VBA ARRAYS

What is an Array?

We know very well that a variable is a container to store a value. Sometimes, developers are in a position to hold more than one value in a single variable at a time. When a series of values are stored in a single variable, then it is known as array variable.

Array Declaration

Arrays are declared the same way a variable has been declared except that the declaration of an array variable uses paranthesis. In the below example, the size of the array is mentioned in the brackets.

```
'Method 1 : Using Dim
Dim arr1() 'Without Size

'Method 2 : Mentioning the Size
Dim arr2(5)  'Declared with size of 5

'Method 3 : using 'Array' Parameter
Dim arr3
arr3 = Array("apple","Orange","Grapes")
```

1. Although, the Array size is indicated as 5, it can hold 6 values as array index starts from ZERO.
2. Array Index Cannot be Negative.
3. VBScript Arrays can store any type of variable in an array. Hence, an array can store an integer, string or characters in a single array variable.

Assigning Values to an Array

The values are assigned to the array by specifying array index value against each one of the values to be assigned. It can be a string.

Example :

Add a button and add the below function

```
Private Sub Constant_demo_Click()
  Dim arr(5)
  arr(0) = "1"            'Number as String
  arr(1) = "VBScript"     'String
  arr(2) = 100       'Number
  arr(3) = 2.45       'Decimal Number
  arr(4) = #10/07/2013#   'Date
  arr(5) = #12.45 PM#     'Time

  msgbox("Value stored in Array index 0 : " & arr(0))
  msgbox("Value stored in Array index 1 : " & arr(1))
  msgbox("Value stored in Array index 2 : " & arr(2))
  msgbox("Value stored in Array index 3 : " & arr(3))
  msgbox("Value stored in Array index 4 : " & arr(4))
  msgbox("Value stored in Array index 5 : " & arr(5))
End Sub
```

When you execute the function the output is shown below:

```
Value stored in Array index 0 : 1
Value stored in Array index 1 : VBScript
```

```
Value stored in Array index 2 : 100
Value stored in Array index 3 : 2.45
Value stored in Array index 4 : 7/10/2013
Value stored in Array index 5 : 12:45:00 PM
```

Multi Dimension Arrays

Arrays are not just limited to single dimenstion and can have a maxinum of 60 dimensions. Two-dimension arrays are the most commonly used ones.

Example :

In the below example, a multi-dimension array is declared with 3 rows and 4 columns.

```
Private Sub Constant_demo_Click()
  Dim arr(2,3) as Variant ' Which has 3 rows and 4 columns
  arr(0,0) = "Apple"
  arr(0,1) = "Orange"
  arr(0,2) = "Grapes"
  arr(0,3) = "pineapple"
  arr(1,0) = "cucumber"
  arr(1,1) = "beans"
  arr(1,2) = "carrot"
  arr(1,3) = "tomato"
  arr(2,0) = "potato"
  arr(2,1) = "sandwitch"
  arr(2,2) = "coffee"
  arr(2,3) = "nuts"

  msgbox("Value in Array index 0,1 : " &  arr(0,1))
  msgbox("Value in Array index 2,2 : " &  arr(2,2))

End Sub
```

When you execute the function the output is shown below:

```
Value stored in Array index : 0 , 1 : Orange
Value stored in Array index : 2 , 2 : coffee
```

Redim Statement

ReDim Statement is used to Declare dynamic-array variables and allocate or reallocate storage space.

```
ReDim [Preserve] varname(subscripts) [, varname(subscripts)]
```

- Preserve - An Optional parameter used to preserve the data in an existing array when you change the size of the last dimension.
- varname - A Required parameter, which denotes Name of the variable, which should follow the standard variable naming conventions.
- subscripts - A Required parameter, which indicates the size of the array.

Example

In the below example, an array has been redefined and then preserved the values when the existing size of the array is changed.

Note : Upon resizing an array smaller than it was originally, the data in the eliminated elements will be lost.

```
Private Sub Constant_demo_Click()
  Dim a() as variant
  i=0
```

```
  redim a(5)
  a(0)="XYZ"
  a(1)=41.25
  a(2)=22

  REDIM PRESERVE a(7)
  For i=3 to 7
  a(i)= i
  Next

  'to Fetch the output
  For i=0 to ubound(a)
    Msgbox a(i)
  Next
End Sub
```

When you execute the function the output is shown below:

```
XYZ
41.25
22
3
4
5
6
7
```

Array Methods :

There are various inbuilt functions within VBScript which help the developers to handle arrays effectively. All the methods that are used in conjuction with arrays are listed below. Please click on the method name to know in detail.

Function	Description
LBound	A Function, which returns an integer that corresponds to the smallest subscript of the given arrays.
UBound	A Function, which returns an integer that corresponds to the Largest subscript of the given arrays.
Split	A Function, which returns an array that contains a specified number of values. Splitted based on a Delimiter.
Join	A Function, which returns a String that contains a specified number of substrings in an array. This is an exact opposite function of Split Method.
Filter	A Function, which returns a zero based array that contains a subset of a string array based on a specific filter criteria.
IsArray	A Function, which returns a boolean value that indicates whether or not the input variable is an array.
Erase	A Function, which recovers the allocated memory for the array variables.