X11/97-3

MUMPS Development Committee

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

Pattern Ranges

September 29, 1996

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.

Anyone reproducing this release is requested to reproduce this introduction.

1. Identification of the proposed change

1.1 Title: Pattern ranges

1.2. MDC Proposer and Sponsor

Alan Frank
35 Gardner St.
Arlington, MA 02174
Phone: 617 647-4884x280 (at CoMed)
Fax: 617 648-2824
alf@world.std.com

SC13/TG2 String Handling
Chair: David Whitten
12160 Abrams #601
Dallas, TX 75243
Fax: 214-454-1050
Phone: 214-437-5255

1.3. Motion: None

1.4. History:

Jan 1997	X11/97-3: Publication of MDC Type A.
Aug 1996	X11/SC13/TG2/96-4: Adds vendor comment to 4.2. Approved as MDC Type A. 20:6:1.
Feb 1996	X11/SC13/TG2/96-1. Corrected history, changed "null string" to "empty string." and improved examples, as requested by the task group. Passed by SC13 as type A. 15:0:4 with no cons cited.
Oct 1995	X11/SC13/TG2/95-10: Attempts to resolve the collating sequence problem. Passed by task group, 7:0:9 and by subcommittee, 20:4:2, as a replacement type B.
May 1995	X11/SC13/95-17, failed 10-10 due to problems with alternate collating sequences
Jan 1995	X11/SC13/TG2/WG2/94-5 reflects the amendments and corrections described in X11/SC13/TG2/94-3. Passed as type B by SC13, 8-7-4.
Jun 1994	X11/SC13/TG2/WG2/94-3, one of two (the other is ^(2)) which reflect changes recommended by working group, passed as amended by SC13/TG2/WG2, 8-0-4.
Jan 1994	X11/SC13/TG2/WG2/94-1 considered by WG2, which recommended a new syntactic approach
Oct 1993	X11/SC13/TG2/WG2/93-2 uses bracket syntax and allows patcodes in strconsts; adopted by TG2, 6-0
Jun 1993	Original proposal[1] with $ANY and $EXCEPT
Jan 1993	Formation of WG2
1992	X11/SC13/TG2/92-6: Suggestion document from Jon Diamond based on ideas from Victor Grishkan

1.5. Dependencies: none

2. Justification of proposed change

2.1. Needs

It is somewhat annoying and nonintuitive to test for a substring matching any of a set of characters, especially when multiple such tests need to be done in one pattern match.

Although one can define additional patcodes using the methods developed for internationalization, this is clumsy for checks which need to be done only once, and is not modular (you can't NEW a patcode definition).

2.2. Existing practice:

To test for a string starting and ending with vowels:

```
        $TRANSLATE(X,"EIOU","AAAA")?1"A".E1"A"
```

To test for a string containing hexadecimal characters:

```
        $TRANSLATE(X,"ABCDEF",999999)?.N
```

[1] Document number not known

3. Description of the proposed change

3.1. General description

Bracket syntax is added to the definition of patcode to allow for a pattern which matches any of a set of characters. Note that due to the use of the sorts after operator in the definition of ranges, the exact list of characters which match one of these patcodes may change depending on the national/cultural environment.

3.2. Annotated examples of use:

The examples above could be replaced by

```
X?1["AEIOU"].E1["AEIOU"] ;starts and ends with a vowel
X?.["A":"F"]N ;any number of occurences of either the letters A to F or digits.
```

3.3. Formalization:

Note that all changes are to the 1995 standard.

In the third paragraph of section 5 (Metalanguage Description) add an additional sentence after "...FF (form feed).":

> Also, where necessary to avoid confusion with the "option" metalanguage operator, OB is used to represent the open bracket character ([ ) and CB is used to represent the close bracket character ( ]).

In section 7.2.3, add charspec to the formal definition of patcode so that it reads:

$$\underline{\texttt{patcode}}\ \texttt{::=}\ \left|\begin{array}{c} \texttt{Y}\ \underline{\texttt{patnonY}}\ \texttt{Y} \\ \texttt{Z}\ \underline{\texttt{patnonZ}}\ \texttt{Z} \\ \underline{\texttt{patnonYZ}} \\ \underline{\texttt{OB}}\ \underline{\texttt{charspec}}\ \underline{\texttt{CB}} \end{array}\right|$$

$$\underline{\texttt{charspec}}\ \texttt{::=}\ \underline{\texttt{strconst}}_1\ [\ :\ \underline{\texttt{strconst}}_2\ ]$$

$$\underline{\texttt{strconst}}\ \texttt{::=}\ \left|\begin{array}{c} \texttt{\$C [HAR] (}\underline{\texttt{L}}\ \underline{\texttt{numlit}}\texttt{)} \\ \underline{\texttt{strlit}} \end{array}\right|$$

Add text before the second paragraph after the metalanguage (the ¶ beginning "Patcodes differing..." so that it reads:

a If a patcode has the form of a charspec, determination of whether a character belongs to the patcode is made as follows: A character belongs to a charspec containing only one strconst if it is contained in the string represented by that strconst. A character belongs to a charspec containing two strconsts if it is (inclusively) between them. Formally, *X* is a member of *S* if *S*[*X*, and *X* is a member of *S1*:*S2* if *S1* does not trail *X* and *X* does not trail *S2*, but the check against the value of *S2* will be omitted if *S2* is the empty string. If *S2* is present, then neither *S1* nor *S2* may contain more than one character.[2]

If a strconst is of the form $C[HAR](...), then it has the same value as the result of the function $CHAR called with the same parameters. Use of upper, lower, or mixed case in the name $CHAR is permitted.

b Otherwise, patcodes differing..."

[2] As usual for MDC standards, this is not specified as an error, because it is ruled out by syntax, not by run-time happenings. This footnote is not part of the proposed change.

4. Implementation effects

4.1. Impact on existing user practices and investments

Although the syntactical elements added by this proposal usually can be emulated by judicious use of $TRANSLATE, it is expected that new code written by programmers concerned with readability will use these new elements.

4.2. Impact on existing vendor practices and investments

The proposer would appreciate comments from vendors as to the impact this would have on their practices. Note that internationalization proposals may already require vendors to redo their pattern match algorithms. One vendor wrote"While this is fairly easy to do, I would... request: avoid piece-meal solutions."

4.3. Techniques and costs for compliance verification

Write a program which uses pattern ranges containing both one and two strconsts. Test various strings against the patterns and confirm that the language implementation returns the correct answer. If alternate character sets are available, try this test in an environment where the sorts after operator returns results differing from U.S. ASCII, and confirm that the pattern match works according to the definition for the character set in use.

4.4. Legal considerations: none

5. Closely related standards activities

5.1. Other X11 proposals under consideration:

Internationalization proposals to allow additional patcodes
Regular expressions (not currently before the MDC)
MDC Type A extension for pattern negation

5.2. Other related standards efforts: none

5.3. Recommendations for coordinating liaison:

We should notify the original proposer (if we can find him) of the progress of this document.

6. Associated Documents: none

7. Issues, Pros and Cons, and Discussion:

From September, 1996, MDC Vote:

Pro:

Significant enhancement to pattern match [10]

Con:

Overloads pattern match [1]	This is a matter of opinion. Judging from the vote results, most MDC members believe that pattern match is a valid and useful place to put this functionality.
Unclear in context of internationalization [6]	No discussion is cited in the minutes to support this claim. The document editor would be pleased to work with anyone who believes that additional changes are needed, to formulate additional proposals to clarify the behavior of pattern ranges in the context of alternate collating sequences and internationalization.

From March, 1996, **Pros** only; no Cons were cited:
Desirable functionality [5]
Asked for by M[UMPS] community [4]
Enhances internationalization [1]
Common in other string pattern/search languages (SNOBOL, grep, etc.) [5]

From October, 1995:
Pro:
Significantly enhances pattern match [3]
Asked for by the user community [3]
Enhances internationalization [1]

Con:
Overloads pattern match [6]
This is a matter of opinion.
There were an additional con cited in January, 1995:

Should be part of consolidated Pattern Match proposal [5]
It was the decision of the working group that the original proposal that came into the MDC was best dealt with by breaking out the atomic functional items and presenting them as separate proposals. This allows each document to be perfected and come to a vote when it's ready, and allows the membership an easy way to vote for the features that they believe should be in the standard without holding up the ones which are deemed unworthy, or in need of additional work.

8. Glossary:

trails "α trails β" means that (" "_α)]]](" "_β) in the appropriate collation sequence; if not specified, it refers to the sequence used for local variables.