

## Re: Question about 2D VB arrays

---

*Source:*

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2006-11/msg00429.html>

---

- *From:* John <[John@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:John@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Mon, 6 Nov 2006 09:22:02 -0800
- 

Dmitriy,

You state the following:

"So if you want to have the same memory layout in both languages, then you just need to switch "meaning" of subscripts to the opposite – if you think that the first dimension in C is a row, then you should think that the first subscript in VB is a column."

When you stated this, it leads one to believe that one of the languages will now be more inefficient when accessing the array. If the array is stored in contiguous memory in column major order (i.e. VB) and accessed by C, the C code will be much more inefficient, due to the fact that it expects its data to be in row major order. There is going to be a big hit due to memory access.

John

"Dmitriy Antonov" wrote:

"John" <[John@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:John@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)> wrote in message [news:9B1BA774-2D4E-4C7D-8026-B4F909F2115A@xxxxxxxxxxxxxxxxxxxx](mailto:news:9B1BA774-2D4E-4C7D-8026-B4F909F2115A@xxxxxxxxxxxxxxxxxxxx)

This following question has started several "heated" discussions within my company between the C/C++ and VB programmers and it deals with 2d visual basic arrays.

Say for instance that I want to define an array that consists of 480 rows by 640 columns. In C++ you would allocate and populate the array in the using

## Re: Question about 2D VB arrays

the  
following syntax:

```
LONG i, j, k = 0;
LONG* larray = new LONG[480* 640];
for (i=0; i<480; i++) { // loop for rows
for (j=0; j<640; j++) { // loop for columns
larray[i * 640 + j] = k; // row * columns + column
}
}
delete [] larray;
```

In memory the array is stored in the following manner

```
Hex
0x0000 0, 1, 2, 3, 4, 5.....639 (Row 1)
0x0A00 641, 642..... (Row 2)
```

Now when I look at how these arrays are allocated and populated within our company in VB it's done in the following manner

```
Dim i as long, j as long, k as long
Dim larray (0 to 639, 0 to 479) as long
```

```
k = 0
For i = 0 To 639 // columns
For j = 0 to 479 // rows
larray(i, j) = k
k = k + 1
Next J
Next I
```

The VB code above seems wrong, allocated wrong and populated wrong!

They  
allocate the array by columns then rows. All the documentation I have  
found  
on  
arrays for visual basic state that they are allocated by rows/columns.

This really screws things up when you call a COM object using a safearray  
as a  
parameter, when you get the number of rows and columns, they are  
backwards  
and the data will be scrambled.

So here are my questions:

- How are arrays supposed to be allocated and populated within VB?
- How are arrays layed out within memory?
- Is VB column or row major order?
- If column major order why does the documentation state they should be

## Re: Question about 2D VB arrays

allocated  
by Dim(Rows, Columns)?

I want an outsiders opinion because I don't think any of the VB programmers I work with really know the answer.

Any explanation would be greatly appreciated.  
Signed,

A confused C/C++ programmer.

I don't think any "official" documentation explicitly defines which subscript is Rows and which one is Columns. This is, IMO, where your problem (or subject for discussion) lies. You, as an author, will define the meaning of each subscript (or index, or dimension). C and VB use opposite layouts for arrays, which is true. I don't see a problem here as long as you know this fact. None of these two layouts, as others mentioned, has definite advantages comparing to another one. So if you want to have the same memory layout in both languages, then you just need to switch "meaning" of subscripts to the opposite – if you think that the first dimension in C is a row, then you should think that the first subscript in VB is a column.

For example, I personally, often prefer to think that the second dimension is Rows, when array is to be redimensioned with Preserve keyword, because I more often need to add or remove Rows without losing existing data, and it is not that often when I need to change number of columns. In cases, where Preserve is not used, then meaning of each subscript is more matter of personal taste and/or organizational policies and/or context of given task.

One more thing to mention – be aware that SafeArrayBound elements of SafeArray descriptor are placed in the order, which is an opposite to how subscripts are defined in VB code. So the second dimension in VB's declaration of 2D array is presented by the first SafeArrayBound element.

BTW. As presented and as mentioned by MichaelC, you are actually comparing 1D array in C with 2D array in VB, which is somewhat incorrect. You could define and populate 1D array in VB in virtually the same way and this would eliminate the "heated discussions" in your company before they started – you would compare apples with apples and not different fruits, as you are doing right now.

Dmitriy.

Re: Question about 2D VB arrays