEST/block structuring Task Group (SC15/TG9)

7.2 October 1992 'NEW svn Additions'

Restructured formalism to use the 'NEW svn' formalism of the Error Processing proposal (X11/SC15/1992-27)
Proposed as SC15/Type B Failed 7-14-5
Pro: Needed for better extrinsic functions
Con: 1. should address $IO [4] 2. $D/$K/$X/$Y not handled as arrays [2]
3. $D/$K/$X/$Y should reflect current state [12] 4. NEW $TEST ineffective [1]

An attempt to divide the issue is being made by presenting separate proposal for the different svns. Con 1 (should address $IO) was voted on in a straw poll, losing 2-1. The issues of CON 2 and 3 center on the fact that for a specific device/$IO, there is an array of values being stored (the svns just being conceptual 'subscripts') – however, since one can SET the individual IO-related svns, I see no reason to prevent them from being NEWed – one culd accomplish the same objective in a simple (albeit *ugly*) set of code:

```
Instead of:
     New $X
One uses:
     New XXX Set XXX=$X Xecute ("New XXX Do newlabel") Set $X=XXX Quit
     newlabel;routine continues on
```

Granted, exfuncs and exvars would need to return a value, but I hope the point is clear: the mechanics for arbitrarily changing these svns is already available within the standard; being able to NEW them does not change that, it just makes certain actions more concise and understandable.

7.3 September 1993 'NEW svn Addition: $REFERENCE'

Initial proposal (NEW svn additions) broken into component parts; individual proposals for $TEST, $REFERENCE, $X/$Y, $DEVICE, $KEY. Tabled in subcommittee; not addressed due to lack of time.

7.4 March 1997, Proposed as SC13/B Passed: 12:0:4

No Cons. Pro: Well defined.

7.5 September 1997, Proposed as SC13/A Passed: 11:5:3

Pro	1. Allows for safer routines	[6]
	2. Eliminates a Kludge, useful for tools	[7]
	3. Implementable	[4]
Con	1. Not implemented	[5]
	2. Benefit not outweighed by cost	[6]

Comments: Con 1: Since this was going up for SC13/A status, I would be surprised if there were any implementations of this proposal; Vendors are not godlike entities. Con 2: My short discussions with vendors has indicated that this would not be a big deal.

7.6 June 1998 Proposed as MDC/A Passed 14:0:5
Pros:
1. Useful for better extrinsic functions

8. Glossary

None.

9. Appendix

None.