

#22 Using "@" in string comparisons.

Written by Dave Terry

Published Date December 14, 1987

Using the wildcard character in place of the the Substring or Position functions.

It is stated many places in the manual that the @ sign can be used as a wildcard character. The examples show it used in several searches and the procedure editor. However, there is one more place it can be used, in string comparisons. Look at the following code:

```
x:="ABCDE"
Flag:=(x="A@")
```

In this case "Flag" would be assigned the value True. You can use this scheme to test whether a variable contains certain characters without having to use either the Substring or Position functions. Below is an example that uses this capability.

Checkbox Arrays

This example shows one way to use a scrollable area in place of a large number of checkboxes. As the user selects an element, a bullet (or any character) is placed to the left of that element and the element is de-selected. The opposite occurs if the element already has a bullet. The button "bCheck" is a *button* type button and just demonstrates that it is easy to check for the elements selected. Note the use of both " @" (space, at sign) and "•@" (bullet, at sign) to check the first character of each element.

- ` This is a dialog layout procedure. In this case the array is loaded from a particular record that has many
- ` subrecords. The scrollable area is named "SList".

Case of

: (Before)

ALL SUBRECORDS([Subs]Sub)

- ` Put the records inot the SList array with 3 spaces in front.

APPLY TO SUBSELECTION([Subs]Sub;SList{[Subs]Sub'Seq}:=" "+[Subs]Sub'F1)

- ` The number of elements in SList is the same as the number of subrecords.

SList0:=**Records in subselection**([Subs]Sub)

: (During)

Case of

: (SList#0) ` An element of SList was selected.

If (SList{SList}=" @")

- ` If the first character is a space...

SList{SList}:="•"+**Substring**(SList{SList};3;100)

- ` In the Geneva font, a bullet happens to be the same size as two spaces.



Else

` If the first character is not a space (it must be a bullet).



```
        SList{SList}:=" "+Substring(SList{SList};2;100)
    End if
    SList:=0    ` De-select the element.
    ` You must force a redraw after changing an element.
    REDRAW(SList)
: (bCheck=1)
    i:=0
    x:=""
    While (i<Slist0)
        i:=i+1
        If (SList{i}="*") ` Starts with a bullet.
            x:=x+", "+String(i)
        End if
    End while
    If (Length(x)>2)
        x:=Substring(x;2;255) ` Get rid of first comma.
        ALERT("You have selected the following elements: "+Char(13)+x)
    Else
        ALERT("You have selected no elements.")
    End if
End case
End case
```

