X11/96-10

MUMPS Development Committee

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

Reverse $Query

October 26, 1995

Produced by the MDC Subcommittee #13
Data Management and Manipulation

Ed de Moel, 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:

Reverse $Query

1.2 MDC Proposer and Sponsor:

Proposer:
Ben Bishop
64 Maolis Road
Nahant, MA 01908
(617) 593-3038
aci@shore.net

Sponsor:
SC13/TG13 MUMPS Data Traversal
Rod Dorman, Chair
Polylogics Consulting
(201) 489-4200
rodd@panix.com

1.3 Motion:

None (final version of document), superseding X11/SC13/95-22.

1.4 History:

Date	Document	Action	
01 Feb 96	X11/96-10	Final publication version	
31 Aug 95	X11/SC13/95-22	Proposed as MDC/A	(passed 28:0:2)
19 Apr 95	X11/SC13/95-14	Proposed as SC13/A	(passed 22:0:3)
01 Dec 94	X11/SC13/TG13/94-7	Initial proposal, SC13/B	(passed 15:0:4)

1.5 Dependencies:

The proposal X11/SC13/TG16/95-4 (User Defined ssvns) refers to the second argument of $Query and therefore depends on this proposal. No proposals have been identified upon which this proposal depends.

2. Justification

2.1 Needs

There are times where reverse collation order is useful and more efficient than forward collation order. Using $Query instead of $Order is, at times, more efficient as well. At present it is possible to reverse collate using $Order, but not with $Query; the intent of this proposal is to remove that asymmetry.

2.2 Existing Practice

$Order provides the capability of going backwards; just as one could emulate the process of forward $Query using $Order, one could emulate reverse $Query using reverse $Order. That mechanism is cumbersome and difficult to use.

3. Description

3.1 General description

Add a second argument to the $Query function which would be either 1 or -1 (as with the $Order function). No second argument defaults to 1. 1 means $Query forward. -1 means $Query backward.

3.2 Annotated Examples of Use

```
S REF=$R,GLO="" S:REF]"" GLO=$Q(@REF,-1) S:GLO="" GLO=$P(REF,"(")
I GLO]"" F  GLO=$Q(@GLO,1) Q:GLO=""  DO  ;walk through the remainder of GLO
. ;do something with each of the nodes on/after REF

S REF="^GLO" F  S REF=$Q(@REF,-1) Q:REF=""  DO  ;^GLO nodes in reverse order
. ;do something interesting here
```

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

- **Add to the end of subclause I.7.1.5.15 ($Query), but before the paragraph beginning "If the value of ...":**

$Q[UERY] (glvn , expr)

Let S be the value of expr. Then $QUERY(glvn,expr) returns:

1) If S = 1, the function returns a result identical to that returned by $QUERY(glvn).

2) if S = -1, the function returns a value which is either the emptystring ("") or a namevalue according to the following steps:

   a) Let glvn be a variable reference of the form Name(s_1,s_2,...,s_q) where s_q may be the empty string. If glvn is unsubscripted, initialize *V* to the form Name(""); otherwise, initialize *V* to glvn.

   b) if the last subscript of *V* is empty, Goto step e.

   c) if $D(*V*)\10=1, append the subscript "" to *V*, i.e. *V* is Name(s_1,s_2,...,s_q,"").

   d) If *V* has no subscripts, return "".

   e) Let s = $O(*V*,-1).

   f) if s = "", truncate the last subscript off *V*, Goto step d.

   g) if s' = "", replace the last subscript of *V* with s.

   h) if $D(*V*)#2 = 1, return *V* formatted as namevalue.

   i) Goto step c.

3) Values of S other than 1 or -1 are reserved for future extension of the $QUERY function.

- **In the paragraph beginning "If the value of ...", replace $QUERY(glvn) with $QUERY(glvn[,expr]) so that the paragraph should read:**

If the value of $QUERY(glvn[,expr]) is not the empty string, and glvn includes an environment, then the namevalue shall include the environment; otherwise the namevalue shall not include an environment.

4. Implementation Effects

4.1 Effect on Existing User Practices and Investments

None expected.

4.2 Effect on Existing Vendor Practices and Investments

Costs of implementation.

4.3 Techniques and Costs for Compliance Verification

None.

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

None.

7. Issues, Pros and Cons, and Discussion

Has been implemented (IBM/VM or Micronetics?)

January 1995, Albuquerque, NM: Proposed as SC13/B, passed 15:0:4
No cons.

June 1995, Chicago, IL: Proposed as SC13/A, passed 22:0:3
2 cons: (a) Proposal incomplete (compliance with vendor impact) and (b) Suboptimal statement of algorithm.
Responses to cons: (a) I have been unable to identify any impact on *existing* vendor practices and investments. Aside from the cost of actually implementing this proposed change, which is implicitly understood as not applying to section 4.2, it is unlikely that a vendor will be much effected by this proposal. (b) The algorithm as presented directly mimics the forward algorithm as it is presented for the single argument $Query in the standard; modifying the original algorithm is beyond the scope of this proposal, and I feel consistency would be more acceptable.

October 1995, New Orleans, LA: Proposed as MDC/A, passed 28:0:2
No cons. Addition of dependence to X11/SC13/TG16/95-4. Corrected spelling of Rodd [sic] Dorman's name.

8. Glossary

None.

9. Appendix

None.