What is an Array?

An array is a set of related data items which can be accessed by a common name. normally the values stored should be of the same data type, but foxpro allows you to store values of different data types in tan array. In foxpro, an array is up of rows and columns.

Arrays in foxpro are of two types

1. **Single Dimensional**
2. **Double Dimensional**

A **single dimensional** array has multiple rows and columns.

Example of an array having 3 rows and 1 column

Column 1	
Element 1	Row 1
Element 2	Row 2
Element 3	Row 3

Each location in the array is called an element.

A **double dimensional** array having 4 rows and 3 columns

Column 1	Column 2	Column3	
			Row 1
			Row 2
			Row 3
			Row 4

Declaring or Creating an Array

Creating an array involves specifying a name for the array, and the size of the array in terms of number of rows and columns. An array can be created with the following command

```
DIMENSION <array name>(<number of rows>,[<number
of columns>])
```

The array name can be 10 character long. The naming conventions are similar to that of memory variables.

Single Dimensional Array

As you notice, specifying the number of columns is optional. If the number of columns is not specified, then the array is assumed to have only 1 column. This is a single dimensional array.

Double Dimensional Array

If however, both the number of rows and number of columns are specified, then the array is treated as a double dimensional array.

Example

DIMENSION Arr1(4)

creates a single dimensional array Arr1, having 4 rows and 1 column.

DIMENSION Arr2(3,6)

creates a double dimensional array Arr2 having 3 rows and 6 columns.

DIMENSION Arr3(3,4), Arr4(5,6)

creates array Arr3 having 3 rows and 4 columns, and Arr4 having 5 rows and 6 columns.

When an array is created, all elements are initialised to a logical .F., This can be seen by using the command **DISPLAY MEMORY**.

Accessing Array Elements

The elements of a single dimensional array can be accessed by specifying the array name, the row number enclosed within parenthesis.

Example

DIMENSION Arr5(3)

Arr5(1)
Arr5(2)
Arr5(3)

The elements of a double dimensional array can be accessed by specifying the array name, row number and the column number enclosed within parenthesis.

Example

DIMENSION Arr(2,3)

Arr(1,1)	Arr(1,2)	Arr(1,3)
Arr(2,1)	Arr(2,2)	Arr(2,3)

The number used to access an element of the array is called a **subscript**.

The number of elements in an array can be calculated by multiplying the number of rows by the number of columns. The maximum number of elements that an array in FoxPro can have is 3600.

Initialising Array Elements

As you have already seen, when an array is created, all elements are automatically initialised to a logical .F. Elements of any array can also be assigned values with the store command or the assignment statement.

Examples

```
DIMENSION Arr6(4)
Arr6(3) = 20
```

The third element of the array Arr6 is assigned a value of 20.

```
DIMENSION Arr7(10)
Arr7 = 10
```

All elements of the array Arr7 are initialised to 10.

Examples

```
DIMENSION Arr9(3,4)
Arr9(2,3) = 154
```

The element on the second row and third column of Arr9, is initialised to 154.

```
DIMENSION Arr8(4,3)
Arr8 = "hello"
```

All elements of the array Arr8 are initialised to hello.

II. Lists

A list appears as a vertical list of options inside a box, often with a scroll bar to the right.

Lists are often used to display the directories, files or fields from which you can choose. The "Open" dialog box which includes a list of file names from which a file name can be chosen, is a typical example of this control object.

Syntax

@ <row, column>
GET <memvar> | <field>
FROM <array> [RANGE <expN1> [, <expN2>]]
| POPUP <pop-up name>

Options

<memvar> | <field>

<memvar> or <field> must be of numeric or character type. If <memvar> or <field> is numeric, the chosen item's position in the list is stored. If <memvar> or <field> is character, the chosen item's prompt is stored.

Initial Option Selection

When a list appears, the value of <memvar> or <field> determines which list option (if any) is initially selected. For example, if the value of <memvar> or <field> is 4, the fourth option in the list is selected when the list is activated by READ. If <memvar> or <field> does not correspond to any of the options in the list (the value is less than 1 or greater than the number of options), no option is initially selected.

FROM <array>

The FROM <array> clause creates a list from an array. If the array is one-dimensional, the contents of the first array element is the first item in the list, the contents of the second array element is the second item and so on.

For example the following creates a list of MS-Office products which the user can choose from.

```
DIMENSION ARR(4)
ARR(1) = "EXCEL"
ARR(2) = "WORD"
ARR(3) = "POWERPOINT"
ARR(4) = "ACCESS"

STORE 1 TO MLIST

@ 2,10 GET MLIST FROM ARR
READ
```

The list would look like as in Figure 1.2

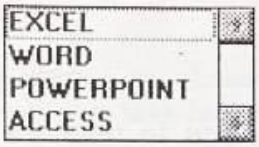

Figure 1.2

RANGE <expN1> [, <expN2>]

List items by default start with the contents of the first array element. You can however designate a different starting element in the array by including RANGE <expN1>. For example, if the array is one-dimensional and <expN1> is 3, the third element in the array is the first item in the list and the fourth element is the second item.

Example

```
DIMENSION ARR(4)
ARR(1) = "EXCEL"
ARR(2) = "WORD"
ARR(3) = "POWERPOINT"
ARR(4) = "ACCESS"

STORE 1 TO MLIST

@ 2,10 GET MLIST FROM ARR RANGE 3
READ
```