Wildcarding resources

If you want to improve your wildcarding skills, I've done some YouTube tutorials, listed below. But also in this file are various wildcard resources, some based (with his kind agreement) on Jack Lyon's book, *Wildcard Cookbook*, which will teach you how to use wildcards from beginning to end.

Wildcarding techniques 1 (39:56)
Becoming familiar with the commands used in wildcard F&R

Wildcarding techniques 2 (17:44)
The command I forgot to show you in Part 1!

Wildcarding techniques 3 (20:45)
Looking at a few real-life wildcard F&Rs

Developing wildcard searches (2:09)
Speed up the process of fault-finding a wildcard search

Special codes for wildcard F&R

^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)
^11 New line
^12 Page *or* section break
^13 Carriage return (*not* ^p)
^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'.

Some useful wildcard expressions

First let me try to explain my terminology:

1) Here is an example of what I call an 'expression':

[0-9]{4}

If you type this into the Find box and click 'Use wildcard' and then 'Find Next', it will jump to the next year, e.g. '2009', well, to the next four-digit number.

2) Ordinary finds and expressions can be grouped, by using ( ), then those groups can be referred to by using \1 for the first group, \2 for the second, etc.

3) Using [ ] specifies a range, i.e. "Find one of these characters.". For example...

[02468] finds an even number

[0-9] finds one digit

[a-z] finds a single lowercase character

[A-Z] finds a single uppercase character

Using @ after a character or a range means "Find *one or more* of these – but find as ***few*** as possible."

Using, say, {2} after a character or a range means "Find *two* of these."

Using, say, {2,4} after a character or a range means "Find *two, three or four* of these."

Using, say, {2,} after a character or a range means "Find *two, or more* of these – but find as ***MANY*** as possible."

Examples:

o{1,} Finds: pr**o**fit, pr**oo**freader, sp**ooo**f!

o@ Finds: pr**o**fit, pr**o**ofreader, sp**o**oof!

o@f Finds: pr**of**it, pr**oof**reader, sp**ooof**!

Useful example expressions:

[a-z]{1,} lowercase word

[A-Z]{1,} uppercase word

[a-zA-Z]{1,} mixed-case word, e.g. Beverley, McTaggert, BBC, hello

N.B. Some websites and books advocate use of [A-z]. This is ***wrong***, (a) it does not find accented characters, such as in 'façade' or 'déjà vu', and (b) it ***includes*** the characters with ASCII codes between 'Z' and 'a', that is: **[**, \, **]**, ^, _, and ` (back tick).

[0-9]{3,} multi-digit number, for 100,000, it would first find just 100, then 000

[,0-9]{5,} multi-digit number with commas, e.g. 100,000 or 4567 or 1234,6785,000

[hw][tw][tw][a-zA-Z0-9:/.\-_=\?&]{6,} URL (if you think of more characters needed, please tell me!)

[A-Ω] uppercase Greek character

[α-ω]	lowercase Greek character
[A-Я]	uppercase Cyrillic character (may be wrong – I don't know Cyrillic!)
[а-я]	lowercase Cyrillic character (may be wrong – I don't know Cyrillic!)
[a-z]@ing	present participles
[a-z]@ed>	past participles

N.B. [a-z]{1,}ing wouldn't work because for, say, 'running', the [a-z]{1,} part would find the whole of 'running', and 'running' is ***not*** followed by 'ing'. Remember, {1,} means 'find as many as possible'.

And because they might prove useful, here is some more search information:

Appendix 1 – Ordinary codes for F&R

Character	Find	Replace
any character	^?	
any digit	^#	
any letter	^$	
caret	^^	
clipboard contents		^c
column break	^n	^n
(Mac only?) Comment	^a	^a
'what you just found'		^&
endnote mark	^e	
field	^d	
footnote mark	^f	
hyphen, nonbreaking	^~	^~
hyphen, optional	^-	^-
line break	^l	^l
manual page break	^m	^m
paragraph mark	^p (or ^13)	^p (**not** ^13)
section break	^b	
space, ordinary	^32	
space, nonbreaking	^s	^s
tab character	^t	^t
white space (i.e. space or tab)	^w	

Appendix 2 – Special codes for non-wildcard F&R

Character	Find
(¼)	^0188
(½)	^0189
(¾)	^0190
(¹)	^0185
(²)	^0178
(³)	^0179
bullet (•)	^0149
caret character (^)	^^
copyright (©)	^0169
degree symbol (°)	^0176
divide (÷)	^0247
ellipsis (…)	^0133
em dash	^+
en dash	^=
endnote	^e
euro (€)	^0128
footnote	^f
hash (#)	^35
hyphen, nonbreaking	^~
hyphen, optional	^-
masculine ordinal (as in Nº) (º)	^0186
medial dot (•)	^0183
multiply (×)	^0215
vertical bar (|)	^0124
paragraph mark	^p
quote, open single curly (‘)	^0145
quote, close single curly (’)	^0146
quote, single unsexed (')	^39 Note 1
quote, open double curly (“)	^0147
quote, close double curly (”)	^0148
quote, double unsexed (")	^34 Note 2
registered trademark (®)	^0174
space	^32
space, nonbreaking	^s
tab character	^t
tilde (~)	^0126
trademark (™)	^0153
Unicode numbers (e.g.)	^u8211 (Unicode 8211 = en dash)

Note 1: You have to use ^39 in non-wildcard Find, but in wildcard Find, you can just use '.
Note 2: Ditto. In a wildcard Find you can just use ".

Appendix 5 – ASCII codes

(N.B. For **Mac users**, it seems that some codes may be different. They are listed here.)

33	**!**	78	**N**	123	**{**	168	¨	213	**Õ**
34	**"**	79	**O**	124	**|**	169	©	214	**Ö**
35	**#**	80	**P**	125	**}**	170	ª	215	**×**
36	**$**	81	**Q**	126	**~**	171	«	216	**Ø**
37	**%**	82	**R**	127	▯	172	¬	217	**Ù**
38	**&**	83	**S**	128	**€**	173	-	218	**Ú**
39	**'**	84	**T**	129	•	174	®	219	**Û**
40	**(**	85	**U**	130	**‚**	175	¯	220	**Ü**
41	**)**	86	**V**	131	***ƒ***	176	°	221	**Ý**
42	*****	87	**W**	132	**„**	177	±	222	**Þ**
43	+	88	**X**	133	**…**	178	²	223	**ß**
44	**,**	89	**Y**	134	†	179	³	224	**à**
45	**-**	90	**Z**	135	‡	180	´	225	**á**
46	**.**	91	**[**	136	**ˆ**	181	**µ**	226	**â**
47	/	92	\	137	**‰**	182	¶	227	**ã**
48	**0**	93	**]**	138	**Š**	183	•	228	**ä**
49	**1**	94	^	139	‹	184	¸	229	**å**
50	**2**	95	_	140	**Œ**	185	¹	230	**æ**
51	**3**	96	`	141	•	186	º	231	**ç**
52	**4**	97	**a**	142	**Ž**	187	»	232	**è**
53	**5**	98	**b**	143	•	188	¼	233	**é**
54	**6**	99	**c**	144	•	189	½	234	**ê**
55	**7**	100	**d**	145	**‘**	190	¾	235	**ë**
56	**8**	101	**e**	146	**’**	191	¿	236	**ì**
57	**9**	102	**f**	147	**“**	192	**À**	237	**í**
58	**:**	103	**g**	148	**”**	193	**Á**	238	**î**
59	**;**	104	**h**	149	•	194	**Â**	239	**ï**
60	<	105	**i**	150	–	195	**Ã**	240	**ð**
61	=	106	**j**	151	—	196	**Ä**	241	**ñ**
62	>	107	**k**	152	**˜**	197	**Å**	242	**ò**
63	**?**	108	**l**	153	**™**	198	**Æ**	243	**ó**
64	**@**	109	**m**	154	**š**	199	**Ç**	244	**ô**
65	**A**	110	**n**	155	**›**	200	**È**	245	**õ**
66	**B**	111	**o**	156	**œ**	201	**É**	246	**ö**
67	**C**	112	**p**	157	•	202	**Ê**	247	**÷**
68	**D**	113	**q**	158	**ž**	203	**Ë**	248	**ø**
69	**E**	114	**r**	159	**Ÿ**	204	**Ì**	249	**ù**
70	**F**	115	**s**	160		205	**Í**	250	**ú**
71	**G**	116	**t**	161	**¡**	206	**Î**	251	**û**
72	**H**	117	**u**	162	**¢**	207	**Ï**	252	**ü**
73	**I**	118	**v**	163	**£**	208	**Đ**	253	**ý**
74	**J**	119	**w**	164	¤	209	**Ñ**	254	**þ**
75	**K**	120	**x**	165	**¥**	210	**Ò**	255	**ÿ**
76	**L**	121	**y**	166	¦	211	**Ó**		
77	**M**	122	**z**	167	**§**	212	Ô		

Appendix 6 – Useful unicode numbers

0160 hard space
0215 ×
0247 ÷
8201 thin space
8211 – (en)
8212 — (em)
8216 ‘
8217 ’
8220 “
8221 ”
8722 − (minus)

U+2153	⅓	1/3
U+2154	⅔	2/3
U+2155	⅕	1/5
U+2156	⅖	2/5
U+2157	⅗	3/5
U+2158	⅘	4/5
U+2159	⅙	1/6
U+215A	⅚	5/6
U+215B	⅛	1/8
U+215C	⅜	3/8
U+215D	⅝	5/8
U+215E	⅞	7/8