

The Secrets of \$TRANSLATE

by Winfried Gerum



Winfried Gerum

are not needed. These are subsequently purged by the outer call of \$TRANSLATE.

If a string after the removal of characters is compared with the original string, you can see whether certain characters are in the string or not:

```
IF $TRANSLATE(X, "+-*/")'=X
```

is equivalent to

```
IF X["+!"(X["-"]!(X["*"]!(X["/"]
```

or for the negation:

```
IF $TRANSLATE(X, "+-*/")=X
```

is equivalent to

```
IF X'["+",X'["-",X'["*",X'["/"]
```

A closer look at the translate process shows that it is possible to exchange characters:

```
SET X=$TRANSLATE(X,"01","10")
SET X=$TRANSLATE(X,"ABCDEabcde",
"abcdeABCDE")
```

The first example exchanges 0 and 1; the second exchanges the sets of lowercase and uppercase characters "a" through "e".

Frequently you need a string of identical characters. With \$JUSTIFY you can have that for any character you wish, as long as you want spaces. Then with \$TRANSLATE you can change that to any other character:

```
S X=$TRANSLATE($JUSTIFY(" ",N),"
","*")
```

If you liberate yourself from the idea that the string must appear as a first argument in order to be edited by \$TRANSLATE, there are a lot of very interesting uses of that function. It may be used to produce permutations of a

given string while simultaneously adding or removing characters. In some of the following examples we assume the maximum length of a string X to be ten characters to save typing space. But all of these examples can be enhanced to work on longer strings. The examples that use date formats assume two-digit days and two-digit months.

```
SET X=$TRANSLATE(2134567890,123
4567890,X)
```

exchanges the first two characters of X.

```
SET X=$TRANSLATE("Dd/Mm/19Yy",
"Mm/Dd/Yy",X)
```

changes a date from MM/DD/YY format to DD/MM/19YY

```
SET X=$TRANSLATE("Mm/Dd/Yy",
"Dd/Mm/19Yy",X)
```

is the reverse transformation DD/MM/19YY to MM/DD/YY.

Similarly,

```
SET X=$TRANSLATE("19YyMmDd",
"Mm/Dd/Yy",X)
```

transforms the format MM/DD/YY into the format 19YYMMDD. The latter format sorts chronologically like the +\$H-format, but it's still readable! The reverse transformation one gets again by exchanging the first and the second argument:

```
SET X=$TRANSLATE("Mm/Dd/Yy",
"19YyMmDd",X)
```

An additional example illustrates the mechanism for inserting characters with \$TRANSLATE:

```
SET X=$TRANSLATE("123-45-6789",
123456789,$TRANSLATE(X," "))
```

\$TRANSLATE may not be one of the most important functions in M, but certainly it is the most versatile.

At first glance, this function simply replaces certain characters with other characters or with no characters:

```
SET X=$TRANSLATE(X,"abcde","ABCDE")
```

That replaces lowercase characters "a" through "e" with their uppercase counterparts.

A slight modification also removes any spaces:

```
SET X=$TRANSLATE(X,"abcde","ABCDE")
```

It may be used to remove any unwanted characters from an input string. For example,

```
SET X=$TRANSLATE(X,"",$")
```

removes formatting characters from a string supposed to be a kind of number.

For a bit more sophistication, write

```
SET X=$TRANSLATE(X,$TRANSLATE(X,
"+-1234567890."))
```

to remove all characters except those listed. The inner call of \$TRANSLATE yields a string with all characters that

that yields the correct format for American Social Security numbers. These consist of nine digits with a hyphen (-) between the 3rd and 4th digits and the 5th and 6th. The inner \$TRANSLATE removes all hyphens whether at the correct position or not. The first nine characters of the resulting string are then used in the outer \$TRANSLATE. The first argument has hyphens at the desired locations. The \$TRANSLATE then replaces the dummy numbers with their correct counterparts.

Advanced programmers will expand the example so that not just hyphens, but all unwanted characters, are removed:

```
SET X=$TR("123-45-6789",123456789,  
$TR(X,$TR(X,123456789)))
```

The recently introduced \$REVERSE intrinsic function is just excess baggage for the language. \$TRANSLATE does the trick of string reversal, too:

```
SET X=$TRANSLATE("0987654321",  
1234567890,X)
```

reverses the string (reversal is just another permutation). The next example even uses the string to be edited in two of the \$TRANSLATE arguments:

```
SET Y="QWERTY",X=$TRANSLATE(Y,X  
_Y,X)
```

that removes any characters other than "QWERTY" from X; it removes any duplicates and it sorts the remaining characters in the order "QWERTY". (Same as \$TR(Y,\$TR(Y,X)))

As you see, \$TRANSLATE can do many things and often it performs many

things with one function call. It is a fast and efficient tool especially if you want to make a routine more user friendly. We hope that these examples inspire many M users to even more tricky uses of that marvelous function.

This column is based on one that was first published in the *MUG-Europe Newsletter*, in April 1990. ■

Winfried Gerum is president of Winner Software in Erlangen, Germany. His column appears regularly in *M Computing*.

Network Integration Services

Expert consulting to help your organization design, implement, and maintain complex networks. Experience with all major LAN platforms (Novell Netware, DECNET, Windows for Workgroups) and M database protocols (OMI, DDP, and vendor specific). Services include strategic planning, LAN audits, and on-site problem resolution.

J. J. Althouse & Associates
645 Lexington Road
Concord, MA 01742
508-369-6076