mod_skypopen
mod_skypopen does not work anymore

http://lists.freeswitch.org/pipermail/freeswitch-users/2017-June/126571.html

About

mod_skypopen was the Skype endpoint implemented by FreeSWITCH, but it does not work anymore. See http://lists.freeswitch.org/pipermail/freeswitch-users/2017-June/126571.html

SHORT BLUEPRINT: STEPS NEEDED TO USE SKYPOPEN

- Compile and install FreeSWITCH
- Compile and install mod_skypopen
- Install (on Linux) Xvfb and Pulseaudio
- Configure Pulseaudio for low cpu usage
- Download and install Skype client (on Linux, this is done by the INTERACTIVE INSTALLER)
- Configure Skype client (on Linux, this is done by the INTERACTIVE INSTALLER)
- Install and edit mod_skypopen config file (on Linux, this is done by the INTERACTIVE INSTALLER)
- Create a script to start the Skype clients with needed parameters (on Linux, this is done by the INTERACTIVE INSTALLER)
- Start Skype clients using the script
- Start FreeSWITCH
- Load mod_skypopen in FreeSWITCH
- Profit!

NEW Interactive INSTALLER for LINUX: using the provided script, you'll end up with a complete and configured, ready for calls, installation of Skypopen on FreeSWITCH!!! (see below)

Overview and Info

What is Skypopen

This software (Skypopen) uses the Skype API but is not endorsed, certified or otherwise approved in any way by Skype.

Skypopen is an endpoint (channel driver) that uses the Skype client as an interface to the Skype network, and allows incoming and outgoing Skype calls to/from FreeSWITCH (that can be bridged, originated, answered, etc. as in all other endpoints, e.g. Sofia-SIP).

Skypopen works in FreeSWITCH (FS) on both Linux and Windows (directly, no Cygwin needed), native at 16khz (Skype client has 16khz audio I/O).

Think of Skypopen as similar to FreeTDM for analog lines. For each channel you need an interface (a Skype client). So, for example, two concurrent calls would need two channels, and therefore two Skype clients running on your FreeSWITCH server.

If your Skype client(s) have Skype credits, then Skypopen works for SkypeOut calls as well.

On Linux the Skype client can use a lot of CPU if you don’t customize your setup. To lower its CPU consumption, you customize the configuration of Pulseaudio.

You start your Skype clients using the provided scripts (the interactive installer for Linux write a script, while mod_skypopen/configs/startskype.bat is for windows).

Skypopen has been contributed to the community by: Giovanni Maruzzelli (gmaruzz at gmail dot com) with lot of help from the core developer’s team, and hints, patches, suggestions, bug reports, features requests from the superlative FreeSWITCH community.

Which Skype Client to use on Linux

Use the dynamic build of the stable Skype client (4.3.37), is the only one that works (the interactive installer automatically download and install the correct Skype client, see below).

Don’t use the build for your distro

Supported Platforms and Distros

Skypopen runs well on Linux and Windows. Does not work on Mac OSX.
Most supported Linux distro are:

- Debian 7 64 bit server (wheezy)
- CentOS 6.5.x 64 bit server (RHEL6)
- Ubuntu SERVER LTS 64 bit (Trusty - 14.04)

If you don't have a specially compelling reason (that will pay you back for time and frustration), do yourself a favor and go with one of the most supported Linux distro, or with Windows.

Skypopen runs on Virtual Machines, there are various installation happily running on EC2 (so, Xen).

On Windows, Skypopen runs well starting from XP up.

**Skype Contacts, Messages, Buddy List, etc**

Skypopen automatically directs the Skype client instance to agree to a request of being added to the Contact List and to make known its "status" (online, offline, etc), without adding the requester to the Contact List of the Skype client instance.

Skypopen automatically directs the Skype client instance to flag as "Seen" all received messages.

**Building, Configuration and Operation**

**Linux**

**Which packages are required to build mod_skypopen on Linux**

You will need the libraries needed by Skype client and by the fake X server.

**REALLY, you need all the packages below!**

You can cut and paste the follow:

**Debian 7.x required packages**

In addition to the packages required to build and run FreeSWITCH, you will need the following:
# Dependencies for building mod_skypopen:
apt-get install \
libx11-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev \
x11proto-core-dev x11proto-input-dev x11proto-kb-dev xtrans-dev

# Infrastructure needed to run skype client:
apt-get install \nxvfb pulseaudio

# Dependencies of skype client:
# Enable i386 libraries installing
dpkg --add-architecture i386
apt-get update
# Install libraries
apt-get install \
fontconfig fontconfig-config gcc-4.7-base:i386 libasound2:i386 \libasound2-plugins:i386 libasynncnc0:i386 libbattr1:i386 libaudio2:i386 \libbavahi-client3:i386 libbavahi-common-data:i386 libbavahi-common3:i386 \libbavocoder53:i386 libbavutil15:i386 libb6c6:i386 libb6c6-i686:i386 libcap2:i386 \libcomerr2:i386 libdbus-1-3:i386 libdirac-encoder0:i386 \libexpat1:i386 libffi6:i386 libflac8:i386 libfontconfig1 libfontconfig1:i386 \libfreetype6:i386 libgmp3:i386 libgpg-error0:i386 libgstreamer-plugins-base0.10-0:i386 \libgstreamer-plugins0.10-0:i386 libgles2:i386 libglib2.0-0:i386 \libgobject-2.0-0:i386 libgstreamer-0.10-0:i386 libgstreamer-plugins-base0.10-0:i386 \libgstreamer-plugins0.10-0:i386 libgstreamer0.10-0:i386 libgstreamer1.0-0:i386 \libgl1:i386 libglu1:i386 libglu1-mesa:i386 libgstreamer-1.0-0:i386 \libgstreamer-plugins-1.0-0:i386 libgstreamer-plugins-1.0-2:i386 \libgstreamer-plugins-1.0-0:i386 libgstreamer-plugins-1.0-2:i386 \libgstreamer-plugins-1.0-2:i386 libgstreamer-plugins0.10-0:i386 \libgstreamer-plugins1.0-0:i386 libgstreamer0.10-0:i386 libgstreamer1.0-0:i386 \libgstreamer1.0-0:i386 libgstreamer-plugins-1.0-0:i386 \libgstreamer-plugins-1.0-0:i386 libgstreamer-plugins-1.0-2:i386 \libgstreamer-plugins-1.0-2:i386 libgstreamer-plugins0.10-0:i386 \libgstreamer-plugins1.0-0:i386 libgstreamer0.10-0:i386 libgstreamer1.0-0:i386 \libgstreamer1.0-0:i386 libgstreamer-plugins-1.0-0:i386 \libgstreamer-plugins-1.0-0:i386 libgstreamer-plugins-1.0-2:i386 \libgstreamer-plugins-1.0-2:i386 libgstreamer-plugins0.10-0:i386 \libgstreamer-plugins1.0-0:i386 libgstreamer0.10-0:i386 libgstreamer1.0-0:i386 \libgstreamer1.0-0:i386 libgstreamer-plugins-1.0-0:i386 \libgstreamer-plugins-1.0-0:i386 libgstreamer-plugins-1.0-2:i386 \libgstreamer-plugins-1.0-2:i386 libgstreamer-plugins0.10-0:i386 \libgstreamer-plugins1.0-0:i386 libgstreamer0.10-0:i386 libgstreamer1.0-0:i386

Ubuntu 14.04.x required packages

In addition to the packages required to build and run FreeSWITCH, you will need the following:
# Dependencies for building mod_skypopen:
apt-get install \
libx11-dev libx11-dev libxau-dev libxcb1-dev libxdmcp-dev \
x11proto-core-dev x11proto-input-dev x11proto-kb-dev xtrans-dev

# Infrastructure needed to run skype client:
apt-get install \
xvfb pulseaudio

# Dependencies of skype client:
apt-get install \
fontconfig fontconfig-config gcc-4.7-base:i386 libasound2:i386 \
libasound2-plugins:i386 libasynchans0:i386 libattr1:i386 libaudio2:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 \
libaba-client3:i386 libbash-common-data:i386 libbash-common3:i386 

CentOS 6.5.x and 7.x required packages

In addition to the packages required to build and run FreeSWITCH, you will need the following:
Build procedure (all Linuxes) for FS and mod_skypopen

Note you must install the hd sounds and music packages, that includes both 8khz and 16khz (skype works at 16 khz native), eg: from FreeSWITCH source directory: "make hd-sounds-install & make hd-moh-install"

After you finished to compile FreeSWITCH, in FreeSWITCH source directory edit modules.conf and uncomment the mod_skypopen line.

edit modules.conf

Then, "make install" again from the FreeSWITCH root directory (this will compile and install just the mod_skypopen module).

make install #yes, again, this time will add mod_skypopen

Pulseaudio Configuration

Debian 7

Configuration files:
edit /etc/default/pulseaudio
PULSEAUDIO_SYSTEM_START=1

edit /etc/pulse/system.pa  # delete or comment out all file contents, and add at the end:
load-module module-null-sink
load-module module-native-protocol-unix auth-anonymous=1

edit /etc/init.d/pulseaudio
# delete --high-priority

edit /etc/pulse/daemon.conf
# add at the end of the file:
daemonize = yes
system-instance = yes
resample-method = trivial
flat-volumes = yes
default-sample-rate = 48000
default-sample-channels = 1
default-fragment-size-msec = 20
alternate-sample-rate = 44100

Add the user running the Skype clients to pulse-access group

cd /etc/pulse
adduser root pulse-access

Ubuntu 14.04
Configuration files:

edit /etc/init/pulseaudio.conf
# uncomment "start on ..."

edit /etc/pulse/system.pa  # delete or comment out all file contents, and add at the end:
load-module module-null-sink
load-module module-native-protocol-unix auth-anonymous=1

edit /etc/init.d/pulseaudio
# delete --high-priority

edit /etc/pulse/daemon.conf
# add at the end of the file:
daemonize = yes
system-instance = yes
resample-method = trivial
flat-volumes = yes
default-sample-rate = 48000
default-sample-channels = 1
default-fragment-size-msec = 20
alternate-sample-rate = 44100

Add the user running the Skype clients to pulse-access group

cd /etc/pulse
adduser root pulse-access

CentOS 6.5 and 7.x
Configuration files:

edit /etc/pulse/system.pa # delete or comment out all file contents, and add at the end:
load-module module-null-sink
load-module module-native-protocol- unix auth-anonymous=1

edit /etc/pulse/daemon.conf
# add at the end of the file:
daemonize = yes
system-instance = yes
resample-method = trivial
flat-volumes = yes
default-sample-rate = 48000
default-sample-channels = 1
default-fragment-size-msec = 20
alternate-sample-rate = 44100

Add the user running the Skype clients to pulse-access group

usermod -G pulse-access -a root

Disable SELinux

edit /etc/sysconfig/selinux # disabled!

Start Pulseaudio, there is no /etc/init.d/pulseaudio, so you must start it "by hand" (eg: from /etc/rc.local):

/usr/bin/pulseaudio --system=true --daemonize=true --log-level=info --log-target=syslog

Interactive Skype Client INSTALLER and CONFIGURATOR

Go to the "install" subdir of mod_skypopen sources, then execute the "install.pl" script.

cd /usr/src/freeswitch/src/mod/endpoints/mod_skypopen/install
./install.pl

It will ask you questions and will let you review all the values you entered before to do any real action.
Then it will download and install (where you told it) the correct Skype client, and will create and install (where and how you told it) all the needed configurations and startup scripts.
You'll end up with a complete and configured installation of mod_skypopen for FreeSWITCH, ready for calls.
From 0 to Skype calls in 30 seconds !!!

Skype 4.3.37 is the only Skype client version that works

Starting first week of August 2014, Microsoft stopped operating the authorization servers for Skype clients prior to version 4.3.37. They require Pulseaudio. Following this page and using the automatic installer you will be all set, no problem.

How to start the Skype clients, then start FreeSWITCH and load mod_skypopen on Linux

First you start the Skype clients with the script created by the interactive configurator, then you start FreeSWITCH, and load mod_skypopen in it.

sh /usr/local/freeswitch/skypopen/skype-clients-startup-dir/start_skype_clients.sh
/usr/local/freeswitch/bin/freeswitch
load mod_skypopen

Yay!

Problems in starting skype clients ?

Check you disabled both SELinux (on CentOS) or AppArmor (on Ubuntu)
**Windows**

**How to build Skypopen on Windows, and which tools and files are needed**

Skypopen has been designed to work well on windows since the beginning, like FreeSWITCH itself.

Skypopen (mod_skypopen) is automatically built when you build FreeSWITCH on Windows.

You will need the Visual C compiler from Microsoft, commercial version, or the free (as in beer) Visual C Express (requires registration). They both give the same results in our case (eg: no need to buy the commercial version just for Skypopen).

After having downloaded the FreeSWITCH sources from svn or the packaged FreeSWITCH source release, follow the instruction on how to build FreeSWITCH on Windows. Using Visual C (Express or not):

1. Open Freeswitch.sln
2. Right click the main solution node at the top of the Solution Explorer
3. Right click and select Build

Then, download and install the 16khz sounds and music, needed by Skypopen (16khz native, like Skype itself):

- Right click on "Download 16khzsounds" and select Build
- Right click on "Download 16khzmusic" and select Build
- Right click on "16khz sound" and select Build
- Right click on "16khz music" and select Build


Then, go forward with the configuration of mod_skypopen (see below).

**Config files location and script to start Skype client instances on Windows**

You'll probably build the "Debug" version. So, go and copy src/mod/endpoints/mod_skypopen/configs/skypopen.conf.xml to Debug/conf/autoload_configs.

Edit it to reflect your configuration.

Then, go into src/mod/endpoints/mod_skypopen/configs/ and edit startskype.bat to suit your configuration. Remember that startskype.bat needs wait.bat to be in its same directory (wait.bat is in the "configs" dir too).

Eventually, launch the Skype client instances using startskype.bat, then launch FreeSWITCH, and from the FreeSWITCH command line:

```bash
load mod_skypopen
```

The Skype instances will ask you for permission to be connected by Skypopen. Give the auth.

If you put in config file the `silent_mode` (recommended, see section on configuration file), another authorization will be asked to you. Give it, and not to ask again!

You're all set.

**Running Skypopen on Windows as a Service**

I assume you have FreeSWITCH configured and working with mod_skypopen (running from the command line). I mean, first you have to make sure all is working as a "normal non-service application".

The difficult thing is to have the Skype client instances started by a service (FreeSWITCH itself is able to be a Windows Service).

You install FreeSWITCH as service (freeswitch.exe -install servicename), then use the "services" applet to run it under "local system" user, starting manually (not automatically).

Then edit to your like (Skype usernames/passwords AND change the name of the service to be started from AICCU to the name of the FreeSWITCH service - on the last line -) and copy the 2 command files contained in the configs/windows-service/ directory to a known location (eg. C:\).


Procedure for creating a service is detailed here: [http://support.microsoft.com/kb/137890](http://support.microsoft.com/kb/137890) (or more shortly here: [http://www.sixxs.net/wiki/Configuring_Windows_Vista#2816.29__Installing_AICCU_Utility_as_a_Service](http://www.sixxs.net/wiki/Configuring_Windows_Vista#2816.29__Installing_AICCU_Utility_as_a_Service))

You create an (empty) service with those tools, then you follow the procedure steps and as "Parameters"->"Application" you put the string "C:\startskype.cmd"

Then, from the "services" applet in Control Center ->administrative tools, you make sure the service is owned by "local system" and that "Access desktop" is ticked.

Start the service manually from the "services" applet.
Maybe it will appear a "the service wants to access the desktop". Go to "show message" to see what Skype wants, and give some configurations if needed.

After a while, when the FreeSWITCH service will be started by our new "startskype.cmd" service, it will appear "the service wants to access the desktop". Go there and give Skype clients authorization to be connected by FreeSWITCH, forever.

Stop both services from the "Services" applet.

Unfortunately I've found no way to have the Skype instances shutdown when the service is shutdown. So, exit the Skype client instances manually, or reboot.

Restart the service, manually from the "Services" applet.

From another machine, make a Skype call to FreeSWITCH.

If all works as expected, stop both services, make sure (via "services" applet) the FreeSWITCH service will retry three times to start, with a minute pause (just to allow for the Skype clients to start and settle their connection with the network, to be on the safe side).

Make the service to start "automatic" (only the "startskype.cmd" service, not the FreeSWITCH service. The FreeSWITCH service is started by the "startskype.cmd" service.

Reboot the machine, don't log in, make another test call to FreeSWITCH via Skype, and... joy :-)!

The trick for me (on Vista) is to use the same "local system" account for both the Skype and the FreeSWITCH services, and "NOT" to use a personal account. Go figure... :-)

**Config File Details (skypopen.conf.xml)**

Following is a sample config file.

In the first section you set defaults, then you override in the "interface" sections. All and each global_settings parameter can be overridden in each of the interface definition.

X11-display is not used on Windows :).

report_incoming_chatmessages is useful when you have multiple channels with the same skypeusername. They will all receive a message for that skype_user. You probably want to have one only MESSAGE event fired on the ESL.

write_silence_when_idle will make mod_skypopen to send packets of silence to the Skype client when there are no audio writes from FreeSWITCH (inactivity, silence, network lag, etc). This proved useful on some platforms to keep the Skype client on his toes, and to avoid loosing packets "synch" that ends up in a ghost sounding voice.

setsockopt will make mod_skypopen to drastically reduce the size of the tcp buffer and set the option NO_TCP_DELAY. This will results in a lower latency and more respect for internal timing, but an higher CPU load.

silent_mode works on Windows, using the latest Skype client (update it! :) ). At very first start of FreeSWITCH each Skype client will ask you an additional authorization, and from now on, the Skype client will never pop up windows on calls or chat. It will save a good deal of cpu cycles! silent_mode is not used by the Linux version (present Skype clients for Linux do not understand it).

Following example config file is for 4 interfaces, 2 of which will be connected to Skype client instances running as the same skypeuser "user5".
<configuration name="skypopen.conf" description="Skypopen Configuration">
  <global_settings>
    <param name="dialplan" value="XML"/>
    <param name="context" value="default"/>
    <param name="destination" value="5000"/>
    <param name="skype_user" value="user5"/>
    <param name="report_incoming_chatmessages" value="false"/>
    <param name="silent_mode" value="false"/>
    <param name="write_silence_when_idle" value="false"/>
    <param name="setsockopt" value="true"/>
  </global_settings>
  <!-- one entry here per each skypopen interface -->
  <per_interface_settings>
    <interface id="1" name="interface1">
      <param name="X11-display" value=":101"/>
      <param name="skype_user" value="user1"/>
      <param name="report_incoming_chatmessages" value="true"/>
    </interface>
    <interface id="2" name="interface2">
      <param name="X11-display" value=":102"/>
      <param name="skype_user" value="user3"/>
      <param name="destination" value="9664"/>
      <param name="report_incoming_chatmessages" value="true"/>
    </interface>
    <interface id="3" name="interface3">
      <param name="X11-display" value=":103"/>
      <param name="report_incoming_chatmessages" value="true"/>
    </interface>
    <interface id="4" name="interface4">
      <param name="X11-display" value=":104"/>
    </interface>
  </per_interface_settings>
</configuration>

Dialplan, and how to use Skypopen

Dialplan

Like other endpoints it's easy to build up useful dialplans using Skypopen. There are a few simple examples given in the "default.xml" dialplan located in mod_skypopen/configs/

You can use the standard format with the interface name:

skypopen/interface1/echo123

to call "echo123" using the skypopen interface named "interface1"

<!-- dial echo123 via skypopen using interface1 interface to go out -->
<extension name="skypopen">
  <condition field="destination_number" expression="^2909$">
    <action application="bridge" data="skypopen/interface1/echo123"/>
  </condition>
</extension>

The "RR" interface, poor man interface grouping

You can also use the "RR" interface

skypopen/RR/echo123

to call "echo123" using the first available (idle, not in a call) skypopen interface, automatically selected (thx Seven Du).

"RR" will choose an available idle interface based on a round robin algorithm (so to distribute outbound calls more fairly between all the available interfaces).
optional "Skype URI"

If you modify your dialplan, you'll be able to call "skype uri"s for terminating calls via the Skype network.

A "Skype URI" is composed by the word 'skype' followed by a slash and a destination: "skype/remote_skypename" or "skype/skypeout_phonenumber"

E.g. "skype/echo123" or "skype/+18007844444".

You can use the "skype uri" of the provided dialplan directly from a SIP softphone, e.g. X-Lite.

And you can call the "2909" extension in the provided dialplan from any kind of phone to be bridged to the Skype Test Call.

Dialplan snippet:

```xml
<!-- dial via SKYPE uri using interface1 interface to go out -->
<extension name="skype_uri">
  <condition field="destination_number" expression="^skype/(.*)$">
    <action application="bridge" data="skypopen/interface1/$1"/>
  </condition>
</extension>
```

With the provided skypopen.conf.xml all incoming Skype calls will be routed to the "5000" extension, the IVR in default FreeSWITCH installation.

API and CLI Commands

Skypopen adds the "sk" and the "skypopen" API/CLI commands, for sending Skype API commands to the Skype client instance related to an interface. Both "sk" and "skypopen" commands can be used through the command line and via API/socket/ESL/whatever.

**sk**

"sk" commands are intended to be used from the FreeSWITCH command line ("sk remove" and "sk reload" can be useful from Event socket as well).

You begin typing "sk console interface_name" to direct the "current console" to sending messages to interface_name. Starting now on, you can type "sk command_string" and command_string will be sent to the Skype client instance related to interface interface_name.

```
sk console interface1
sk MESSAGE alice25 Hej Alice, this is a Skype chat message for you!
```

**sk list**

"sk list" gives the list and status of all the Skypopen interfaces (a star marks the interface from which "RR" - see below - will start hunting an IDLE one).

```
sk list
```

**sk balances**

"sk balances" gives the list and balances (in account currency) of all the Skypopen interfaces (e.g: the credits you have for Skypeout, calling PSTN). Patch added by Ivan Trifonov

```
sk balances
```

We got patches from Muhammad Shahzad and Seven Du for adding and removing interfaces on the fly.
sk reload

sk reload => this re-reads Skypopen configuration file skypopen.conf.xml and adds ONLY new interfaces it found in conf. All existing interfaces are not touched.

sk reload

sk remove

sk remove <skype-user> => this remove skypopen interface associated with given Skype user account, if it is idle.

freeswitch@internal> sk remove Bob

sk remove <#interface-numeric-id> => this remove skypopen interface associated with given Skype interface_id, if is idle (remember, you must add the pound sign in front of the interface-numeric-id, # is mandatory).

freeswitch@internal> sk remove #10

sk remove <#interface-name> => this remove skypopen interface associated with given Skype interface_name, if is idle (remember, you must add the pound sign in front of the interface-name, # is mandatory).

freeswitch@internal> sk remove #interface10

skypopen

"sk remove" and "sk reload" (see before) can be useful from API/socket/ESL/whatever as well.

"skypopen" commands are intended to be used by programs (API/socket/ESL/whatever) and have the format: "skypopen interface_name command_string". They send the command_string to the Skype client instance related to interface_name.

skypopen interface2 MESSAGE bob34 Ciao Bob! This is my Skype chat message for you from FreeSWITCH :-) This allow you to use directly the entire power of the Skype API (http://developer.skype.com/public-api-reference), for eg to send chat messages, interact with the buddy list, etc etc. Typing "console loglevel 9" at the FreeSWITCH command line allows you to see the Skype API answers from the Skype client instance.

skypopen_chat

It supports full utf8 chat text, although the FreeSWITCH command line only accepts ASCII. Please use ESL or API to send utf8 text.

skypopen_chat interface_name remote_skypename chat_text

eg:

skypopen_chat interface2001 gmaruzz5 ciao bello

chat

Skypopen answers to the FreeSWITCH standard "chat" command too, and use its arguments to execute a skypopen_chat command (it will find the first available interface that uses the "local_skypename"). So, if you got a messaging applications that uses chat command with Sofia/SIP or Jingle, no need to recode it with special cases for Skype:).

chat SKYPE|local_skypename|remote_skypename|chat_text

It uses SKYPE as protocol specification. e.g., from command line:

chat SKYPE|user2|gmaruzz5|ciao amore

channel Variables

DTMFs: skype_add_outband_dtmf_also_when_bridged

When a skype call is bridged (eg: to a sip call) the default behavior is NOT to pass the incoming Skype DTMFs as out of band (signaling) DTMFs to the other call, because the audio (inband) DTMFs will be always present. Explanation: Skype sends DTMFs both as inband audio and as signaling, and you cannot disable the inband (audio) DTMFs.

When is desirable that on top of the audio DTMFs the other call receives outband (signaling) DTMFs too (eg: because is an ivr or a conference that do not recognize inband DTMFs), please set the channel variable 'skype_add_outband_dtmf_also_when_bridged' to 'true'.

E.g., in dialplan you can:

<action application="set" data="skype_add_outband_dtmf_also_when_bridged=true"/>
DTMFs: `skype_get_inband_dtmf`

Normally, by default, `mod_skypopen` do not detect inband (audio) incoming DTMFs, because it get it via signaling from the skype peer.

When you have a call to skypeout numbers or an incoming call via the skypein service, you do not get DTMFs in signaling, you get it only as audio.

You set the channel variable `skype_get_inband_dtmf` to 'true' if you want `mod_skypopen` to detect inband (audio) DTMFs coming from skype, eg: in skypein/skypeout calls.

E.g., in dialplan you can:

```xml
<action application="set" data="skype_get_inband_dtmf=true"/>
```

Please note that skype peers will send you BOTH audio and signaling DTMFs, so if you set this variable to 'true' on a "normal" skype-skype call, you may get double digits.

Events

`Skypopen` generates (fires) the standard FreeSWITCH events on voice calls (like the other endpoints) and MESSAGE (chat) events on incoming... chat (like Sofia and Jingle).

Voice Calls

Standard CODEC and CHANNEL_* events.

See Event_list

MESSAGE (Chat)

The Event type generated by an incoming chat is of type MESSAGE (like in Jingle and Sofia).

The most interesting fields are:

- login: the interface name that received the chat
- hint: the skype display name of the sender
- from: the sender's skypename (username)
- date: the date of the received message, urlencoded
- during-call: bool, the message was received while a voice call was ongoing?

And obviously the body, encoded in utf8, that contains the chat's text.

To see it "under the hood" you can connect to the event socket via telnet:

```
# telnet localhost 8021
Content-Type: auth/request
auth ClueCon

Content-Type: command/reply
Reply-Text: +OK accepted

events plain message
Content-Type: command/reply
Reply-Text: +OK event listener enabled plain
```

This is an example of MESSAGE event:
Adding and removing interfaces on the fly

We got a patch from Muhammad Shahzad for adding and removing interfaces on the fly. It basically adds two sk commands:

```
sk reload => this re-reads Skypopen configuration file skypopen.conf.xml and adds ONLY new interfaces it found in conf. All existing interfaces are not touched.
sk remove <skype-user> => this remove skypopen interface associated with given Skype user account, if it is idle.
```

Advanced Configuration

Multiple concurrent incoming calls to the same Skype username

Solutions based on "transfer" the call to another Skype username do not work anymore with new Skype client

Running multiple instances (MAX 6) of the same Skype username on Linux, for BOTH incoming and outbound calls

The new Skype client limits the concurrent instances of the same username to 6

**MAX 6 INSTANCES PER SKYPENAME**

(you can have many skypenames, each one with a max of 6 concurring instances, eg: myskypenameA is used by 6 concurrent skype clients, myskypenameB is used by 6 concurrent skype clients, myskypenameC is used by 6 concurrent skype clients, total 36 concurrent instances on your machine)

On Linux, you can run multiple instances of the same Skype username (eg: multiple Skype clients that register with the Skype networks as the same username, eg: 6 instances of the username "Bob"). On Windows, is not possible.

Please use the INTERACTIVE INSTALLER, it will setup all (download and install the correct Skype client, script to start the Skype clients, Skype client configs, skypopen.conf.xml config file).

The trick is to launch the client instances with separate Xservers (as always with multiple clients) AND separate directories for the Skype client's database (eg: by default it resides in the ".Skype" directory in the user's home).

You can find an example configuration file and launch scripts in configs/multiple-instance-same-skype-username/ directory.

Testing with the "Echo" application
Because the Skype clients have their own AEC (Acoustic Echo Cancellation), if you test Skypopen with the echo() application (press 2 on the demo IVR) you will have a bizarre effect of your voice going up and down in volume "pulsations".

This is caused by the combined AEC of the Skype clients, that "think" the echo() application is the echo generated because you are in an empty room and are using loudspeakers.

On windows AEC can be disabled editing the config.xml file. On Linux, is not possible to disable it (disabling it with the API command has no effect).

So, the test with the echo() application is only good to give you an idea of the latency. If you want to have an idea of the audio quality, use the "delayed echo" at extension 9195 of the default dialplan. Or the echo that is at end of "screaming monkeys" (kindly provided by John Todd, press 5 on the default IVR), but you'll add a transcoding to ulaw 8khz and a round trip to loligo.com servers.

**TO DO**

Requests, suggestions, ideas (feel free to add here, but best is to add on Jira - see below 'BUGS and Feature Requests'):

**TROUBLESHOOTING**

**Problems in starting skype clients ?**

Check you disabled both SELinux (on CentOS) or AppArmor (on Ubuntu)

**Error and warnings at the starting of Skype clients on Linux**

When starting the Skype clients on Linux, on certain distros you can have warnings and errors on the console, like those:

```bash
Couldn't open RGB_DB '/usr/share/X11/rgb'
error opening security policy file /usr/lib64/xserver/SecurityPolicy
```

or this:

Those are completely harmless, and after you are sure your script correctly starts the Skype clients, you can also redirect those outputs to /dev/null, if so you like.

**Use cases, integration, special implementations**

**Skypopen Directory-dialing example**

An example of using text-to-speech to read a list of online Skype contacts with dynamic extensions, then dialing out when the user keys in the contact's extension.

**Passing Caller ID to Remote Skype ID - by Zohair**

I am using xml curl[http://wiki.freeswitch.org/wiki/Mod_xml_curl] and in my php file I added this :
This way Caller Id will be sent before the call as a skype chat message which will only work if remoteskypeid has added the skype id which you are using, otherwise it won't work.

How To Report BUGS and Feature Requests

You can file bug reports, hints, suggestions, feature requests, improvements, patches, etc to https://freeswitch.org/jira/browse/FS/component/10242/ open an account there if you don't have it (it's free ;-)).

That's the best way to give us info on bugs:

1) from the FS CLI: "console loglevel 9"
2) load mod_skypopen
3) reproduce the bug
4) attach the *complete, since beginning* console output as a file attachment to the Jira bug
5) add any comment you may deem useful

If the bug reports crashes, core dumps, etc, please first read this guide: Reporting Issues to GitHub then file a Jira with all relevant info to mod_skypopen.

You can also write to the FreeSWITCH users' and developers' mailing lists: http://lists.freeswitch.org/mailman/listinfo

And you can drop in the IRC channels #freeswitch and #freeswitch-dev on irc.freenode.org to ask questions and discuss issues. The original developer of Skypopen is called "gmaruzz" in the IRC channel.

As a last resort, absolutely not the preferred way, you can write email to Giovanni Maruzzelli - gmaruzz @ gmail dot com

Common Problems

1. When starting pulseaudio with "alternate-sample-rate = 44100" in /etc/pulse/daemon.conf it will complain and not start:

   E: conf-parser.c: [/etc/pulse/daemon.conf:90] Unknown lvalue 'alternate-sample-rate' in section 'n/a'.

   R: Probably your version of PulseAudio doesn't implement this config option, try upgrading PulseAudio or removing this setting.