X11/95-116

MUMPS Development Committee

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

^$JOB Device Information

June 4, 1995

Produced by the MDC Subcommittee #15
Programming Structures

Ed de Moel, Chairman
MUMPS Development Committee

Art Smith, Chairman
Subcommittee #15

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.

Anyone reproducing this release is requested to reproduce this introduction.

1. Identification

1.1 Title:

^$JOB Device Information

1.2 MDC Proposer and Sponsor:

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

Sponsor:
SC15/TG13 ssvn Syntax
Alan Frank, Chair
Matchups
alf@world.std.com

1.3 Motion:

None (final version of document), superseding X11/SC15/95-12.

1.4 History:

Date	Document	Action
31 Aug 95	X11/95-116	Final publication version
19 Apr 95	X11/SC15/95-12	Proposed as MDC/A Passed (18:9:10)
01 Dec 94	X11/SC15/94-33	Proposed as SC15/A Passed (13:4:5)
20 Apr 94	X11/SC15/94-26	Proposed as SC15/B Passed (12:10:5)

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

There is currently no mechanism to determine the list of devices which have been OPENed by a specific process. Use of ^$JOB to accomplish this goal would also allow implementors to provide access to this information for use in system status utilities and the like.

2.2 Existing Practice

There is no standard mechanism for determining which devices a process has OPENed but not yet closed. A number of vendors use custom $Z functions to provide this information, while others use the semi-standard $View function to walk through the implementation-specific structures which hold the information.

3. Description

3.1 General description

First, restructure the subsection for ^$JOB so that all ^$JOB characteristics are located in one section. Next, define the characteristic nodes "$PRINCIPAL" and "$IO" which hold the current device-state information for the process. The "OPEN" characteristic node has descendent subscripts providing the list of open devices associated with the process.

3.2 Annotated Examples of Use

```
;close all but the current and principal devices:
N IO S IO="" F  S IO=$O(^$JOB($J,"OPEN",IO)) Q:IO=""  DO
. I IO'=$IO,IO'=$P C IO ;close the device if neither current nor principal

;device status information for all jobs:
N PIO,CIO,IO,JOB,PP,CC S JOB="" F  S JOB=$O(^$JOB(JOB)) Q:JOB=""  DO
. S PIO=$G(^$JOB(JOB,"$P")),CIO=$G(^$JOB(JOB,"$IO"))
. W !,$J(JOB,4),?8
. W $S(PIO=CIO:"*p ",1:" p ")_PIO ;principal device
. I CIO]"",PIO'=CIO W !?8," * "_CIO ;current device
. S IO="" F  S IO=$O(^$JOB(JOB,"OPEN",IO)) Q:IO=""  DO
. . I IO'=CIO,IO'=PIO W !?8,"   "_IO
```

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

- **Reorganize Subsection 7.1.3.4 (^$JOB) so that all the $JOB related information is together and organized alphabetically by characteristic section; and placing the default environment section (7.1.3.9) at the end.**

7.1.3.4 ^$J[OB] (processid)

<general description of ^$JOB ssvn>

7.1.3.4.1 Characteristic: Character Set Profile

<section on character set profile characteristics>

etc for the additional sections.

- **Add the new subsection of 7.1.3.4 (^$JOB) defining device information within ^$JOB:**

7.1.3.4.* Characteristic: Devices

^$JOB(processid, expr V "$PRINCIPAL") =	devicexpr	(principal device)
^$JOB(processid, expr V "$IO") =	devicexpr	(current device)
^$JOB(processid, expr V "OPEN", devicexpr)		(for each OPENed device)

These nodes specify the device information associated with process processid. The node "$PRINCIPAL" is the value of $PRINCIPAL for that process. The node "$IO" is the value of $IO (current device being used) for that process. The devicexpr nodes beneath "OPEN" are the device identifiers for all the devices which are currently OPENed for that process.

4. Implementation Effects

4.1 Effect on Existing User Practices and Investments

None identified.

4.2 Effect on Existing Vendor Practices and Investments

None identified.

4.3 Techniques and Costs for Compliance Verification

Check if ^$JOB($JOB,"$PRINCIPAL") is the same value as $PRINCIPAL.
Check is ^$JOB($JOB,"$IO") is the same as $IO, including while switching between devices.
Open a number of devices, and make sure that all the open devices are included under ^$JOB($JOB,"OPEN",*) -- and make sure they are not there when they are closed.

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

X11/TG18	ssvn coordination

6. Associated Documents

X11/94-23	MDC/A	LIBRARY Proposal
X11/SC13/95-13	SC13/B	Local variables in ^$JOB

7. Issues, Pros and Cons, and Discussion

- June 1994, proposed as SC15/B (as amended), passed 12-10-5

Initial proposal to clean up ^$JOB definition and add device related nodes. The $P node was considered to be renamed $PRINCIPAL since $IO was fully spelled out, but the vote was for it to remain $P. $IO remained the current device value, but the sub-nodes (opened devices) were moved to the node "OPEN".

Pros: a. Only way to find list of open devices
b. Useful for standard jobexam-type utility
Cons: a. Inconsistent $P vs $IO
b. Those nodes are redundant with svns
c. Excess baggage

Response to cons: (a) The choice between $P and $PRINCIPAL was brought up in task group, and $P was the preferred value; $IO was also kept as $IO. (b) although these nodes are redundant within a single process, they do, however, facilitate possible examination from another job ('jobexam') and permits capturing process state information by a single MERGE command. (c) is a con which can never really be addressed: one person's clutter is another's work-saving tool; I've written job examination utilities and ^$JOB, providing the information for which it was created as in this example, would have made that task much easier.

January 1995, Albuquerque, NM: Proposed as SC15/A, passed 13:4:5

A non-substantive change was made to make $P be $PRINCIPAL

June 1995, Chicago, IL: Proposed as MDC/A, passed 18:9:10

Pros	Cons
1. Provides access to open devices	1. functionality provided using objects
2. Correct mechanism for info	2. section 4.2 incorrect
	3. redundant access to $I and $P

Notes on Cons: 1. objects don't exist yet, so it isn't. 2: section 4.2 ("effect on existing vendor practices and investments") is intended to identify impact on existing vendor implementations -- there is an implied understanding that this does *not* include the work needed to actually implement the change (which is obvious), but is intended to include prior work which may need to be discarded (or re-worked) because of this proposal. No such impact has been identified in relation to this proposal. 3: Although there is redundancy when referring to those svns for the current processid, this proposal permits potential access to this information for other processids.

8. Glossary

None.

9. Appendix

None.