

## Re: can .range return a 1D array?

---

*Source:*

<http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.programming/2008-05/msg03027.html>

---

- *From:* "RB Smissaert" <[bartsmissaert@xxxxxxxxxxxxxxxxxxxxx](mailto:bartsmissaert@xxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Thu, 15 May 2008 20:47:01 +0100
- 

This is some useful code (as it is very fast) to change the LBound of arrays that are declared like this: Dim arr()

```
Option Explicit
Private Declare Function VarPtrAry _
Lib "msvbvm60" _
Alias "VarPtr" (Ary() As Any) As Long
Private Declare Sub CopyMemory _
Lib "kernel32" _
Alias "RtlMoveMemory" (Dest As Any, Src As Any, _
ByVal cBytes As Long)

Function GetArrayDims(arr As Variant) As Integer

'-----'
'copied from Francesco Balena at: '
'http://www.devx.com/vb2themax/Tip/18265'
'-----'

Dim ptr As Long
Dim VType As Integer
Const VT_BYREF = &H4000&

' get the real VarType of the argument
' this is similar to VarType(), but returns also the VT_BYREF bit
CopyMemory VType, arr, 2

' exit if not an array
If (VType And vbArray) = 0 Then
Exit Function
End If

' get the address of the SAFEARRAY descriptor
' this is stored in the second half of the
' Variant parameter that has received the array
CopyMemory ptr, ByVal VarPtr(arr) + 8, 4

' see whether the routine was passed a Variant
' that contains an array, rather than directly an array
```

## Re: can .range return a 1D array?

' in the former case ptr already points to the SA structure.  
' Thanks to Monte Hansen for this fix

```
If (VType And VT_BYREF) Then  
' ptr is a pointer to a pointer  
CopyMemory ptr, ByVal ptr, 4  
End If
```

' get the address of the SAFEARRAY structure  
' this is stored in the descriptor

' get the first word of the SAFEARRAY structure  
' which holds the number of dimensions  
' ...but first check that saAddr is non-zero, otherwise  
' this routine bombs when the array is uninitialized  
' (Thanks to VB2TheMax aficionado Thomas Eyde for  
' suggesting this edit to the original routine.)

```
If ptr Then  
CopyMemory GetArrayDims, ByVal ptr, 2  
End If
```

End Function

```
Sub SetLBound(Ary() As Variant, lNewLBound As Long)
```

' "As Variant" for example only — use your specific type  
' Note that this won't work for string() or UDT() with strings  
' Sets Ary's LBound to NewBound, returns previous LBound.

'-----

```
Dim i As Integer  
Dim AryPtr As Long  
Dim PrevLBound As Long  
Dim iDims As Integer
```

```
iDims = GetArrayDims(Ary)
```

```
If iDims = 0 Then  
Exit Sub  
End If
```

```
AryPtr = VarPtrAry(Ary) ' address of address of safearray struct  
CopyMemory AryPtr, ByVal AryPtr, 4
```

```
AryPtr = AryPtr + 20 ' pointer to safearray.bounds.lLbound  
CopyMemory PrevLBound, ByVal AryPtr, 4
```

'no point altering lBound to the existing lBound  
If PrevLBound = lNewLBound Then  
Exit Sub  
End If

Re: can .range return a 1D array?

```
For i = 1 To iDims
CopyMemory ByVal AryPtr + (i - 1) * 8, lNewLBound, 4
Next
```

End Sub

```
Sub test()
```

```
Dim arr()
```

```
ReDim arr(0 To 10)
```

```
MsgBox LBound(arr), , "Original LBound"
```

```
SetLBound arr, 1
```

```
MsgBox LBound(arr), , "New LBound"
```

End Sub

RBS

"Alan Beban" <unavailable@xxxxxx> wrote in message  
[news:%23iAXR9qtIHA.3564@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:%23iAXR9qtIHA.3564@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx)

Dave Peterson wrote:

And if arr is a single row 2-D array, then

As will:

```
with application
arr = .transpose(.Transpose(arr))
end with
```

just another way...

Which reminds me . . .

```
Both arr = Application.Transpose(Application.Transpose(arr)) and
arr = Application.Index(arr,1,0)
```

will return a 1-based array, regardless of the base of the array that was input.

If the add-in functions I referred to in my previous post are available to the workbook

arr = OneD(arr) will return a 1-based or 0-based array, depending on the base of the array

Re: can .range return a 1D array?

Re: can .range return a 1D array?

that was input.

Alan Beban