

# Re: fastest way to change case of string

---

*Source:*

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2006-05/msg02204.html>

---

- *From:* "RB Smissaert" <[bartsmissaert@xxxxxxxxxxxxxxxxxxxx](mailto:bartsmissaert@xxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Fri, 26 May 2006 21:17:29 +0100
- 

I have Quicksort-Code wich works on onedimensional StringArrays, using the same approach, swapping only the BSTR-Pointers.

I am interested in that. Could you post it here?

I would recommend either alternatives to the 2D-Array (e.g. Recordsets, wich can be sorted in a comfortable way)

Had a look at that, but couldn't find a fast way to write the array to a Recordset. All I could see was a simple (nested) loop.

or a separate Index-Array (of Type Long), wich acts as an additional layer of indirection

Not an option in my case. I need the actual arrays to be sorted.

What I can do though (and have done) is make a separate index array, sort that and then move the rows of the original array according to the index array. The more columns the more this will pay off as there will be less data to be swapped.

RBS

"Schmidt" <[sss@xxxxxxxx](mailto:sss@xxxxxxxx)> wrote in message [news:uS8vp8PgGHA.4932@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:uS8vp8PgGHA.4932@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

"RB Smissaert" <[bartsmissaert@xxxxxxxxxxxxxxxxxxxx](mailto:bartsmissaert@xxxxxxxxxxxxxxxxxxxx)> schrieb im Newsbeitrag [news:OSgaz5LgGHA.3652@xxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:OSgaz5LgGHA.3652@xxxxxxxxxxxxxxxxxxxxxxxxxxxx)

Looking at this code it seems it a similar approach could be used to sort a variant 2-D array. The underlying quicksort would be the same, but it would be

the data swapping bit that should be different.

The arrays I am particularly interested in are arrays where I want to sort

## Re: fastest way to change case of string

on a column holding Long numbers with the other columns holding strings, non-integer and integer numbers.

This is mentioned in Matthew Curland's book, but I think it needs a helper file that is on the CD, which I don't have.

Also, the example is for 1-D arrays and not 2-D arrays.

Would you by any chance have code already that does this?

I have Quicksort-Code wick works on onedimensional StringArrays, using the same approach, swapping only the BSTR-Pointers.

But if you have a 2D-Array, a single Row takes  $\text{ColumnCount} * \text{LenB}(\text{ArrayType})$  Bytes.

There are no "RowPointers" in such a scenario, wick could be swapped efficiently, transferring only the 4 "Pointer-Bytes" – so the Array-Span-Approach makes no sense for such an array.

I would recommend either alternatives to the 2D-Array (e.g. Recordsets, wick can be sorted in a comfortable way) or a separate Index-Array (of Type Long), wick acts as an additional layer of indirection.

In your Quicksort you would then compare using this Index-Array.

Actually you have something like this

If Arr2D(RowIdx, CompareIdx) < ...

For indirect sorting you're not allowed. to access the RowIdx directly inside the compares– instead you will have to write:

If Arr2D(IdxArr(RowIdx), CompareIdx) < ...

But this way you are able, to swap only the Entries in the IndexArray, and so you achieve a better "over-all-sort-performance".

Of course, access to the "sorted" original 2D-Array (wick remains untouched) is then only possible, using the indirection over IdxArr(RowIdx), instead of using the RowIdx directly.

Olaf