

Re: working with byte arrays

Source: <http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2004-11/1503.html>

From: Peter (-)

Date: 11/13/04

Date: Sat, 13 Nov 2004 10:31:29 +0100

Thanks, Rick

I like this solution because it is simple and easy, Thanks again

I haven't make any performance tests yet but I could think that it would be faster than my current solution because the whole comparisson is done inside InstrB(no VB loop is running).

Thanks for the hint to check the size of the arrays, in my project all bytearray's start at index 0 therefore I just need to concentrate on Ubound.

Regards

Peter

"Rick Rothstein" <rickNOSPAMnews@NOSPAMcomcast.net> schrieb im Newsbeitrag news:%231WmiwNyEHA.3024@TK2MSFTNGP14.phx.gbl...

>> *I have also made a function*

>> *IsEqualByte(a() as byte,b() as byte) as boolean*

>> *because you can not do something like*

>> *if a=b then...*

>> *My IsEqualByte just compares each byte of both arrays, to my surprise*

> *a byte*

>> *array comparisson with 1MB of size was very fast*

>> *for i=0 toUbound(a) 'as I use the bytearray's as a replacement for*

> *strings*

>> *all arrays start at index 0*

>> *if a(i)<>b(i) then...*

>> *next i*

>> *If you know a better way to compare two byte arrays then please let me*

> *know.*

>

> *Not sure if this is "better", but you could use your just-gained*

> *knowledge of InstrB...*

>

> *Function AreArraysEqual(A() As Byte, B() As Byte) As Boolean*

> *If InstrB(1, A, B) = 1 And _*

> *LBound(A) = LBound(B) And Ubound(A) = Ubound(B) Then*

> *AreArraysEqual = True*

> *End If*

> *End Function*

>

> *or, as a one-liner...*

```
>  
> Function AreArraysEqual(A() As Byte, B() As Byte) As Boolean  
> AreArraysEqual = (InstrB(1, A, B) = 1 And _  
> LBound(A) = LBound(B) And UBound(A) = UBound(B))  
> End Function  
>  
>
```