STRML$ – Move Zero-terminated String

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

1. Invocation

To move a zero-terminated string code:

```
CALL STRML$ USING source dstn 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, which must be at least as long as string1. The length of the string that was moved, string1, **including** the terminating byte of LOW-VALUE, is returned in the PIC 9(4) COMP field length.

2. STOP Codes and Exception Conditions

No STOP codes are generated by STRML$.

No exceptions are returned by STRML$.

3. Programming Notes

Repeated calls to STRML$ provide a very convenient way to concatenate zero-terminated strings.

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

The destination string MUST be at least as large as the source string otherwise unpredictable results will occur. No checking is performed by STRML$.

4. Examples

The following example concatenates three zero-terminated strings:

```
PROGRAM STRML
DATA DIVISION
*
77    X-1   PIC X(?)
      VALUE "c:\test\"
      VALUE #00
77    X-2   PIC X(?)
      VALUE "Filename"
      VALUE #00
77    X-3   PIC X(?)
      VALUE ".abc"
      VALUE #00
01    X-DSTN
  02  X-DSTX       OCCURS 256 PIC X
*
77    Z-LEN PIC 9(4) COMP
77    Z-INX PIC 9(4) COMP
*
PROCEDURE DIVISION
*
      MOVE 1 TO Z-INX
      CALL STRML$ USING X-1 X-DSTX(Z-INX) Z-LEN
      SUBTRACT 1 FROM Z-LEN
      ADD Z-LEN TO Z-INX
      CALL STRML$ USING X-2 X-DSTX(Z-INX) Z-LEN
      SUBTRACT 1 FROM Z-LEN
      ADD Z-LEN TO Z-INX
      CALL STRML$ USING X-3 X-DSTX(Z-INX) Z-LEN
```

5. Copy-Books
No copy-books are required.

6. See Also
STRMX$ Move zero-terminated string to fixed-length.
STRM$ Move fixed-length string to zero-length string