]	Terminates the character class

Backslash in Regular Expressions

The backslash character has several uses. First, if it is followed by a nonalphanumeric character, it takes away any special meaning that character may have. This use of backslash as an escape character applies both inside and outside character classes.

For example, the * character normally means "match zero or more of the preceding subpattern". If you want to match a * character, you write * in the pattern. This escaping action applies whether or not the following character would otherwise be interpreted as a metacharacter, so it is always safe to precede a nonalphanumeric with backslash to specify that it stands for itself. In particular, if you want to match a backslash, you write \\.

Note: Because Igor also has special uses for backslash (see **Escape Sequences in Strings** on page IV-14), you must double the number of backslashes you would normally use for a Perl or grep pattern. Each pair of backslashes sends one backslash to, say, the **Grep** command.

For example, to copy lines that contain a backslash followed by a *z* character, the Perl pattern would be "\\z", but the equivalent Igor Grep expression would be /E="\\\\z".

Igor's input string parser converts "\\" to "\" so, when you write /E="\\\\z", the regular expression engine sees /E="\\z".

This difference is important enough that the PCRE and Igor Patterns (using Grep /E syntax) are both shown below when they differ.

Only ASCII numbers and letters have any special meaning after a backslash. All other characters are treated as literals.

If you want to remove the special meaning from a sequence of characters, you can do so by putting them between \Q and \E. This is different from Perl in that $ and @ are handled as literals in \Q...\E sequences in PCRE, whereas in Perl, $ and @ cause variable interpolation. Note the following examples:

Igor Pattern	PCRE Pattern	PCRE Matches	Perl Matches
\\Qabc$xyz\\E	\Qabc$xyz\E	abc$xyz	abc followed by the contents of $xyz
\\Qabc\\$xyz\\E	\Qabc\$xyz\E	abc\$xyz	abc\$xyz
\\Qabc\\E\\$\\Qxyz\\E	\Qabc\E\$\Qxyz\E	abc$xyz	abc$xyz

The \Q...\E sequence is recognized both inside and outside character classes.

Backslash and Nonprinting Characters

A second use of backslash provides a way of encoding nonprinting characters in patterns in a visible manner. There is no restriction on where nonprinting characters can occur, apart from the binary zero that terminates a pattern, but when a pattern is being prepared by text editing, it is usually easier to use one of the following escape sequences than the binary character it represents:

Igor Pattern	PCRE Pattern	Character Matched
\\a	\a	Alarm, that is, the BEL character (hex 07)
\\cx	\cx	“Control-x”, where x is any character
\\e	\e	Escape (hex 1B)
\\f	\f	Formfeed (hex 0C)
\\n	\n	Newline (hex 0A)
\\r	\r	Carriage return (hex 0D)