X11/96-44

MUMPS Development Committee

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

Improve mnemonicspace Handling

March 23, 1996

Produced by the MDC Subcommittee #12
Environment

Ed de Moel, Chairman
MUMPS Development Committee

Larry Ruh, Chairman
Subcommittee #12

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 which 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 with 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:

Improve mnemonicspace Handling

1.2 MDC Proposer and Sponsor:

Proposer:
Jon Diamond
Hoskyns Group

Sponsor:
SC12/TG9, General Device Issues
Larry Ruh, Chair

1.3 Motion:

None. Final publication version superseding X11/SC12/96-5.

1.4 History:

Date	Document	Action
Aug 96	X11/96-44	Final publication version.
Mar 96	X11/SC12/96-5	Proposed as MDC Type A — Passed: 21:0:5
Oct 96	X11/SC12/95-12	Proposed as MDC Type A – Withdrawn due to Section 4 problems.
Jun 95	X11/SC12/TG9/95-8	Proposed as SC12 Type A — Passed: 3-0-4
Mar 95	X11/SC12/TG9/95-4	Discussed at MDCC-E
Jan 95	X11/SC12/TG9/95-2	Proposed as SC12 Type B — Passed: 4-0-2
Jan 95	X11/SC12/TG9/95-1	Proposed as SC12 Type B – Modified to add mnemonicspace information.
Jun 94		Discussed in SC14/TG6 (TCP/IP)

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

Some I/O bindings and mnemonicspace creators have found limiting capability in feedback from certain conditions and would like to enhance the general capability of the language. Of specific interest is the retrieval of label information from magnetic tapes, etc.

In addition there is no feedback as to the current mnemonicspace in effect for the device.

2.2 Existing Practice

None known.

3. Description

3.1 General description

The proposal is to allow $KEY to have values other than as a result of a READ command and provide access to mnemonicspace information.

3.2 Annotated Examples of Use

None currently (examples of using state information?).

3.3 Formalization

- **Add to the description of $KEY in 7.1.4.10 after the second para:**

```
Certain mnemonicspaces may also specify that $KEY contains values as a
result of other I/O commands.
```

- **Add two new nodes to the ^$DEVICE ssvn (7.1.3.2):**

```
^$DEVICE( devicexpr , expr V "MNEMONICSPACE" ) = mnemonicspace

This node identifies the mnemonicspace currently in effect for the device.
If there is no mnemonicspace in effect then this node has the value of the
empty string.

^$DEVICE(devicexpr,expr V "MNEMONICSPEC",expr V mnemonicspace) = emptystring

This node identifies a mnemonicspace that has been associated with the
device through the OPEN and USE commands.  All nonempty string values are
reserved for future extension of the standard.
```

4. Implementation Effects

4.1 Effect on Existing User Practices and Investments

None expected; there are no mechanisms currently in place for determining the mnemonicspaces associated with a device -- so this does not invalidate existing practices.

4.2 Effect on Existing Vendor Practices and Investments

None expected beyond implementation costs. The information which this proposal makes available has to be available to the implementation -- this just makes it available for inspection to the user.

4.3 Techniques and Costs for Compliance Verification

Open a device with a list of mnemonicspaces -- make sure they appear in ^$DEVICE(*,"MNEMONICSPEC") as expected. Use each mnemonicspace with the USE command and make sure the appropriate value is returned in the ^$DEVICE(*,"MNEMONICSPACE") node.

4.4 Legal Considerations

None identified.

5. Closely Related Standards Activities

5.1 Other X11 Proposals Under Consideration

TCP/IP binding (document # unknown).

5.2 Other Related Standards Efforts

None identified.

5.3 Recommendations for Coordinating Liaison

None.

6. Associated Documents

None Identified.

7. Issues, Pros and Cons, and Discussion

- **January 1995, Albuquerque** Proposed as SC12/B passed 4-0-2
  Pros: • Allows applications to find out current mnemonicspace.
  • Allows $KEY to provide other information

- **March 1995, MDCC-E meeting** Discussed
  The MNEMONICSPEC node seems to be ambiguous. If it contains all possible mnemonicspaces ever assigned to the device then there could be string length problems. It is recommended that this node be removed. The structure has been modified to avoid the problem.

- **June 1995, Chicago** Proposed for SC12/A passed 3-0-4
  Pros: • Allows applications to find out current mnemonicspace.
  • Allows $KEY to provide other information

- **October 1996, New Orleans** Proposed for MDC/A -- withdrawn due to incorrect format.

- **March 1996, Boston** Proposed for MDC/A passed 21:0:5
  Pros: • Allows applications to find out current mnemonicspace.
  • Allows $KEY to provide other information

8. Glossary

None.

9. Appendix

None.