

MUMPS Development Committee

**Extension to the MUMPS Language Standard
Type A Release of the MUMPS Development Committee**

Initialising Intrinsic

June 1993

**Produced by the MDC Subcommittee #15
Programming Structures**

**Thomas C. Salander, Chairman
MUMPS Development Committee**

**Kate Schell, Chairman
Subcommittee #15**

The reader is hereby notified that the following language 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 MUMPS specifications. 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 MUMPS specifications, the reader is further reminded that changes are likely to occur in the specification released herein prior to a complete republication of MUMPS specifications.

© Copyright 1994 by the MUMPS Development Committee. This document may be reproduced in any form so long as acknowledgment of the source is made.

Anyone reproducing this release is requested to reproduce this introduction.

1. IDENTIFICATION

1.1 Title

Initialising Intrinsic

1.2 MDC proposer and sponsor

Proposer: Jon Diamond, Hoskyns Group, 130 Shaftesbury Avenue, London W1V 7DN, ENGLAND
Tel: +44 71 434 8226 Fax: +44 71 437 6223 Email: jdiamond@hoskyns.co.uk

1.3 History of MDC actions

Date	Doc#	Action
Jun 93	X11/SC15/93-27	Approved as MDC Type A (37:0:4) Technique for compliance verification added.
Feb 93	X11/SC15/93-12	Approved as SC15 Type A
Dec 92		Approved by MDCC-E (7:0:1)
Oct 92	X11/SC15/92-26	Passed, with amendments, as SC15 Type B (14:4:9)
Jun 92	X11/SC15/91-17R	Passed as SC15 Type B (24:0:4) <i>1. Definition of \$IO expanded to remove possible confusion.</i>
Feb 92	X11/SC15/91-17	Not discussed in SC15 due to lack of time <i>1. \$IO is defined in terms of \$PRINCIPAL, which is defined (X11/89-5) in terms of \$IO. \$PRINCIPAL formalised more closely in terms of current practice.</i>
28 Nov 91	X11/SC15/91-...	Approved by MDCC-E (8:0:0) with recommended change to \$PRINCIPAL.
Jun 91	X11/SC1/91-27	Formalism included and some changes to the exact values. Approved as SC1 Type C (25:2:0)

2. JUSTIFICATION

2.1 Needs

Currently the MUMPS language specification is silent on a number of issues. One of these relates to the initial values of a number of intrinsic special variables. These are

\$IO
\$TEST

and the device-specific ones

\$DEVICE
\$KEY
\$X
\$Y

Furthermore the formulation for \$PRINCIPAL (X11/89-5) is somewhat loose in this area.

2.2 Existing practice

Implementations perform different actions.

3. DESCRIPTION OF THE PROPOSED CHANGE

3.1 General Description

The proposal is that

\$TEST is set to 0 when a process is created

\$IO is set , either to "" or to \$PRINCIPAL which has (presumably) been OPENed and USED automatically. If \$IO="" then the device is usable. Which choice is taken is implementation-specific.

If no device is opened when a process is created \$DEVICE, \$KEY are set to "" and \$X and \$Y to 0 if \$IO is "", otherwise they are set to some implementation-dependent values

When a device is OPENed and USED for the first time the values of the device-specific variables are set either to null or some implementation-dependent values (ie are not undefined). The reasoning for the implementation-dependent value is that the act of OPEN/USE may cause these values to be set by the underlying system, eg these values may be restored to the values they had on the previous CLOSE

This proposal may not be backwards compatible with some implementations. This issue should be addressed by the Backwards Incompatibility task group. However the reason for the alternatives for \$IO etc is to reflect existing practice wherever possible.

3.2 Annotated Examples of Use

None

3.3 Formalization

Amendments to RMDS

In section 4.1.3.10 add the following to the end of the paragraph on \$DEVICE

When the process is initiated \$DEVICE is given the value of the empty string if \$IO is given a value which is the empty string, otherwise it is given an implementation-dependent value.

In section 4.1.3.10 add the following to the end of the paragraph on \$KEY

When the process is initiated \$KEY is given the value of the empty string if \$IO is given a value which is the empty string, otherwise it is given an implementation-dependent value.

In section 4.1.3.10 add the following to the end of the paragraph on \$IO

When the process is initiated \$IO is given the value of \$PRINCIPAL if an implicit OPEN and USE for the device specified by \$PRINCIPAL is executed by the implementation. If the implementation does not execute this OPEN and USE then \$IO is given the value of the empty string.

In section 4.1.3.10 add the following to the end of the paragraph on \$TEST

When the process is initiated \$TEST is given the value 0 (false).

In section 4.1.3.10 add the following to the end of the paragraph on \$X

When the process is initiated \$X is given the value 0 if \$IO is given a value which is the empty string, otherwise it is given an implementation-dependent value.

In section 4.1.3.10 add the following to the end of the paragraph on \$Y

When the process is initiated \$Y is given the value 0 if \$IO is given a value which is the empty string, otherwise it is given an implementation-dependent value.

In section 4.1.3.10 replace the \$PRINCIPAL definition with

\$P[RINCIPAL] \$PRINCIPAL identifies the principal I/O device.

The principal I/O device is defined in the following fashion:

a. If the process is initiated by another MUMPS process then \$PRINCIPAL is given the value of \$PRINCIPAL of the initiating process, unless overridden by implementation-specific JOB parameters

b. If the process is initiated from a specific device then \$PRINCIPAL is given the identifier of the device

c. Otherwise \$PRINCIPAL is given an implementation-specific value

\$PRINCIPAL is constant throughout the active life of a process.

4. IMPLEMENTATION IMPACTS

4.1 Impact on Existing User Practices and Investments

These changes are not fully backwards compatible, since the standard is silent on these issues. Some applications may therefore require changes.

4.2 Impact on Existing Vendor Practices and Investments

Some vendors will have to change their implementations, but this should require a small amount of effort.

4.3 Techniques and Costs for Compliance Verification

On initiation of a process if \$I="" then \$X and \$Y should be 0 and \$D and \$K be "". \$T should always be 0. \$I should either be "" or \$P.

Compliance verification for the initial setting of \$I or \$P cannot be achieved or the values of \$X,\$Y,\$D and \$K if \$I="".

4.4 Legal Considerations

None

5. CLOSELY RELATED STANDARDS ACTIVITIES

None.

6. ASSOCIATED DOCUMENTS

X11/89-5 \$PRINCIPAL