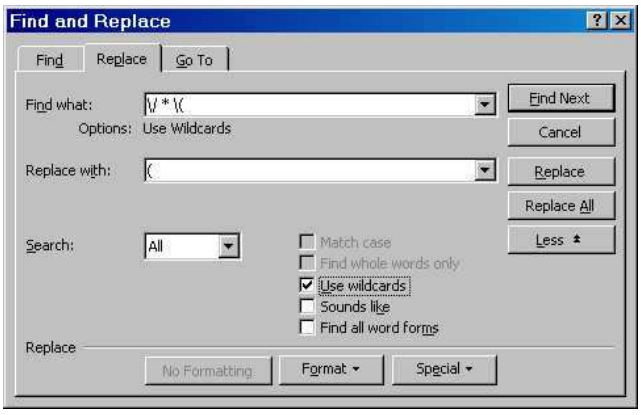
TECH EXCHANGE No. 7 – Microsoft Word Tips and Techniques (Part 2)

In this part we will take a look at a couple of more advanced Word techniques and also start our introduction to macros and programming Microsoft Word using Visual Basic for Applications (VBA). In the following text we will use the notation **Menu | Menu Item** to mean select **Menu Item** from menu **Menu**. We will start with a quick text selection tip - The F8 key for extending a selection. Place the cursor at the beginning of the text you wish to mark, then press F8 and you'll see that each subsequent press (after the first, which appears to do nothing), selects first one word, then the rest of the sentence, then the next sentence or paragraph and finally the whole document. For those interested the included table describe what the common '**Fn**' keys do in Word.

Keyboard Shortcut	Description
F1	Help
F2	Move selected text
F3	Insert AutoText
ALT-F3	Make current selection AutoText Entry
F4	Repeat last action
F5	Open Find and Replace dialog
F6	Move to next pane or frame
F7	Run spell checker
F8	Extends a selection (see article)
ALT-F8	Open macro dialog
F9	Update fields
F10	Select File menu
F11	Move to next field
F12	Open Save As dialog

Tables are a very versatile tool in Word. As well as offering a way of displaying columnar data, they provide a way of aligning graphics in a document. For example say we have four images that we wish to layout as a 2z2 grid. First use **Table | Insert Table** ... to insert a 2 row, 2 column table into our document. Then use **Inset | Picture | From File ...** to insert an images into each of the four table cells. We would then select the table and use **Format | Borders and Shading ...** to turn off all the table borders.

We are all familiar with searching for a section of text and replacing it with another, but what if the text we are searching for has the same pattern each time but differs in its content? Here we can use word's wildcard pattern-matching feature. To put this into context lets take an example - suppose we have a list of items in two languages with an associated price. For example '**dog / pero (450€)**'. Our task, if you should choose to accept it, is to remove the Spanish leaving just the English and a price. So for the above example we would be left with '**dog (450€)**'. Select Edit | Find or press Ctrl-F to display the find dialog box. Click the '**More**' button to get the expanded dialog box as shown in the screen shot. You will need to place a tick in the 'Use wildcards' checkbox. Now comes the clever bit. In the **Find What** field type '\/ ***\(**' and in the **Replace With** field type '**(**' (without the quotes in both cases). For those interested see included table for the supported wildcard characters. In English we are replacing everything from a '/' to the '(' with a '(', remember a '*' means zero or more characters.



Character	What It Matches
?	Any single character
*	Zero or more characters
[character-list]	Any single character in character-list
[!character-list]	Any single characters not in character-list
{n}	Exactly n occurrences of the previous character
{n,}	At least n occurrences of the previous character
{n,m}	From n to m occurrences of the previous character
@	One or more occurrences of the previous character
<	The beginning of a word
>	The end of a word
\	Match literal

These are actually a subset of much larger set that makes up '**Regular Expressions**'.

So what's a macro? A macro is a small program that contains a list of instructions that you wish performed. Like DOS batch files, macros combine several operations into a single procedure that you can invoke quickly to automate repetitive tasks in an application. By far the easiest way to create a macro is to use the Macro Recorder. With this method you just run through the task you wish to automate - which can include selecting text, menu commands and dialog box options. The Macro Recorder translates everything into the appropriate VBA statements. A VBA project is created and these statements are copied to an area called a module where you can replay the entire procedure anytime you like. Use the following steps to record a macro. Select **Tools | Macro | Record New Macro**. Using the displayed dialog give the macro a name, shortcut key and description. Clicking the **Ok** button starts the recording. Perform the tasks you want to include in the macro. When you have finished select **Tools | Macro | Stop Recording**. You can run and edit the macros you have recorded via the Macros dialog - select **Tools | Macro | Macros or press Alt+F8**. There is also a toolbar that gives you quick access to the macro functionality - select **View | Toolbars | Visual Basic**.

With the release of Office 97, all the big four - Word, Excel, Access and PowerPoint - have VBA at their core. Other companies have licensed VBA from Microsoft to include in their own products. VBA takes over where macro languages left off. So what can we do with VBA? In the next article we will look at some simple tasks we can do with VBA. This week's recommended site is the Magazine Portal (http://www.magportal.com). That's it for this week, please send any questions or comments to techexchange@endhousesoftware.com.