Cobol Course

The STRING verb

Introduction

Aims

The aim of this unit is to provide to provide you with a solid understanding of the STRING verb.

Objectives

By the end of this unit you should:By the end of this unit you should:

1. Understand how the STRING verb works.
2. Be able to use the STRING to concatenate text strings in your programs

Prerequisites

You should be familiar with the material covered in the unit;

- Data Declaration
- Iteration
- Selection
- Tables
- Edited Pictures
- The INSPECT verb

To top of page

The STRING verb

Introduction

The STRING verb is used for string concatenation. That is, it is used to join together the contents of two or more source strings or partial source string to create a single destination string.

The examples below show how the STRING is used.

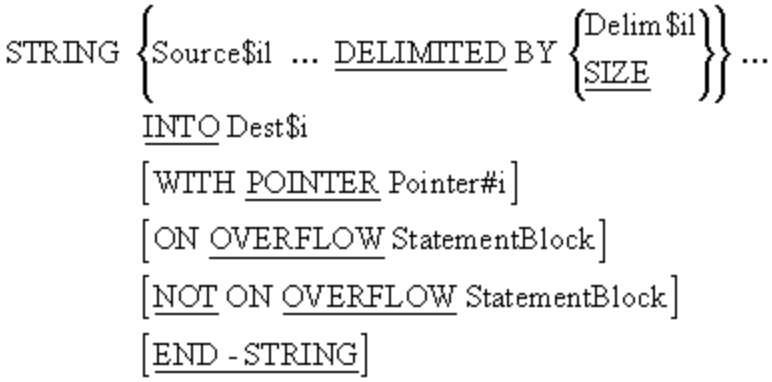
The first example concatenates the entire contents of the identifiers Indent1 and Ident2 with the literal "10" and puts the resulting sting into DestString.

The second example concatenates the entire contents of Ident1, with the partial contents of Ident2 (all the characters up to the first space) and Ident3 (all the characters up to the word "frogs").

```
STRING Ident1, Ident2, "10" DELIMITED BY SIZE
    INTO DestString
END-STRING

STRING
   Ident1 DELIMITED BY SIZE
   Ident2 DELIMITED BY SPACES
   Ident3 DELIMITED BY "Frogs"
INTO Ident4 WITH POINTER StrPtr
END-STRING.
```

STRING syntax



Delim$il	Character or set of characters in a source string that terminates data transfer to the destination string.
Pointer#i	Points to the position in the destination string where the next character will go.

How the STRING works

The STRING statement moves characters from the source string to the destination string according to the rules for alphanumeric to alphanumeric moves.

However, no space filling occurs and unless characters in the destination string are explicitly overwritten they remain undisturbed.

As usual with alphanumeric to alphanumeric MOVEs, data movement is from left to right.

The leftmost character of the source is moved to the leftmost position of the destination then the next leftmost of the source to the next leftmost of the destination and so on.

When a number of source strings are concatenated, characters are moved from the leftmost source string first.

When a WITH POINTER phrase is used, its value determines the starting character position for insertion into the destination string.

The ON OVERFLOW clause executes if there are still valid characters left in the source strings but the destination string is full.

Data movement termination

Data movement from a particular source string ends when either;

- the end of the source string is reached
- the end of the destination string is reached
- the delimiter is detected.

STRING termination

The STRING statement ends when either;

- all the source strings have been processed
- the destination string is full
- the pointer points outside the string.

STRING rules

1. Where a literal can be use, a Figurative Constant can be used except the ALL (literal).
2. When a Figurative Constant is used its size is one character.
3. The destination item Dest$i must be an elementary data item without editing symbols or the JUSTIFIED clause.
4. Pointer#i must be an integer item and its description must allow it to contain a value one greater than the size of the destination string. For instance, a pointer declared as PIC 9 is too small if the destination string is 10 characters long.

STRING clauses

ON OVERFLOW
If the ON OVERFLOW clause is used then the statement following it will be executed if there are still characters left to pass across in the source field(s) but the destination field has been filled.

WITH POINTER
The WITH POINTER phrase allows an identifier/dataname to be kept which holds the position in the Destination String where the next character will go.

When the WITH POINTER phrase is used, the program must set the pointer to an initial value greater than 0 and less than the length of the destination string before the STRING statement executes.

If the WITH POINTER phrase is **not** used, operation on the destination field starts from the leftmost position.

DELIMITED BY SIZE
The DELIMITED BY SIZE clause means that the whole of the sending field will be added to the destination string.

To top of page

STRING examples

Introduction

This section starts with some examples to reinforce the material you have just read and it ends with a few test/examples to let you see if you have understood everything so far.

Example 1 (animation)

The animation in this example shows how the STRING works by stepping though each clause in the STRING statement.

Click on the diagram below to step through the animation. Left click or PageDown to go to the next step and right click or press PageUp to go to the previous step.

Example 2 (animation)

The WITH POINTER phrase is often very useful because it means that we don't have to STRING all the source strings together in one go.

Using the WITH POINTER we can build the destination string by executing a number of separate STRING statements or by executing a STRING statement a number of times.

In this example we show how a destination string may be built a piece at a time by executing several separate STRING statements. At the end of this unit in the combined example you will see how a STRING statement may be used within a loop to build a destination string.

Click on the diagram below to step through the animation. Left click or PageDown to go to the next step and right click or press PageUp to go to the previous step.

Self assessment questions/examples

The examples below consist of data delarations and a number of STRING statements.

For each STRING example, write out the text that would be displayed by the DISPLAY statement. Assume that the data starts off fresh for each STRING statement.

Treat the examples as an opportunity to test your understanding of the STRING verb. Before you click on the answers in the frame below write down your own answer.

Data Declarations.

```
01 StringFields.
   02 Field1   PIC X(18) VALUE "Where does this go".
   02 Field2   PIC X(30) VALUE "This is the destination string".
   02 Field3   PIC X(15) VALUE "Here is another".
   02 Field4   PIC X(15) VALUE SPACES.

01 StrPointers.
   02 StrPtr  PIC 99.
   02 NewPtr  PIC 9.
```

STRING examples.

```
1. STRING Field1 DELIMITED BY SPACES INTO Field2.
   DISPLAY Field2.

2. STRING Field1 DELIMITED BY SIZE INTO Field2.
```

```
   DISPLAY Field2.

3. MOVE 6 TO StrPtr.
   STRING Field1, Field3 DELIMITED BY SPACE
       INTO Field2 WITH POINTER StrPtr
       ON OVERFLOW DISPLAY "String Error"
       NOT ON OVERFLOW DISPLAY "No Error"
   END-STRING.
   DISPLAY Field2.

4. STRING Field1, Field2, Field3
       DELIMITED BY SPACES INTO Field4
   END-STRING.
   DISPLAY Field4

5. MOVE 4 TO NewPtr.
   STRING Field1 DELIMITED BY "this"
          Field3 DELIMITED BY SPACE
          "END"  DELIMITED BY SIZE
          INTO Field2
   END-STRING.
   DISPLAY Field2.

6. MOVE 4 TO NewPtr.
   STRING Field1 DELIMITED BY "this"
          Field3 DELIMITED BY SPACE
          "Tom"  DELIMITED BY SIZE
       INTO Field2 WITH POINTER NewPtr
       ON OVERFLOW DISPLAY "String Error"
       NOT ON OVERFLOW DISPLAY "No Error"
   END-STRING.
   DISPLAY Field2.
```

Answers to SAQs

Left click or PageDown to go to the next answer and right click or press PageUp to go to the previous answer.

To top of page

Copyright Notice

Last updated : April 1998
e-mail : CSISwebeditor@ul.ie