X11/SC13/94-33

1 Identification Of Proposed Change

1.1 Title:

Kill Data And Kill Subscripts of *glvn*s

1.2 MDC Proposer and Sponsor

David J. Marcus
Micronetics Design Corporation
1375 Piccard Drive
Rockville, Maryland 20850

tel: 301-258-2695
fax: 301-840-8943
email: djm@rios.mnetx.com

1.3 Motion

Recommend this document for elevation to MDC Type A.

1.4 History

June, 1994	X11/SC13	Response to Motion#1: Mr. Stenn's document is not available to author, nor was it used as input. No further action taken. Response to Motion#2: Document reference numbers are not known to author and hence cannot be complied with.
February, 1994	X11/SC13	Motion#1: Include reference in Section 6 to document by Harlan Stenn passed by voice vote. Motion#2: Include document references and numbers in History section passed by voice vote. Editorial changes only.
October, 1993	X11/SC13	Proposed for elevation to SC Type A Passed 14:2:1. Restored original KILL command (was inadvertantly deleted). Limited the naming of the kill subscripts/value commands to a single name for each. Clarified that exclusive KILL is limited to unsubscripted locals (wording used in MDC standard *did not* make this clarification, only the BNF did).

		Clarified that the argumentless KILL (kill all) is limited to locals). The MDC standard *did not* make this clarification!!
June, 1993		added formalization section. Passed as SC13 T:pe B.
March, 1993	X11/SC13	No vote. Document ruled to be in improper format. Formalization section required for Type B proposals.
	X11/SC13/TG11	Passed (2-0-0): Recommend elevation to SC Type B.
October, 1992	X11/SC13/TG11	This is the initial proposal.

1.5 Dependencies

None.

2 Justification of Proposed Change

2.1 Needs

The MUMPS Language does not allow deleting of a ***glvn***'s descendents using a simple (single) command. Additionally, MUMPS does not support deleting the value of a node without affecting the sub-nodes (descendents).

2.2 Existing Practice in Area of the Proposed Change

To delete sub-nodes of a ***glvn*** requires a MUMPS program which uses $ORDER to traverse the next level of sub-nodes and KILL them individually.

To delete the current value of a node requires a MUMPS program to use $ORDER to traverse the next level of sub-nodes and MERGE them into a temporary ***glvn***. The the entire node of the original ***glvn*** must be KILLed and then MERGEd back from the temporary. The final step is to KILL the temporary ***glvn***.

Clearly, neither of the above solutions are desirable. Certainly a LOCK is required in the case of a global (although it is not even clear that it is possible to use a LOCK if other parts of the application are not cooperatively using this specific LOCK).

3 Description of Proposed Change

3.1 General Description of the Proposed Change

Define two new commands, KVALUE (abbreviated KV) and KSUBSCRIPTS (abbreviated KS). The KV command kills the value portion of the *glvn* without affecting its subscripts (if any). Thus the $DATA value of the ***glvn*** changes so that it ends in a zero (a 10 if it was 11 or 0 if it was 1). The KS command kills the subscripts (if any) of the *glvn* without affecting its value. Thus the $DATA value of the ***glvn*** becomes a 0 (if it was 10) or a 1 (if it was an 11).

It should be obvious that the KS and KV together are equivalent to the KILL command.

3.2 Annotated Examples of Use

KVALUE ^ABC(3)
Clears any value of ^ABC(3) but does not affect presence of subscripts.

KSUBSCRIPTS MYVAR
Deletes any subscripts of MYVAR without affecting its value.

KV (A,B,C)
This is analogous to the exclusive KILL command. The value portion of all local variables other than A, B, and C would be cleared (KILLED).

KS (A,B,C)
Only the subscripts of all local variables other than A, B, and C would be KILLed.

3.3 Formalization

I-5.2.11 KILL Command

Replace: K[ILL]

With:

```
K[ILL]
KV[ALUE]
KS[UBSCRIPTS]
```

Replace: In the first sentence of this section (beginning with: The three argument forms of the KILL ...), add after the word KILL the words: KILL, KVALUE, and KSUBSCRIPTS

Replace: The entire 3rd paragraph begining with: "KILL is defined using a subsidiary function K(V) where V is a glvn." and the 3 subsections labeled: a) b) and c)

With:

KILL, KVALUE, and KSUBSCRIPTS are defined using a subsidiary function K(*V,val,subs*) where *V* is a glvn, *val* is 0 or 1, and *subs* is 0 or 1.

a) Search for the name *V* in the NAME-TABLE. If no such entry is found, the function is completed. Otherwise, extract the DATA-CELL pointer and proceed to step b.

b) If *val*=1 and *subs*=1 then in the DATA-CELL identified in step 'a':

1) Let N be the number of subscripts in *V*. If *V* is unsubscripted, let N be 0.
2) If N is 0, then delete all tuples. The function is completed.
3) Otherwise (if N > 0), delete all tuples of degree N or greater whose first N subscripts are the same as those in *V*. The function is completed.

c) If *val*=1 and *subs*=0 then in the DATA-CELL identified in step 'a':

1) If *V* is unsubscripted, delete the tuple of degree 0 (if found). The function is completed.
2) Otherwise, let N be the number of subscripts in *V*. Delete (if found) only the tuple of degree N whose first N subscripts are the same as those in *V*. The function is completed.

d) If *val*=0 and *subs*=1 then in the DATA-CELL identified in step 'a':

1) Let N be the number of subscripts in *V*. If *V* is unsubscripted, let N be 0.
2) Delete all tuples of degree N+1 or greater whose first N subscripts are the same as those in *V*. The function is completed.

Replace: The entire 4th paragraph begining with: "Note that as a result of procedure K ..."

With:

Note that as a result of procedure K(*V*,1,1), $D(*V*)=0, i.e., the value of *V* is undefined and *V* has no descendents.

Note that as a result of procedure K(*V*,1,0), $D(*V*)=0 if *V* had no descendents before procedure K was applied, or $D(*V*)=10 if *V* had descendents before procedure K was applied, i.e., only the value of *V* is deleted.

Note that as a result of procedure K(*V*,0,1), $D(*V*)=1 if *V* had a value before procedure K was applied, or $D(*V*)=0 if *V* had no descendents before procedure K was applied, i.e., only the descendents of *V* are deleted.

Replace: The entire section begining with: "The actions of the three forms of KILL are then defined as:
With:

a) Selective Kill
Apply procedure:
K(glvn,1,1), if KILL,
K(glvn,1,0), if KVALUE ,
K(glvn,0,1), if KSUBSCRIPTS.

b) Exclusive Kill
For all names, *V*, in the locals NAME-TABLE except those in the argument list, apply procedure:
K(*V*,1,1), if KILL,
K(*V*,1,0), if KVALUE ,
K(*V*,0,1), if KSUBSCRIPTS.
Note that the names in the argument list of an exclusive kill are restricted to unsubscripted locals.

c) Kill All
For all names, *V*, in the locals NAME-TABLE, apply procedure:
K(*V*,1,1), if KILL,
K(*V*,1,0), if KVALUE ,
K(*V*,0,1), if KSUBSCRIPTS.
Note that 'kill all' applies the 'K' procedure to the local variable NAME-TABLE only.

4 Implementation Effects

4.1 Affect on Existing User Practices and Investments

Provide significant simplification.

4.2 Affect on Existing Vendor Practices and Implementations

None.

4.3 Techniques and Costs for Compliance Verification

MUMPS program to check $DATA after executing KVALUE and KSUBSCRIPTS. The following routine is offered as an example of such:

```
TESTKILL ;DJM;
         NEW  D SETUP            ;start with known locals
         KS A,B,C                ;KSUBSCRIPTS inclusive
         I $D(A)'=1 D ERROR(1)   ;'A' should have remained unchanged
         I $D(B)'=0 D ERROR(2)   ;'B(1)' should have been deleted
         I $D(C)'=1 D ERROR(3)   ;Only 'C(1)' should have been deleted
         D SETUP
         KV A,B,C                ;KVALUE inclusive
         I $D(A)'=0 D ERROR(4)   ;'A' should have been deleted
         I $D(B)'=10 D ERROR(5)  ;'B' should have remained unchanged
         I $D(C)'=10 D ERROR(6)  ;Only 'C' should have been deleted
         D SETUP
         KS (A,B)                ;KSUBSCRIPTS exclusive
         I $D(A)'=1 D ERROR(7)   ;'A' should have remained unchanged
         I $D(B)'=10 D ERROR(8)  ;'B(1)' should have remained unchained
         I $D(C)'=1 D ERROR(9)   ;Only 'C(1)' should have been deleted
         D SETUP
         KV (A,B)                ;KVALUE exclusive
         I $D(A)'=1 D ERROR(10)  ;'A' should have remained unchanged
         I $D(B)'=0 D ERROR(11)  ;'B(1)' should have remained unchanged
         I $D(C)'=10 D ERROR(12) ;Only 'C' should have been deleted
         W !,"End." Q
ERROR(N) W !,"Error #",N Q
SETUP    S A=1,B(1)=1,C=1,C(1)=1 ;variables with $d=1, 10, 11
         Q
```

4.4 Legal Considerations

None.

5 Closely Related Standards Activities

5.1 Other X11 Proposals Under Consideration

None (SET $DATA proposal was withdrawn).

5.2 Other Related Standards Efforts

None.

5.3 Recommendations for Coordinating Liaison

None.

6 Associated Documents

None.

7 Glossary

None.

8 Appendix

None.