

Sockets

/socklisten

Added in 1.9.0

/socklisten [-dpE] [bindip] <name> [port] [certificate]

Listens on the specified port for connections to that port. If a port is not specified, the port is selected randomly from the range specified in [Options](#) -> [DCC.Options](#) -> [Port Range](#).

Switches

- d - Switch indicates that you specified an ip address as the bind address.
- p - Enables [UPnP](#) support for the listening socket, if that is available.
- E - Creates a SSL listening socket. **(AdiIRC only)**

Parameters

- [bindip] - Ip address to bind to. (used with -d)
- <name> - Name to give the listening connection.
- [port] - Port to listen on.
- [certificate] - the SSL certificate [pfx](#) file to use with -E, if not specified, the global client certificate is used. **(AdiIRC only)**

/sockopen

Added in 1.8.10

/sockopen [-dentl46C] [bindip] <name> <address/hostname> <port> [certificate]

Opens a socket connection to the specified ipaddress/hostname and port.

Switches

- d - The specified IP address is the bind ip address.
- e - Creates an SSL connection.
- n - Disables the [Nagle algorithm](#).
- t - **TODO**
- l - Accept invalid remote SSL/TLS certificates. **(AdiIRC only)**
- 4 - Connect to Ipv4 addresses.
- 6 - Connect to Ipv6 addresses.
- C - Indicates that [certificate] is specified. **(AdiIRC only)**

Parameters

- [bindip] - Bind ip to be used.
- <name> - Socket connection name.
- <address/hostname> - Remote ip address or hostname to connect to.
- <port> - Remote port to connect to.
- [certificate] - the SSL certificate [pfx](#) file to use with -C, if not specified, the global client certificate is used. **(AdiIRC only)**

Example

```
alias example {  
    ; Initiate a connection with "irc.libera.chat" on port 6669, Secured connection.  
    /sockopen -e IRC irc.libera.chat +6697  
}
```

```
on *:sockopen:IRC:{  
    ; Send our USER and NICK irc commands.
```

```

/sockwrite -n IRC USER Tester $+ $rand(1,100) Test Test:mSL Testing $+ $crlf
/sockwrite -n IRC NICK Tester $+ $rand(1,100) $crlf
}

; Show all receiving data in a window.
on *:sockread:IRC:{
/window -de @IRC
/var %x
/sockread %x
/aline -p @IRC $iif(%x,$v1,-)
}

```

/sockclose

Added in 1.8.10

/sockclose <name>

Closes the connection with the specified name. If you specify a [wildcard](#) name, all connections that match the [wildcard](#) are closed.

Parameters

<name> - Socket name to close.

/sockread

Added in 1.9.0

/sockread [-fn] [numbytes] <%var|&binvar>

Reads bytes from the receive buffer into the specified variable.

If you specify a %var variable, a line of text ending with a Carriage Return/LineFeed is read into %var. The [\\$crlf](#) are stripped off (this may result in %var being [\\$null](#) if the line only consisted of [\\$crlf](#)).

If you specify a &binvar then [numbytes] of info is read into the binary variable. If no [numbytes] is specified it defaults to 4096 bytes.

Switches

-f - Forces AdilIRC to fill the %var variable with whatever text is in the receive buffer, even if it does not end in a [\\$crlf](#).

-n - allows you to read a [\\$crlf](#) terminated line into a &binvar. If the incoming line does not contain a \$crlf, no bytes will be read into &binvar, unless you specify the -f switch, which forces the read (same as when reading into %vars).

Parameters

[numbytes] - Number of bytes to read into &binvar.

<%var|&binvar> - %var or &binvar to read the data into.

/sockwrite

Added in 1.8.10

/sockwrite [-tnba] <name> [numbytes] <text|%var|&binvar>

Queues data to be sent on the specified connection. AdilIRC will then try to send that info as quickly as it can. Once it has finished sending the info, it triggers the on sockwrite event so you can send more info if you need to.

You can use a [wildcard](#) name to send the same information at once to all connections that match the [wildcard](#).

If the write fails, it will set \$sock().wserr to the error value, and triggers the on sockwrite event with \$sockerr set.

Switches

- t - Forces AdIRC to send anything beginning with a & as normal text instead of interpreting it as a binary variable.
- n - Appends a [\\$CrLf](#) to the line being sent if it is not a &binvar and if it does not already have a [\\$CrLf](#).
- b - Indicates that you are specifying the numbytes value which is the number of bytes you want sent.
- a - Disables UTF-8 encoding of characters in the range 0-255, as long as the text contains no characters > 255.

Parameters

<name> - Name of the connection to send to.
 [numbytes] - Number of bytes to send. (used with -b)
 <text[%var]&binvar> - Plain text, %var or &binvar to send.

/sockpause

Added in 1.9.0

/sockpause [-r] <name>

Pauses/restarts a socket when reading incoming data.

Switches

-r - Restarts a socket instead of pausing.

Parameters

<name> - Socket to modify.

/sockmark

Added in 1.9.0

/sockmark <name> [text]

Assigns temporary data to a socket to be referenced later using [\\$sock\(<name>\).mark](#). Leaving the [text] parameter \$null will clear the socket mark. The sockmark command can be used to assign the same data to multiple sockets using a [wildcard](#) pattern.

Parameters

<name> - Socket to mark.
 [text] - Text to add to the socket mark.

Example

```
; Non-functioning example, shows practical usage only
on *:text:/^!foo (\S+)$/Si:#{
  ; if ( .. validate input .. ) {
  sockopen sock1 www.example.com 80
  sockmark sock1 $regml(1)
  }
on *:sockopen:sock1:{
  ; submit the information we got from the user
  sockwrite -nt $sockname GET /foobar.php?q= $+ sock($sockname).mark HTTP/1.0
  sockwrite -nt $sockname Host: www.example.com
  sockwrite -nt $sockname $CrLf
  }
; on *:sockread:sock1: {
;   ....
; }
```

/sockrename

Added in 1.9.0

/sockrename <name> <newname>

Assigns a new name to an existing connection.

Parameters

<name> - Socket to rename.

<newname> - New socket name.

/sockaccept

Added in 1.9.0

/sockaccept <name>

Accepts the current connection to your listening port and assigns it a name to identify it.

Parameters

<name> - Name to assign the connection.

/sockudp

Added in 2.3

/sockudp [-bntkd] [bindip] <name> [port] <ipaddress> <port> [numbytes] [text|%var|&binvar]

Opens a UDP connection to a remote ipaddress/port and sends a UDP packet.

Switches

-b - Indicates that you are specifying the numbytes value which is the number of bytes you want sent.

-n - Appends a [\\$CrLf](#) to the line being sent if it is not a &binvar and if it does not already have a [\\$CrLf](#).

-t - Forces AdilIRC to send anything beginning with a & as normal text instead of interpreting it as a binary variable.

-k - Forces the socket to stay open, so it can receive UDP data.

-d - The specified IP address is the bind ip address.

Parameters

[bindip] - Bind ip to be used.

<name> - Socket connection name.

[port] - Bind port to be used.

<ipaddress> - Remote ipaddress to connect to.

<port> - Remote port to connect to.

[numbytes] - Number of bytes to send. (used with -b)

[text|%var|&binvar] - Text, %variable or &binvar to send.

Example

```
alias gettime {
    ; Set a NULL byte binary variable.
    bset &null 1 0

    ; Open a UDP connection to Time-a.nist.gov = 129.6.15.28
    sockudp -k gettime 129.6.15.28 37 &null
}

on *:udpread:gettime: {

    ; Read the reply.
    sockread -f &time

    var %time $bvar(&time,1,$bvar(&time,0))
}
```

```

; Convert to binary, remove spaces.
var %bin $regsubex(%time, /(\d+)\s?/g, $base(\1, 10, 2, 8))

; Get the current unix time in decimal system.
var %time = $base(%bin, 2, 10)

; Print the time and close the socket.
echo -ag Currnt Time/Date: $asctime($calc(%time - 2208988800), yyyy-mm-dd hh:nn:ss TT)
sockclose $sockname
}

```

/socklist

Added in 1.9.0

/socklist [-tul] [name]

Lists all matching open sockets.

Switches

- t - Lists only tcp sockets.
- u - Lists only udp sockets.
- l - Lists only listening sockets.

Parameters

[name] - List socket matching name, can be a [wildcard](#).

/bindip

Added in 1.9.0

/bindip [on|off] <ipaddress/adapter>

Bind or toggles ip address/network adapter for DCC transfers.

/bindip with no parameters shows current status.

Parameters

- on - Sets bind on.
- off - Sets bind off.

<ipaddress/adapter>* - IpAddress or Network Adapter to bind to.

Example

```

;Binds ip to 192.168.0.1
/bindip 192.168.0.1

```

```

;Finds local ip address on Local Network Adapter and binds to it.
/bindip Local Network Adapter

```

```

;Disable binding.
/bindip off

```

\$sock

Added in 1.9.0

\$sock(name,[N])

Returns information about a the Nth socket connection.

Parameters

name - Name to lookup, can be a [wildcard](#).
[N] - If N = 0, number of matches, otherwise the Nth match. (optionally)

Properties

.name - Name of the socket.
.addr - Hostname the socket is connected to.
.sent - Number of bytes sent.
.rcvd - Number of bytes received.
.sq - Number of bytes in send queue.
.rq - Number of bytes in receive queue.
.status - Status of the connection: paused/active.
.ip - IP Addresss the socket is connected to.
.port - Port the socket is connected to.
.ls - Number of seconds since the socket last sent data.
.lr - Number of seconds since the socket last received data.
.mark - Socket mark set by [/sockmark](#).
.type - Socket type udp/tcp.
.to - Number of sockets since it was opened.
.wsmsg - Return the last socket error message.
.wserr - Return the last socket error.
.pause - Returns [\\$true](#) if socket is paused, otherwise [\\$false](#).
.ssl - Returns [\\$true](#) if socket is a SSL connection, otherwise [\\$false](#).
.sport - Returns the source port from the last received UDP packet.
.saddr - Returns the source address from the last received UDP packet.
.bindip - Returns the local ip address the socket is bound to.
.bindport - Returns the local port the socket is bound to.
.starttls - **TODO**
.certfile - Returns the server certificate file if the socket is a listening socket, otherwise the client certificate file. (**AdiIRC only**)

Example

```
; Print number of sockets.  
//echo -ag $sock(*, 0)  
  
; Print the first socket.  
//echo -ag $sock(*, 1)
```

\$sockbr

Added in 1.9.0

\$sockbr

Returns the number of bytes read during a [on SOCKREADevent](#).

Example

```
; Open a socket connection to 'google.com' port '80'.  
/sockopen Example www.google.com 80  
  
; Listen for the on SOCKOPEN event.  
on *:SOCKOPEN:Example:{  
    ; Write a GET request.  
    sockwrite -nt $sockname GET / HTTP/1.0  
    sockwrite -n $sockname  
}  
  
; Listen for the on SOCKREAD event.  
on *:SOCKREAD:Example:{  
    ; Create a local variable to hold the data.
```

```

var %i

; Read the data waiting into the local variable.
sockread %i

; Print number of bytes read.
echo -ag Read $sockbr bytes.

; Close the connection.
sockclose $sockname
}

```

\$sockerr

Added in 1.9.0

\$sockerr

Returns any error during a socket command or socket [event](#).

Example

```

; Open a socket connection to 'google.com' port '80'.
alias Example {
  /sockopen Example www.google.com 80

  ; Check if there was an error.
  if ($sockerr) {
    echo -ag There was an error: $sockerr
    halt
  }
}

; Listen for the on SOCKOPEN event.
on *:SOCKOPEN:Example:{
  ; Write a GET request.
  sockwrite -nt $sockname GET / HTTP/1.0
  sockwrite -n $sockname

  ; Check if there was an error.
  if ($sockerr) {
    echo -ag There was an error: $sockerr
    halt
  }
}

; Listen for the on SOCKREAD event.
on *:SOCKREAD:Example:{
  ; Check if there was an error.
  if ($sockerr) {
    echo -ag There was an error: $sockerr
    halt
  }
}

; Create a local variable to hold the data.
var %i

; Read the data waiting into the local variable.
sockread %i

; Print data read.
echo -ag Data = %i
}

```

\$sockname

Added in 1.9.0

\$sockname

Returns the name of a socket during [on SOCKCLOSE](#), [on SOCKLISTEN](#), [on SOCKOPEN](#), [on SOCKREAD](#) and [on SOCKWRITE events](#).

Example

```
; Open a socket connection to 'google.com' port '80'.
/sockopen Example www.google.com 80

; Listen for the on SOCKOPEN event.
on *:SOCKOPEN:*:{
; Print the name of the socket.
echo -ag Sockname is $sockname

; Close the socket.
sockclose $sockname
}
```

\$bindip

Added in 1.9.0

\$bindip(N|ipaddress)

Returns information off a network adapter.

Parameters

N|ipaddress - If N = 0, returns the total number of active network adapters else the Nth network adapter or a network adapter matching the ipaddress.

Properties

.name - Returns the name of matched network adapter.

.ip - Returns the internal ipaddress associated with the network adapter.

.loopback - Returns [\\$true](#) if the associated internal ipaddress is loopback address otherwise [\\$false](#).

\$portfree

Added in 1.9.0

\$portfree(N,[ipaddress])

Returns [\\$true](#) if the specified port number is not in use, otherwise [\\$false](#).

Parameters

N - The port number to check.

[ipaddress] - If specified, only the interface with that IP address is checked for used ports, otherwise all active interfaces are checked. An adapter name can also be used instead of an IP address.

Example

```
; Check if port 4242 is free.
//echo -ag Port 4242 is $iif($portfree(4242), open, not open)
```

\$iptype

Added in 1.9.3

\$iptype(text)

Returns "ipv4" or "ipv6" if text is is a valid IP address format.

Parameters

text - The text to validate.

Properties

-expand - Expand a ipv6 address.
.compress - Compress a ipv6 address.

Example

```
; Check a IPv4 address.  
//echo -ag 8.8.8.8 is $iptype(8.8.8.8)  
  
; Check a IPv6 address.  
//echo -ag FE80:0000:0000:0000:0202:B3FF:FE1E:8329 is $iptype(FE80:0000:0000:0000:0202:B3FF:FE1E:8329)
```

on SOCKLISTEN

Added in 1.9.0

on <level>:SOCKLISTEN:name:<commands>

Triggers when someone tries to connect to a port that you are listening on using [/socklisten](#).

See also [on SOCKOPEN](#), [on SOCKCLOSE](#), [on SOCKREAD](#), [on SOCKWRITE](#), [/socklisten](#), [/sockopen](#), [/sockaccept](#), [\\$sock](#), [\\$sockbr](#), [\\$sockerr](#), [\\$sockname](#).

Parameters

<level> - The [User Access Level](#) for the event to trigger.
<name> - The [\\$sockname](#) to listen to.
<commands> - The commands to be performed when the event listener's criteria is met.

Example

```
; Listen for requests to port 4242.  
/socklisten adiirc 4242  
  
; When someone is connecting, accept the connection.  
on *:SOCKLISTEN:adiirc:{  
    /sockaccept name  
}
```

on SOCKOPEN

Added in 1.9.0

on <level>:SOCKOPEN:<name>:<commands>

Triggers when a [/sockopen](#) command is successful and a connection has been made.

See also [on SOCKOPEN](#), [on SOCKLISTEN](#), [on SOCKREAD](#), [on SOCKWRITE](#), [/socklisten](#), [/sockopen](#), [/sockaccept](#), [\\$sock](#), [\\$sockbr](#), [\\$sockerr](#), [\\$sockname](#).

Parameters

<level> - The [User Access Level](#) for the event to trigger.
<name> - The [\\$sockname](#) to listen to.
<commands> - The commands to be performed when the event listener's criteria is met.

```
; Open a connection to AdiIRC.com, port 80.  
/sockopen adiirc adiirc.com 80
```

```
on *:SOCKOPEN:adiirc:echo -ag Connection to AdiIRC successful | sockclose adiirc
```

on SOCKWRITE

Added in 1.9.0

on <level>:SOCKWRITE:name:<commands>

Triggers when AdiIRC has finished sending all of the data that you previously queued for sending or when the socket is ready for more writing.

See also [on SOCKOPEN](#), [on SOCKCLOSE](#), [on SOCKLISTEN](#), [on SOCKWRITE](#), [on UDPREAD](#), [/socklisten](#), [/sockopen](#), [/sockaccept](#), [\\$sock](#), [\\$sockbr](#), [\\$sockerr](#), [\\$sockname](#).

Parameters

<level> - The [User Access Level](#) for the event to trigger.
<name> - The [\\$sockname](#) to listen to.
<commands> - The commands to be performed when the event listener's criteria is met.

on SOCKREAD

Added in 1.9.0

on <level>:SOCKREAD:name:<commands>

Triggers when there is info waiting to be read on the specified connection.
You can read this info using the [/sockread](#) command.

See also [on SOCKOPEN](#), [on SOCKCLOSE](#), [on SOCKLISTEN](#), [on SOCKWRITE](#), [on UDPREAD](#), [/socklisten](#), [/sockopen](#), [/sockaccept](#), [\\$sock](#), [\\$sockbr](#), [\\$sockerr](#), [\\$sockname](#).

Parameters

<level> - The [User Access Level](#) for the event to trigger.
<name> - The [\\$sockname](#) to listen to.
<commands> - The commands to be performed when the event listener's criteria is met.

Example

```
; Open a connection to AdiIRC.com, port 80.  
/sockopen adiirc adiirc.com 80
```

```
; Make a request for the index page.  
on *:SOCKOPEN:adiirc:{  
    sockwrite -n adiirc GET / HTTP/1.0  
    sockwrite -n adiirc Host: www.adiirc.com  
    sockwrite -n adiirc  
}
```

```
on *:SOCKREAD:adiirc:{  
    var %s  
    sockread %s  
    echo -ag Read line %s  
}
```

on SOCKCLOSE

Added in 1.9.0

on <level>:SOCKCLOSE:name:<commands>

Triggers when a connection is closed by the remote connection. (not by [/sockclose](#))

Parameters

<level> - The [User Access Level](#) for the event to trigger.

<name> - The [\\$sockname](#) to listen to.

<commands> - The commands to be performed when the event listener's criteria is met.

See also [on SOCKLISTEN](#), [on SOCKCLOSE](#), [on SOCKREAD](#), [on SOCKWRITE](#), [/socklisten](#), [/sockopen](#), [/sockaccept](#), [\\$sock](#), [\\$sockbr](#), [\\$sockerr](#), [\\$sockname](#).

Example

```
; Open a connection to AdiIRC.com, port 80.
/sockopen adiirc adiirc.com 80
```

```
; Make a request for the index page.
on *:SOCKOPEN:adiirc:{
    sockwrite -n adiirc GET / HTTP/1.0
    sockwrite -n adiirc Host: www.adiirc.com
    sockwrite -n adiirc
}
```

```
on *:SOCKCLOSE:adiirc:echo -ag adiirc.com closed the connection
```

on UDPREAD

Added in 2.3

on <level>:UDPREAD:name:<commands>

Triggers when there is info waiting to be read on the specified UDP connection.
You can read this info using the [/sockread](#) command.

See also [on SOCKREAD](#), [on SOCKOPEN](#), [on SOCKCLOSE](#), [on SOCKLISTEN](#), [on SOCKWRITE](#), [/sockudp](#), [\\$sock](#), [\\$sockbr](#), [\\$sockerr](#), [\\$sockname](#).

Parameters

<level> - The [User Access Level](#) for the event to trigger.

<name> - The UDP [\\$sockname](#) to listen to.

<commands> - The commands to be performed when the event listener's criteria is met.

Example

```
on *:udpread:name:{
    if ($sockerr) { echo -s An error occurred while trying to read data: $sock($sockname).wsmsg | re
turn }
    else {
        sockread %a
        echo -s received: %a
    }
}
```