

Re: Batch Replace function for Large strings

Source: <http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.programming/2005-03/7468.html>

From: Peter T (*peter_t_at_discussions*)

Date: 03/23/05

Date: Wed, 23 Mar 2005 17:53:18 -0000

As Tom suggested it would be quicker to do on the worksheet, however if strings are longer than about 900, I find Edit Search & Replace complains that the formula is too long.

With just a quick look at your code, if not all FindArray strings exist in InputString you could avoid unnecessary and time consuming Replace's by first checking:

```
If InStr(1, InputString, FindArray(i)) Then
```

There's probably some cut off where using this test increases time, in other words if most FindArray strings will be found, and a total waste of time if all 50 will be found & need replacing.

Regards,
Peter T

"R Avery" <robert.avery@gmail.com> wrote in message
news:1111592353.075990.209600@f14g2000cwb.googlegroups.com...

> *String processing in VBA is very slow when strings are large. I have a*
> *function that I've been using for doing batch replace ops (below), but*
> *it chokes on large strings with many replacements to do (like 50),*
> *because it has to do 50 passes of the string to perform the*
> *replacements.*

>
> *Has anyone written a fast function designed to do the same thing for*
> *large strings but only makes one pass through the data?*

>
>
>

> *Public Function BatchReplace(ByVal InputString As String, FindArray As*
> *Variant, _*

> *ReplaceArray As Variant, Optional MatchCase As Boolean = False) As*
> *String*

> *' Performs a batch of find/replace ops on a single string.*

> *Dim strErrMsg As String*

> *If UBound(FindArray) - LBound(FindArray) <> UBound(ReplaceArray) -*
> *LBound(ReplaceArray) Then*

```
> GoTo errUnequalArrays
> End If
>
> On Error GoTo errBadReplace
> Dim i As Integer
> For i = LBound(FindArray) To UBound(FindArray)
> If MatchCase Then
> InputString = Replace(InputString, FindArray(i),
> ReplaceArray(i), , , vbBinaryCompare)
> Else
> InputString = Replace(InputString, FindArray(i),
> ReplaceArray(i), , , vbTextCompare)
> End If
> Next i
>
> BatchReplace = InputString
>
> Exit Function
>
> errUnequalArrays:
> strErrMsg = "Error. The number of entries in the FindArray and
> ReplaceArray do not match."
> Err.Raise Number:=vbObjectError + 1000, source:="BatchReplace",
> Description:=strErrMsg
> Exit Function
>
> errBadReplace:
> strErrMsg = "Error. An unknown error occurred during the
> replacement operations."
> Err.Raise Number:=vbObjectError + 1001, source:="BatchReplace",
> Description:=strErrMsg
> Exit Function
>
> End Function
>
```