1. Identification

1.1 Title

User-definable I/O handling

1.2. MDC proposer and sponsor

Jon Diamond, Diamond Consulting Ltd

1.3 Motion

Proposed as MDC Type A superseding X11/SC12/TG9/98-~~3~~ 7

1.4 History of MDC actions

Date	Doc# SC12/98-13	Action
Sep 98	This document X11/~~98-13~~	Proposed as MDC Type A
Mar 98	X11/SC12/TG9/98-~~3~~ 7	Approved as SC12 Type A (13:0:4)
Nov 97	X11/SC12/TG9/97-8	Proposed as replacement SC12 Type B (8:0:11)
Oct 96	X11/SC12/TG9/96-7	Amended by TG
May 96	X11/SC12/TG9/96-5 and /96-6	Discussion documents (Option 1 and 2) regarding directions
Mar 96	X11/SC12/TG9/96-1	Discussed in TG, not brought to SC
Oct 95	X11/SC12/TG9/95-12	Not approved as SC12 Type A (6:4:4)
June 95	X11/SC12/TG9/95-5	Approved as SC12 Type B
Feb 95	X11/SC12/TG9/95-5	Discussed at MDCC-E
June 94	X11/94-12	Read/Write mentioned by InterSystems

1.5 Dependencies

None

2. Justification

2.1 Needs

One of the other issues that has concerned me for some time was the fact that although we are creating bindings to various standards and device types we do not expect every implementor to provide as part of their system. We can expect most, if not all, implementors to interface to Xwindows, X3.64, possibly Magtapes and Serial files. But will they provide the interfaces to TCP/IP and GKS?

One major feature is still required, in additional to providing user-definability for controlmnemonics is to allow I/O to be redirected via user-provided routines.

2.2 Existing practice

InterSystems has already implemented Read/Write redirection.

3. *Description Of The Proposed Change*

3.1 General Description

The change provides for READ and WRITE commands, which are related to data transfer to/from the device to be redirected to call a mnemonicspace command routine.

The definition of this routine is provided for in the Portable ControlmnemonIc proposal. This I/O proposal builds on this for the READ and WRITE commands which do not use controlmnemonics.

3.2 Annotated Examples of Use

Following

```
OPEN device:::^yyy
```

then

```
READ x          is equivalent to  SET x=$$%READ^yyy()
READ ^X#5       is equivalent to  SET ^X=$$%READ^yyy(5)
READ A:30       is equivalent to  SET A=$$%READ^yyy(,30)
WRITE 3+5       is equivalent to  DO %WRITE^yyy(8)
WRITE 3,5       is equivalent to  DO %WRITE^yyy(3),%WRITE^yyy(5)
W #!?20         is equivalent to DO %WRITEFF^yyy,%WRITENL^yyy,%WRITETAB^yyy(20)
R !!?20,"Prompt: ",x#3:10
                        is equivalent to
DO %WRITENL^yyy(1),%WRITENL(1), WRITETAB^yyy(20,1),%WRITE^yyy("Prompt: ")
set x=$$%READ^yyy(3,10)
```

3.3 Formalization

Amendments to Canvass ANSI/MDC X11.1-1994:

In 8.2.25 replace the definition of format with

```
format ::=          | positionformat                          |
                    | / controlmnemonic [ ( L expr ) ]        |

                    | nlformat | ... [ tabformat ]            |
positionformat ::=  | fformat |                               |
                    |                                         |
                    | tabformat                               |

nlformat ::=        |
```

ffformat ::= #

tabformat ::= ? **intexpr**

Replace *the following paragraph from the Portable ControlmnemonicsMDC Type A proposal (X11/95-95):*

The labels %READ and %WRITE within a user-defined mnemonicspace command routine are reserved for future enhancement of this standard.

with

iocommand ::= | READ |
| WRITE |

If a label of the form %*command*, where *command* is an iocommand, exists in a mnemonicspace command routine then execution of an iocommand of the form

a. W[RITE] ffformat
b. W[RITE] nlformat
c. W[RITE] tabformat
d. W[RITE] expr
e. W[RITE] * intexpr
f. R[EAD] glvn [readcount] [timeout]
g. R[EAD] ffformat
h. R[EAD] nlformat
i. R[EAD] tabformat
j. R[EAD] strlit
k. R[EAD] * glvn [timeout]

is respectively computationally equivalent, with the exception of the effect on $TEST and the naked indicator, to

a. DO %WRITEFF^*routine*()
b. DO %WRITENL^*routine*()
c. DO %WRITETAB^*routine*(intexpr)
d. DO %WRITE^*routine*(expr)
e. DO %WRITES^*routine*(intexpr)
f. SET glvn=$$%READ^*routine*(intexpr$_1$[,intexpr$_2$])
g. DO %WRITEFF^*routine*(1)
h. DO %WRITENL^*routine*(1)
i. DO %WRITETAB^*routine*(intexpr),1)
j. DO %WRITE^*routine*(strlit,1)
k. SET glvn=$$%READS^*routine*([,intexpr$_2$])

where *routine* is the user-defined mnemonicspace command routine;

intexpr$_1$ is the intexpr from readcount, or absent if no readcount is present,

if timeout is present, intexpr$_2$ is the intexpr from timeout,

During the execution of any user-defined mnemonicspace command routine READ and WRITE re-direction for the device which caused the routine to be executed is disabled.

Upon completion of execution of a routine associated with a user-defined mnemonicspace:
a. the naked indicator and
b. $TEST
are restored to their original values.

Editor's note: This change is backwards incompatible with previously MDC Type A approved documents, since it takes up more labels in the mnemonicspace command routine than had been previously reserved (%READ and %WRITE only).

4. *Implementation impacts*

4.1 Impact on Existing User Practices and Investments

This should encourage the usage of new device types in applications in the knowledge that these are portable. This would dramatically speed up their implementation in applications such as the VA software which mandate the use of portable code.

4.2 Impact on Existing Vendor Practices and Investments

Some additions would be required, however these should be small compared to the implementation of the portable controlmnemonics proposal.

4.3 Techniques and Costs for Compliance Verification

No significant costs are anticipated.

4.4 Legal Considerations

None

5. *Closely related standards activities*

5.1 Other X11 Proposals (Type A or Type B) Under Consideration

None

5.2 Other Related Standards Efforts

None

5.3 Recommendations For Co-ordinating Liaison

None

6. *Associated documents*

None

7. *ISSUES etc.*

Jun 98
Pros: Implemented, Very useful
Cons: NONE

Nov 97
TG reverses its mind on preserving $T and naked.
The editor disagrees with this change, but has nevertheless changed the document.

Oct 96
Change made by TG to not preserve **$T** and naked indicator on return from user-defined mnemonicspace routines.

Comp.lang.mumps discussion
The proposal has been modified to add additional WRITE labels for the various functions.

Write with timeout
This is expected to be handled in one the following ways:
- if $TIMER variable is used then no change is required to the proposal, since the routine can pick it up as normal
- if OPEN/USE option (deviceparameter) then again the routine can pick up the information from ^$DEVICE($I,"WRITETIMEOUT") or whatever
- if a [timeout] is introduced for the WRITE command itself then another parameter (the 4th in position) will be used. [An alternative would be to reserve the second parameter for this and move the current second/third to the third/fourth.]

Mar 96
Micronetics flushes input buffer when output performed on READ
Direction to the editor:
- must be a mechanism to signal whether invoked from READ or WRITE
- must allow for READ * and WRITE *
- it would be wise to anticipate the ramifications of the implementation of a WRITE time-out
Task group voted 0:6:2 on motion to recommend elevation to Type A in SC

Oct 95
$T needed clarifying. Straw polls: Stack $T(except timeout) 10:0, Stack $T always 0:10, Stack naked 6:0
A paragraph, at the bottom of 3.3 above, has been included to clarify these effects.
Pros: Implemented, Very useful
Con: contents of $T unclear

June 95
Straw poll to include R * or W * formats 0:2:6. (Therefore not included in subsequent revision)

Mar 95 MDCC-E meeting
W #?10 and W "#?10" were pointed out as being equivalent. The proposal was changed to add a new label %WRITEF for format handling