

Re: How to trim cell information

Source:

<http://www.tech-archive.net/Archive/Excel/microsoft.public.excel.programming/2007-10/msg03694.html>

- *From:* Ron Rosenfeld <ronrosenfeld@xxxxxxxxxx>
 - *Date:* Mon, 22 Oct 2007 22:27:54 -0400
-

On Mon, 22 Oct 2007 16:11:43 -0400, "Rick Rothstein \ (MVP - VB)" <rickNOSPAMnews@xxxxxxxxxxxxxxxxxxxx> wrote:

I could modify mine to handle the absence of a file type extension.

I think you should do so. That way the OP (or others following the thread) will have a choice. Besides, if the OP (or others readers) come from a Regular Expression background, they will more than likely find your solution far more appealing than mine.

Rick

To include filenames that might not include a type extension, a starting point is to modify sPat (pattern) so that the characters after the last dot "." are optional:

```
=====
Option Explicit
Function reFilename(str As String) As String
Dim re As Object
Dim mc As Object
Const sPat As String = "\\([^\\,]+)(\\.[^.]*)?$"

Set re = CreateObject("vbscript.regexp")
re.Pattern = sPat
If re.test(str) = True Then
Set mc = re.Execute(str)
reFilename = mc(0).submatches(0)
End If
End Function
=====
```

HOWEVER, this may no longer give accurate matches. Windows long file names may

Re: How to trim cell information

include the dot "." So really you'd need to exclude all possible file type extensions, but not other endings that are not.

I'm no expert on allowable file type extensions in Windows.

What I did below was assume that a final "." followed by three or four characters that are letters, numbers or underscore would represent a file type extension. But that other lengths or including other characters would not.

It may well be that other characters should be added to the allowable list. This can be easily done.

In any event, it does become a lot more complex with long filenames that might or might not include an extension:

```
=====
Option Explicit
Function reFilename(str As String) As String
Dim re As Object
Dim mc As Object
Const sPat As String = "\\([^\|]+)(\\.\\w{3,4})(\\.\\.\\.)*"

Set re = CreateObject("vbscript.regexp")
re.Pattern = sPat
If re.test(str) = True Then
Set mc = re.Execute(str)
reFilename = mc(0).submatches(0) & mc(0).submatches(3)
End If
End Function
=====
```

It'd become even more complicated if we allowed other file systems!

Best,
--ron
.