The Asterisk command line interface (CLI) is reached by using the Linux shell command

asterisk -r or **rasterisk**

If you want debugging output, add one or many **v**:s

asterisk -vvvvvr

The Asterisk server has to be running in the background for the CLI to start.

If you want to run a CLI command in a shell script, use the **x** option

asterisk -rx "logger reload"

For help in the CLI mode, use the **core show help** command (formerly **help**). To get help on various applications you can use in the extensions.conf config file, use the **core show applications** command (formerly **show applications**).

General commands

- **!<command>**: Executes a given shell command
- **abort halt**: Cancel a running halt
- **add extension**: Add new extension into context
- **add ignorepat**: Add new ignore pattern
- **add indication**: Add the given indication to the country
- **agent show**: Show status of Asterisk Agents
- **debug channel**: Enable debugging on a channel
- **dont include**: Remove a specified include from context
- **help**: Display help list, or specific help on a command
- **include context**: Include context in another context
- **load**: Load a dynamic module by name
- **logger reload**: Reopen log files. Use after rotating the log files.
- **mixmonitor {start|stop|list}**: Execute a MixMonitor command.
- **no debug channel**: Disable debugging on a channel
- originate: originate a call.
- **remove extension**: Remove a specified extension
- **remove ignorepat**: Remove ignore pattern from context
- **remove indication**: Remove the given indication from the country
- **save dialplan**: *Overwrites* your current extensions.conf file with an exported version based on the current state of the dialplan. A backup copy of your old extensions.conf is *not* saved. The initial values of global variables defined in the [globals] category retain their previous initial values; the current values of global variables are not written into the new extensions.conf. Using "save dialplan" will result in losing any comments in your current extensions.conf.

- **dialplan save (1.4)**: BROKEN, doesn't parse correctly. *Overwrites* your current extensions.conf file with an exported version based on the current state of the dialplan. A backup copy of your old extensions.conf is *not* saved. The initial values of global variables defined in the [globals] category retain their previous initial values; the current values of global variables are not written into the new extensions.conf. Using "save dialplan" will result in losing any comments in your current extensions.conf.
- **set verbose**: Set level of verboseness
- **show applications**: Shows registered applications
- **show application**: Describe a specific application
- **show channel**: Display information on a specific channel
- **show channels**: Display information on channels
- **show codecs**: Display information on codecs
- **show conferences**: Show status of conferences
- **show dialplan**: Show dialplan
- **show hints**: Show registered hints
- **show image formats**: Displays image formats
- **show indications:** Show a list of all country/indications
- **show locals**: Show status of local channels
- **show manager command**: Show manager commands
- **show manager connect**: Show connected manager users
- **show parkedcalls**: Lists parked calls
- **show queues**: Show status of queues, see details here
- **show switches**: Show alternative switches
- **show translation**: Display translation matrix
- **soft hangup**: Request a hangup on a given channel – in Asterisk 1.6.2: "channel request hangup <name>"
- **show voicemail users**: List defined voicemail boxes
- **show voicemail zones**: List zone message formats
- **devstate change**: Change state of a custom device (new in Asterisk 1.6.0)

Server management

- **restart gracefully**: Restart Asterisk gracefully, i.e. stop receiving new calls and restart at empty call volume
- **restart now**: Restart Asterisk immediately
- **restart when convenient**: Restart Asterisk at empty call volume
- **reload**: Reload configuration
- **stop gracefully**: Gracefully shut down Asterisk, i.e. stop receiving new calls and shut down at empty call volume
- **stop now**: Shut down Asterisk immediately
- **stop when convenient**: Shut down Asterisk at empty call volume
- **dialplan reload**: Reload extensions and *only* extensions (formerly extensions reload)
- **unload**: Unload a dynamic module by name
- **show modules**: List modules and info about them

- **show uptime**: Show uptime information
- **show version**: Display Asterisk version info

AGI commands

- **show agi**: Show AGI commands or specific help
- **dump agihtml**: Dumps a list of agi command in html format

Database handling commands

- **database del**: Removes database key/value
- **database deltree**: Removes database keytree/values
- **database get**: Gets database value
- **database put**: Adds/updates database value
- **database show**: Shows database contents
- **database showkey**: Shows database contents: An alternative to showing keys by family with *database show*, this command shows **all** the families with a particular key

IAX Channel commands

- **iax2 debug**: Enable IAX debugging
- **iax2 no debug**: Disable IAX debugging
- **iax2 set jitter**: Sets IAX jitter buffer
- **iax2 show cache**: Display IAX cached dialplan
- **iax2 show channels**: Show active IAX channels
- **iax2 show netstats**: Show network and jitter buffer statistics for active IAX calls
- **iax2 show peers**: Show defined IAX peers
- **iax2 show registry**: Show IAX registration status
- **iax2 show stats**: Display IAX statistics
- **iax2 show users**: Show defined IAX users
- **iax2 trunk debug**: Request IAX trunk debug
- **iax debug**: Enable IAX debugging
- **iax no debug**: Disable IAX debugging
- **iax set jitter**: Sets IAX jitter buffer
- **iax show cache**: Display IAX cached dialplan
- **iax show channels**: Show active IAX channels
- **iax show peers**: Show defined IAX peers
- **iax show registry**: Show IAX registration status
- **iax show stats**: Display IAX statistics
- **iax show users**: Show defined IAX users
- **init keys**: Initialize RSA key passcodes
- **show keys**: Displays RSA key information

H323 channel commands

- **h.323 debug**: Enable chan_h323 debug
- **h.323 gk cycle**: Manually re-register with the Gatekeper
- **h.323 hangup**: Manually try to hang up a call
- **h.323 no debug**: Disable chan_h323 debug
- **h.323 no trace**: Disable H.323 Stack Tracing
- **h.323 show codecs**: Show enabled codecs
- **h.323 show tokens**: Manually try to hang up a call
- **h.323 trace**: Enable H.323 Stack Tracing

SIP channel commands

- Debugging
  - Enable
    - **sip debug**
    - **sip set debug on** (valid on 1.6.2.7)
  - Disable
    - **sip no debug**
    - **sip set debug off** (valid on 1.6.2.7)
- **sip reload**: Reload sip.conf (added after 0.7.1 on 2004-01-23)
- **sip show channels**: Show active SIP channels
- **sip show channel**: Show detailed SIP channel info
- sip show inuse: List all inuse/limit
- **sip show peers**: Show defined SIP peers (clients that register to your Asterisk server), see details
- **sip show registry**: Show SIP registration status (when Asterisk registers as a client to a SIP Proxy)
- **sip show subscriptions**: Lists all sip presence (busy lamp indication) subscriptions
- **sip show users**: Show defined SIP users

Zap channel commands

- **zap destroy channel**: Destroy a channel
- **zap show channels**: Show active zapata channels
- **zap show channel**: Show information on a channel
- **zap show status**: lists all the Zaptel spans. A span will apear here whether or not its channels are configured with chan_zap.
- **zap show cadences**: Show the configured ring cadences (available e.g with Zap/1r2).
- **zap set swgain**(<= 1.6): set the (software) gain for a channel. Temporary equivalents of rxgain and txgain in zapata.conf.
- **zap set hwgain**(<=1.6): set the hardware gain for channels that support it.

- **zap set dnd**(<=1.6) set a channel’s do-not-disturb mode on or off.

The following commands are available if the channel is built with support for libpri:

- **pri debug span**: Enables PRI debugging on a span
- **pri intense debug span**: Enables REALLY INTENSE PRI debugging
- **pri no debug span**: Disables PRI debugging on a span
- **pri show spans**: List spans and their status.
- **pri show span**: Information about a span.
- **pri show debug**: show where debug is enabled.

See section 3 of AT&T tr41459_99 to better understand what the output generated by the pri debug command means.

Bristuff adds **bri debug** which is an alias for pri debug.

TODO: SS7 support in 1.6.

Console channel commands

- dial: Dials the given extension, if specified, from the console. Can be used to initiate a call, or to dial digits during an existing call.
- **answer**: Answer a call if one is currently ringing on the console.
- **hangup**: Hangup the call if there is currently one on the console.

MGCP channel commands

- **mgcp audit endpoint**: Audit specified MGCP endpoint
- **mgcp debug**: Enable MGCP debugging
- **mgcp no debug**: Disable MGCP debugging
- **mgcp show endpoints**: Show defined MGCP endpoints

skinny channel commands

- **skinny debug**: Enable Skinny debugging
- **skinny no debug**: Disable Skinny debugging
- **skinny show lines**: Show defined Skinny lines per device

CAPI channel commands

- **capi debug**: Enable CAPI debugging

- **capi no debug**: Disable CAPI debugging
- **capi info**: Show CAPI info

Sirrix ISDN channel commands

- **srx reload**: Reload channel driver configuration; active calls are not being terminated!
- **srx show ccmsgs**: Disable / enable output of incoming callcontrol messages.
- **srx show chans**: Show info about B-Channels
- **srx show globals**: Show info about global settings
- **srx show groups**: Show info about configured groups
- **srx show layers**: Show info about ISDN stack (Layer 1, 2, 3)
- **srx show sxpvts**: Show private info about active channels
- **srx show timers**: Show info about running timers

Batch files with CLI

If you meant “can Asterisk read a series of commands from a file” the answer is no, but something like the following may do:

```
cat batch-file\
| awk ‘{printf “/usr/sbin/asterisk -r -x \”%s\”\n”, $0}’\
| sh
```

The above is very slow, though. A faster option is to use socat and write the commands directly to the Asterisk socket.

```
#!/bin/sh
while read line
do
      echo -n "$line"
      sleep 0.001
done \
| socat STDIN UNIX-CONNECT:/var/run/asterisk/asterisk.ctl
```

Another version without socat :

```
#!/bin/sh
while read line; do
  echo "$line" | tr '\n' '\0'
  sleep 0.001
done | nc -U /var/run/asterisk/asterisk.ctl
```

The short sleep is only needed to guarantee that every line is written in a separate write() call. It will not print any output from any command, though, or even report an error. And you’ll have to end your “programs” with a “quit” line.