Regular Expressions

Regular expressions is a powerful tool to help match text using a pattern, they can be used in many places such as searching log files, searching text-buffers, highlight matching and scripting, .

See also $regex, $regsub, $regsubex, $regml.

Regular Expressions Engines

As of 2.9 AdiIRC has built-in support for PCRE2 expressions provided that Visual C++ 2015 x86 is installed for 32 bit AdiIRC or Visual C++ 2015 x64 is installed for 64 bit AdiIRC.

AdiIRC will automatically use PCRE2 if available, although it can be disabled by typing /setoption **Misc UsePcre False**, in this case AdiIRC will fallback to .NET regular expressions.

The .NET regular expression engine is different from the PCRE engine which mIRC uses, some differences are converted from PCRE to .NET while others are not possible.

Read more about PCRE2 regular expressions.

http://www.regular-expressions.info/pcre2.html
http://www.regular-expressions.info/tutorial.html

Read more about .NET regular expressions:

http://regexhero.net/reference/
https://msdn.microsoft.com/en-us/library/hs600312%28v=vs.110%29.aspx
https://msdn.microsoft.com/en-us/library/az24scfc%28v=vs.110%29.aspx
http://www.regular-expressions.info/dotnet.html

Modifiers

/g /G - Enables global match.
/i /I - Enables case in-sensitive match.
/S - Strips any control codes before matching ($hfind will ignore this).
/s - Enables single line match.
/m /M /c /C - Enables multi line match.
/x /X - Eliminates unescaped white space from the pattern.
/U - Enables non greedy mode. (Tries to replace greedy patterns with non greedy patterns + > +?, * -> *?)
/u - Enables UTF8 instead of ASCII regular expression.
/F - Allow/include empty capture groups.
/E /D - TODO

Differences between .NET and PCRE

When PCRE support is not available, AdiIRC translate some PCRE patterns into .NET patterns.

(*UTF8)/(*UTF) -> Enables UTF8 instead of ASCII regular expression.
(?R) -> .*
(?2) -> .*
(?1) -> .*
++ -> +
[:alnum:] -> a-zA-Z0-9
[:alpha:] -> a-zA-Z
[:ascii:] -> \x00-\x7F
[:blank:] -> \s\t
[:cntrl:] -> \x00-\x1F\x7F
[:digit:] -> 0-9
[:graph:] -> \x21-\x7E
[:lower:] -> a-z
[:print:] -> \x20-\x7E

[:punct:] -> !"#$%&'()*+,\-./:;<=>?@[\\\]^_`{|}~
[:space:] -> \s\t\r\n\v\f
[:upper:] -> A-Z
[:word:] - > A-Za-z0-9_
[:xdigit:] -> A-Fa-f0-9
\cc -> \x003
\co -> \x00F
\cb -> \x002
\x\{([A-Fa-f0-9]{1,4})\} -> \uXXXX
\Q \E tries to escapes all characters in between

\K is not available in .NET, use (<=abc)d instead.

These have no .NET counterpart:

code (?{…})
recursive (R), (R1), (R&name)
define (DEFINE).

List of differences between .NET and PCRE https://stackoverflow.com/questions/3417644/translate-perl-regular-expressions-to-net