

---

# SIP Trunking Configuration Guide for Asterisk 1.8.7

---

© 2011, Cox Communications, Inc. All rights reserved.

This documentation is the confidential and proprietary intellectual property of Cox Communications, Inc. Any unauthorized use, reproduction, preparation of derivative works, performance, or display of this document, or software represented by this document is strictly prohibited.



## Table of Contents

1	Audience .....	4
2	Introduction .....	4
2.1	tekVizion Labs .....	5
3	SIP Trunking Network Components .....	6
3.1	Hardware Components .....	7
3.2	Software Requirements .....	7
4	Features .....	8
4.1	SIP Registration Method .....	8
4.2	Features Supported .....	8
5	Caveats and Limitations .....	9
6	Configuration .....	10
6.1	Configuration Checklist .....	10
6.2	IP Address Worksheet .....	11
6.3	Asterisk Detailed Configuration Steps .....	11
6.3.1	Configuration Files .....	12
6.3.2	asterisk.conf .....	12
6.3.3	System IP Address .....	12
6.3.4	sip.conf .....	14
6.3.5	SIP Trunk .....	15
6.3.6	SIP Extension .....	17
6.3.7	features.conf .....	18
6.3.8	extensions.conf .....	19
6.3.9	IncomingCalls .....	20
6.3.9.1	Basic Incoming Call .....	20
6.3.9.2	Call Forward NoAnswer & Busy .....	20
6.3.9.3	Call Forward All .....	20
6.3.9.4	Simultaneous Ring Group .....	20
6.3.9.5	Sequential Ring Group .....	21
6.3.9.6	Auto Attendant .....	21
6.3.10	Out Going Calls .....	21
6.3.10.1	Anonymous Calls .....	22
6.3.10.2	Authorization/Account Codes .....	22
6.3.10.3	Parked Calls .....	22
6.3.10.4	Dial Plan .....	22
6.3.10.5	Station to Station Dialing .....	23
6.3.11	Block 900 .....	23
6.3.12	Send CLI .....	23
6.3.13	Calling Line ID Spoofing .....	23
7	Sample conf files .....	24
7.1	asterisk.conf .....	24
7.2	sip.conf .....	24
7.3	features.conf .....	26
7.4	extensions.conf .....	26



## Table of Figures

Figure 1 - Cox Fiber Network .....	4
Figure 2 - SIP Trunk Lab Reference Network.....	6
Figure 3 IP Address of Server.....	13

## Table of Tables

Table 1 – PBX Configuration Steps .....	10
Table 2 – IP Addresses.....	11

# 1 Audience

This document is intended for the SIP trunk customer's technical staff and Value Added Retailer (VAR) having installation and operational responsibilities.

# 2 Introduction

This Configuration Guide describes configuration steps for Cox SIP trunking to an Asterisk IP-PBX. Cox SIP trunking is a scalable and efficient IP trunking telecommunication solution for your business that provides all the traditional services such as Direct Inward Dialing, Hunting, Calling Name, Calling Number, Local/Long Distance and Business Continuity options, including:

- Burstable Trunk Capacity – Dynamically increases call capacity during peak busy periods so your customers never receive a busy signal.
- Call Forward Always – On the trunk group pilot number for all calls in case of an outage (i.e., flood, fire, loss of power, etc.).
- Call Forward Not Reachable – On the trunk group pilot number that operates on a per-call contingency basis to forward the call to any PSTN number (i.e., call center or alternate office location) during temporary call completion impairments.
- Route Exhaustion – Automatic reroute of trunk group calls to any PSTN phone number (i.e., a call center) if calls can't be completed to the PBX.
- Support for geo-redundant PBX deployments and automatic reroute of SIP trunks to the backup customer data center.

All calls are routed over Cox's national fiber network with guaranteed Quality of Service (QoS); calls never traverse the Internet.

## Cox National IP Backbone

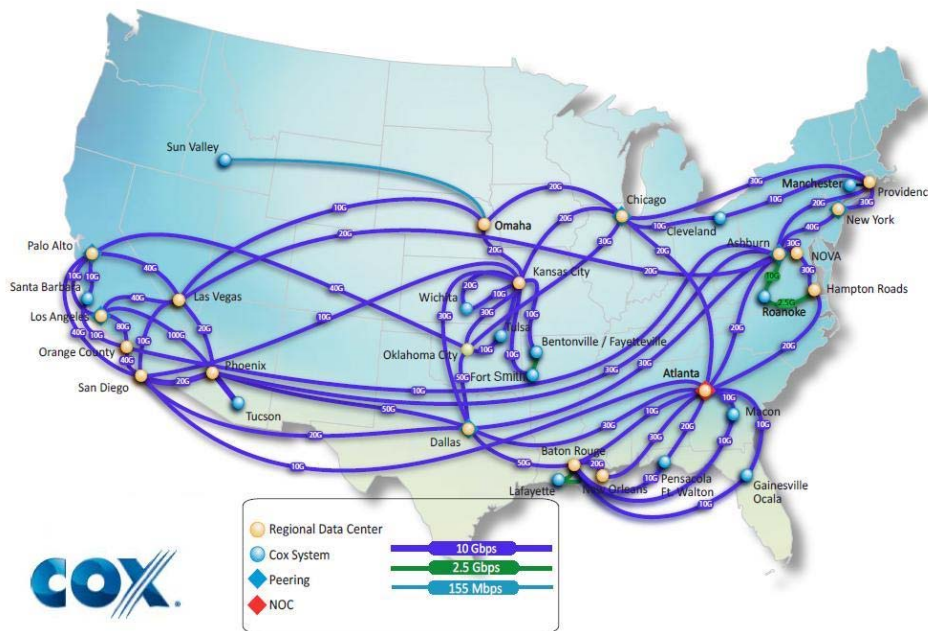


Figure 1 - Cox Fiber Network



## 2.1 tekVizion Labs

tekVizion Labs™ is an independent testing and Verification facility offered by tekVizion PVS, Inc. ("tekVizion"). tekVizion Labs offers several types of testing services including:

- Remote Testing – provides secure, remote access to certain products in tekVizion Labs for pre-Verification and ad hoc testing
- Verification Testing – Verification of interoperability performed on-site at tekVizion Labs between two products or in a multi-vendor configuration ("solution Verification")
- Product Assessment – independent assessment and verification of product functionality, interface usability, assessment of differentiating features as well as suggestions for added functionality, stress and performance testing, etc.

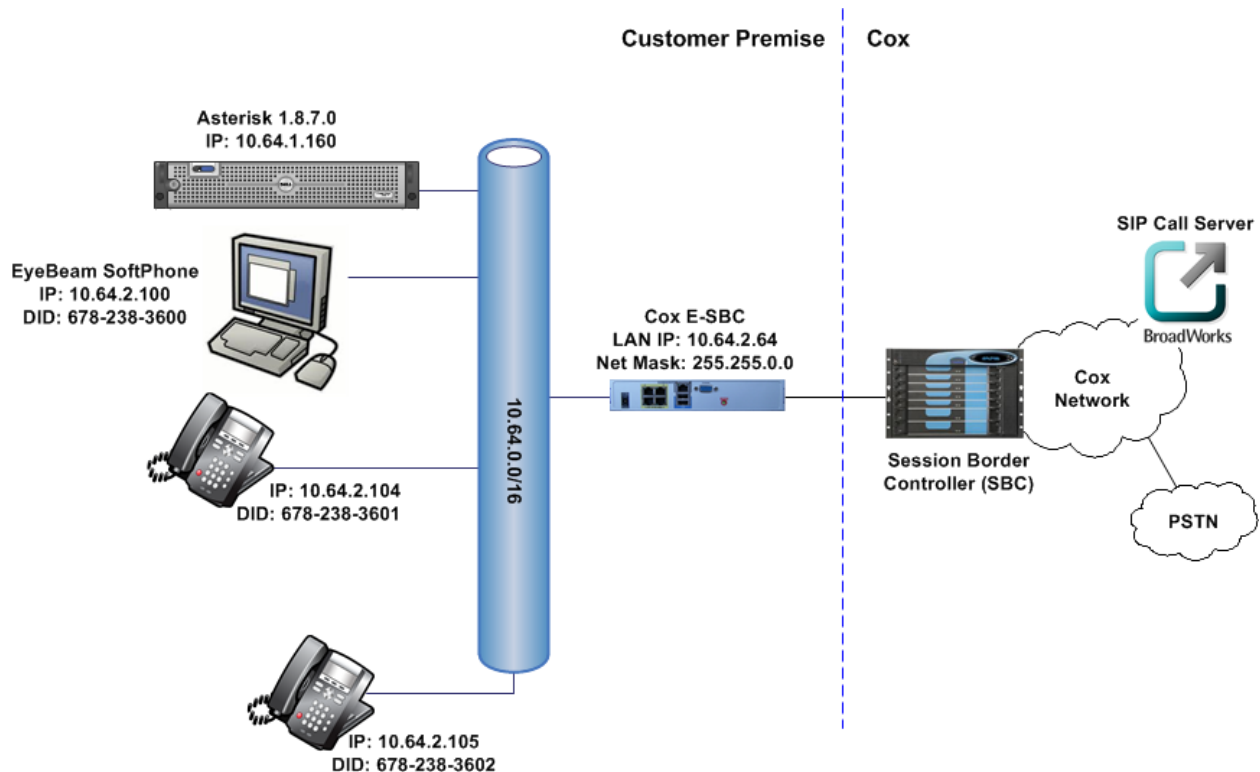
tekVizion is a systems integrator specifically dedicated to the telecommunications industry. Our core services include consulting/solution design, interoperability/Verification testing, integration, custom software development and solution support services. Our services help service providers achieve a smooth transition to packet-voice networks, speeding delivery of integrated services. While we have expertise covering a wide range of technologies, we have extensive experience surrounding our FastForward>> practice areas which include: SIP Trunking, Packet Voice, Service Delivery, and Integrated Services.

The tekVizion team brings together experience from the leading service providers and vendors in telecom. Our unique expertise includes legacy switching services and platforms, and unparalleled product knowledge, interoperability and integration experience on a vast array of VoIP and other next-generation products. We rely on this combined experience to do what we do best: help our clients advance the rollout of services that excite customers and result in new revenues for the bottom line. tekVizion leverages this real-world, multi-vendor integration and test experience and proven processes to offer services to vendors, network operators, enhanced service providers, large enterprises and other professional services firms. tekVizion's headquarters, along with a state-of-the-art test lab and Executive Briefing Center, is located in the Telecom Corridor® in Richardson, Texas.

*(For more information on tekVizion and its practice areas, please visit tekVizion Labs's web site at [www.tekVizionlabs.com](http://www.tekVizionlabs.com).)*

### 3 SIP Trunking Network Components

The network for the SIP trunk reference configuration is illustrated below and is representative of an Asterisk configuration.



**Figure 2 - SIP Trunk Lab Reference Network**

**Note:** Asterisk does not offer DHCP server for dynamic IP address assignment for the SIP phones; however, the Cox Enterprise Session Border Controller (E-SBC) requires a static LAN IP address that must be manually assigned by the LAN network administrator. The DHCP server is provisioned on the Ethernet switch. The DHCP's IP address pool is constrained so that the E-SBC can be assigned an IP address outside of the pool.

The lab network consists of the following components:

- Asterisk IP-PBX for voice features, SIP proxy and SIP trunk termination.
- Various SIP phones on the local LAN.
- The Cox E-SBC is the Edgewater Networks ([www.edgewaternetworks.com](http://www.edgewaternetworks.com)) EdgeMarc appliance. The EdgeMarc is the service demarcation point between customer's LAN network and Cox's WAN network and provides firewall/NAT traversal, B2BUA and SIP Application-level gateway. The EdgeMarc has diverse routes to a primary and secondary Acme SBC.
- Acme Packet Net-Net 9200 Session Border Controllers (SBC).



### **3.1 Hardware Components**

- Server to Install Asterisk
- EdgeMarc 4550 E-SBC

### **3.2 Software Requirements**

- Asterisk 1.8.7
- EdgeMarc 4550 9.12.5 Release



## 4 Features

---

### 4.1 SIP Registration Method

Cox Network requires SIP REGISTER support to allow the IP-PBX to originate calls from the IP-PBX and to send calls to the PBX from the PSTN. Asterisk supports SIP Register with authentication. Cox implementation team provides the Pilot number and the authentication key, which should be provisioned in Asterisk. How to configure these in Asterisk is shown in [Section 6.3.5](#)

### 4.2 Features Supported

- Basic calls using G.711ulaw
- Calling Party Number Presentation
- Anonymous call
- Call Transfer
- Call Forwarding
- Call Hold and Resume
- Call Pickup
- Call Waiting
- DND
- Call Park
- Hunt groups (Simultaneous and Sequential Ring)
- Three-Way Calling
- PBX Auto Attendant to Off-net Numbers
- PBX Account Codes
- PBX Authorization Codes
- E911 Call
- RFC2833 transcoding
- PBX-Defined Caller ID (spoofing)11





## 5 Caveats and Limitations

---

- Call Pickup. There seems to be a race condition that causes the Asterisk IP-PBX to send out an UPDATE message that is then challenged by Cox Network with a 401 unauthorized. Asterisk IP-PBX does not respond to the challenge. The concern is that the call may be torn down if the 401 unauthorized is not responded to. At the time of testing the call was not disconnected. The call was terminated after 7 minutes.
- Incoming Call to IVR shows the same behavior as Call Pickup occasionally.
- Asterisk 1.8.7.0 hairpins both call legs during call transfer and call forwards, meaning the SIP sessions are not released after transfer. The sessions are released when the calls are released.
- Asterisk supports FAX. Fax is not tested ATA not available for analog port.
- Modem call not tested. Same reason as FAX.



## 6 Configuration

---

### 6.1 Configuration Checklist

In this section we present an overview of the steps that are required to configure Asterisk 1.8.7.0 for SIP Trunking as well as all features that were tested. PBX features including Call Forwarding, Simultaneous Ring, Sequential Ring, Auto Attendant, etc. are not required to configure the Cox SIP Trunking service.

**Table 1 – PBX Configuration Steps**

Step	Description	Reference
Step 1	System IP Address	<a href="#">Section 6.3.3</a>
Step 2	Configure SIP Trunk	<a href="#">Section 6.3.5</a>
Step 3	Configure SIP Extensions	<a href="#">Section 6.3.6</a>
Step 4	Incoming Calls	<a href="#">Section 6.3.9</a>
Step 5	Basic Incoming Call	<a href="#">Section 6.3.9.1</a>
Step 6	Call Forward No Answer and Busy	<a href="#">Section 6.3.9.2</a>
Step 7	Call Forward All	<a href="#">Section 6.3.9.3</a>
Step 8	Simultaneous Ring Group	<a href="#">Section 6.3.9.4</a>
Step 9	Sequential Ring Group	<a href="#">Section 6.3.9.5</a>
Step 10	Auto Attendant	<a href="#">Section 6.3.9.6</a>
Step 5	Out Going Calls	<a href="#">Section 6.3.10</a>
Step 12	Anonymous Calls	<a href="#">Section 6.3.10.1</a>
Step 13	Authorization/Account Codes	<a href="#">Section 6.3.10.2</a>
Step 14	Parked Calls	<a href="#">Section 6.3.10.3</a>
Step 15	Dial Plan	<a href="#">Section 6.3.10.4</a>
Step 16	Station to Station Dialing	<a href="#">Section 6.3.10.5</a>
Step 17	Block 900 numbers	<a href="#">Section 6.3.11</a>
Step 18	Sample asterisk.conf	<a href="#">Section 7.1</a>
Step 19	Sample sip.conf	<a href="#">Section 7.2</a>
Step 20	Sample features.conf	<a href="#">Section 7.3</a>
Step 21	Sample extensions.conf	<a href="#">Section 7.4</a>



## 6.2 IP Address Worksheet

The specific values listed in the table below and in subsequent sections are used in the lab configuration described in this document, and are for **illustrative purposes only**. The customer must obtain and use the values for your deployment. **<modify this table as required list out all IP address with explanation of each>**

**Table 2 – IP Addresses**

Component	Cox Lab Value	Customer Value
<b>EdgeMarc E-SBC</b>		
• LAN IP Address	10.64.2.64	
• LAN Subnet Mask	255.255.0.0	
<b>Asterisk 1.8.7.0 IP PBX</b>		
<ul style="list-style-type: none"> <li>• System IP Address</li> </ul> <p>The Internet Connection will typically be on the same subnet as the LAN IP Address of the E-SBC. If this is not the case, then Layer 3 routing must be in place.</p>	10.64.1.160	
<ul style="list-style-type: none"> <li>• Default Gateway</li> </ul> <p>The Default Gateway must be the LAN Network default Gateway. This will allow the administrator to log in via his/her workstation if the workstation is on a different network</p>	10.64.1.1	

## 6.3 Asterisk Detailed Configuration Steps

Equipment used for configuration setup:

- Asterisk software version release 1.8.7



### 6.3.1 Configuration Files

In Asterisk 1.8.7.0 there are 126 configuration files that are named <filename.conf> . In each of the .conf files a ; is used to document useful information about each parameter. This document only covers the .conf files that were modified for this test. These files are listed below.

- o [asterisk.conf](#)
- o [sip.conf](#)
- o [features.conf](#)
- o [extensions.conf](#)

All parameters that are commented out will be removed from its prospective .conf file for documentation purposes.

### 6.3.2 asterisk.conf

Asterisk.conf is the configuration file for system parameters. The parameters that are used in asterisk.conf are shown below.

```
[directories](!)1
astetcdir => /etc/asterisk
astmoddir => /usr/lib/asterisk/modules
astvarlibdir => /var/lib/asterisk
astdbdir => /var/lib/asterisk
astkeydir => /var/lib/asterisk
astdatadir => /var/lib/asterisk
astagidir => /var/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk
astrundir => /var/run/asterisk
astlogdir => /var/log/asterisk
```

```
[options]2
documentation_language = en_US
```

```
[compat]3
pbx_realtime=1.6
res_agi=1.6
app_set=1.6
```

### 6.3.3 System IP Address

The IP address of the server that Asterisk 1.8.7.0 is installed on is 10.64.1.160 with a subnet mask of 255.255.0.0. Asterisk is typically installed on a Linux based Operating System. Please refer to <http://www.asterisk.org/> for installation information. In this example Asterisk is installed on a server running Cent OS. Follow the commands below to acquire the IP Address of the server which is used by the E-SBC for SIP Trunking Devices and the Default Dial Rules.

.

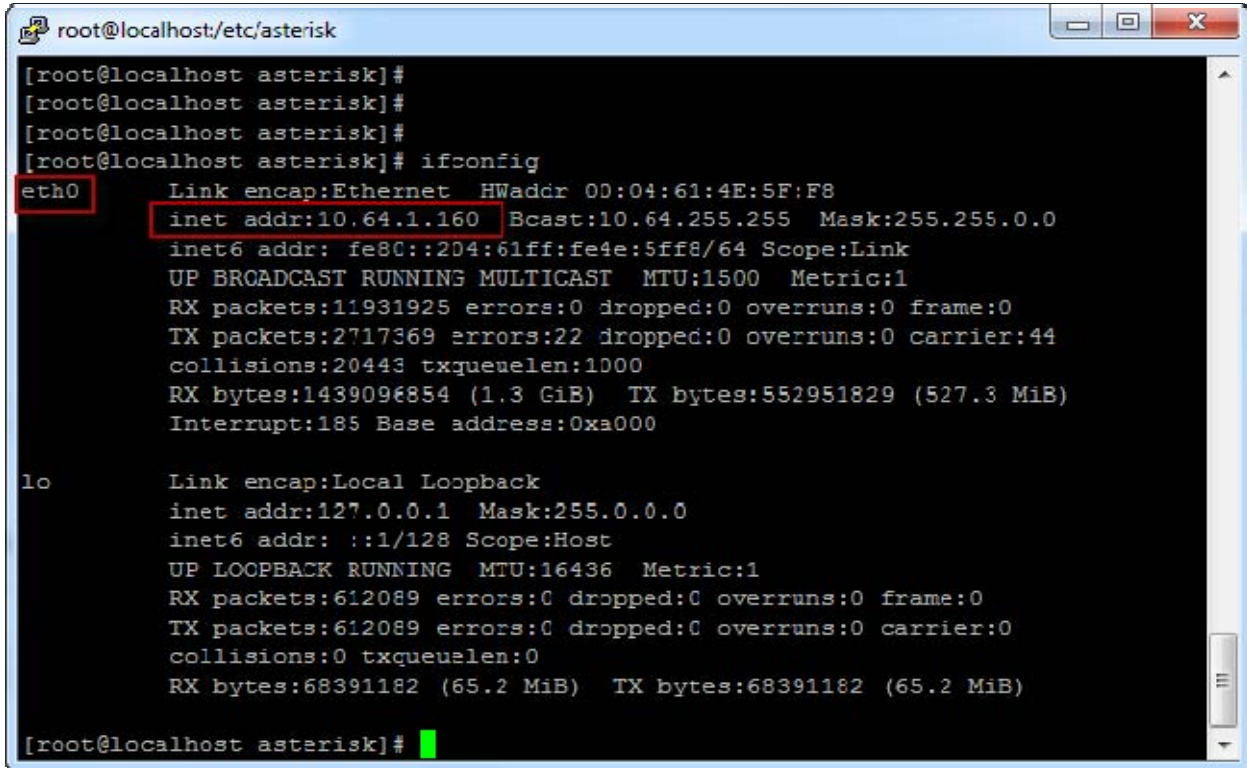
---

<sup>1</sup> There are 10 parameters below directories that define the directory paths used.

<sup>2</sup> There are many different parameters below options that are commented out. The only parameter that must be defined is documentation\_language

<sup>3</sup> The context compat is an entry that defines backward compatibility.

1. Open a terminal window
2. Type ifconfig.



```
root@localhost/etc/asterisk
[root@localhost asterisk]#
[root@localhost asterisk]#
[root@localhost asterisk]#
[root@localhost asterisk]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:04:61:4E:5F:F8
          inet addr:10.64.1.160  Bcast:10.64.255.255  Mask:255.255.0.0
          inet6 addr: fe80::204:61ff:fe4e:5ff8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11931925 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2717369 errors:22 dropped:0 overruns:0 carrier:44
          collisions:20443 txqueuelen:1000
          RX bytes:1439096854 (1.3 GiB)  TX bytes:552951829 (527.3 MiB)
          Interrupt:185 Base address:0xa000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:612089 errors:0 dropped:0 overruns:0 frame:0
          TX packets:612089 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:68391182 (65.2 MiB)  TX bytes:68391182 (65.2 MiB)

[root@localhost asterisk]#
```

Figure 3 IP Address of Server

### 6.3.4 sip.conf

The configuration file sip.conf covers the configuration of SIP trunks and SIP endpoints. Creating a SIP trunk is covered [Section 6.3.5](#). Creating extensions is covered in [Section 6.3.6](#). The general section of the file is addressed below.

```
[general]1  
context=default2  
allowoverlap=no3  
udpbindaddr=0.0.0.04  
tcpenable=no5  
tcpbindaddr=0.0.0.06  
srvlookup=yes7  
tos_sip=cs38  
tos_audio=ef9  
relaxdtmf=yes10  
trustpid = no11  
sendrpid = yes12  
sendrpid = pai13
```

---

<sup>1</sup> The general context is where the parameters for SIP are contained.

<sup>2</sup> Set context to default. This is the default context for incoming calls...In [fn19](#) the context is set to cox\_in. This will override the default context for incoming calls.

<sup>3</sup> Set allowoverlap to no. Disable overlap dialing support.

<sup>4</sup> Set udpbindaddr to the LAN Ip Address of the E-SBC as defined in [Figure 2 Table 2](#). For this example 0.0.0.0 is used.

<sup>5</sup> Set tcpenable to no. By setting tcpenable to no UDP is used

<sup>6</sup> Set tcpbindaddr to 0.0.0.0 With [fn5](#) set to no [fn6](#) is not used.

<sup>7</sup> Set srvlookup to yes. This field is set to yes by default and was not changed during this test cycle. srvlookup is not required for this configuration.

<sup>8</sup> Set tos\_sip to cs3. This is default and is not verified for this test

<sup>9</sup> Set tos\_audio to ef. This is default and is not verified for this test

<sup>10</sup> Set relaxdtmf to yes. Relax dtmf handling

<sup>11</sup> Set trustpid to no. Set to yes if Remote Party ID should be trusted

<sup>12</sup> Set sendrpid to yes. Set to yes

<sup>13</sup> Set sendrpid to pai. Use the "P-Asserted-Identity" header



### 6.3.5 SIP Trunk

To create a SIP trunk in Asterisk the sip.conf file is used. In sip.conf sip extensions are created as well.

1. The line below sets up the registration in the sip.conf file

```
register => 67823836001:67823836002@10.64.2.643/67823836004~36005
```

---

<sup>1</sup> Username The actual SIP Registration Username will be provided by your Cox Account Representative and must be kept confidential! The Trunk Group Pilot Number (Username) that is used here is 6782383600 and is for illustration purposes only!

<sup>2</sup> Password The actual SIP Registration Password will be provided by your Cox Account Representative and must be kept confidential! The Trunk Group Pilot password that is used here is 6782383600 and is for illustration purposes only!

<sup>3</sup> This is the static LAN IP address of the Cox E-SBX. Please use the actual E-SBC LAN IP for your network. The IP Address used in this configuration is 10.64.2.64 Note: Remember that the E-SBC LAN IP address may/will be different from this example. Please see **Figure 2** and **Table 2** for the IP address scheme that is used in this example.

<sup>4</sup> This field must match **fn1**

<sup>5</sup> This field sets expires timer for the registration.



2. Set the trunk context of 6782383600 that was defined in **fn4** of this section.

```
[6782383600]1
type = friend2
username = 67823836003
fromdomain = 10.64.2.644
realm = 10.64.2.645
host = 10.64.2.646
dtmfmode = rfc28337
secret = 67823836008
nat = yes9
canreinvite = no10
insecure = invite11
qualify = yes12
disallow = all13
allow = ulaw14
allow = alaw
allow = g729
amaflags = default15
trustpid = no16
sendrpid = yes17
sendrpid = pai18
context = cox_in19
```

---

<sup>1</sup> This field must match **fn4**

<sup>2</sup> Set type to friend

<sup>3</sup> Set username to the value set in **fn1**

<sup>4</sup> Set fromdomain to the value set in **fn3**

<sup>5</sup> Set realm to the value set in **fn3**

<sup>6</sup> Set host to the value set in **fn3**

<sup>7</sup> Set dtmfmode to rfc2833

<sup>8</sup> Set secret to the value set in **fn2**

<sup>9</sup> Set nat to yes

<sup>10</sup> Set canreinvite to no

<sup>11</sup> Set insecure to invite

<sup>12</sup> Set qualify to yes

<sup>13</sup> Set disallow to all

<sup>14</sup> Set allow for each codec. In this example there are 3 lines ulaw, alaw and g729. Only use g729 if you have licenses for this codec.

<sup>15</sup> Set amaflags to default

<sup>16</sup> Set trustpid to no

<sup>17</sup> Set sendrpid to yes

<sup>18</sup> Set sendrpid to pai

<sup>19</sup> Set context to cox\_in. This will be used in extensions.conf file for incoming calls.





### 6.3.6 SIP Extension

The sip.conf file is used to define sip extensions. In the default sip.conf file there are several different types of sip phones defined.

1. Set the phone context of 2000

```
[2000]1
type=friend2
regexten=12343
callcounter=yes4
call-limit=25
context=testDialPlan6
busylevel=27
secret=12348
callerid="Jane Smith" <6782383600>9
host=dynamic10
callgroup=111
pickupgroup=112
mailbox=2000@default13
```

---

<sup>1</sup> Set the Phone context to 2000. This is the extension number of the phone to be configured

<sup>2</sup> Set type to friend

<sup>3</sup> Set regexten to 1234. This value will match what is in the phone trying to register.

<sup>4</sup> Set callcounter to yes

<sup>5</sup> Set call-limit to 2. This is how many calls the phone can receive

<sup>6</sup> Set context=testDialPlan. If context is omitted from the configuration of the sip extension it uses default context. For this testing context is omitted and default context is used. If context is specified as in this example then testDialPlan must be defined in extensions.conf. This would be used for example, if a phone needs to be marked as anonymous.

<sup>7</sup> Set the busylevel to 2. When the 2<sup>nd</sup> call is received the call is marked as busy.

<sup>8</sup> Set secret to 1234. This value will match what is in the phone trying to register.

<sup>9</sup> Set callerid to "Jane Smith" <6782383600>. Set the calling name value between "calling name". Set the callerid value between the < callerid number >

<sup>10</sup> Set host to dynamic

<sup>11</sup> Set callgroup to 1. This places the extension being configured in callgroup 1.

<sup>12</sup> Set pickupgroup to 1. This places the extension being configured in pickupgroup 1

<sup>13</sup> Set mailbox to 2000@default. This sets up the voice mail box for extension 2000



### 6.3.7 features.conf

features.conf is used to set up call park and blind transfer configuration parameters. There may be other features that can be configured by using features.conf.

```
[general]1
parkext => 7002
parkpos => 701-7203
context => parkedcalls4)
parkinghints = yes5
parkingtime => 456
comebacktoorigin = yes7
parkedcalltransfers = caller8
findslot => next9
parkedmusicclass=default10
xfersound = beep11
xferfailsound = beeperr12
pickupexten = *813
[featuremap]14
blindxfer => #115
parkcall => #7216
[applicationmap]17
```

---

<sup>1</sup> The general context is where the parameters for features are contained

<sup>2</sup> Set parkext to 700. This is the extension used when the extension places the call to the park queue, by transferring caller to extension 700. To enable the extension to park the call, the extensions dial plan must have include => parkedcalls as shown in [Section 6.3.10.3](#).

<sup>3</sup> Set parkpos to 701 -720. This range is used for this example. This provides the number of possible parked calls

<sup>4</sup> Set context to parked calls. This sets which context parked calls are in.

<sup>5</sup> Set parkinghints to yes. Add hints priorities automaticall for parking slots.

<sup>6</sup> Set parkingtime to 45. This value is set in seconds. This is the amount of time that the call will stay parked before recalling the originating station. If [fn6](#) is set to yes.

<sup>7</sup> Set comebacktoorigin to yes. If this value is set to yes then when the timer in FootNote58 times out the caller goes to the original party that placed the call on Park.

<sup>8</sup> Set parkedcalltransfers to caller. Enables or disables DTMF based transfers when picking up a parked call

<sup>9</sup> Set findslot to next. This will allow the extension that parks the call to automatically select the next parking slot

<sup>10</sup> Set parkedmusicclass to default. This is the MOH class to use for the parked channel.

<sup>11</sup> Set xfersound to beep. This is used to indicate an attended transfer is complete.

<sup>12</sup> Set xferfailsound to beeper. This is used to indicate a failed transfer.

<sup>13</sup> Set pickupexten to \*8. This is the default dial code used for call pickup as long as the extensions are in the same group as [fn12](#)

<sup>14</sup> The featuremap context is defined for phone features.

<sup>15</sup> Set blindxfer to #1. This is part of the file as tested.

<sup>16</sup> Set parkcall to #72. This is part of the file as tested.

<sup>17</sup> The applicationmap context is part of the file as tested.



### 6.3.8 extensions.conf

The configuration file extensions.conf covers the configuration of Incoming and outgoing Dial Plans. Configuration of incoming dial plan includes the way extensions behave for different types of features. These features include Ring No Answer, Call Forward Busy, Call Forward All, Huntgroups etc...

```
[general]1  
static=yes2  
writeprotect=no3  
clearglobalvars=no4  
[globals]5
```

---

<sup>1</sup> The general section is used for certain variables. The variables defined in fn [2,3,4](#) and [5](#) are the ones used for this configuration example.

<sup>2</sup> Set static to yes. If static is set to yes and writeprotect is equal to no, you can save dialplan by CLI command dialplan save. If static is set to no, or omitted, then the pbx\_config will rewrite this file when extensions are modified. Remember that all comments made in the file will be lost when that happens.

<sup>3</sup> Set writeprotect to no.

<sup>4</sup> Set clearglobalvars to no. If clearglobalvars is not set, then global variables will persist through reloads, and even if deleted from the extension.conf or one of its included files, will remain set to the previous value.

<sup>5</sup> The globals section contains global variables that can be referenced in the dial plan with the global dialplan function. For this configuration example this is not used.



### 6.3.9 IncomingCalls

To configure incoming calls the extensions.conf file is used.

[cox\_in]<sup>1</sup>

#### 6.3.9.1 Basic Incoming Call

exten =><sup>2</sup> \_6782383601<sup>3,14</sup>,Dial(SIP/2001<sup>5</sup>,30<sup>6</sup>,t<sup>7</sup>,r<sup>8</sup>)

#### 6.3.9.2 Call Forward NoAnswer & Busy

exten => \_6782383602,1,Dial(SIP/2002,30,t,r)<sup>9</sup>  
exten => \_6782383602,2,NoOp(Call Status \${DIALSTATUS})<sup>10</sup>  
exten => \_6782383602,3,GoToIf(\$["\${DIALSTATUS}"="NOANSWER"]<sup>11</sup>?5<sup>12</sup>)  
exten => \_6782383602,4,GoToIf(\$["\${DIALSTATUS}"="BUSY"]<sup>13</sup>?6<sup>14</sup>)  
exten => \_6782383602,5,Dial(SIP/6782344101<sup>15</sup>@6782383600<sup>16</sup>)  
exten => \_6782383602,6,Dial(SIP/6782344101<sup>17</sup>@6782383600)<sup>18</sup>

#### 6.3.9.3 Call Forward All

exten => \_6782383606,1,Dial(SIP/6782344101<sup>19</sup>@6782383600)

#### 6.3.9.4 Simultaneous Ring Group

exten => \_6782383605,1,Dial(SIP/2000&SIP/2001&SIP/2002<sup>20</sup>,30,t,r)

<sup>1</sup> cox\_in is context defined in Section 6.3.5 [fn19](#)

<sup>2</sup> exten => This is the syntax that declares a dialplan rule.

<sup>3</sup> \_6782383601 is the ten digit number in the TO header in the INVITE from the Cox Network

<sup>4</sup> The number 1 represents the rule order number. This number must be incremented by 1 if there is a matching line below, like the example in Section 6.3.9.2 above.

<sup>5</sup> Dial(SIP/2001 Tells the call to try SIP Extension 2001

<sup>6</sup> 30 represents the number of seconds that extension 2001 rings. If there is a voicemail number the caller will be forwarded to voice mail.

<sup>7</sup> t Allows the caller the ability to blind transfer to another extension.

<sup>8</sup> r Is responsible for sending ringback.

<sup>9</sup> This is the same configuration as [Section 6.3.9.1](#).

<sup>10</sup> DIALSTATUS Shows call status if the console is being monitored.

<sup>11</sup> NOANSWER is a condition that is acted on with the GoToIf statement. This gives the ability to branch on different conditions.

<sup>12</sup> 5 is a pointer to the rule order number.

<sup>13</sup> BUSY is a condition that is acted on with the GoToIf statement. This gives the ability to branch on different conditions.

<sup>14</sup> 6 is a pointer to the rule order number.

<sup>15</sup> Dial(SIP/6782344101 is the number that is used to forward a NOANSWER status to.

<sup>16</sup> This is the trunk context that is defined in Section 6.3.5 [fn4](#)

<sup>17</sup> Dial(SIP/6782344101 is the number that is used to forward a BUSY status to.

<sup>18</sup> This is the trunk context that is defined in Section 6.3.5 [fn4](#)

<sup>19</sup> Dial(SIP/6782344101 is the PSTN number that will be dialed when Asterisk receives a call for 6782383606

<sup>20</sup> For this example when DID 6782383605 is dialed, extension 2000,2001 and 2002 will ring at the same time for 30 seconds. Once the 30 seconds is over the call is terminated if not answered. If the call should follow a different path after the 30 second timer expires another rule is added as shown in [Section 6.3.9.2](#)



### 6.3.9.5 Sequential Ring Group

```
exten => _6782383600,1,Dial(SIP/2000,10,t,r)1
exten => _6782383600,2,Dial(SIP/2001,15,t,r)2
exten => _6782383600,3,Dial(SIP/2002,30,t,r)3
```

### 6.3.9.6 Auto Attendant

```
exten => _6782383604,1,Answer4
exten => _6782383604,2,Read5(digito6,demo-instruct7,18,,,109)
exten => _6782383604,3,NoOp(${digito})10
exten => _6782383604,4,GoTolf("${digito11" = "1"]?7)12
exten => _6782383604,5,GoTolf("${digito13" = "2"]?8)14
exten => _6782383604,6,Hangup15
exten => _6782383604,7,Dial(SIP/2001,10,t,r)16
exten => _6782383604,8,Dial(SIP/6782344101@6782383600,10,t,r)17
```

## 6.3.10 Out Going Calls

To configure outgoing calls the extensions.conf file is used. In [Section 6.3.6](#) context testDialPlan **fn6** is discussed as well as the context of default.

[testDialPlan]<sup>18</sup>

```
exten => _XXXXXXXXXX19,1,Set(CALLERID(all)="UNKNOWN")20
exten => _XXXXXXXXXX,2,Dial(SIP/${EXTEN}21@6782383600)
```

[default]<sup>22</sup>

<sup>1</sup> 6782383600 is the pilot number for Sequential Ring Group. Extension 2000 is dialed first and rings for 10 seconds.

<sup>2</sup> Extension 2001 is dialed second and rings for 15 seconds.

<sup>3</sup> Extension 2002 is dialed third and rings for 30 seconds. If there is no answer the call is dropped.

<sup>4</sup> 6782383604 is the pilot number for the Auto Attendant (AA). The call is answered with Answer.

<sup>5</sup> Read is a function that inspects the contents of the variable digito.

<sup>6</sup> digito is the variable that the users entry is stored in.

<sup>7</sup> demo-instruct is the file that is being played for purposes of this explanation.

<sup>8</sup> 1 is collect one digit. The four commas must be in place.

<sup>9</sup> Wait for 10 seconds to collect the digit.

<sup>10</sup> Displays the contents of the variable digito

<sup>11</sup> digito is compared to the value 1

<sup>12</sup> GoTolf gives a rule the ability to branch on different conditions. If digito equal 1 then go to rule 7.

<sup>13</sup> digito is compared to the value 2

<sup>14</sup> GoTolf gives a rule the ability to branch on different conditions. If digito equal 2 then go to rule 8

<sup>15</sup> If digito does not match either 1 or 2 then Asterisk terminates the call based on the configuration.

<sup>16</sup> Rule 7 is to dial extension 2001

<sup>17</sup> Rule 8 is to dial PSTN 6782344101

<sup>18</sup> testDialPlan is the context defined in [Section 6.3.6 fn6](#). For this example the testDial Plan is set up for extension 2000 to place Anonymous calls to the PSTN.

<sup>19</sup> The 10 X's represent any 10 digits 0 – 9 in value.

<sup>20</sup> Set(CALLERID(all)="UNKNOWN"). Sets the caller id to UNKNOWN only when testDialPlan is invoked.

<sup>21</sup> Dial(SIP/\${EXTEN}. Dials the 10 digit number that was dialed.

<sup>22</sup> default Is the default context that is referenced in [Section 6.3.6 fn6](#)



### 6.3.10.1 Anonymous Calls

Please reference [Section 6.3.10](#) fn <sup>19 20 21</sup> for information on the dial rules below. The example given below configures a 10 digit PSTN call to be anonymous. Anonymous call feature is configured on a per dial plan basis per call type.

```
exten => _XXXXXXXXXX,1,Set(CALLERID(all)="UNKNOWN")
exten => _XXXXXXXXXX,2,Dial(SIP/${EXTEN}@6782383600)
```

### 6.3.10.2 Authorization/Account Codes

In this example configuration Authorization/Account Codes feature is configured on a per dial plan basis per call type. In this example the Authorization/Account Code is configured on a 10 digit call.

```
exten => _XXXXXXXXXX1,1,Authenticate2(11113,a4)
exten => _XXXXXXXXXX,2,Dial(SIP/${EXTEN}@6782383600)5
```

### 6.3.10.3 Parked Calls

This value comes from the context configured in features.conf for parked calls. This value must be present in the extensions dial plan to allow parked calls. Please reference [Section 6.3.7 fn4](#).

```
include => parkedcalls
```

### 6.3.10.4 Dial Plan

The dial rules below are the rules required for an extension using the default dial plan to place the following call types 0, 00, X11 service numbers and emergency 911, Local 10digit, Long Distance and Toll Free 1+10digit, , Casual Dial 101+5269+1+10digits and 011 International Numbers

```
exten => _0,1,Dial(SIP/${EXTEN}@6782383600)6
exten => _00,1,Dial(SIP/${EXTEN}@6782383600)7
exten => _X11,1,Dial(SIP/${EXTEN}@6782383600)8
exten => _XXXXXXXXXX,1,Dial(SIP/${EXTEN}@6782383600)9
exten => _1XXXXXXXXXX,1,Dial(SIP/${EXTEN}@6782383600)10
exten => _10152691XXXXXXXXXX,1,Dial(SIP/${EXTEN}@6782383600)11
exten => _01XXXXX.,1,Dial(SIP/${EXTEN}@6782383600)12
```

<sup>1</sup> This pattern matches 10 digits where each digit can have the value of 0-9.

<sup>2</sup> Authenticate is an application that matches the code in fn3 below.

<sup>3</sup> 1111 is a four digit code that is used in this example. This code can be any 4 digits as long as it does not conflict with an existing dial plan.

<sup>4</sup> a will set a code to the password. The variable \$(ACCOUNTCODE) will be set.

<sup>5</sup> If user inputs the correct four digit code the 10 digit call will be completed.

<sup>6</sup> This is a dial rule for an Operator Assisted call.

<sup>7</sup> This is a dial rule for an International Operator Assisted call.

<sup>8</sup> This is a dial rule for Service Numbers and Emergency Number like 911.

<sup>9</sup> This is a dial rule for Local 10 digit numbers.

<sup>10</sup> This is a dial rule for Long Distance numbers as well as 1+10 toll free numbers.

<sup>11</sup> This is a dial rule for Casual Dialing the 4 digit CAC 5269

<sup>12</sup> This is a dial rule for International Numbers.



### 6.3.10.5 Station to Station Dialing

For this example Asterisk's has three SIP extensions. They are 2000, 2001 and 2002. The dial rules below must be in place to allow the extension to be called.

```
exten => 2000,1,Dial(SIP/${EXTEN})
exten => 2001,1,Dial(SIP/${EXTEN})
exten => 2002,1,Dial(SIP/${EXTEN})
```

### 6.3.11 Block 900

```
exten => _1900XXXXXXX,1,Noop(Call from ${EXTEN})1
exten => _1900XXXXXXX,2,Hangup2
```

### 6.3.12 Send CLI

The Calling Line ID (CLID) is configured to send phone user's 10-digit DID number. The Calling Line ID is set when configuring the SIP extension. Please refer to [Section 6.3.6 fn9](#) for explanation.

```
callerid="Jane Smith" <6782383600>
```

### 6.3.13 Calling Line ID Spoofing

Calling ID Spoofing is the practice of sending a CLID that is not within the Cox assigned DID telephone number range(s). A legitimate use of CLID spoofing is the PBX customer wants to display their toll free number on non-emergency PBX originated calls such the dialed party can call the company back. To enable CLID spoofing support, the PBX customer must:

1. Contact your Cox sales representative to request spoofed CLID support on the Cox network, otherwise Cox will block calls that present a spoofed CLID.
2. Enter the toll free or other telephone number instead of the 10-digit DID number to be displayed for each extension as described in Section 6.3.11 above.
3. For this example 8006781234 is used instead of 6782383600.

```
callerid="Jane Smith" <8006781234>
```

---

<sup>1</sup> If the dial plan receives 1900 and any 7 digits do nothing.

<sup>2</sup> For this example the 2<sup>nd</sup> option is to Hangup on the originating extension.



## 7 Sample conf files

---

### 7.1 asterisk.conf

The asterisk.conf file that is shown below has all of the default comments removed.

```
[directories](!)
astetcdir => /etc/asterisk
astmoddir => /usr/lib/asterisk/modules
astvarlibdir => /var/lib/asterisk
astdbdir => /var/lib/asterisk
astkeydir => /var/lib/asterisk
astdatadir => /var/lib/asterisk
astagidir => /var/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk
astrundir => /var/run/asterisk
astlogdir => /var/log/asterisk
```

```
[options]
documentation_language = en_US
```

```
[compat]
pbx_realtime=1.6
res_agi=1.6
app_set=1.6
```

### 7.2 sip.conf

The sip.conf file that is shown below has all of the default comments removed.

```
[general]
context=default
allowoverlap=no
udpbindaddr=0.0.0.0
tcpenable=no
tcpbindaddr=0.0.0.0
srvlookup=yes
tos_sip=cs3
tos_audio=ef
relaxdtmf=yes
trustpid = no
sendrpid = yes
sendrpid = pai
```

```
register => 6782383600:6782383600@10.64.2.64/6782383600~3600
```

```
[authentication]
[6782383600]
type = friend
username = 6782383600
fromdomain = 10.64.2.64
```





realm = 10.64.2.64  
host = 10.64.2.64  
dtmfmode = rfc2833  
secret = 6782383600  
nat = yes  
canreinvite = no  
insecure = invite  
qualify = yes  
disallow = all  
allow = ulaw  
allow = alaw  
allow = g729  
amaflags = default  
trustpid = no  
sendrpid = yes  
sendrpid = pai  
context = cox\_in

[2000]  
type=friend  
regexten=1234  
callcounter=yes  
call-limit=1  
busylevel=1  
secret=1234  
callerid="Jane Smith" <6782383600>  
host=dynamic  
callgroup=1  
pickupgroup=1  
;context=testDialPlan  
mailbox=2000@default

[2001]  
type=friend  
callerid="Don Smith" <6782383601>  
secret=2001  
host=dynamic  
callgroup=1  
pickupgroup=1  
defaultuser=2001  
mailbox=2001@default  
disallow=all  
allow=ulaw  
allow=alaw  
allow=g729  
call-limit=2  
busylevel=1

[2002]  
type=friend  
secret=2002  
host=dynamic



```
callerid="James Smith" <6782383602>
callgroup=1
pickupgroup=1
defaultuser=2002
mailbox=2002@default
```

### 7.3 features.conf

The features.conf file that is shown below has all of the default comments removed.

```
[general]
parkext => 700
parkpos => 701-720
context => parkedcalls
parkinghints = yes
parkingtime => 45
comebacktoorigin = yes
parkedcalltransfers = caller
findslot => next
parkedmusicclass=default
xfersound = beep
xferfailsound = beeper
pickupexten = *8
```

```
[featuremap]
blindxfer => #1
atxfer => *2
parkcall => #72
```

```
[applicationmap]
```

### 7.4 extensions.conf

The features.conf file that is shown below has all of the default comments removed.

```
[general]
static=yes
writeprotect=no
clearglobalvars=no
```

```
[globals]
```

```
[cox_in]
```

```
exten => _6782383601,1,Dial(SIP/2001,10,t,r)
exten => _6782383601,2,voicemail(2001,u)
```



```
exten => _6782383602,1,Dial(SIP/2002,30,t,r)
exten => _6782383602,2,NoOp(Call Status ${DIALSTATUS})
exten => _6782383602,3,GoTolF("${DIALSTATUS}"="NOANSWER"?5)
exten => _6782383602,4,GoTolF("${DIALSTATUS}"="BUSY"?6)
exten => _6782383602,5,Dial(SIP/6782344101@6782383600)
exten => _6782383602,6,Dial(SIP/6782344101@6782383600)

exten => _6782383606,1,Dial(SIP/6782344101@6782383600)

exten => _6782383605,1,Dial(SIP/2000&SIP/2001&SIP/2002,30,t,r)

exten => _6782383600,1,Dial(SIP/2000,10,t,r)
exten => _6782383600,2,Dial(SIP/2001,15,t,r)
exten => _6782383600,3,Dial(SIP/2002,30,t,r)

exten => _6782383604,1,Answer
exten => _6782383604,2,Read(digito,demo-instruct,1,,10)
exten => _6782383604,3,NoOp(${digito})
exten => _6782383604,4,GoTolF("${digito}" = "1"?7)
exten => _6782383604,5,GoTolF("${digito}" = "2"?8)
exten => _6782383604,6,Hangup
exten => _6782383604,7,Dial(SIP/2001,10,t,r)

exten => _6782383603,1,Dial(SIP/2003,30,t,r)

[testDialPlan]
exten => _XXXXXXXXXX,1,Set(CALLERID(all)="UNKNOWN")
exten => _XXXXXXXXXX,2,Dial(SIP/${EXTEN}@6782383600)

[default]

exten => _0,1,Dial(SIP/${EXTEN}@6782383600)
exten => _00,1,Dial(SIP/${EXTEN}@6782383600)
exten => _X11,1,Dial(SIP/${EXTEN}@6782383600)
exten => _XXXXXXXXXX,1,Dial(SIP/${EXTEN}@6782383600)
exten => _1XXXXXXXXXX,1,Dial(SIP/${EXTEN}@6782383600)
exten => _10152691XXXXXXXXXX,1,Dial(SIP/${EXTEN}@6782383600)
exten => _01XXXXX.,1,Dial(SIP/${EXTEN}@6782383600)
exten => _1900XXXXXXXX,1,Noop(Call from ${EXTEN})
exten => _1900XXXXXXXX,2,Hangup

exten => 2000,1,Dial(SIP/${EXTEN})
exten => 2001,1,Dial(SIP/${EXTEN})
exten => 2002,1,Dial(SIP/${EXTEN})

include => parkedcalls

exten => 8500,1,VoicemailMain
exten => 8500,n,Goto(s,6)
```