HowTo Create SIP Echo Service

This HowTo describes the creation of an Echo service built with sipXecs and PJSIP.

You will be able to call an extension and hear your voice echoed back to you. We will be using the pjsua application provided with the PJSIP project.

About PJSIP

PJSIP is a SIP stack supporting many SIP features. It is extremely portable, provides Python bindings and has a small footprint.

pjsua

As part of PJSIP, the project makes available the pjsua application, a command-line SIP UA.

Echo Service

You will need a sipXecs user account to use with the echo service. The pjsua application can run on any computer with adequate connectivity to the sipXecs server. I have built and tested pjsua on both 32 and 64 bit Linux.

sipXecs configuration

Create an account to be used with the echo service. You will need the extension and the SIP password of this user to configure pjsua.

pjsua Configuration

pjsua can be launched with a configuration file. The parameters - with identical syntax - can also be given on the command line. Run 'pjsua --help' to see a list of all accepted parameters.

pjsua echo.conf

1. we don't want the host's audio device
   --null-audio

1. SIP parameters
   --realm example.com
   --registrar sip:example.com # DNS SRV, or FQDN
   --id sip:181@example.com
   --username 181
   --password secret

1. needed for SRV DNS resolution
   --nameserver 192.168.1.1

1. default of 55 will be rejected as being too short by sipX
   --reg-timeout 3600

1. auto-answer all calls with "200 OK"
   --auto-answer 200

1. limit call duration
   --duration 1200

1. automatically loop incoming RTP to outgoing RTP
   --auto-loop

1. mix WAV file into the audio stream
   #--play-file /...

Start pjsua

pjsua menu

+h4. h4. h4. h4. h4. h4. h4. h4. h1. +

Call Commands:	Buddy, IM & Presence:	Account:
m Make new call	+b Add new buddy .	+a Add new accnt
M Make multiple calls	-b Delete buddy	-a Delete accnt.
a Answer call	!b Modify buddy	!a Modify accnt.
h Hangup call (ha=all)	i Send IM	rr (Re-)register
H Hold call	s Subscribe presence	ru Unregister
v re-inVite (release hold)	u Unsubscribe presence	> Cycle next ac.
] Select next dialog	t ToGgle Online status	< Cycle prev ac.
[Select previous dialog ---+		
x Xfer call	Media Commands:	Status & Config:
X Xfer with Replaces		
1. Send DTMF string	cl List ports	d Dump status
dq Dump curr. call quality	cc Connect port	dd Dump detailed
	cd Disconnect port	dc Dump config
S Send arbitrary REQUEST	V Adjust audio Volume	f Save config

----

q QUIT sleep N: console sleep for N ms

h4. h4. h4. h4. h4. h4. h4.

Start pjsua with configuration file

/path/to/pjsua --config-file echo.conf

pjsua should print a 'status=200' message:

09:20:38.650 pjsua_acc.c sip:181@example.com: registration success, status=200 (OK), will re-register in 3371 seconds

and the echo extension (181 in our example) should appear in the list of registered UA's in sipXconfig.

Call the Echo service

Call the extension chosen above (181) from any phone and you should hear your voice echoed back.

Test SIP and RTP connectivity

pjsua being a complete SIP UA, you can also setup the call from the remote end, of course. This allows you to test both SIP and RTP connectivity from a remote, unattended, headless machine. pjsua supports STUN (-~~stun-srv) and ICE (use-ice) as well as manually setting the IP address (~~-ip-addr).

```
pjsua> m
pjsua> sip:111@example.com
```

Extension 111 should ring; when you answer the call you should be able to hear your voice echoed back (assuming you are using the same configuration file as above).

Conference Bridge

pjsua can also provide a minimalistic conference bridge. Add the following to your config and all callers will automatically be added to the conference. See also the *cl*, *cd*, *cc* and *V* commands of pjsua.

1. add incoming calls to conference
   --auto-conf

See also pjmedia conference bridge sample for a more complete conference bridge app built with PJSIP.

Author(s): mike