Scripting

AdiIRC has a powerful scripting language which can be used to extend/change the clients behavior.

It's modeled after the popular mIRC scripting language, make sure you read the compatibility section.

Choose a category to read more:

Example scripts

Simple kickcounter script:

```
alias kick {
    if (!%kickcount) %kickcount = 0

    inc %kickcount

    if (!$3) {
        kick # $$2 Kick number %kickcount
        halt
    }
}
```

Kickban example

```
alias kb {
  if (!$1) {
    echo /kb - Nick missing
    return
  }

  var %msg = $iif(#, $2-, $3-)
  var %chan = $iif(#, #, $2)

  ; Set this for default ban reason, or remove for no default reason
  ; Can be shortened to %msg = $iif(%msg, %msg, GTFO)
  if (!%msg) %msg = GTFO

  if ($me isop %chan) {
```

```
    MODE %chan +b $wildsite
    KICK %chan $1 %msg
  }
  else echo You are not oper on %chan
}
```

Simple calculator script:

```
alias calc {
  if (!$1) {
    echo /calc - Parameters missing
    return
  }

  ; typing /calc -p <expression> sends output to channel
  if ($1 == -p) {
    msg # Calculating : $2-
    msg # Result is : $calc($2-)
  }
  else {
    echo Calculating : $1-
    echo Result is : $calc($1-)
  }
}
```

Colored version

```
alias calc {
  if (!$1) {
    echo /calc - Parameters missing
    return
  }

  # typing /calc -p <expression> sends output to channel
  if ($1 == -p) {
    msg # $chr(3) $+ 4Calculating : $2-
    msg # $chr(3) $+ 4Result is : $calc($2-)
  }
  else {
    echo $chr(3) $+ 4Calculating : 4$1-
    echo $chr(3) $+ 4Result is : $calc($1-)
  }
}
```

CTCP flood detection example

```
CTCP *:*:*:{
  if (!%count) set -u10 %count 1
  else inc -u10 %count 1

  if (%count > 4) ignore -tu30 $wildsite
}
```

Mass mode example

```
alias mass {
  if (!$2) {
    echo /mass - Parameters missing [+/-<mode> <nick> <nick> <nick>]
    return
  }

  %len = 2

  ; equal to while (%len <= $count(%1-, $chr(32)))
  while (%len <= $0) {
```

```
    if ($(%len) ison #) mode # $1 $($ $+ %len)
    inc %len
  }
}
```

Shows info about servers, channels and users

```
on *:JOIN:#: {
  var %s = $scon(0), %c = 0, %u = 0, %t = 0, %c2 = 0;

  while (%t < %s) {
    inc %t
    scon $scon(%t);
    inc %c $chan(0)

    %c2 = 0
    while (%c2 < $chan(0)) {
      inc %c2
      inc %u $nick($chan(%c2), 0)
    }
  }

  /echo You are on ( $+ %s $+ ) servers, ( $+ %c $+ ) channels with ( $+ %u $+ ) users
}
```

It is possible to use scripts as functions.
These functions are fully nested like the client functions.

Lets say you make a /mycalc like this.

```
alias mycalc {
  return $calc($$1 + $$2);
}
```

Then you can call this function with eiter /mycalc <number> <number> the normal way or $mycalc(<number, <number>)
Typing /testcalc will show the result.

```
alias testcalc {
  echo -a $1 + $2 is $mycalc($1, $2);
  echo -a 5 + 4 is $mycalc(5, 4);
}
```

Simple convert temperature C to F or F to C
/temp C 20 will print 68 F

```
alias temp {
  if ($1 == C) echo $calc(($2 * 9/5) + 32) F
  else if ($1 == F) echo $round($calc(($2 - 32) * 5/9), 1) C
  else echo Temp missing
}
```

Announce song changes in a channel or to a user

```
On *:SONG:{
  nmsg <network> <channel/Nick> $1-
}
```

Announce to several channels with:

```
On *:SONG:{
  nmsg <network> <channel1/Nick>,<channel2/Nick> $1-
  nmsg <network2> <channel3/Nick> $1-
```

```
}
```

Automatically find the summary of a imdb url

```
On *:TEXT:*:#k: {
  var %reg = $regex($1-, http://www\.imdb\.com(/title/[a-z0-9]+/))
  if (!%reg) { return }
  sockclose imdb
  unset %data
  %imdb = $regml(1) $+ plotsummary
  %imdbchan = #
  sockopen imdb www.imdb.com 80
}

on *:sockopen:imdb: {
  sockwrite -n imdb GET %imdb HTTP/1.1
  sockwrite -n imdb Host: www.imdb.com
  sockwrite -n imdb User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/536.5 (KHTML, like
 Gecko) Chrome/19.0.1084.24 Safari/536.5
  sockwrite -n imdb Referer: http://www.imdb.com
  sockwrite -n imdb Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/pl
ain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,text/css,*/*;q=0.1
  sockwrite -n imdb Accept-Language: en-us, en;q=0.50
  sockwrite -n imdb Connection: Close $+ $crlf $+ $crlf
}

on *:sockread:imdb:{
  sockread %text
  %data = %data $+ %text
}

on *:sockclose:imdb: {
  if ($regex(%data, <p class="plotSummary">([\s\S]*?)</p>)) {
    msg %imdbchan $regml(1)
  }

  unset %data
}
```

mIRC scripting compatibility

Although mostly compatible, AdiIRC scripting is not fully compatible, there are missing features, various bugs and differences.

As such, asking for AdIRC scripting help should not be done in mIRC specific channels or forums, unless it's a script developed for mIRC and the question is related to mIRC only.

You can ask for AdiIRC scripting help on the Scripting forum or in the #adiirc channel on chat.freenode.net.

Submitting scripting bug reports

First verify the script works correctly in mIRC unless it uses AdIRC specific features, then narrow down the bug to the smallest possible lines of code before submitting.

Include as much information about the problem as possible, such as observed behavior and what the expected behavior should be.

More information

You can find many AdiIRC scripts in the Scripting forum or mIRC scripts in places such as http://hawkee.com.

Not everyone will work, if you find one that doesn't, see above on how to report a bug.

For more on mIRC scripting and how it works, check out http://en.wikichip.org/wiki/Introduction_-_mIRC and https://web.archive.org/web/20170322040538/http://www.mirc.org/mishbox/index.htm

Known issues

Scripts checking for $version will probably not work since $version returns the AdiIRC version, not mIRC version. (Removing the version check will fix that)
Scripts reading $mircini for mIRC specific options will not work in AdiIRC since AdiIRC has different options and uses a completely different options layout.
[] brackets doesn't work inside identifiers and /if expressions without surrounding the brackets with spaces.
DLL's designed to inject them self into mIRC's memory/internals doesn't work (and never will).
DLL's designed for mIRC are 32 bit and will never work in 64 bit version of AdiIRC.
Some identifiers/commands/features are not implemented (some never will be), check the Identifiers and Commands pages.
Small number of identifiers/commands parameters are not implemented, usually with features not available in AdiIRC yet (or ever), search for TODO in the search box.
Some identifiers/commands may have slightly different behavior/results in various places, but usually nothing to worry about.
Since there are no 64 bit version of the Wscript.Shell COM object, scripts using it wont work in 64 bit AdiIRC unless tsc64 is installed.
There might be other COM objects with no 64 bit version, use 32 bit AdiIRC if necessary.