Using Scripting

Understanding scripting will give you a great set of new abilities and ways to use Snak. A script is a mini-program with variables and functions. An alias is one kind of a script, an entry in the Tools menu is another. An alias complements the built-in commands and is used the same way.

You can even replace some of the built in commands with aliases to provide enhanced functionality. The /msg and /query scripts in the "nicks" script file is an example of this. They redefine the built in commands of the same names to allow you to use abbreviated nicks for the people you send private messages to.

This release of Snak supports almost all of the ircII scripting language, but a few features are still missing.

Most built-in functions are implemented, and all the operators are functional.

Most event handlers are supported, except for handlers relating to DCC and CTCP which are not yet implemented.

The unsupported language features are:

Keywords:
/Window, /Bind

Built-in functions:
$userhost, $winnum, $winnam, $connect, $listen, $mychannels, $myservers, $curpos, $onchannel, $pid, $ppid, $chanusers, $$, $D, $H, $O, $P, $Q, $R, $S, $T, $U, $V, $W

I prioritize the implementation of these features according to the needs of the scripts that people want to run. Please let me know which language features you need most for your favorite scripts to work.

Introduction

Scripts are stored in files in the script folder, which is automatically created if missing. Script files are regular text files that can be easily edited with a text editor, like BBEdit or even SimpleText. If you edit them in a word processor, please take care to save the files as pure text without formatting.

When you open a connection, the program will create a script file in the script folder for that connection if necessary, and this script file will contain commands to read in the standard script files. It is designed this way so that you can modify the behaviour and use of each connection. For example, one connection could be used to run a bot and another could be used normally.

Snak comes with a number of script files from the regular distribution package of ircII - the grandfather of all irc clients. ircII originated on UNIX systems and it is only now that irc clients with similar capabilities have become available on the Mac.

Also included is a complete script package called PurePak.

Aliases can be complex multi-line mini-programs or they can simply be used to abbreviate frequently used commands. It is too big a subject to cover in detail here, so you are encouraged to study the example script files that come with the program to learn details. Here I will present some examples and a short reference.

Some examples

Snak comes with a number of script files from the regular distribution of ircII, and you are encouraged to open them in a text editor like SimpleText to see what you can do.

You will notice the useful /oops alias in the file called "alias" which uses an user defined variable.

In the file "action" you will find an example of the "if" statement. This example sets a string containing the possessive (his/her/the) depending on a gender variable.

That gender variable is defined in the file "basical", so if you are a woman, you may want to change the gender to "F" in order to get the correct possesive.

For more elaborate examples, look in the script files that came with the program.

The /J <channel> alias

A simple example is the /j alias. It is simply used as an abbreviation for the normal /join command. If you look in the script file "alias" you'll see that alias j is defined as /join. /join is a built in command and can be used on the input line with a parameter, and so can /j.

When you type /j, the program will replace the alias with its definition, which in this case is /join

The /op <nick> alias

A slightly more elaborate example is /op. In order to convey operator status on someone you normally have to type

/mode <channelname> +o <nickname>

However the /op alias is defined as "/mode $C +o $0" and can be used to simplify this.

You can now convey op status by typing "/op <nick>". The alias contains two "variables" $C and $0, and when the programs finds a variable in an alias the variable will be replaces by its value before executing the command. $C will be replaced by the name of

the current channel, and $0 will be replaced by the <nickname> that you typed after the /op alias.

The /oops <nick> alias

The alias /oops is is used to correct a message that went to, let's say John when it should have gone to Mary. The syntax is simply "/oops <intended nick>"

/oops is defined as

alias oops

^assign alias.oops $B

msg $. Sorry, that wasn't meant for you.

msg $0 $alias.oops

Two new variables appear here : $B which always contain the text of your last message, and $. which is the last nick you messaged. The ^ character makes the assignment "silent", meaning no message is output.

The first line copies the text of the last message to an intermediate variable called "alias.oops".
Second line sends a message to the last nick we messaged (John) to say Sorry.
Third line resends the original text (stored in the intermediate variable) to the nick that was the parameter to the alias (Mary)

The "$0" in the above is a "scripting variable" and it will be replaced by the corresponding argument from the input line. $0 will be replaced by the first argument, $1 will be replaced by the second argument and so on.

The consoleClick script

There Aliases can be run from the input field, and you can also run a particular script when double click on a user in either the user list or the notify list. The default command for a double click in the notify list is "ConsoleClick". ConsoleClick is an alias, which is defined as "/msg $E $0-".

This looks cryptic, but remember that a /msg always takes two pieces of input : the nick and the message. When Snak interprets the alias it replaces $E with the currently selected nick, and $0- with the entire contents of the input line.

The Possessive script variable

In the file "action" you can see an example of the how to affect the execution of the alias depending on outside values. As mentioned before, scripting is really a form of programming, and a programming language must have conditional statements.

Currently only the "if" statement is supported. The syntax for this statement is

IF (<variable-expression>) <true-command/s>

[<false-command/s>]

and the example in the file is
if (GENDER)

if ([$GENDER] == [F])

assign POSSESSIVE her

assign POSSESSIVE his

assign POSSESSIVE the

This is actually two "if" statements inside each other. The outermost one test if the variable GENDER is defined at all. If not then it assigns "the" to the variable POSSESSIVE.

If GENDER is defined then the program runs the innermost "if" which tests its value, and assigns "her" to POSSESSIVE if it is "F". Otherwise POSSESSIVE will be set to "his"

Scripting Variables

Snak supports numerical variables $0 thru $9
The numerical variables can be used individually or in ranges:
$n- gives argument n through the last argument
$n-m gives arguments n through m
$-m gives the first argument through m
$* gives the entire contents of the input line. Same as $0-

$.
Nick of the last person to whom you sent a message
$,
Nick of the last person who sent you a message
$:
Nick of the last person who joined the channel
$:
Nick of the last person who sent a message to the channel

$? Brings up a dialog where you can enter text.

Syntax is $?="explanation"

$B
Text of the last message you sent
$C The name of the channel
$E
Nick of the first selected user in the userlist
$F
User and host information about the first selected user in the userlist

$I
Name of the channel you were last invited to join
$N
Your nick
$K returns the character (/) that is used to make the program process some input as a command