

#7 Loading 4th Dimension's Scrollable Area Variable

Written by Ron Dell'Aquila

Published Date

August 1, 1987

Modified Date

March 4, 1988

This technote describes how to load a scrollable area variable with formatted information.

Unlike included layouts, the scrollable area allows you to view multiple lists simultaneously. The scrollable area variable allows you to highlight any row and the ability to limit the number of lines displayed.

Subfile displayed in an Scrollable Area

alpha.....	\$12.00
beta.....	\$5.00
alpha.....	\$3.00
del ta.....	\$7.00
beta.....	\$23.00
del ta.....	\$12.00

Highlighted Line: 2

Highlighted Text: beta.....\$5.00

In the above illustration, the scrollable area is called "vArray".

Define a scrollable area by creating a variable of the type "Scrollable Area" on your layout. In this example, the variable is named "vArray".

To load the array, use **APPLY TO SUBSELECTION** to call a global procedure which will increment an array line pointer while stepping through the subrecords.

Scrollable areas do not use the TAB (**CHAR(09)**) horizontal position character, so it is up to the user to position the information so it aligns correctly. The formatting is controlled entirely by the field formatting commands **STRING** and **SUBSTRING** in conjunction with a monospaced character font such as Courier or Monaco.

```
i:=0 `index counter
```

```
APPLY TO SUBSELECTION([File1]subfile;LoadArray ) `calls the following Global procedure
```

```
vArray0:=Records in subselection([File1]subfile) `define the size of the array in lines
```

```
vArray:=1 `highlight the first line of the array
```

The previous layout procedure's **APPLY TO SELECTION** will step down each subrecord in the subselection and apply the following global procedure. *LoadArray* will increment the array line pointer (i), then assign the current subrecord's information to the array line as selected by the array line pointer, (i).

```
`Global Procedure:LoadArray
```

```
i:=i+1
```

```
vArray[i]:=[File1]subfile'Alpha+(30-(Length([File1]subfile'Alpha))*".")+ (15-
```

```
Length(String([File1]subfile'Numeric;"$#,##0.00"))*".")+String([File1]subfile'Numeric;"$#,##0.00")
```

In the above procedure, variable vArray{n} is concatenated using one line for optimum speed . The same results can be obtained by assigning 2 local variables then concatenating them together when assigning vArray.

