Sofia Configuration Files

About

Sofia is a FreeSWITCH™ module (mod_sofia) that provides SIP connectivity to and from FreeSWITCH in the form of a User Agent.

A "User Agent" ("UA") is an application used for handling a certain network protocol; the network protocol in Sofia's case is SIP. Sofia is the general name of any User Agent in FreeSWITCH using the SIP network protocol.

For example, Sofia receives calls sent to FreeSWITCH from other SIP User Agents (UAs), sends calls to other UAs, acts as a client to register FreeSWITCH with other UAs, lets clients register with FreeSWITCH, and connects calls (i.e., to local extensions).

To add a SIP Provider (Sofia User Agent) to your FreeSWITCH, please see the Interoperability Examples and add the SIP Provider information in an .xml file stored under conf/sip_profiles/

Sofia allows for multiple User Agents

A "User Agent" ("UA") is an application used for running a certain network protocol, and a Sofia UA is the same thing but the protocol in that case is SIP.

When FreeSWITCH starts, it reads the conf/autoload_configs/sofia.conf.xml file. That file contains a "X-PRE-PROCESS" directive which instructs FreeSWITCH to subsequently load and merge any conf/sip_profiles/*.xml files. Each *.xml file so loaded and merged should contain a complete description of one or more SIP Profiles. Each SIP Profile so loaded is part of a "User Agent" or "UA"; in FreeSWITCH terms, UA = User Agent = Sofia Profile = SIP Profile.

Note that the individual UAs so loaded are all merged together by FreeSWITCH and must not interfere with each other: In particular, **each UA must have its own unique port on which it accepts connections** (the default port for SIP is 5060).

Multiple User Agents (Profiles) and the Dialplan

Why might you want to create multiple User Agents? Here's an example.

In my office, I use a firewall. This means that calls I make to locations outside the firewall must use a STUN server to transverse the NAT in the firewall, while calls within the office don't need to use a STUN server. In order to accommodate these requirements, I've created two different UAs. One of them uses a STUN server and for that matter also connects up to the PSTN through a service provider. The other UA is purely for local SIP calls.

Now I've got two UAs defined by my profiles, each of which can handle a call. When dialing a SIP address or telephone number, which UA is used? That determination is made in the dialplan. One syntax for making a call via Sofia in the dialplan is

```
sofia/profile_name/destination
```

So, the task becomes rather straightforward. Dialplans use pattern matching and other tricks to determine how to handle a call. My dialplan examines what I've dialed and then determines what profile to use with that call. If I dial a telephone number, the dialplan selects the UA that connects up to the PSTN. If I dial a SIP address outside the firewall, the dialplan selects that same UA because it uses the STUN server. But if I dial a SIP address that's inside the firewall, the dialplan selects the "local" UA.

To understand how to write dialplans, use pattern matching, etc., see Dialplan

The Relationship Between SIP Profiles and Domains

The following content was written in a mailing list thread by Anthony Minessale in response to questions about how SIP profiles relate to domain names in FreeSWITCH.

The best thing to do is take a look at these things from a step back.

The domains inside the XML registry are completely different from the domains on the internet and again completely different from domains in sip packets. The profiles are again entirely different from any of the above. Its up to you to align them if you so choose.

The default configuration distributed with FreeSWITCH sets up the scenario most likely to load on any machine and work out of the box. That is the primary goal of that configuration, so, It sets the domain in both the directory, the global default domain variable and the name of the internal profile to be identical to the IP addr on the box that can reach the internet. Then it sets the sip to force everything to that value. When you want to detach from this behavior, you are probably on a venture to do some kind of multi-home setup.

Aliases in the <aliases> tag are a list of keys you want to use to use that lead to the current profile your are configuring. Think of it as the /etc/hosts file in Unix, only for profiles. When you define aliases to match all of the possible domains hosted on a particular profile, then when you try to take a user@host.com notation and decide which profile it came from, you can use the aliases to find it providing you have added <alias name="host.com"/> to that profile.

The <domains> tag is an indicator telling the profile to open the XML registry in FreeSWITCH and run through any domains defined therein. The 2 key attributes are:

```
alias: [true/false]  (automatically create an alias for this domain as mentioned above)
parse: [true/false] (scan the domain for gateway entries and include them into this profile)
name: [<string>] (either the name of a specific domain or 'all' to denote parsing every domain in the directory)
```

As you showed in your question the default config has

```
<domain name="all" alias="true" parse="false"/>
```

If you apply what you have learned above, it will scan for every domain (there is only one by default) and add an alias for it and not parse it for gateways. The default directory uses global config vars to set the domain to match the local IP addr on the box. So now you will have a domain in your config that is your IP addr, and the internal profile will attach to it and add an alias so that value expands to match it.

This is explained in a comment at the top of directory/default.xml:

```
    FreeSWITCH works off the concept of users and domains just like email.
    You have users that are in domains for example 1000@domain.com.

    When freeswitch gets a register packet it looks for the user in the directory
    based on the from or to domain in the packet depending on how your sofia profile
    is configured.  Out of the box the default domain will be the IP address of the
    machine running FreeSWITCH.  This IP can be found by typing "sofia status" at the
    CLI.  You will register your phones to the IP and not the hostname by default.
    If you wish to register using the domain please open vars.xml in the root conf
    directory and set the default domain to the hostname you desire.  Then you would
    use the domain name in the client instead of the IP address to register
    with FreeSWITCH.
```

So having more than one profile with the default of

```
<domain name="all" alias="true" parse="false"/>
```

is going to end up aliasing the same domains into all profiles who call it and cause an overwrite in the lookup table and probably an error in your logs somewhere. If you had parse="true" on all of them, they would all try and register to the gateways in all of your domains.

If you look at the stock config, external.xml is a good example of a secondary profile, it has

```
<domain name="all" alias="false" parse="true"/>
```

so no aliases, and yes parse ... the exact opposite of the internal so that all the gateways would register from external and internal would bind to the local IP addr.

So, you probably want to use separate <domain name="example.com"/> per domain per profile you want to bind it to in more complicated setups.

Structure of a Profile

Each profile may contain several different subsections. At the present time there's no XSD or DTD for sofia.conf.xml — and any volunteer who can create one would be very welcome indeed.

```
<!ELEMENT configuration (global_settings?, profiles)>
<!ELEMENT global_settings (param+)>
<!ELEMENT profiles (profile+)>
<!ELEMENT profile (aliases, gateways, domains, settings)>
<!ELEMENT aliases (alias*)>
<!ELEMENT gateways (gateway*)>
<!ELEMENT gateway (param+, variables?)>
<!ELEMENT variables (variable+)>
<!ELEMENT domains (domain*)>
<!ELEMENT settings (param+)>
<!ELEMENT alias EMPTY>
<!ELEMENT param EMPTY>
<!ELEMENT variable EMPTY>
<!ELEMENT domain EMPTY>
<!ATTLIST configuration name CDATA #REQUIRED description CDATA #REQUIRED>
<!ATTLIST profile name CDATA #REQUIRED domain CDATA #IMPLIED>
<!ATTLIST gateway name CDATA #REQUIRED>
<!ATTLIST alias name CDATA #REQUIRED>
<!ATTLIST param name CDATA #REQUIRED value CDATA #REQUIRED>
<!ATTLIST variable name CDATA #REQUIRED data CDATA #REQUIRED direction CDATA #IMPLIED>
<!ATTLIST domain name CDATA #REQUIRED alias (true | false) #IMPLIED parse (true | false) #IMPLIED>
```

Gateway

Each profile can have several gateways:

```
<gateways>
 <gateway>
   elements...
 </gateway>
 <gateway>
   elements...
 </gateway>
</gateways>
```

A gateway has an attribute "name" by which it can be referred. A gateway describes how to use a different UA to reach destinations. For example, the gateway may provide access to the PSTN, or to a private SIP network. The reason for defining a gateway, presumably, is because the gateway requires certain information before it will accept a call from the FreeSWITCH User Agent.

Variables can be defined on a gateway. Inbound variables are set on the channel of a call received from a gateway, outbound variables are set on the channel of a call sent to a gateway.

An example gateway configuration would be:

```
 <gateway name="gateway012">
   <param name="realm" value="sip.voipcarrier.com" />
   <param name="username" value="WBrandes" />
   <param name="password" value="myvoiceismypassword" />
   <param name="register" value="true" />
   <param name="caller-id-in-from" value="true" />
   <param name="ping" value="5" />
   <param name="ping-max" value="3" />
   <param name="retry-seconds" value="5" />
   <param name="expire-seconds" value="60" />
   <variables>
     <variable name="verbose_sdp" value="true"/>
     <variable name="absolute_codec_string" value="PCMU,PCMA" direction="outbound"/>
     <variable name="customer_id" value="3532" direction="inbound"/>
   </variables>
 </gateway>
```

To reach a particular gateway from the dial plan, use

```
sofia/gateway/<gateway_name>/<dialstring>
```

FreeSWITCH can also subscribe to receive notification of events from the gateway. For more information see Presence - Use FreeSWITCH as a Client

Parameters

The following is a list of param elements that are children of a gateway element:

```
<!-- /// account username *required* /// -->
<param name="username" value="foo"/>
```

Note: The username param for the gateway is not to be confused with the username param in the Profile settings config!

```
<!--/// auth realm: *optional* same as gateway name, if blank ///-->
<!-- realm -->
<param name="realm" value="sip.example.com[:port]"/>

<!--/// account password *required* ///-->
<param name="password" value="a password"/>

<!--/// username to use in from: *optional* same as  username, if blank ///-->
<param name="from-user" value="fooman"/>

<!--/// domain to use in from: *optional* same as  realm, if blank; can also be set to "auto-aleg-full" or "auto-
aleg-domain" ///-->
<param name="from-domain" value="asterlink.com"/>

<!--/// replace the INVITE from user with the channel's caller-id ///-->
<!--/// this is false by default because outbound calls generally need username in the INVITE ///-->
<!--Use the callerid of an inbound call in the from field on outbound calls via this gateway -->
<param name="caller-id-in-from" value="false"/>

<!--/// extension for inbound calls: *optional* Same as username, if blank. To use what's in ${sip_to_user}, set
it to the value "auto_to_user" ///-->
<param name="extension" value=""/>
```

Note: *extension* parameter influence the contents of channel variable *Caller-Destination-Number* and *destination_number*. If it is blank, *Caller-Destination-Number* will always be set to gateway's username. If it has a value, *Caller-Destination-Number* will always be set to this value. If it has value *auto_to_user*, *Caller-Destination-Number* will be populated with value *${sip_to_user}* which means the real dialled number in case of an inbound call.

```
<!--/// proxy host: *optional* same as realm, if blank ///-->
<param name="proxy" value="proxy.example.com"/>

<!--/// expire in seconds: *optional* 3600, if blank ///-->
<param name="expire-seconds" value="3600"/>
```

```
<!-- suppress CNG on this profile or per call with the 'suppress_cng' variable -->
<param name="suppress-cng" value="true"/>

<!--/// false: do not register. true: register (default) ///-->
<param name="register" value="false"/>

<!-- which transport to use for register -->
<param name="register-transport" value="udp"/>

<!--extra sip params to send in the contact-->
<param name="contact-params" value="tport=tcp"/>

<!-- check gateway availability: *optional* -->
<param name="ping" value="15"/>

<!-- *optional* -->
<param name="ping-max" value="10"/>

<!--  *optional* -->
<param name="ping-min" value="1"/>
```

ping-min means "how many successful pings we must have before declaring a gateway up".

The interval between **ping-min** and **ping-max** is the "safe area" where a gateway is marked as UP. So if we have, for example, min 3 and max 6, if the gateway is up and we move counter between 3,4,5,6 the gateway will be up.

If from 6 we loose 4 (so counter == 2) pings in a row, the gateway will be declared down.

Please note that on sofia startup the gateway is always started as UP, so it will be up even if ping-min is > 1 . the "right" way starts when the gateway goes down.

```
<!--  *optional* - default is false-->
<param name="extension-in-contact" value="true"/>

<!--  *optional* -->
<param name="cid-type" value="rpid"/>
```

Param "register," is used when this profile acts as a client to another UA. By registering, FreeSWITCH informs the other UA of its whereabouts. This is generally used when FreeSWITCH wants the other UA to send FreeSWITCH calls, and the other UA expects this sort of registration. If FreeSWITCH uses the other UA only as a gateway (e.g., to the PSTN), then registration is not generally required.

Param "distinct-to" is used when you want FS to register using a distict AOR for header To. It requires proper setting of related parameters. For example if you want the REGISTER to go with:

```
From: <sip:someuser@somedomain.com>
To: <sip:anotheruser@anotherdomain.com>
```

Then set the parameters as this:

```
<param name="distinct-to" value="true"/>
<param name="auth-username" value="someuser"/>
<param name="from-user" value="someuser"/>
<param name="from-domain" value="somedomain.com"/>
<param name="password" value="somepassword"/>
<param name="username" value="anotheruser"/>
<param name="realm" value="anotherdomain.com"/>
```

The latter param, "ping" is used to check gateway availability. By setting this option, FreeSWITCH will send SIP OPTIONS packets to gateway. If gateway responds with 200 or 404, gateway is pronounced up, otherwise down. [N.B. It appears that other error messages can be returned and still result in the gateway being marked as 'up'?] If any call is routed to gateway with state down, FreeSWITCH will generate NETWORK_OUT_OF_ORDER hangup cause. Ping frequency is defined in seconds (value attribute) and has a minimum value of 5 seconds.

Param "extension-in-contact" is used to force what the contact info will be in the registration. If you are having a problem with the default registering as gw+gateway_name@ip you can set this to true to use extension@ip. If extension is blank, it will use username@ip.

if you need to insert the FROM digits to the Contact URI User Part when sending call to gateway
BEFORE

```
From: "8885551212" <sip:88855512120@8.8.8.8>
Contact: <sip:gw+mygateway@7.7.7.7:7080>
```

try adding these to gateway params

```
<param name="extension" value="8885551212"/>
<param name="extension-in-contact" value="true"/>
```

AFTER

```
From: "8885551212" <sip:88855512120@8.8.8.8>
Contact: <sip:8885551212@7.7.7.7:7080>
```

Variables

In addition to the parameters you can optionally set variables to set on either incoming or outgoing calls through this gateway. You set a direction, which sets it on both incoming and outgoing calls if omitted.

```
<gateway>
  ...params...
  <variables>
    <variable name="inbound_var_name"  value="this is inbound"  direction="inbound"/>
    <variable name="outbound_var_name" value="this is outbound" direction="outbound"/>
    <variable name="both_var_name"     value="this on any direction"/>
  </variables>
</gateway>
```

These channel variables will be set on all calls going through this gateway in the specified direction.

However, see below for a special syntax to set profile variables rather than channel variables.

```
<variable name="p:caller_id_name" value="Gateway"/>
```

Settings

Settings include other, more general information about the profile, including whether or not STUN is in use. Each profile has its own settings element. Not only is this convenient — it's possible to set up one profile to use STUN and another, with a different gateway or working behind the firewall, not needing STUN — but it's also crucial. That's because each profile defines a SIP User Agent, and each UA must have its own unique "sip-port." By convention, 5060 is the default port, but it's possible to make calls to, e.g., "foo@sip.example.com:5070", and therefore you can define any port you please for each individual profile.

The conf directory contains a complete sample sofia.conf.xml file, along with comments. See Git examples: Internal, External

Basic settings

alias

```
<param name="alias" value="sip:10.0.1.251:5555"/>
```

This seems to make the SIP profile bind to this IP & port as well as your SIP / RTP IPs and ports.

Anthony had this to say about aliases in a ML thread:

```
Aliases in the <aliases> tag are a list of keys you want to use to use that lead
to the current profile your are configuring.  Think of it as the /etc/hosts file
in unix only for profiles.   When you define aliases to match all of the possible
domains hosted on a particular profile, then when you try to take a user@host.com
notation and decide which profile it came from, you can use the aliases to find
it providing you have added <alias name="host.com"/> to that profile.
```

shutdown-on-fail

```
<param name="shutdown-on-fail" value="true"/>
```

If set to true and the profile fails to load, FreeSWITCH will shut down. This is useful if you are running something like Pacemaker and OpenAIS which manage a pair of FreeSWITCH nodes and automatically monitor, start, stop, restart, and standby-on-fail the nodes. It will ensure that the specific node is not able to be used in a "partially up" situation.

user-agent-string

This sets the User-Agent header in all SIP messages sent by your server. By default this could be something like "FreeSWITCH-mod_sofia/1.0.trunk-12805". If you didn't want to advertise detailed version information you could simply set this to "FreeSWITCH" or even "Asterisk PBX" as a joke.

Take care when setting this value as certain characters such as '@' could cause other SIP proxies could reject your messages as invalid.

```
<param name="user-agent-string" value="FreeSWITCH Rocks!"/>
```

log-level

```
<param name="log-level" value="0"/>
```

debug

```
<param name="debug" value="0"/>
```

sip-trace

```
<param name="sip-trace" value="no"/>
```

context

Dialplan context in which to dump calls that come in to this profile's ip:port

```
<param name="context" value="public"/>
```

sip-port

Port to bind to for SIP traffic:

```
<param name="sip-port" value="$${internal_sip_port}"/>
```

sip-ip

IP address to bind to for SIP traffic. **DO NOT USE HOSTNAMES, ONLY IP ADDRESSES**

```
<param name="sip-ip" value="$${local_ip_v4_or_v6}"/>
```

rtp-ip

IP address to bind to for RTP traffic. **DO NOT USE HOSTNAMES, ONLY IP ADDRESSES**

```
 <param name="rtp-ip" value="$${local_ip_v4_or_v6}"/>
```

- Multiple rtp-ip support: if more rtp-ip parameters are added, they will be used in round-robin as new calls progress.
- IPv6 addresses **are not supported** under Windows at the time of writing. See FS-4445

ext-rtp-ip

This is the IP behind which FreeSWITCH is seen from the Internet, so if FreeSWITCH is behind NAT, this is basically the public IP that should be used for RTP.
Possible values are:
Any variable from vars.xml, e.g. $${external_rtp_ip}:

```
<param name="ext-rtp-ip" value="$${external_rtp_ip}"/>
```

"specific IP address"

```
<param name="ext-rtp-ip" value="1.2.3.4"/>
```

"when used for LAN and WAN to avoid errors in the SIP CONTACT sent to LAN devices, use"

```
<param name="ext-rtp-ip" value="autonat:1.2.3.4"/>
```

"auto": the guessed IP will be used (guessed by looking in the IP routing table which interface is the default route)

```
<param name="ext-rtp-ip" value="auto"/>
```

"auto-nat": FreeSWITCH will use uPNP or NAT-PMP to discover the public IP address it should use

```
<param name="ext-rtp-ip" value="auto-nat"/>
```

"stun:DNS name or IP address": FreeSWITCH will use the STUN server of your choice to discover the public IP address

```
<param name="ext-rtp-ip" value="stun:stun.freeswitch.org"/>
```

"host:DNS name": FreeSWITCH will resolve the DNS name as the public IP address, so you can use a dynamic DNS host

```
<param name="ext-rtp-ip" value="host:mypublicIP.dyndns.org"/>
```

ATTENTION: AS OF 2012Q4, 'ext–' prefixed params cited above when populated with to-be-resolved DNS strings -- e.g. name="ext–sip–ip" value="stun:stun.freeswitch.org" or name="extrtp–ip" value="host:mypublicIP.dyndns.org" -- are resolved to IP addresses once only at FS load time and const thereafter. FS is blind to (unaware of) any subsequent changes in your environment's IP address. Thus, these ext– vars may become functionally incompatible with the environment's current IP addresses with *unspecified results* in call flow at the network layer. FS restart is ***required*** for FS to capture the now-current, working IP address(es).

ext-sip-ip

This is the IP behind which FreeSWITCH is seen from the Internet, so if FreeSWITCH is behind NAT, this is basically the public IP that should be used for SIP.
Possibles values are the same as those for ext-rtp-ip, and it is usually set to the same value.

```
<param name="ext-sip-ip" value="$${external_sip_ip}"/>
```

tcp-keepalive

Set this to interval (in milliseconds) to send keep alive packets to user agents (UAs) registered via TCP; do not set to disable.

tcp-pingpong

tcp-ping2pong

dialplan

The dialplan parameter is very powerful.

In the simplest configuration, it will use the XML dialplan. This means that it will read data from mod_xml_curl XML dialplans (e.g., callback to your webserver), or failing that, from the XML files specified in freeswitch.xml dialplan section. (e.g. default_context.xml)

```
<param name="dialplan" value="XML"/>
```

You can also add enum lookups into the picture (since mod_enum provides dialplan functionality), so enum lookups override the XML dialplan

```
<param name="dialplan" value="enum,XML"/>
```

Or reverse the order to enum is only consulted if XML lookup fails

```
<param name="dialplan" value="XML,enum"/>
```

It is also possible to specify a specific enum root

```
<param name="dialplan" value="enum:foo.com,XML"/>
```

Or use XML on a custom file

```
<param name="dialplan" value="XML:/tmp/foo.xml,XML,enum"/>
```

Where it will first check the specific XML file, then hit normal XML which also do a mod_xml_curl lookup assuming you have that configured and working.

Media related options

See also: Proxy Media

resume-media-on-hold

When calls are in no media this will bring them back to media when you press the hold button. To return the calls to bypass-media after the call is unheld, enable bypass-media-after-hold.

```
<param name="media-option" value="resume-media-on-hold"/>
```

bypass-media-after-att-xfer

This will allow a call after an attended transfer go back to bypass media after an attended transfer.

```
 <param name="media-option" value="bypass-media-after-att-xfer"/>
```

bypass-media-after-hold

This will allow a call to go back to bypass media after a hold. This option can be enabled only if resume-media-on-hold is set. Available from git rev 8fa385b.

```
 <param name="media-option" value="bypass-media-after-hold"/>
```

inbound-bypass-media

Uncomment to set all inbound calls to no media mode. It means that the FreeSWITCH server only keeps the SIP messages state, but have the RTP steam go directly from end-point to end-point

```
<param name="inbound-bypass-media" value="true"/>
```

inbound-proxy-media

Uncomment to set all inbound calls to proxy media mode. This means the FreeSWITCH keeps both the SIP and RTP traffic on the server but does not interact with the RTP stream.

```
<param name="inbound-proxy-media" value="true"/>
```

disable-rtp-auto-adjust

```
<param name="disable-rtp-auto-adjust" value="true"/>
```

ignore-183nosdp

```
<param name="ignore-183nosdp" value="true"/>
```

enable-soa

```
<param name="enable-soa" value="true"/>
```

Set the value to "false" to diable SIP SOA from sofia to tell sofia not to touch the exchange of SDP

t38-passthru

```
<param name="t38-passthru" value="true"/>
```

The following options are available

- 'true' enables t38 passthru
- 'false' disables t38 passthru
- 'once' enables t38 passthru, but sends t.38 re-invite only once (available since commit 08b25a8 from Nov. 9, 2011)

Codecs related options

Also see:

- Codec Negotiation
- Supported Codecs

inbound-codec-prefs

This parameter allows to change the allowed inbound codecs per profile.

```
 <param name="inbound-codec-prefs" value="$${global_codec_prefs}"/>
```

outbound-codec-prefs

This parameter allows to change the outbound codecs per profile.

```
 <param name="outbound-codec-prefs" value="$${outbound_codec_prefs}"/>
```

codec-prefs

This parameter allows to change both inbound-codec-prefs and outbound-codec-prefs at the same time.

```
 <param name="codec-prefs" value="$${global_codec_prefs}"/>
```

inbound-codec-negotiation

set to 'greedy' if you want your codec list to take precedence

```
 <param name="inbound-codec-negotiation" value="generous"/>
```

if 'greedy' doesn't work for you, try 'scrooge' which has been known to fix misreported ptime issues with DID providers such as CallCentric.

A rule of thumb is:

- 'generous' permits the remote codec list have precedence and 'win' the codec negotiation and selection process
- 'greedy' forces a win by the local FreeSWITCH preference list
- 'scrooge' takes 'greedy' a step further, so that the FreeSWITCH wins even when the far side lies about capabilities during the negotiation process

sip_codec_negotiation is a channel variable version of this setting

inbound-late-negotiation

Uncomment to let calls hit the dialplan *before* you decide if the codec is OK.

```
 <param name="inbound-late-negotiation" value="true"/>
```

bitpacking

This setting is for AAL2 bitpacking on G.726.

```
<param name="bitpacking" value="aal2"/>
```

disable-transcoding

Uncomment if you want to force the outbound leg of a bridge to only offer the codec that the originator is using

```
<param name="disable-transcoding" value="true"/>
```

renegotiate-codec-on-reinvite

```
<param name="renegotiate-codec-on-reinvite" value="true"/>
```

STUN

If you need to use a STUN server, here are common working examples:

ext-rtp-ip

```
<param name="ext-rtp-ip" value="stun:stun.fwdnet.net"/>
```

stun.fwdnet.net is a publicly-accessible STUN server.

ext-sip-ip

```
<param name="ext-sip-ip" value="world_reachable.real.numeric.ip"/>
```

stun-enabled

Simple traversal of UDP over NATs (STUN), is used to help resolve the problems associated with SIP clients, behind NAT, using private IP address space in their messaging. Use stun when specified (default is true).

```
<param name="stun-enabled" value="true"/>
```

stun-auto-disable

Set to true to have the profile determine stun is not useful and turn it off globally

```
<param name="stun-auto-disable" value="true"/>
```

NATing

apply-nat-acl

When receiving a REGISTER or INVITE, enable NAT mode automatically if IP address in Contact header matches an entry defined in the RFC 1918 acces s list. "acl" is a misnomer in this case because access will not be denied if the user's contact IP doesn't match.

```
<param name="apply-nat-acl" value="rfc1918"/>
```

aggressive-nat-detection

This will enable NAT mode if the network IP/port from which the request was received differs from the IP/Port combination in the SIP Via: header, or if the Via: header contains the received parameter (regardless of what it contains.) **Note 2009-04-05:** Someone please clarify when this would be useful. It seems to me if someone needed this feature, chances are that things are so broken that they would need to use NDLB-force-rport

```
<param name="aggressive-nat-detection" value="true"/>
```

VAD and CNG

VAD stands for Voice Activity Detector. FreeSWITCH is capable of detecting speech and can stop transmitting RTP packets when no voice is detected.

vad

```
<param name="vad" value="in"/>
<param name="vad" value="out"/>
<param name="vad" value="both"/>
```

suppress-cng

Suppress Comfort Noise Generator (CNG) on this profile or per call with the 'suppress_cng' variable

```
<param name="suppress-cng" value="true"/>
```

NDLB (A.K.A. No device left behind)

NDLB-force-rport

This will force FreeSWITCH to send SIP responses to the network port from which they were received. **Use at your own risk!** For more information see NAT Traversal.

```
<param name="NDLB-force-rport" value="true|safe"/>
```

- safe = param that does force-rport behavior only on endpoints we know are safe to do so on. This is a dirty hack to try to work with certain endpoints behind sonicwall which does not use the same port when it does nat, when the devices do not support rport, while not breaking devices that acutally use different ports that force-rport will break

NDLB-broken-auth-hash

Used for when phones respond to a challenged ACK with method INVITE in the hash

```
<param name="NDLB-broken-auth-hash" value="true"/>
```

NDLB-received-in-nat-reg-contact

add a ;received="<ip>:<port>" to the contact when replying to register for nat handling

```
<param name="NDLB-received-in-nat-reg-contact" value="true"/>
```

NDLB-sendrecv-in-session

By default, "a=sendrecv" is only included in the media portion of the SDP. While this is RFC-compliant, it may break functionality for some SIP devices. To also include "a=sendrecv" in the session portion of the SDP, set this parameter to true.

```
<param name="NDLB-sendrecv-in-session" value="true"/>
```

NDLB-allow-bad-iananame

- Introduced in rev. 15401, this was enabled by default prior to new param.

Will allow codecs to match respective name even if the given string is not correct.

i.e., Linksys and Sipura phones will pass G.729a by default instead of G.729 as codec string therefore not matching.

If you wish to allow bad IANA names to match respective codec string, add the following param to your SIP profile.

```
 <param name="NDLB-allow-bad-iananame" value="true"/>
```

Refer to RFC 3551, RFC 3555 and the IANA list(s) for SDP

Call ID

inbound-use-callid-as-uuid

On inbound calls make the uuid of the session equal to the SIP call id of that call.

```
<param name="inbound-use-callid-as-uuid" value="true"/>
```

outbound-use-uuid-as-callid

On outbound calls set the callid to match the uuid of the session

```
<param name="outbound-use-uuid-as-callid" value="true"/>
```

This goes in the "..sip_profiles/external.xml" file.

TLS

Please make sure to read SIP TLS before enabling certain features below as they may not behave as expected.

tls

TLS: disabled by default, set to "true" to enable

```
   <param name="tls" value="$${internal_ssl_enable}"/>
```

tls-only

disabled by default, when enabled prevents sofia from listening on the unencrypted port for this connection. This can stop many generic brute force scripts and if all your clients connect over TLS then can help decrease the exposure of your FreeSWITCH server to the world.

```
<param name="tls-only" value="false"/>
```

tls-bind-params

additional bind parameters for TLS

```
<param name="tls-bind-params" value="transport=tls"/>
```

tls-sip-port

Port to listen on for TLS requests. (5061 will be used if unspecified)

```
<param name="tls-sip-port" value="$${internal_tls_port}"/>
```

tls-cert-dir

Location of the agent.pem and cafile.pem ssl certificates (needed for TLS server)

```
<param name="tls-cert-dir" value="$${internal_ssl_dir}"/>
```

tls-version

TLS version ("sslv2", "sslv3", "sslv23", "tlsv1", "tlsv1.1", "tlsv1.2"). NOTE: Phones may not work with TLSv1

```
<param name="tls-version" value="$${sip_tls_version}"/>
```

When not set defaults to: "tlsv1,tlsv1.1,tlsv1.2"

tls-passphrase

If your agent.pem is protected by a passphrase stick the passphrase here to enable FreeSWITCH to decrypt the key.

```
<param name="tls-passphrase" value=""/>
```

tls-verify-date

If the client/server certificate should have the date on it validated to ensure it is not expired and is currently active.

```
<param name="tls-verify-date" value="true"/>
```

tls-verify-policy

This controls what, if any security checks are done against server/client certificates. Verification is generally checking certificates are valid against the cafile.pem. Set to 'in' to only verify incoming connections, 'out' to only verify outgoing connections, 'all' to verify all connections, also 'subjects_in', 'subjects_out' and 'subjects_all' for subject validation (subject validation for outgoing connections is against the hostname/ip connecting to). Multiple policies can be split with a '|' pipe, for example 'subjects_in|subjects_out'. Defaults to none.

```
<param name="tls-verify-policy" value="none"/>
```

tls-verify-depth

When certificate validation is enabled (tls-verify-policy) how deep should we try to verify a certificate up the chain again the cafile.pem file. By default only depth of 2.

```
<param name="tls-verify-depth" value="2"/>
```

tls-verify-in-subjects

If subject validation is enabled for incoming connections (tls-verify-policy set to 'subjects_in' or 'subjects_all') this is the list of subjects that are allowed (delimit with a '|' pipe), note this only effects incoming connections for outgoing connections subjects are always checked against hostnames/ips.

```
<param name="tls-verify-in-subjects" value=""/>
```

DTMF

rfc2833-pt

```
<param name="rfc2833-pt" value="101"/>
```

dtmf-duration

```
<param name="dtmf-duration" value="100"/>
```

dtmf-type

Set the parameter in the SIP profile:

```
<param name="dtmf-type" value="info"/>
```

or

```
<param name="dtmf-type" value="rfc2833"/>
```

or

```
<param name="dtmf-type" value="none"/>
```

OR set the variable in the SIP gateway or user profile (NOT in the channel, it must be before CS_INIT):

```
<variables>
  <variable direction="inbound|outbound|both" name="dtmf_type" value="info">
</variables>
```

Note the "_" instead of "-" in profile param (this is var set in dialplan). (24.10.2010: "both" don't seem to me work in my tests, "outbound" does)

Note: for inband DTMF, Misc. Dialplan Tools start_dtmf must be used in the dialplan.
Also, to change the outgoing routing from info or rfc2833 to inband, use Misc._Dialplan_Tools_start_dtmf_generate

RFC 2833

pass-rfc2833

Default: false

If true, it passes RFC 2833 DTMF's from one side of a bridge to the other, untouched. Otherwise, it decodes and re-encodes them before passing them on.

```
<param name="pass-rfc2833" value="true"/>
```

liberal-dtmf

Default: false
For DTMF negotiation, use this parameter to just always offer 2833 and accept both 2833 and INFO. Use of this parameter is not recommended since its purpose is to try to cope with buggy SIP implementations.

```
<param name="liberal-dtmf" value="true"/>
```

SIP Related options

enable-timer

This enables or disables support for RFC 4028 SIP Session Timers.

```
<param name="enable-timer" value="false"/>
```

session-timeout

session timers for all call to expire after the specified seconds Then it will send another invite (re-invite). If not specified defaults to 30 minutes. Some gateways may reject values less than 30 minutes. This values refers to Session-Expires in RFC 4028 (The time at which an element will consider the session timed out, if no successful session refresh transaction occurs beforehand)

```
<param name="session-timeout" value="1800"/>
```

Note: If your switch requires the timer option; for instance, Huawei SoftX3000, it needs this optional field and drops the calls with "Session Timer Check Message Failed", then you may be able to revert back the commit that took away the Require: timer option which is an optional field by:

```
git log -1 -p 58c3c3a049991fedd39f62008f8eb8fca047e7c5 libs/sofia-sip/libsofia-sip-ua | patch -p1 -R
touch libs/sofia-sip/.update

make mod_sofia-clean
make mod_sofia-install
```

enable-100rel

This enable support for 100rel (100% reliability - PRACK message as defined in RFC3262) This fixes a problem with SIP where provisional messages like "180 Ringing" are not ACK'd and therefore could be dropped over a poor connection without retransmission. *2009-07-08:* Enabling this may cause FreeSWITCH to crash, see FSCORE-392.

```
<param name="enable-100rel" value="true"/>
```

minimum-session-expires

This sets the "Min-SE" value (in seconds) from RFC 4028. This value must not be less than 90 seconds.

```
<param name="minimum-session-expires" value="120"/>
```

sip-options-respond-503-on-busy

When set to true, this param will make FreeSWITCH respond to incoming SIP OPTIONS with 503 "Maximum Calls In Progress" when FS is paused or maximum sessions has been exceeded.

When set to false or when not set at all (default behavior), SIP OPTIONS are always responded with 200 "OK".

```
<param name="sip-options-respond-503-on-busy" value="false"/>
```

Setting this param to true is especially useful if you're using a proxy such as OpenSIPS or Kamailio with dispatcher module to probe your FreeSWITCH servers by sending SIP OPTIONS.

sip-force-expires

Setting this param overrides the *expires* value in the 200 OK in response to all inbound SIP REGISTERs towards this sip_profile.

This param can be overridden per individual user by setting a *sip-force-expires* user directory variable.

sip-expires-max-deviation

Setting this param adds a random deviation to the *expires* value in the 200 OK in response to all inbound SIP REGISTERs towards this sip_profile.

Result will be that clients will not re-register at the same time-interval thus spreading the load on your system.

For example, if you set:

```
<param name="sip-force-expires" value="1800"/>
<param name="sip-expires-max-deviation" value="600"/>
```

then the expires that is responded will be between 1800-600=1200 and 1800+600=2400 seconds.

This param can be overridden per individual user by setting a *sip-expires-max-deviation* user directory variable.

outbound-proxy

Setting this param will send all outbound transactions to the value set by outbound-proxy.

```
 <param name="outbound-proxy" value="127.0.0.1"/>
```

send-display-update

Tells FreeSWITCH not to send display UPDATEs to the leg of the call.

```
 <param name="send-display-update" value="false"/>
```

RTP Related options

auto-jitterbuffer-msec

Set this to the size of the jitterbuffer you would like to have on all calls coming through this profile.

```
<param name="auto-jitterbuffer-msec" value="120"/>
```

rtp-timer-name

```
<param name="rtp-timer-name" value="soft"/>
```

rtp-rewrite-timestamps

If you don't want to pass through timestamps from 1 RTP stream to another, rtp-rewrite-timestamps is a parameter you can set in a SIP Profile (on a per call basis with rtp_rewrite_timestamps chanvar in a dialplan).

The result is that FreeSWITCH will regenerate and rewrite the timestamps in all the RTP streams going to an endpoint using this SIP Profile.

This could be necessary to fix audio issues when sending calls to some paranoid and not RFC-compliant gateways (Cirpack is known to require this).

```
<param name="rtp-rewrite-timestamps" value="true"/>
```

rtp-timeout-sec

The number of seconds of RTP inactivity (media silence) before FreeSWITCH considers the call disconnected, and hangs up. It is recommended that you use session timers instead. If this setting is omitted, the default value is "0", which disables the timeout.

```
<param name="rtp-timeout-sec" value="300"/>
```

rtp-hold-timeout-sec

The number of seconds of RTP inactivity (media silence) for a call placed on hold by an endpoint before FreeSWITCH considers the call disconnected, and hangs up. It is recommended that you use session timers instead. If this setting is omitted, the default value is "0", which disables the timeout.

```
<param name="rtp-hold-timeout-sec" value="1800"/>
```

rtp-autoflush-during-bridge

Controls what happens if FreeSWITCH detects that it's not keeping up with the RTP media (audio) stream on a bridged call. (This situation can happen if the FreeSWITCH server has insufficient CPU time available.)

When set to "true" (the default), FreeSWITCH will notice when more than one RTP packet is waiting to be read in the incoming queue. If this condition persists for more than five seconds, RTP packets will be discarded to "catch up" with the audio stream. For example, if there are always five extra 20 ms packets in the queue, 100 ms of audio latency can be eliminated by discarding the packets. This will cause an audio glitch as some audio is discarded, but will improve the latency by 100 ms for the rest of the call.

If rtp-autoflush-during-bridge is set to false, FreeSWITCH will instead preserve all RTP packets on bridged calls, even if it increases the latency or "lag" that callers hear.

```
<param name="rtp-autoflush-during-bridge" value="true"/>
```

rtp-autoflush

Has the same effect as "rtp-autoflush-during-bridge", but affects NON-bridged calls (such as faxes, IVRs and the echo test).

Unlike "rtp-autoflush-during-bridge", the default is false, meaning that high-latency packets on non-bridged calls will not be discarded. This results in smoother audio at the possible expense of increasing audio latency (or "lag").

Setting "rtp-autoflush" to true will discard packets to minimize latency when possible. Doing so may cause errors in DTMF recognition, faxes, and other processes that rely on receiving all packets.

```
<param name="rtp-autoflush" value="true"/>
```

Auth

These settings deal with authentication: requirements for identifying SIP endpoints to FreeSWITCH.

challenge-realm

Choose the realm challenge key. Default is auto_to if not set.

auto_from - uses the from field as the value for the SIP realm. auto_to - uses the to field as the value for the SIP realm. <anyvalue> - you can input any value to use for the SIP realm.

If you want URL dialing to work you'll want to set this to auto_from.

If you use any other value besides auto_to or auto_from you'll loose the ability to do multiple domains.

Note: comment out to restore the behavior before 2008-09-29

```
<param name="challenge-realm" value="auto_from"/>
```

accept-blind-auth

accept any authentication without actually checking (not a good feature for most people)

```
<param name="accept-blind-auth" value="true"/>
```

auth-calls

Users in the directory can have "auth-acl" parameters applied to them so as to restrict users access to a predefined ACL or a CIDR.

```
<param name="auth-calls" value="$${internal_auth_calls}"/>
```

Value can be "false" to disable authentication on this profile, meaning that when calls come in the profile will *not* send an auth challenge to the caller.

log-auth-failures

Write log entries (Warning) on authentication failures (Registration & Invite). useful for users wishing to use fail2ban. note: Required SVN#15654 or higher

```
<param name="log-auth-failures" value="true"/>
```

auth-all-packets

On authed calls, authenticate *all* the packets instead of only INVITE and REGISTER(Note: OPTIONS, SUBSCRIBE, INFO and MESSAGE are not authenticated even with this option set to true, see http://jira.freeswitch.org/browse/FS-2871)

```
<param name="auth-all-packets" value="false"/>
```

Registration

disable-register

disable register which may be undesirable in a public switch

```
<param name="disable-register" value="true"/>
```

multiple-registrations

Valid values for this parameter are "contact", "true", "false". value="true" is the most common use. Setting this value to "contact" will remove the old registration based on sip_user, sip_host and contact field as opposed to the call_id.

```
<param name="multiple-registrations" value="contact"/>
```

max-registrations-per-extension

as per Jira FS-2720

accept-blind-reg

this lets anything register comment the next line and uncomment one or both of the other 2 lines for call authentication

```
<param name="accept-blind-reg" value="true"/>
```

inbound-reg-force-matching-username

Force the user and auth-user to match.

```
<param name="inbound-reg-force-matching-username" value="true"/>
```

force-publish-expires

Force custom presence update expires delta (-1 means endless)

```
<param name="force-publish-expires" value="true"/>
```

force-register-domain

all inbound registrations will look in this domain for the users. Comment out to use multiple domains

```
<param name="force-register-domain" value="$${domain}"/>
```

force-register-db-domain

all inbound reg will stored in the db using this domain. Comment out to use multiple domains

```
<param name="force-register-db-domain" value="$${domain}"/>
```

send-message-query-on-register

Can be set to 'true', 'false' or 'first-only'. If set to 'true' (this is the default behavior), mod_sofia will send a message-query event upon registration. mod_voicemail uses this for counting messages.

```
<param name="send-message-query-on-register" value="true"/>
```

If set to 'first-only', only the first REGISTER will trigger the message-query (it requires the UA to increment the NC on subsequent REGISTERs. Some phones, snom for instance, do not do this).

The final effect of the message-query is to cause a NOTIFY MWI message to be sent to the registering UA (it is used to satisfy terminals that expect MWI without subscribing for it).

unregister-on-options-fail

If set to True with nat-options-ping the endpoint will be unregistered if no answer on OPTIONS packet.

```
<param name="unregister-on-options-fail" value="true"/>
```

nat-options-ping

With this option set FreeSWITCH will periodically send an OPTIONS packet to all NATed registered endpoints to keep alive connection. If set to True with unregister-on-options-fail the endpoint will be unregistered if no answer on OPTIONS packet.

```
<param name="nat-options-ping" value="true"/>
```

all-reg-options-ping

With this option set FreeSWITCH will periodically send an OPTIONS packet to all registered endpoints to keep alive connection. If set to True with unregister-on-options-fail the endpoint will be unregistered if no answer on OPTIONS packet.

```
<param name="all-reg-options-ping" value="true"/>
```

registration-thread-frequency

Controls how often registrations in the FreeSWITCH are checked for expiration.

```
<param name="registration-thread-frequency" value="30"/>
```

inbound-reg-in-new-thread

For each inbound register, launch a new thread to process it, e.g. for when using heavier backends

```
<param name="inbound-reg-in-new-thread" value="true"/>
```

ping-mean-interval

Controls the mean interval FreeSWITCH™ will send OPTIONS packet to registered user, by default 30 seconds.

```
<param name="ping-mean-interval" value="30"/>
```

ping-thread-frequency

Controls the frequency, withn the mean time, in which to send pings. by default is 1 second.

<param name="ping-thread-frequency" value="1"/>

Example: with interval set to 30, and frequency set to 1, for a 1000 registered users, FS will ping 33 users a second, and start over every 30 seconds. (1000 divided by 30 = 33)

Subscription

force-subscription-expires

force suscription expires to a lower value than requested

```
<param name="force-subscription-expires" value="60"/>
```

force-subscription-domain

all inbound subscription will look in this domain for the users. Comment out to use multiple domains

```
<param name="force-subscription-domain" value="$${domain}"/>
```

Presence

manage-presence

Enable presence.

If you want to share your presence (see dbname and presence-hosts) set this to "true" on the first profile and enable the shared presence database. Then on subsequent profiles that share presence set this variable to "passive" and enable the shared presence database there as well.

```
<param name="manage-presence" value="true"/>
```

dbname

Used to share presence info across sofia profiles

Name of the db to use for this profile

```
<param name="dbname" value="share_presence"/>
```

presence-hold-state

By default when a call is placed on hold, monitoring extensions show that extension as ringing. You can change this behavior by specifying this parameter and one of the following values. Available as of commit 1145905 on April 13, 2012.

- confirmed - Extension appears busy.
- early (default) - Extension appears to be ringing.
- terminated - Extension appears idle.

```
<param name="presence-hold-state" value="confirmed"/>
```

presence-hosts

A list of domains that have a shared presence in the database specified in dbname. People who use multiple domains per profile can't use this feature anyway, so you'll want to set it to something like "_DISABLED_" in this case to avoid getting users from similar domains all mashed together. For multiple domains also known as multi-tenant calling 1001 would call all matching users in all domains. Don't use presence-hosts with multi-tenant.

```
<param name="presence-hosts" value="$${domain}"/>
```

presence-privacy

Optionally globally hide the caller ID from presence notes in distributed NOTIFY messages. For example, "Talk 1002" would be the presence note for extension 1001 while it is on a call with extension 1002. If the presence privacy tag is set to true, then it would distribute the presence note as "On The Phone" (without the extension to which it is connected). So any subscriber's to 1001's presence would not be able to see who he/she is talking to. http://jira.freeswitch.org/browse/FS-849 This also hides the number in the status "hold", "ring", "call" and perhaps others. http://jira.freeswitch.org/browse/FS-4420

```
<param name="presence-privacy" value="true"/>
```

send-presence-on-register

Specify whether or not to send presence information when users register. Default is not to send presence information. Valid options:

- false
- true
- first-only

```
<param name="send-presence-on-register" value="true"/>
```

CallerID Related options

caller-id type

choose one, can be overridden by inbound call type and/or sip_cid_type channel variable

Remote-Party-ID header:

```
   <param name="caller-id-type" value="rpid"/>
```

P-*-Identity family of headers:

```
   <param name="caller-id-type" value="pid"/>
```

neither one:

```
   <param name="caller-id-type" value="none"/>
```

pass-callee-id

(defaults to true) Disable by setting it to false if you encounter something that your gateway for some reason hates X-headers that it is supposed to ignore

```
<param name="pass-callee-id" value="false"/>
```

Other (TO DO)

hold-music

```
<param name="hold-music" value="$${hold_music}"/>
```

disable-hold

This allows to disable Music On Hold (added in GIT commit e5cc0539ffcbf660637198c698e90c2e30b05c2f, from Fri Apr 30 19:14:39 2010 -0500).

This can be useful when the calling device intends to send its own MOH, but nevertheless sends a REINVITE to FreeSWITCH triggering its MOH.

This can be done from dialplan also with rtp_disable_hold channel variable.

```
<param name="disable-hold" value="true"/>
```

apply-inbound-acl

set which access control lists, defined in acl.conf.xml, apply to this profile

```
<param name="apply-inbound-acl" value="domains"/>
```

apply-register-acl

```
<param name="apply-register-acl" value="domains"/>
```

apply-proxy-acl

```
<param name="apply-proxy-acl" value="myproxies"/>
```

This allows traffic to be sent to FreeSWITCH via one or more proxy servers.

The proxy server should add a header named X-AUTH-IP containing the IP address of the client. FreeSWITCH trusts the proxy because its IP is listed in the proxy server ACL, and uses the value of the IP in this header as the client's IP for ACL authentication (acl defined in apply-inbound-acl).

record-template

```
<param name="record-template" value="$${base_dir}/recordings/${caller_id_number}.${target_domain}.${strftime(%Y-%
m-%d-%H-%M-%S)}.wav"/>
```

max-proceeding

max number of open dialogs in proceeding

```
<param name="max-proceeding" value="1000"/>
```

bind-params

if you want to send any special bind params of your own

```
<param name="bind-params" value="transport=udp"/>
```

disable-transfer

disable transfer which may be undesirable in a public switch

```
<param name="disable-transfer" value="true"/>
```

manual-redirect

```
 <param name="manual-redirect" value="true"/>
```

enable-3pcc

enable-3pcc determines if third party call control is allowed or not. Third party call control is useful in cases where the SIP invite doesn't include a SDP (late media negotiation).

enable-3pcc can be set to either 'true' or 'proxy', true accepts the call right away, proxy waits until the call has been answered then sends accepts

```
<param name="enable-3pcc" value="true"/>
```

nonce-ttl

TTL for nonce in sip auth

```
<param name="nonce-ttl" value="60"/>
```

This parameter is set to 60 seconds if not set here. It's used to determine how long to store the user registration record in the sip_authentication table. The expires field in the sip_authentication table is this value plus the expires set by the user agent.

sql-in-transactions

If set to true (default), it will instruct the profile to wait for 500 SQL statements to accumulate or 500ms to elapse and execute them in a transaction (to boost performance).

```
<param name="sql-in-transactions" value="true"/>
```

odbc-dsn

If you have ODBC support and a working dsn you can use it instead of SQLite

```
<param name="odbc-dsn" value="dsn:user:pass"/>
```

mwi-use-reg-callid

```
<param name="mwi-use-reg-callid" value="false">
```

username

If you wish to hide the fact that you are using FreeSWITCH in the SDP message (Specifically the o= and and s= fields) , then set the username param under the profile. This has no relation whatsoever with the username parameter when we're dealing with gateways. If this value is left unset the system defaults using FreeSWITCH as the username parameter with the o= and s= fields.

```
<param name="username" value="AnyValueHere" />
```

Example:

```
.
v=0.
o=root 1346068950 1346068951 IN IP4 1.2.3.4.
s=root.
c=IN IP4 1.2.3.4.
t=0 0.
m=audio 26934 RTP/AVP 18 0 101 13.
a=fmtp:18 annexb=no.
a=rtpmap:101 telephone-event/8000.
a=fmtp:101 0-16.
a=ptime:20.
```

when you set <param name="username" value="root" />

Directory of Users

To allow users to register with the server, the user information must be specified in the conf/directory/default/*xml file. To dynamically specify what users can register, use Mod xml curl

Default Configuration File

From Git

Reloading

If you've made a change in sofia.conf.xml, there are two ways to get FreeSWITCH to use the new values.

- shutdown and restart FreeSWITCH (or)
- unload and load mod_sofia

If you've only made changes to a particular profile, you may simply (**WARNING**: will drop all calls associated with this profile):

- sofia profile <profilename> restart reloadxml

Security Features

- SIP TLS for secure signaling.
- SRTP for secure media delivery.
- The Auth section above for authentication settings.