1. Identification of the proposed change

1.1. Title

Corrections to FORMAT Library Function

1.2. MDC Proposer and Sponsor

This proposal originates from Ed de Moel. Motions regarding the status of this document will be made by Taskgroup 2 (String Processing) of Subcommittee 13 (Data Management and Manipulation).
Ed de Moel can be reached at:
· 800 Nelson Street, Rockville, Maryland 20850-2051
· home phone: 301 762 8333
· telefax: 301 762 8999
· email: demoel@radix.net

1.3. Motion

It is to be established whether or not the corrections in this document are of a substantive nature. For those corrections in this document that are deemed to be of an editorial nature, the motion is to be "**direct the document editor of the MDC standard to apply the correction in question**". If there are corrections that are deemed to be of a substantive nature, the motion is to be "**withdraw X11/SC13/1998-10 and its successor X11/1999-1 as an MDC Type A extension, and remand these documents for remedial work to task group 2 of subcommittee 13**".

1.4. History of MDC actions

Date	Document	Action
September 1999	X11/SC13/TG2/1999-1	Introduction of corrections
Summer 1999	X11/1999-1	Final MDC Type A Write-up
Sep 1998	X11/SC13/1998-10	FORMAT Limrary Function, accepted as an MDC Type A Extension (12:1:3).

1.5. Dependencies

None.

2. Justification of Proposed Change

2.1. Needs

The needs for the FORMAT Library Function are well presented in the proposals that established this library function. While attempting a trial implementation of this library function, a number of corrections appeared to be necessary. This proposal addresses the corrections that are needed to be able to implement the FORMAT Library Function.

2.2. Existing Practice in Area of the Proposed Change

The author of this document is not aware of anycurrent implementations of the FORMAT Library Function.

3. Description of the proposed change

3.1. General Description of the Proposed Change

This document proposes various corrections to the specification of the FORMAT Library Function.

3.2. Annotated Examples of Use

Not applicable.

3.3. Formalization

In the discussion below, the term "original proposal" will be used to refer to the version of the FORMAT Library Function that currently has MDC Type A status.

In the original document, there are references two ssvns: ^$SYSTEM("FORMAT") and ^$FORMAT. The proposal itself does not define the meaning of these structures, nor am I aware of any other document that defines these entities.

The standard stipulates that the first subscript of ^$SYSTEM must evaluate to a valid value for the special variable $SYSTEM, i.e. something like 42,serialnumber_and_version. As a result, it is proposed to replace all occurrences of ^$SYSTEM("FORMAT",.... by ^$SYSTEM(system,"FORMAT",....
Since these references occur only in the code that is provided as a sample for implementors, it is likely that this correction will be deemed to be of an editorial nature.

Since there is no definition of the ssvn ^$FORMAT, it is proposed to replace all occurrences of ^$FORMAT(... by ^$JOB($JOB,"FORMAT",....
Since these references occur only in the code that is provided as a sample for implementors, it is likely that this correction will be deemed to be of an editorial nature.

The original document uses `FC=`... in its examples, but the formalism does not explain the meaning of FC. Since FC (the "fill string") is an essential part of the desired functionality, it is proposed to add to the formalism. Also, it is my impression that the definition of FL=... is not "as intended by the original author".

In the definition of fspec, add the possible option: `| FC = filstr |`
In the definition of fspec, delete the option FL=...

The formalism first states that sepchar is a string that evaluates to a single graphic, and then provides the metalanguage that states that sepchar ::= expr V graphic ... (ellipsis is part of the formalization).
The examples assume that sepchar is a multi-character value.

Strike the sentence that reads `sepchar is a string which resolves to a single instance of a graphic.` Replace all occurrences of sepchar with sepstr.

In the examples, a couple of quotes are missing (double quotes within a string, terminating quotes on other strings). Since the examples are not part of the formalism, these omissions are deemed to be of an editorial nature, and this document will not attempt to resolve this issue.

When the sample code was run, a syntax error became apparent: example number 7 leads to an "indirection error". Also, the sample code does not always NEW all its used local variables.

The list of "fmask" characters in the example section is much longer than the list in the formalization. The sample code supports the characters space, -, (,), +, =, c, d, f, n, s and x. Since there are no examples that use the format-letter "m", and the formalism does not address this letter either, it is recommended to drop this letter from the list of options in the definition of fmask. The characters (and) need to be added to the formalism.

The formalism specifies that fillon is a boolean switch that turns the padding with fill characters on or off, with a default value of "off". None of the examples sets the value of this switch to "on", yet several examples are intended to produce fill characters. Since the formalism also offers the option to let the fill string be "just spaces", and since the sample code does not support this switch either, it is recommended to drop this switch from the specification.

As a result, the formalism should be corrected to:

```
                 | CS=curstr  |   Currency Specifier
                 | DC=dechar  |   Decimal Character
                 | EC=erchar  |   Error Character
fspec   ::=      | FC=filstr  |   Fill String
                 | FM=fmask   |   Format Mask
                 | SC=sepstr  |   Separator Character(s) Left
                 | SR=sepstr  |   Separator Character(s) Right

curstr  ::=  expr

dechar  ::=  expr V graphic

erchar  ::=  expr

filstr  ::=  expr

sepstr  ::=  expr

                        | c |       Currency (Left Justified)
                        | d |       Decimal (Singular occurrence per mask)
                        | f |       Float either type at the end of the f run
                        | l |       Left Justified Numeric
                        | n |       Numeric
fmask   ::=  expr V     | s | ...   Separator
                        | x |       Spacer
                        | - |       Display negative only if negative
                        | + |       Display sign always
                        | SP|       Insert space for the spaces in the mask
                        | ( |       Start of negative value
                        | ) |       End of negative value
```

The description of the function starts with:

... returns a formatted string. **It will return expr unchanged.**

Since the description of the function does not at all explain the meaning of expr in this context, this second sentence is confusing to say the least, and could easily be construed to mean that the function is a no-op, regardless of the values of its parameters. Recommend to strike this sentence.

While executing the samples, example number 4 produced a string of only asterisks. The reason was that near the end of subroutine TRANSV1 in the sample code, the name of a local variable was misspelled. replace the segment:

```
 . . SET:$GET(CP)="" CP=$LENGTH(CS)
 . . SET GC=$EXTRACT(CS,CP),CP=CP-1
 . . SET:CP<1 SP=$LENGTH(CS)
```

with

```
 . . SET:$GET(CP)="" CP=$LENGTH(CS)
 . . SET GC=$EXTRACT(CS,CP),CP=CP-1
 . . SET:CP<1 CP=$LENGTH(CS)
```

This type of correction is generally considered to be of an editorial nature.

While executing the samples, both examples 3 and 4 display the characters of the currency string in reverse order. In order to resolve this, replace all three occurrences of the segment:

```
 . . SET:$GET(CP)="" CP=$LENGTH(CS)
 . . SET GC=$EXTRACT(CS,CP),CP=CP-1
 . . SET:CP<1 CP=$LENGTH(CS)
```

with

```
 . . SET GC=$EXTRACT(CS,$LENGTH(CS))
 . . SET CS=GC_$EXTRACT(CS,1,$LENGTH(CS)-1)
```

This type of correction is generally considered to be of an editorial nature.

Example 6 returns a series of asterisks. The comment with the example states "floating sign", implying that the sign should appear immediately preceding the number. The formalism, however, states explicitly that:

"Should a sign not be specified (+, () or -), the absolute value of IN is returned."

Since the code matches neither the example nor the formalism, it is hard to make a recommendation for a correction.

Also, the value of variable `Z`, which is passed as a parameter is `“cfsfffsffndnn”` rather than `“FM=""cfsfffsccndnn""”`, as the formalism requires. The function does not provide any error report...

The return values from the sample code for examples 8, 10, and 11 contain too many instances of the fill character. The return value from example 14 does not pad the decimal fraction with zeroes for those positions where the format mask requires digits. The example for “left adjusted” formatting returns only asterisks, because the sample code does not address the format-letter “l”. As a result, it is recommended to replace the complete sample code with the following:

```
FORMAT(V,S) ;
 New lo,mask,out,p,pos,spec,up,v1,v2,val,x
 ;
 Set lo="abcdefghijklmnopqrstuvwxyz"
 Set up="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
 ;
 ; Array spec() contains the formatting directives
 ;
 ; First set defaults
 ;
 Set spec("CS")="$" ; Currency symbol
 Set spec("DC")="." ; Decimal separator
 Set spec("EC")="*" ; Error character
 Set spec("SC")="," ; Separator characters > 1
 Set spec("FC")=" " ; Fill string
 ;
 ; Other specifiers may be
 ;  FC = Fill String
 ;  FM = Format Mask
 ;  FO = Fill On/Off
 ;  SR = Separator characters < 1
 ;
 ; Then Inherit properties from System,
 ; overwriting the defaults
 ;
 Set x="" For  Set x=$Order(^$System($System,"FORMAT",x)) Quit:x=""  Do
 . Set spec(x)=^$System($System,"FORMAT",x)
 . Quit
 ;
 ; Then Inherit properties from current process
 ; overwriting the system and the defaults
 ;
 Set x="" For  Set x=$Order(^$Job($Job,"FORMAT",x)) Quit:x=""  Do
 . Set spec(x)=^$Job($Job,"FORMAT",x)
 . Quit
 ;
 ; Then look at actual parameters
 ; overwriting anything else
 ;
 Set S=$Get(S) For  Quit:S=""  Do
 . New e,i,str,v
 . Set x=$Piece(S,"=",1)
 . Set i=$Length(x)+2,str=0,v=""
 . Set:x="" i=1
 . For i=i:1:$Length(S)+1 Do  Quit:'i
 . . Set e=$Extract(S_":",i)
 . . If 'str,e=":" Set S=$Extract(S,i+1,$Length(S)),i=0 Quit
 . . Set v=v_e Quit:e'=""""
 . . Set str=1-str
 . . Quit
 . If i>$Length(S) Set S=""
 . If x'="",v'="" Set @("spec($Translate(x,lo,up))="_v) Quit
 . Set $ECode=",M28,"
 . Quit
 ;
 ; Make certain that DC and EC are non-empty
 ; and not longer than 1 character
 ;
 Set spec("DC")=$Extract(spec("DC")_".",1)
 Set spec("EC")=$Extract(spec("EC")_"*",1)
 ;
 Set val=$Get(V),(mask,out)=$Get(spec("FM"))
 If mask="" Quit val
 ;
 ; Currency string
 ;
 Set x=spec("CS")
 Set pos=0 For  Set pos=$Find(mask,"c",pos) Quit:pos<1  Do
 . Set $Extract(out,pos-1)=$Extract(x,1)
 . Set x=$Extract(x,2,$Length(x))_$Extract(x,1)
 . Quit
 ;
 ; Sign
 ;
 Set x=$Select(val>0:"+",val<0:"-",1:" ")
 Set pos=0 For  Set pos=$Find(mask,"+",pos) Quit:pos<1  Do
 . Set $Extract(out,pos-1)=x
 . Quit
 Set pos=0 For  Set pos=$Find(mask,"-",pos) Quit:pos<1  Do
 . Set $Extract(out,pos-1)=$Select(x="-":x,1:" ")
 . Quit
 If x'="-" Set out=$Translate(out,"()","  ")
 ;
```

```
; Decimal separator
;
Set pos=$Find(mask,"d")
Do:pos'<1
. Set $Extract(out,pos-1)=spec("DC")
. For  Set pos=$Find(mask,"d",pos) Quit:pos<1  Do
. . Set $Extract(out,pos-1)=spec("EC")
. . Quit
. Quit
;
; Right (default, format letter "n") or
; left (format letter "l") adjustment?
;
If mask["l",mask["n" Set $ECode=",M28,"
;
; Left and Right Separators
;
Set v1=$Piece(val,".",1),v2=$Piece(val,".",2)
Set v1=$Translate(v1,"-")
If mask'["l" Do
. Set x="" For p=1:1:$Length(v1) Set x=$Extract(v1,p)_x
. Set v1=x
. Quit
;
Set pos=$Find(mask,"d") Set:pos<1 pos=$Length(mask)+2
;
; Integer part and Left separators
;
Set x=spec("SC")
Set p(1)=pos-2,p(2)=-1,p(3)=1
Set:mask["l" p(1)=1,p(2)=1,p(3)=pos-2
For p=p(1):p(2):p(3) Do
. If "fln"[$Extract(mask,p) Do
. . Set $Extract(out,p)=$Extract(v1,1)
. . Set v1=$Extract(v1,2,$Length(v1))_spec("FC")
. . If $Translate(v1,spec("FC"))="" Set x=spec("FC")
. . Quit
. If $Extract(mask,p)="s" Do
. . Set $Extract(out,p)=$Extract(x,1)
. . Set x=$Extract(x,2,$Length(x))_$Extract(x,1)
. Quit
;
; Fractional part and Right separators
;
Set x=$Get(spec("SR"),spec("SC"))
Set:v2="" v2=0
For p=pos:1:$Length(mask) Do
. If "fn"[$Extract(mask,p) Do
. . Set $Extract(out,p)=$Extract(v2,1)
. . Set v2=$Extract(v2,2,$Length(v2))_"0"
. . Quit
. If $Extract(mask,p)="s" Do
. . Set $Extract(out,p)=$Extract(x,1)
. . Set x=$Extract(x,2,$Length(x))_$Extract(x,1)
. . Quit
. Quit
```

```
;
; Fill String
;
Set x=$Get(spec("FC"))
For p=1:1:$l(mask) Do
. Quit:"nf"'[$Extract(mask,p)
. Quit:$Extract(out,p)'=" "
. Set $Extract(out,p)=$Extract(x,1)
. Set x=$Extract(x,2,$Length(x))_$Extract(x,1)
. Quit
;
; Justification
;
For x="+ | +","- | -","( | (","  )|) " Do
. New find,repl
. Set find=$Piece(x,"|",1),repl=$Piece(x,"|",2)
. For  Quit:out'[find  Do
. . Set out=$Piece(out,find,1)_repl_$Piece(out,find,2,$l(out)+2)
. . Quit
. Quit
;
Quit out
```

4. Implementation impacts

4.1. Impact on Existing User Practices and Investments

Since the author of this proposal is not aware of any current implementations of this library function, it is not deemed likely that this proposal will affect any user practices at all.

4.2. Impact on Existing Vendor Practices and Investments

The author expects that the corrections in this document will make it easier for implementors to implement this function.

4.3. Techniques and Costs for Compliance Verification

See the “original document”.

4.4. Legal considerations

None expected.

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. List of Associated Documents

None.

7. Issues, Pros and Cons, and Discussion

7.1. September 1999, San Diego, California

Introduction of these corrections.

8. Glossary

None.