

Command Line Interface (fs_cli)

- 0. About
- 1. Requirements
- 2. Install
 - 2.1 Server
 - 2.2 Client
- 3. Usage
 - 3.1 Available switches
 - 3.2 Examples
 - 3.2.1 Simple
 - 3.2.2 Using profiles
 - 3.2.3 Sending a command and then logging off
- 4. Available commands
 - 4.1 FreeSWITCH API
 - 4.2 Forward slash (/) commands
- 5. Configuration
 - 5.1 switch.conf.xml
 - 5.2 mod_console
 - 5.3 /etc/fs_cli.conf and ~/.fs_cli
 - 5.3.1 Format
 - 5.3.2 Configuration options
- 6. Wish List
- 7. See Also

0. About

The `fs_cli` program is a Command-Line Interface that allows a user to connect to a running FreeSWITCH™ instance. The `fs_cli` program can connect to the FreeSWITCH™ process on the local machine or on a remote system. (Network connectivity to the remote system is, of course, required.)

The `fs_cli` program uses FreeSWITCH™'s [Event Socket Library \(ESL\)](#) to tap into FreeSWITCH™'s event system to send commands issued by the user, and to collect the server responses to send to the display. The [Event Socket Library \(ESL\)](#), a C-based socket library, was developed for use with `fs_cli`, although a programmer could use this library for any C language program that needs to connect to the event socket. With the `-x` switch (see below) `fs_cli` can issue a command to the server, get a response, and then disconnect.

The `fs_cli` program can connect to FreeSWITCH™, whether it has been [started as a service](#) (`TODO`) or [on the console](#) (either background or foreground) (`TODO`), regardless of operating system.

1. Requirements

`fs_cli` **requires** `mod_event_socket` to be loaded in order to connect to the FreeSWITCH™ server.

✔ Normally, the easiest way to [check if a module is loaded](#) is to use `fs_cli`

```
$ fs_cli

freeswitch@tr2> module_exists mod_event_socket
true

# or
$ fs_cli -x 'module_exists mod_event_socket'
true
```

but in this case, for obvious reasons, check whether the `mod_event_socket` is enabled in `modules.conf.xml` (see [Configuring FreeSWITCH](#)).

The default `mod_event_socket` configuration binds to `::` (i.e., to listen to connections from **any** host), which will work on IPv4 or IPv6.

⚠ `::` means that `mod_event_socket` will listen to connections from **any** host (see vanilla `<conf_dir>/autoload_configs/event_socket.conf.xml` configuration file in the [SignalWire GitHub repository](#)). There are obvious security risks, so one would want to change this (e.g., to `localhost` only, `::1`), and perhaps also limit access via a firewall and/or [ACL](#), as well as never using the default password.

2. Install

2.1 Server

The FreeSWITCH™ server will build and install the `fs_cli` client by default.

2.2 Client

The client can also be built without needing to build the entire FreeSWITCH™ server.

To build:

```
make current
cd libs/esl
make
```

To run:

```
./fs_cli
```

3. Usage

3.1 Available switches

command syntax

<code>-?, -h --help</code>	Usage Information
<code>-H, --host=hostname</code>	Host to connect (default is 127.0.0.1)
<code>-P, --port=port</code>	Port to connect (default is "8021")
<code>-u, --user=user@domain</code>	user@domain
<code>-p, --password=password</code>	Password (default is "ClueCon")
<code>-i, --interrupt</code>	Allow Control-c to interrupt
<code>-x, --execute=command</code>	Execute Command and Exit
<code>-l, --loglevel=command</code>	Log Level (default is debug)
<code>-U, --log-uuid</code>	Include UUID in log output
<code>-S, --log-uuid-short</code>	Include shortened UUID in log output
<code>-q, --quiet</code>	Disable logging
<code>-r, --retry</code>	Retry connection on failure every two seconds until connected (or until 2 minutes has passed)
<code>-R, --reconnect</code>	Reconnect if disconnected
<code>-d, --debug=level</code>	Debug Level (0 - 7)
<code>-b, --batchmode</code>	Batch mode
<code>-t, --timeout</code>	Timeout for API commands (in milliseconds)
<code>-T, --connect-timeout</code>	Timeout for socket connection (in milliseconds)
<code>-n, --no-color</code>	Disable color

3.2 Examples

3.2.1 Simple

```
fs_cli
```

`fs_cli` which connects to local machine using default username, password, and debug level.

3.2.2 Using profiles

```
fs_cli my_profile
```

Launches `fs_cli` using profile named "my_profile" found in `.fs_cli_conf` file (see [section 5.3](#) below).

3.2.3 Sending a command and then logging off

```
fs_cli -x "sofia status profile internal"
```

Launches `fs_cli` and sends a command before logging off. The output of the above command looks like this:

```
$ fs_cli -x "sofia status profile internal"
=====
Name internal
Domain Name N/A
Auto-NAT false
DBName sofia_reg_internal
Pres Hosts 10.0.0.5,10.0.0.5
Dialplan XML
Context public
Challenge Realm auto_from
RTP-IP 10.0.0.5
SIP-IP 10.0.0.5
URL sip:mod_sofia@10.0.0.5:5060
BIND-URL sip:mod_sofia@10.0.0.5:5060;transport=udp,tcp
WS-BIND-URL sip:mod_sofia@10.0.0.5:5066;transport=ws
WSS-BIND-URL sips:mod_sofia@10.0.0.5:7443;transport=wss
HOLD-MUSIC local_stream://moh
OUTBOUND-PROXY N/A
CODECS IN OPUS,G722,PCMU,PCMA,VP8
CODECS OUT OPUS,G722,PCMU,PCMA,VP8
TEL-EVENT 101
DTMF-MODE rfc2833
CNG 13
SESSION-TO 0
MAX-DIALOG 0
NOMEDIA false
LATE-NEG true
PROXY-MEDIA false
ZRTP-PASSTHRU true
AGGRESSIVENAT false
CALLS-IN 0
FAILED-CALLS-IN 0
CALLS-OUT 0
FAILED-CALLS-OUT 0
REGISTRATIONS 0
```

4. Available commands

4.1 FreeSWITCH API

While connected, the user can issue any command in the [FreeSWITCH API](#) (which are all the commands exposed in the [enabled modules](#) and [mod_commands](#)).

See the [console](#) commands for example, exposed by [mod_console](#).

```
freeswitch@tr2> console
USAGE:
-----
console help
console loglevel [[0-7] | <loglevel_string>]
console uuid [on|off|toggle]
console json [on|off|toggle]
console colorize [on|off|toggle]
-----
```

4.2 Forward slash (/) commands

Additionally, there are several commands that can be issued using a forward slash (/) character.

"slash" command	Description	Examples	Notes
/quit		/quit	
/bye	These all result in disconnecting from the FreeSWITCH command line.	/bye	
/exit		/exit	
/event	Subscribe to FreeSWITCH events .	/event all	This command corresponds to the <code>event</code> command in <code>mod_event_socket</code> .
/noevents	Unsubscribe from all events (previously subscribed to using <code>/event</code>).	/noevents	This command corresponds to the <code>noevents</code> command in <code>mod_event_socket</code> .
/nixevent	Suppress the specified type of event. Useful when you want to allow <code>/event all</code> followed by <code>nixevent <some_event></code> to see all but one type of event.	/nixevent HEARTBEAT	This command corresponds to the <code>nixevent</code> command in <code>mod_event_socket</code> .
/log	Set log level of the FreeSWITCH daemon. 0 - CONSOLE 1 - ALERT 2 - CRIT 3 - ERR 4 - WARNING 5 - NOTICE 6 - INFO 7 - DEBUG	/log alert /log 1	This command corresponds to the <code>log</code> command in <code>mod_event_socket</code> . TODO So what is the point of console <code>loglevel <level></code> ? The default log level is stated when <code>fs_cli</code> is started (which is 7 for debug), but <code>console loglevel</code> will report NOTICE (i.e., 5). However it is, <code>/log</code> seems to take precedence over the <code>console</code> commands.
/nolog	Disable logging.	/nolog	This command corresponds to the <code>nolog</code> command in <code>mod_event_socket</code> .
/uuid	Filter logs for a single call (specified by its UUID).	/uuid 6936d2ad-bea3-40b3-9de3-34024404e8d4	
/filter	Specify what events to listen to by event header value. <code>/filter <EventHeader> <ValueToFilter></code>	/filter Event-Name HEARTBEAT /filter Unique-ID d29a070f-40ff-43d8-8b9d-d369b2389dfe	This command corresponds to the <code>filter</code> command in <code>mod_event_socket</code> .
/filter delete	Delete previously set event filters. <code>/filter delete <EventHeader> <ValueToFilter></code> If <code><ValueToFilter></code> is omitted, the command will delete all filters previously set for the specified event header.	/filter delete Event-Name HEARTBEAT /filter delete Unique-ID d29a070f-40ff-43d8-8b9d-d369b2389dfe /filter delete Unique-ID	This command corresponds to the <code>filter delete</code> command in <code>mod_event_socket</code> .
/logfilter <string>	Filter the logs by the given <code><string></code> . (The log levels are set by <code>/log</code> , see above.) TODO What is the exact syntax? Does it accept regexes as well? I know that it takes anything that follows <code>/logfilter</code> literally, because adding double quotes to a string with space will actually look for double quotes in the log entries.	/logfilter Codec Activated	
/logfilter	Disable all logfilters.	/logfilter	
/help	List available <code>fs_cli</code> commands.	/help	
/debug <debug_level>	There are 8 debug levels (from 0 to 7) and the higher the number the more verbose the logs will become. <code>/debug <debug_level></code> is equivalent to <pre>/log 7 /debug <debug_level></pre>	/debug 3	
/debug	Same as <code>/debug 0</code>	/debug	

For command-line editing, see "Command-Line Editing" section of [mod_console](#). The details are the same as of [SVN r13964](#).

5. Configuration

TODO The statements in this section need confirmation. For example, tried to de-colorize the console logs by setting the relevant section to false in `console.conf.xml` (see [mod_console](#)) and in `switch.conf.xml`, while having no `fs_cli.conf` anywhere in the system after a [vanilla FreeSWITCH](#) install.

5.1 `switch.conf.xml`

`switch.conf.xml` contains the core FreeSWITCH configuration, see more there.

5.2 `mod_console`

`mod_console` and its configuration file, `console.conf.xml` also affect `fs_cli`, and some of the options are redundant with `switch.conf.xml`.

5.3 `/etc/fs_cli.conf` and `~/.fs_cli`

TODO This section definitely needs confirmation; the [vanilla FreeSWITCH](#) install doesn't have any `fs_cli.conf` (only `~/.fs_cli_history`), and the only intact `fs_cli.conf` file I could find is [this](#).

Use `fs_cli.conf` to override existing configuration (`/etc/fs_cli.conf` for system-wide settings and `~/.fs_cli.conf` for user-specific settings).

5.3.1 Format

The config file uses a simple INI-style layout and allows for multiple profiles. This allows one to access many FreeSWITCH™ systems from a single workstation.

```
[default]
; Put me in /etc/fs_cli.conf or ~/.fs_cli.conf
;override any default options here
loglevel => 6
log-uuid => false
quiet => false
key_f1 => help
key_f2 => status
key_f3 => show channels
key_f4 => show calls
key_f5 => sofia status
key_f6 => reloadxml
key_f7 => /log console
key_f8 => /log debug
key_f9 => sofia status profile internal
key_f10 => sofia global siptrace on
key_f11 => sofia global siptrace off
key_f12 => version

[profile1]
host      => 192.168.1.10
port      => 8021
password  => secret_password
debug     => 7

[profile2]
host      => 192.168.1.11
port      => 8021
password  => someother_password
loglevel  => info

[my_profile]
```

5.3.2 Configuration options

Option	Description	Example
host => <hostname>	Host to connect (default is 127.0.0.1)	host => 127.0.0.1

port => <port>	Port to connect (default is 8021)	port => 8021
user => <username>	user@domain	
password => <password>	Password (default is "ClueCon")	password => ClueCon
interrupt => true false	Allow Control-c to interrupt	
execute => <command>	Execute command and exit	
loglevel => console alert crit err warning notice info debug	Set log Level (default is debug)	
log-uuid => true false	Include UUID in log output	
log-uuid-short => true false	Include shortened UUID in log output	
quiet => true false	Disable logging	
retry => true false	Retry connection on failure every two seconds until connected (or until 2 minutes has passed)	
reconnect => true false	Reconnect if disconnected	
debug => <0 .. 7>	Debug Level (0 - 7)	
batchmode => true false	Batch mode TODO What does this mean?	
timeout => <milliseconds>	Timeout for API commands (in milliseconds)	
connect-timeout => <milliseconds>	Timeout for socket connection (in milliseconds)	
no-color => true false	Disable color	
key_f<n>	Set F1 - F12 keys for a certain functionality. <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">i Default key-bindings F1 = help F2 = status F3 = show channels F4 = show calls F5 = sofia status F6 = reloadxml F7 = console loglevel 0 F8 = console loglevel 7 F9 = sofia status profile internal F10 = sofia profile internal siptrace on F11 = sofia profile internal siptrace off F12 = version</div>	See example at 5.3.1 Format above.

6. Wish List

- Option to see all output to include FS console output.
- Option to see all output from all fs_cli instances connected to the FS box, plus the console.
- Option to connect to more than one FS box.

7. See Also

- [mod_event_socket](#)
- [Event Socket Library](#)
- [Event list](#)
- [Fs_gui](#)