X11/94-47

MUMPS Development Committee

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

NEW svn Addition: $TEST

June 1994
Produced by the MDC Subcommittee #15
Programming Structures

Jamie Crumley, Chairman
MUMPS Development Committee

Kate Schell, Chairman
Subcommittee #15

1. Identification

1.1 Title: NEW svn Addition: $TEST

1.2 MDC Proposer and Sponsor:

Proposer:
Ben Bishop
64 Maolis Road
Nahant, MA 01908
aci@shore.net

Sponsor:
SC15/TG9 Routine Structure
Ben Bishop, Chair
Atlantic Consultants, Inc.
aci@shore.net

1.3 Motion:

Subcommittee 15 moves to elevate this proposal to MDC Type A status.

1.4 History:

Date	Document	Action
01 Dec 94	X11/94-47	MDC/A
20 Apr 94	X11/SC15/94-13	Proposed as MDC/Type A (Passed: 34-5-4)
01 Feb 94	X11/SC15/94-7	Proposed as SC15/Type A (Passed: 18-2-5)
25 Oct 93	X11/SC15/TG9/93-9	Revised, proposed as SC15/Type B (Passed: 12-7-4)
20 Oct 92	X11/SC15/TG9/92-3	Proposed as SC15/Type B (Failed: 7-14-5)
01 Oct 92	X11/SC15/TG9/92-2	Interim document using NEW svn formalization
01 Sep 92	X11/SC15/TG9/92-1	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 selected svn's (system variables) and restoring them when the subroutine/function is completed.

2.2 Existing Practice

The existing practice is to make no assumptions about selected system variables when calling a subroutine. Although extrinsic functions currently stack the value of $Test, other state variables could inadvertently be changed by the function. Alternatively, programmers would require absolute knowledge about the side effects of subroutines being used—hindering modification and maintenance.

3. Description

3.1 General description

With the "Error Processing" proposal (X11/SC15/92-27), selected svns are now permitted to be used with the NEW command. In addition to $ETRAP and $ESTACK, this proposal will add $TEST to the list of svns which can be NEWed.

3.2 Annotated Examples of Use

```
GO    If 1 Do TEST Write  !,"$TEST should equal 1,
      $TEST="_$TEST
      Else  Write  !,"This should not print;  $TEST="_$TEST
      Q
TEST  New $TEST ;save the existing value of $TEST
      If 0   ;$TEST should now be equal to 0
      Q      ;this will restore the 'new'd value of $TEST
```

3.3 Formalization (References are to the X11.1-1994 Canvass Document)

To section 8.2.14 'NEW', add to the list of svns permitted newsvn:

```
                    |    . . .    |
      newsvn  ::=   |   $T[EST]   |
                    |    . . .    |
```

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

3) If the argument specifies $TEST, 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; there is no backward incompatibility issue with this addition.

4.2 Effect on Existing Vendor Practices and Investments

None expected.

4.3 Techniques and Costs for Compliance Verification

Create a subroutine which modifies $TEST (i.e. IF '$TEST); compare the value of $TEST before and after calling this subroutine, as well as a copy of the subroutine with 'NEW $TEST' placed as the first .command in the subroutine. $TEST should not change in the second version.

This testing subroutine could be written as follows:

```
GO    If 1 Do TEST Write  !,"$TEST should equal 1,
      $TEST="_$TEST
      Else Write  !,"This should not print; $TEST="_$TEST
      Q
TEST  New $TEST ; save the existing value' of $TEST
      If 0 ;$TEST should now be equal to 0
Q     ;this will restore the  'new'd value of $TEST
```

4.4 Legal Considerations

None identified.

5. Closely Related Standards Activities

5.1 Other XII Proposals Under Consideration

None.

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

7.1 September 1992

Initial proposal; creation of $TEST/Block structuring Task Group.

7.2 October 1992

Restructured formalism to use the 'NEW svn' formalism of the Error Processing proposal (X11/SC 15/92-27). Proposed as SCI5 Type B: Failed (7-14-5)

Cons: 1. [4] Should address $IO Pro: 1. Needed for better extrinsic functions
2. [2] $D/$K/$X/$Y not handled as arrays
3. [12] $D/$K/SX/$Y should reflect current state
4. [1] NEW $TEST ineffective

An attempt to divide the issue is being made by presenting separate proposals for.the different svns. Con 1 (should address $IO) was voted on in a straw poll, losing 2-1. The issues of Con 2 centers on the fact that for a specific device/$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 could 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

Initial proposal (NEW svn additions) broken into component parts: individual proposals for $TEST, $REFERENCE, $X/$Y, $DEVICE, $KEY.

7.4 October 1993 Passed SC15/B 12-7-4

7.5 February 1994 Passed SC15/A 18-2-5

7.6 June 1994 Passed MDC/A 34-5-4

Pro: a) needed for better extrinsic functions a) Band-Aid fix to serious problem
b) needed for better subroutines

8. Glossary

None.

9. Appendix

None.