FreeSWITCH supports a scheduler API that permits the application to create scheduled events that will fire at some point in the future. It also supports a few sample applications that make use of the scheduler as documented below:

Contents

- 1 Scheduled hangup dialplan application
- 2 Scheduled hangup API function
- 3 Scheduled transfer dialplan application
- 4 Scheduled transfer API function
- 5 Scheduled broadcast dialplan application
- 6 Scheduled broadcast API function
- 7 See Also

Scheduled hangup dialplan application

```
<action application="sched_hangup" data="+10 normal_clearing bleg"/>
```

- The cause code is optional and the optional bleg keyword will only hangup the channel the current channel is bridged to if the call is in a bridge.

Scheduled hangup API function

```
sched_hangup +10 <uuid_string> normal_clearing
```

- The cause code is optional

Scheduled transfer dialplan application

```
<action application="sched_transfer" data="+10 1000 XML default"/>
```

- The last 2 args (dialplan and context) are optional

Scheduled transfer API function

```
sched_transfer +10 <uuid_string> 1000 XML default
```

- The last 2 args (dialplan and context) are optional

Scheduled broadcast dialplan application

```
<action application="sched_broadcast" data="+10 playback::/tmp/foo.wav"/>
<action application="sched_broadcast" data="+10 playback::/tmp/foo.wav bleg"/>
<action application="sched_broadcast" data="+10 playback::/tmp/foo.wav both"/>
<action application="sched_broadcast" data="+10 playback!normal_clearing::/tmp/foo.wav"/>
<action application="sched_broadcast" data="+10 phrase::warning"/>
<action application="sched_broadcast" data="+10 phrase!normal_clearing::goodbye"/>
```

- The optional !<cause_code> can be added to make the channel hangup after broadcasting the file.
- The optional bleg/both parameter specifies which leg of the call to broadcast the file on. If not specified, the broadcast only occurs on the a-leg.

Scheduled broadcast API function

```
sched_broadcast +10 <uuid_str> playback::/tmp/foo.wav
sched_broadcast +10 <uuid_str> playback::/tmp/foo.wav bleg
sched_broadcast +10 <uuid_str> playback::/tmp/foo.wav both
sched_broadcast +10 <uuid_str> playback!normal_clearing::/tmp/foo.wav
```

- The optional !<cause_code> can be added to make the channel hangup after broadcasting the file.
- The option bleg/both parameter specifies which leg of the call to broadcast the file on. If not specified, the broadcast only occurs on the a-leg.

Note: The new C functions in the core are documented in the Doxygen.

See Also

- mod_commands