Appendix 3 – Codes for wildcard F&R

Jack Lyon of the Editorium supplies a free wildcard summary that you can download from: *www.editorium.com/wildcard_reference.pdf*

Jack recommends printing it on 8.5×11″ cardstock, both front and back (each side will be different), and then cutting the card-stock in half lengthwise (at 4.25″). That will give you a handy reference card to keep by your computer and another card to give to a friend. You might also be interested in Jack's *Wildcard Cookbook*, which will teach you how to use wildcards from beginning to end: *editorium.com/archive/wildcard-cookbook-for-microsoft-word/*

^1 Picture – use [^g] on Mac
^2 Auto-referenced endnote (*not* ^e)
^2 Auto-referenced footnote (*not* ^f)
^5 Comment mark
^9 Tab (although ^t seems to work OK) – [\^9] on Word 2004
^11 New line
^12 Page *or* section break
^13 Carriage return (*not* ^p) – [\^13] (on Word 2004) (I think that just \^13 might also work)
^14 Column break
^19 Opening field brace (when the field braces are visible)
^21 Closing field brace (when the field braces are visible)

? Finds any single character: 'c?t' finds 'cat', 'cut', and 'cot'.
* Finds any string of (zero or more) characters: 'b*d' finds 'bad', 'bread', and 'bewildered' – and it finds 'bd'.
[ ] Finds one of the specified characters: 'b[ai]t' finds 'bat' and 'bit' but not 'bet'.
[-] Finds any single character in the specified range (which must be in ascending order): '[l-r]ight' finds 'light', 'might ', 'night', and 'right' (and 'oight', 'pight', and 'qight', if they exist).
[!] Finds any single character except those specified: 'm[!u]st' finds 'mist' and 'most' but not 'must'. 't[!ou]ck' finds 'tack' and 'tick' but not 'tock' or 'tuck'.
[!x-z] Finds any single character except those in the specified range: 't[!a-m]ck' finds 'tock' and 'tuck' but not 'tack' or 'tick'.
{n} Finds exactly n occurrences of the previous character or expression: 're{2}d' finds 'reed' but not 'red'.
{n,} Finds at least n occurrences of the previous character or expression: 're{1,}d' finds 'red' and 'reed' (and 'reeeed'!).
{n,m} Finds from n to m occurrences of the previous character or expression: '10{1,3}' finds '10', '100', and '1000', but not '10000'.
@ Finds one or more occurrences of the previous character or expression, if there are any: 'me@t' finds 'met' and 'meet'.
< Finds the beginning of a word: '<inter' finds 'interest' and 'interrupt' but not 'splinter'.
> Finds the end of a word: 'in>' finds 'in' and 'main' but not 'inspiring'.