

STRMX\$ – Move Zero-terminated String

The STRMX\$ routine can be used to move a zero-terminated (i.e. LOW-VALUE terminated) string to a fixed-length destination area and optionally returns the length of the string just moved.

1. Invocation

To move a zero-terminated string code:

```
CALL STRMX$ USING source dstn length [actual_length]
```

where source is the PIC X(?) source string (i.e. the zero-terminated string to be moved); dstn is the PIC X(?) destination area; length a PIC 9(4) COMP, or literal, specifying the length of the destination area. The optional actual_length parameter is a PIC 9(4) COMP field, into which the actual length of the string, **including the terminating binary-zero**, is returned.

2. STOP Codes and Exception Conditions

No STOP codes are generated by STRMX\$.

No exceptions are returned by STRMX\$.

3. Programming Notes

The source string **MUST** be terminated by a byte of LOW-VALUES otherwise unpredictable results will occur. No checking is performed by STRMX\$.

If the destination field is longer than the zero terminated string, it will be padded with trailing SPACES. The binary-zero terminator is **not** moved. If the destination field is shorter than the zero terminated string, the string will be truncated **but no warning will be given**. To test for truncation the actual_length parameter must be supplied and the result compared to the length value.

4. Examples

The following example will return a length of 5 in Z-LEN2:

```
DATA DIVISION
*
77      X-TEST  PIC X(20)
        VALUE   "test"
        VALUE   #00
77      Z-LEN2  PIC 9(4) COMP
*
77      X-DEST  PIC X(40)
*
PROCEDURE DIVISION
*
        CALL STRMX$ USING X-TEST X-DEST 40 Z-LEN2
```

and X-DEST will contain the string “test” followed by 36 SPACE characters.

5. Copy-Books

No copy-books are required.

6. See Also

STRML\$ Move zero-terminated string to zero-terminated string
 STRM\$ Move fixed-length string to zero-length string