

Configure Asterisk with Cisco IP Phones

With VoIP technology growing so rapidly, the marketplace so too has an ever growing selection of hardware options available for system implementations. The Cisco IP Phone series has solidified its position in the community as the leader of high quality and reliable hardware with feature rich firmware and a good range of supported protocols. Superb call quality, a large easy-to-read display screen and user-friendly button layouts/designs make the Cisco phone models of old and late some of the best hardware available for IP-based phone systems. However, for all their outstanding advantages, configuring these phones to work with a non-Cisco managed VoIP network can be a bit of a chore. To further complicate this task, the information available on configuring the newer and more user-preferred devices (touch screens, color backlit display, etc.) is extremely sparse and largely unconfirmed.

After heading up the deployment of 3 small Asterisk based PBX systems, all of which were to make use of various different Cisco IP Phone models; a great deal of information about these phones has been collected. From that information, the steps to configure the following phone models has been confirmed: 7940, 7960, 7945, 7965, and 7970. A Cisco 7975 was not available for testing, however, the 79X5 models have standardized configuration templates and the information for these units should apply (Still trying to acquire one to confirm this). One of the most valuable and available resources in setting up systems has been peer communities. Both local and online resources will assist in working the kinks out of your system. Also, bookmark voip-info.org. A large number of Asterisk / Cisco configuration issues are identified and often fixed here. I'd also like to give a special thanks to Sean Walberg for suggesting Wireshark (tshark) for helping to debug connection and other TCP/IP issues (along with an immense amount of guidance and other troubleshooting tips) and to Angel Castaneda for always providing helpful suggestions to get me pointed in the right direction. Now, with the introduction out of the way, let's begin.

1) Asterisk

The first component of the system will obviously be the Asterisk IP PBX server. For the sake of this guide I'm going to assume that this has been installed on a server with default settings. If assistance is required in getting an Asterisk server setup or configured with defaults, please consult one of the several sites on Asterisk assistance.

The system that was used for the making of this guide is a turn-key installation package as

distributed by [AsteriskNOW](#). The AsteriskNOW install is a complete installation package available for free that runs off of the [CentOS](#) Linux distribution. It comes pre-configured with a working asterisk installation (with all major / common libraries installed) and the [FreePBX](#) GUI for administering the Asterisk server via a web interface.

For more information on this customized distribution, please see the [AsteriskNOW Homepage](#).

2) TFTP Server

A TFTP (Trivial File Transfer Protocol) Server is not normally required for the setup of an Asterisk IP PBX. The Cisco IP Phones require configuration files downloaded from a TFTP server for their internal settings. We will get into the configuration files later, for now we will just be setting up the TFTP server.

The first step is to connect to the Asterisk server via SSH as root. Once connected to the server, install the TFTP software:

```
yum install tftp-server
```

The TFTP server is an 'xinet' application and therefore xinetd will also need to be installed if it is not already. (AsteriskNOW installs xinetd by default). By default, the folder used by the TFTP server is /tftpboot. Changing this is not necessary.

Once installed run the following commands to ensure that both xinetd and the TFTP server are initialized on server startup:

```
chkconfig xinetd on  
chkconfig tftp on
```

NOTE: The TFTP server for the Cisco IP Phones can be located on a different server if desired. I chose to install it on the same server as Asterisk for 1) simplicity, and 2) to ensure the system was self-contained. If the TFTP is on a different address, ensure that the appropriate IP address is used when referring to the TFTP server for the remainder of the process.

Cisco IP Phones using SIP firmware will use DHCP-option (port) 69 for connecting to the TFTP server and obtaining its configuration. Phones loaded with firmware that makes use of a Cisco Call Manager application (usually SCCP or MGCP) may look for configuration information on port 150 (the TFTP port used by the CCM).

3) Firmware

All Cisco IP Phones have several different types of firmware available. Each type of firmware is programmed to support a different VoIP protocol. The three supported protocols are SCCP or ‘Skinny’, SIP (Session Initiation Protocol), and MGCP. The protocols that are most commonly seen in system deployments are SIP or SCCP.

The firmware that is distributed by Cisco is a licensed software product, and as a result, Cisco requires that a license be purchased and verified before allowing downloads. Although not “legally” permitted other firmware downloads can often be found through creative internet searching.

In addition to the three supported protocols, there are also several different versions of each firmware. In most cases it is recommended that the most recent version of firmware for the needed protocol be used. Older firmware will usually work, but often with limited functionality. A helpful note when phones have issues upgrading their firmware; downgrading the firmware before upgrading will often load the newest versions when direct upgrades fail (eg. Phone currently with SCCP 8.x: if unable to upgrade to SIP 8.x directly, loading SIP 7.x or SCCP 7.x before upgrading to SIP 8.x). Multiple downgrades / upgrades may be needed to load the proper firmware version.

The Cisco IP Phone firmware is organized differently depending on the model of phone being used. XML configured devices (7970, 7945, etc.) can make use an archived firmware file (.cop) or its un-archived version. The un-archived firmware consists of several files with varying extensions. All firmware will have an application loader file that directs the downloading of the software. It is identified by the protocol it provides, the phone model it is for, and the version of its firmware with a .loads file extension (eg. SIP70.8-4-2S.loads: SIP = protocol, 70 = model 7970, version = 8.4.2S). The second format of firmware is for the 7960 / 7940 or non-XML phones. These devices make use of a pair of files for its firmware. The files are named by similar convention; identifying protocol and version. This firmware format is not device specific so the same firmware can be used on both the 7960 and 7940 (eg. POS3-8-12-00.sbn: S = SIP, 8-12-00 = version).

All of the firmware files that are going to be used will need to be placed in the directory being used for the TFTP server (usually /tftpboot). It is recommended that firmware in .cop file format be un-archived and uploaded as its individual files. This can be done by simply renaming the .cop file to a tarball archive (.tar) file and then extracting the files it contains. Once all the necessary firmware for the phones being used has been placed on the TFTP server, configuration files will complete the setup.

4) Global Configuration Files

Anyone setting up an Asterisk PBX with Cisco IP Phones will quickly discover that there are several different models available. In this guide a variety of models will be discussed, all of which have been successfully configured and confirmed as working. To my knowledge, all of the major Cisco IP Phone models are able to work with an Asterisk server in some capacity.

**DISCLAIMER: only confirmed working units will be discussed, ability of other models to work is pure speculation.*

In searching the internet for information on configuring Asterisk with Cisco IP Phones, a great deal of the information available is for the Cisco 7960s and 7940s. These seem to be the most commonly used models with Asterisk IP PBX servers. Both of these models use the same firmware and near identical configuration files. The Cisco 7960 provides 6 programmable lines where the 7940 provides only 2.

The Cisco 7914 is an expansion module available for the 7960 and the 7940 that adds additional line capacity. Currently a 7914 unit is unavailable to test and confirm setup information with. Current information indicates that this should work without

Although seen less commonly, the Cisco 7965 and Cisco 7945 are similar in physical appearance with some more advanced functionality and a full color display screen. Additionally, these models make use of more advanced XML configuration templates instead of text based data. Specifics on these configuration methods will be discussed later.

Finally, the Cisco 7970 is beginning to see some increased popularity among Asterisk deployments. The 7970 offers a full color touch screen display, 5 soft keys (1 more than its predecessors), and 8 programmable lines.

Each group of phone models is slightly different although the main functions provided by all units are the same. It should also be pointed out that the SIP firmware for these phones is NOT perfect. The firmware does have limitations, although new versions seem to have near complete functionality.

Regardless of the model(s) of phones used in the deployment, there will be a few files required for configuration. In general, most Cisco IP Phones will require an application loader (txt) file, a generic configuration file, a dialling plan, a ringer list, and a phone settings file. The application loader file, generic configuration file, dialling plan and ringer list can all be re-used for every phone on the system to simplify deployment. For the sake of this guide, we will assume these files are being re-used.

i. OS79xx.txt

OS79xx.txt is a one line text file that is needed for the application loader during boot of the Cisco 7960 or 7940 phones. The one line of text contained in this file is the filename of the firmware that these phones are to make use for connection. This file is not used for any of the phones that make use of XML configuration files.

ii. XMLDefault.cnf.xml

The XMLDefault.cnf.xml is the base file for global settings for all of the Cisco IP Phones to be used. This file will also contain the general information on firmware to use for the various different IP phone models. An example file is as follows:

```
<Default>
<callManagerGroup>
  <members>
    <member priority="0">
      <callManager>
        <ports>
          <ethernetPhonePort>2000</ethernetPhonePort>
          <mgcpPorts>
            <listen>2427</listen>
            <keepAlive>2428</keepAlive>
          </mgcpPorts>
        </ports>
        <processNodeName></processNodeName>
      </callManager>
    </member>
  </members>
</callManagerGroup>
<loadInformation8 model="IP Phone 7940">POS3-8-12-00</loadInformation8>
<loadInformation7 model="IP Phone 7960">POS3-8-12-00</loadInformation7>
<loadInformation435 model="Cisco 7945">SIP45.8-4-2S</loadInformation435>
<loadInformation436 model="Cisco 7965">SIP45.8-4-2S</loadInformation436>
<loadInformation30006 model="IP Phone 7970">SIP70.8-0-3S</loadInformation30006>
<authenticationURL></authenticationURL>
<directoryURL></directoryURL>
<idleURL></idleURL>
<informationURL></informationURL>
<messagesURL></messagesURL>
<servicesURL></servicesURL>
</Default>
```

The main section of this generic configuration file is its application loader assistance. The various different <loadInformation>...</loadInformation> tags are used by the application loader to determine which firmware each device will download and apply.

iii. SIPDefault.cnf

SIPDefault.cnf is another very important configuration file for specifying global settings. The SIPDefault.cnf file is only used by the older model devices (7960, 7940, etc.). A copy of the file will look as follows:

Firmware to use

image_version: "POS3-8-12-00"

proxy1_address: "xxx.xxx.xxx.xxx"
proxy2_address: "xxx.xxx.xxx.xxx"
proxy3_address: "xxx.xxx.xxx.xxx"
proxy4_address: "xxx.xxx.xxx.xxx"

Proxy Server Port
proxy1_port: "5060"
proxy2_port: "5060"
proxy3_port: "5060"
proxy4_port: "5060"

Asterisk server address(es)
and port(s)

proxy_emergency: ""
proxy_emergency_port: "5060"
proxy_backup: ""
proxy_backup_port: "5060"
outbound_proxy: ""
outbound_proxy_port: "5060"

Advanced and failover
server locations

nat_enable: "0"
nat_address: ""
voip_control_port: "5060"
start_media_port: "16348"
end_media_port: "20134"
nat_received_processing: "1"
dyn_dns_addr_1: ""
dyn_dns_addr_2: ""
dyn_tftp_addr: "xxx.xxx.xxx.xxx"
tftp_cfg_dir: "./"

Firewall, NAT
Traversal, DNS and
TFTP settings

proxy_register: "1"
timer_register_expires: "120"
preferred_codec: "none"
tos_media: "5"
enable_vad: "0"
dial_template: "dialplan"
network_media_type: "auto"
autocomplete: "1"
telnet_level: "2"

Asterisk server communication
and registration settings

cnf_join_enable: "1"
semi_attended_transfer: "0"
call_waiting: "1"
anonymous_call_block: "0"
callerid_blocking: "0"
dnd_control: "0"

Calling features enabling

dtmf_inband: "1"
dtmf_outofband: "avt"
dtmf_db_level: "3"
dtmf_avt_payload: "101"
timer_t1: "500"
timer_t2: "4000"
sip_retx: "10"
sip_invite_retx: "6"
timer_invite_expires: "180"

Quality of Service (QoS) and
SIP protocol settings

sntp_mode: "directedbroadcast"
sntp_server: "xxx.xxx.xxx.xxx"
time_zone: "CST"
time_format_24hr: "1"
dst_offset: "1"
dst_start_month: "April"
dst_start_day: ""
dst_start_day_of_week: "Sun"
dst_start_week_of_month: "1"
dst_start_time: "2"
dst_stop_month: "Nov"
dst_stop_day: "1"

Time and DST settings

```
dst_stop_day_of_week: "Sunday"
dst_stop_week_of_month: ""
dst_stop_time: "2"
dst_auto_adjust: "1"

messages_uri: "*99"
services_url: "http://example.domain.ext/services/menu.xml"
directory_url: "http://example.domain.ext/services/directory.php"
logo_url: "http://example.domain.ext/imagename.bmp"

http_proxy_addr: ""
http_proxy_port: 80
remote_party_id: 0
```

Additional function keys and
logo destinations

The first section of the configuration file is the most basic of items that the devices will need. This includes the firmware to use, the server(s) to connect to, and the port(s) on which to connect. Although not entirely tested, research has shown that the configuration will support at least 6 listed addresses.

The second section regarding failover locations and advanced call handling servers is not required for the phones to work but can be used to configure advanced functionality.

The NAT and firewall settings should be altered to suit the network that the server is on. Common practices will generally leave NAT disabled and only enable it on the phones that will require NAT traversal. This function is enabled in the device specific configurations that will be covered later. A NAT address does not need to be specified. The media ports do not need to change unless the network does not allow the default ports to be used. However, it is recommended that these ports be changed to non-default ports to increase in server security. If dynamic DNS is not being used, its settings should be left blank. It is recommended that the IP address of the TFTP server be entered into the dynamic TFTP location to ensure the devices do not default to their local firewall for the TFTP connection.

For server registration settings, the defaults will work fine in most cases. The main concern in this section is to ensure that `proxy_register` is set to "1". This will require the IP phones to authenticate to the server before being allowed to use it. The `telnet_level` should be changed to suit the systems needs and disabled if not necessary (default of 2 is privileged, requiring a username and password; 1 is openly enabled). This section will also specify the dialplan XML file to use. Enter the name of the XML file that contains the dialplan information without the .xml file extension.

Some access to basic calling features can also be specified in the `SIPDefault.cnf` file. These include things like the Do-Not-Disturb and Call Waiting features. Enabling or disabling these features merely controls the users' access to the feature via their IP phone; the features themselves need to be

enabled in the Asterisk server in order to be used by the phones.

The QoS and SIP protocol settings are used to specify timings and levels for various different internal functions of the Cisco phones. The default settings for these items will work fine for most situations; advanced users may want to change these after consulting the Cisco documentation on them.

In the date and time settings, a time server is for the phones to use. There are several publicly available time servers that can be used for providing the time to the Cisco phones or a local time server can be specified if preferred. Additionally, the `time_zone` and `time_format_24hr` need to be specified. The settings for `time_format_24hr` are '0' for 12hr format display and '1' for 24hr formatting. If the location of the system is in time zone that makes use of daylight savings time, these settings can also be inputted.

The last group of settings is the additional function keys provided on Cisco IP Phones. These are keys for quick access to messages, services and directory listings. These do not need to be specified but do allow for very useful extra functions to be added to the phones (*Another document is planned for explanation of the services XML structure used by Cisco*). In addition, a logo can be provided here as well. The logo must be in BMP file and no bigger than 90x56 pixels. Larger images will be resized as best the software is able. For best results use colors a 4 color palette of #ffffff, #000000, #404040, and #808080.

If a proxy server will be needed for making external HTTP requests, this can be specified as well using the `http_proxy_addr` and `http_proxy_port` variables.

iv. dialplan.xml

The dialplan.xml file is a simple file that has very limited purpose. The XML contained in this file holds information on the timeouts for various different patterns of dialled numbers.

```
<DIALTEMPLATE>
  <TEMPLATE MATCH="*" Timeout="5"/> <!-- Anything else -->
</DIALTEMPLATE>
```

v. RINGLIST.DAT / ringlist.xml

The ringlist file is another file that contains very little data, but is recommended for configuration (even if empty). The ringlist.dat file is responsible for providing the Cisco 7940s and 7960s with knowledge of what ringtones are available to them. These ringtones provided to the phones will also need to be uploaded to the TFTP server. (Ringtone audio files must be in .raw or .pcm file format.)

FlintPhone	FlintPhone.raw
HarpSynth	HarpSynth.raw
Jamaica	Jamaica.raw
Klaxons	Klaxons.raw
KotoEffect	KotoEffect.raw
MusicBox	MusicBox.raw
Ohno	Ohno.raw
Piano 1	Piano1.raw
Piano 2	Piano2.raw

The ringlist.xml file provides the same functionality for the newer XML based Cisco phones. The same file audio files can be used; the structure needs to be altered to the Cisco XML standard. An example ringlist.xml will look similar to the following:

```
<CiscoIPPhoneRingList>
  <Ring>
    <DisplayName>Drums 1</DisplayName>
    <FileName>Drums1.raw</FileName>
  </Ring>
  <Ring>
    <DisplayName>Drums 1</DisplayName>
    <FileName>Drums2.raw</FileName>
  </Ring>
  <Ring>
    <DisplayName>Mr Burns</DisplayName>
    <FileName>mayihelp.raw</FileName>
  </Ring>
  <Ring>
    <DisplayName>Sax 1</DisplayName>
    <FileName>Sax1.raw</FileName>
  </Ring>
  <Ring>
    <DisplayName>Sax 2</DisplayName>
    <FileName>Sax2.raw</FileName>
  </Ring>
</CiscoIPPhoneRingList>
```

vi. List.xml (optional)

An optional configuration file that is only applicable to the XML based phones, List.xml provides the Cisco phones with color screens with background display images. Each <ImageItem> contains two properties, one for the thumbnail location (Image) and the other for the full sized image location (URL). The subfolder in which things are contained IS important. For the Cisco 7975, 7965, and 7945; all images and the List.xml file must be placed in Desktops/320x212x16/ within the TFTP server's root folder. The Cisco 7970 and 7971 will look for this information in Desktops/320x212x12/. As the folder names suggest, background images should be 320x212 pixels in size. The last number of the folder name is the color depth supported by the phone screen, however the phone software will adjust this on its own to suit its display. Thumbnail images should be sized to 80x53 pixels. An example List.xml file would look similar to the following:

```
<CiscoIPPhoneImageList>
  <ImageItem Image="TFTP:Desktops/320x212x16/TN-fish.png"
    URL="TFTP:Desktops/320x212x16/fish.png"/>
  <ImageItem Image="TFTP:Desktops/320x212x16/TN-moon.png"
    URL="TFTP:Desktops/320x212x16/moon.png"/>
  <ImageItem Image="TFTP:Desktops/320x212x16/TN-kitten.png"
    URL="TFTP:Desktops/320x212x16/kitten.png"/>
</CiscoIPPhoneImageList>
```

5) Device Configuration Files

Due to differences in the firmware of the different types of Cisco IP Phones, there are also differences in the device specific configuration files. Upon switching to standardized XML tags, newer model Cisco phones are now configured purely with XML files while the older models used combination of XML and text data files to provide all the configuration information. For the models discussed here, all XML device configurations are named *SEPxxxxxxxxxxxx.cnf.xml* and text data configurations are named *SIPxxxxxxxxxxxx.cnf* where 'xxxxxxxxxxxx' is replaced with the MAC address of the phone that the configurations belong to (ensuring that any alphabetical characters are uppercase).

eg. Phone MAC Address 0123456789AB would use config files:

SEP0123456789AB.cnf.xml

SIP0123456789AB.cnf (if required by device)

In terms of the models discussed in this document, only the Cisco 7960 and 7940 models will use a SIP.cnf file in addition to its SEP.cnf.xml file. All other devices are configured with an SEP.cnf.xml file only.

7940 / 7960

As indicated earlier, a large volume of information is available on configuring the 7960 and 7940 models. These phones make use of only two configuration files and being the simpler of the two files, the SEP file will be reviewed first. This file ensures that the phone's application loader will find and apply the correct firmware for the phone. A copy of this file will be similar to the following:

```
<device>
<deviceProtocol>SIP</deviceProtocol>
<loadInformation model="IP Phone 7960">POS3-8-12-00</loadInformation>
</device>
```

The only two settings required in this file will be obvious settings; the first to specify that the phone will be using the SIP protocol, the other to provide the filename of the firmware the phone will need.

The second component of the configuration is the SIP.cnf file. This file specifies the actual SIP settings for the phone. An example SIP configuration file for a 7960 is as follows:

```
proxy1_address: "xxx.xxx.xxx.xxx"
proxy2_address: "xxx.xxx.xxx.xxx"
proxy3_address: "xxx.xxx.xxx.xxx"
proxy4_address: "xxx.xxx.xxx.xxx"
```

```
line1_name: "EXT NUMBER"
line1_shortcode: "DISPLAYED ON SCREEN"
line1_displayname: "DISPLAY NAME"
line1_authname: "EXT NUMBER"
line1_password: "EXT PASSWORD"
```

Line 1 Settings

```
line2_name: ""
line2_shortcode: ""
line2_displayname: ""
line2_authname: "UNPROVISIONED"
line2_password: "UNPROVISIONED"
```

Line 2 Settings

```
line3_name: ""
line3_shortcode: ""
line3_displayname: "" Name
line3_authname: "UNPROVISIONED"
line3_password: "UNPROVISIONED"
```

```
line4_name: ""
line4_shortcode: ""
line4_displayname: ""
line4_authname: "UNPROVISIONED"
line4_password: "UNPROVISIONED"
```

Additional line settings.
7940 has 2 lines available
7960 has 6 lines available

```
line5_name: ""
line5_shortcode: ""
line5_displayname: ""
line5_authname: "UNPROVISIONED"
line5_password: "UNPROVISIONED"
```

```
line6_name: ""
line6_shortcode: ""
line6_displayname: ""
line6_authname: "UNPROVISIONED"
line6_password: "UNPROVISIONED"
```

```
proxy_emergency: ""
proxy_emergency_port: "5060"
proxy_backup: ""
proxy_backup_port: "5060"
outbound_proxy: ""
outbound_proxy_port: "5060"
```

Firewall and NAT
Traversal settings

```
nat_enable: "0"
nat_address: ""
voip_control_port: "5060"
start_media_port: "16348"
end_media_port: "20134"
nat_received_processing: "0"
```

```
phone_label: "Phone Title "
time_zone: CST
logo_url: "http://domain.ext/imagefile.bmp"
```

Screen display settings

```
telnet_level: "2"
phone_prompt: "Cisco7960"
phone_password: "password"
enable_vad: "0"
network_media_type: "auto"
user_info: phone
```

Telnet and
network settings

The first few lines of the configuration make up a list of PBX servers that the phone will attempt to register with (in the order listed). The configuration is known to accept at least 6 addresses,

however only one is required.

The second section is the SIP settings for each line extension. The `lineX_name` and `lineX_authname` will both be set to the extension number that will be used for that particular line. The `lineX_password` is the registration secret password set for the extension that is being registered. This password is set in the Asterisk server extension settings. Display name is the callerID display name that is sent out from the phone (this is usually overridden by Asterisk). Lastly, the `lineX_shortcode` is the text that will appear on the Cisco IP Phone screen next to the line button being programmed. The example configuration file above is for a Cisco 7960. The same file can be used for a 7940 by removing lines 3 through 6.

Additional proxy information can be specified for failover and advanced call handling. This is done in the section of the configuration for backup, emergency, and outbound proxy addresses. These settings are not necessary, leaving them blank is fine. If these settings have already been defined in the global `SIPDefault.cnf`, these should not be included in the device configuration files.

The next component is the NAT and firewall sections. The NAT enabling will depend on the configuration of the system. NAT should only be required if the phone is going to be connecting to the Asterisk server through an internet connection from an external network, and will have `nat_enable` set to 1. Phones connecting on the local network will have `nat_enable` set to 0. Even if NAT is enabled, a NAT address does not need to be specified. Firewall ports are also specified here. The VoIP control port is the port that the phone will communicate to the Asterisk server on. The standard control port of Asterisk extensions is 5060 although this can be changed for each extension, so be sure to confirm the port used by the extensions being registered. The media port start and end specifies the range of ports that will be used to transmit the realtime audio of a conversation to the phone. It is recommended that this be changed from its defaults to a port series that will not affect other local network communications. It is also important that the media ports provided match the ones specified in the Asterisk server settings.

The display settings contain information regarding the display screen on the phone. The `phone_label` refers to the text displayed in the top bar on the right hand side of the phone screen (It is recommended a space is placed after the phone label text for visual padding from the 'SIP' icon). The `time_zone` will cause the time from the network time server to be formatted and converted to reflect the time zone specified. A background can also be specified as well by adding a value for `logo_url`. As before, the logo must be in BMP file and no bigger than 90x56 pixels. For best results use colors a 4 color palette of #ffffff, #000000, #404040, and #808080. Specifying a `logo_url` in a device

configuration file will override any logo that has been specified in the SIPDefault.cnf file.

The telnet settings make the phone accessible via telnet from computers. By default `telnet_level` is set to '0' or disabled. The setting supports two operating modes. Mode '1' is open availability to the phone via telnet with no username or password required. The second mode is '2' for privileged. The privileged mode is recommended, and requires that the next two settings; `phone_prompt` and `phone_password`, be set.

7945 / 7965 / 7970

For purely XML configured devices, the SIP.cnf file will not be needed. The single XML file contains all data for the phone that it pertains to, using the same MAC address file naming as the SIP files. An example SEP.cnf.xml file for a Cisco 7945 looks like the following:

```
<device>
<deviceProtocol>SIP</deviceProtocol>
<sshUserId>admin</sshUserId>
<sshPassword>password</sshPassword>
<devicePool>
  <dateTimeSetting>
    <dateTemplate>D/M/YA</dateTemplate>
    <timeZone>Central Standard/Daylight Time</timeZone>
    <ntp>
      <ntp>
        <name>xxx.xxx.xxx.xxx</name>
        <ntpMode>Unicast</ntpMode>
      </ntp>
    </ntp>
  </dateTimeSetting>
  <callManagerGroup>
    <members>
      <member priority="0">
        <callManager>
          <ports>
            <ethernetPhonePort>2000</ethernetPhonePort>
            <sipPort>5060</sipPort>
            <securedSipPort>5061</securedSipPort>
          </ports>
          <processNodeName>xxx.xxx.xxx.xxx</processNodeName>
        </callManager>
      </member>
    </members>
  </callManagerGroup>
</devicePool>
<sipProfile>
  <sipProxies>
    <backupProxy></backupProxy>
    <backupProxyPort>5060</backupProxyPort>
    <emergencyProxy></emergencyProxy>
    <emergencyProxyPort></emergencyProxyPort>
    <outboundProxy></outboundProxy>
    <outboundProxyPort></outboundProxyPort>
    <registerWithProxy>true</registerWithProxy>
  </sipProxies>
  <sipCallFeatures>
    <cnfJoinEnabled>true</cnfJoinEnabled>
    <callForwardURI>x-serviceuri-cfwdall</callForwardURI>
    <callPickupURI>x-cisco-serviceuri-pickup</callPickupURI>
    <callPickupListURI>x-cisco-serviceuri-opickup</callPickupListURI>
    <callPickupGroupURI>x-cisco-serviceuri-gpickup</callPickupGroupURI>
  </sipCallFeatures>
</sipProfile>
</device>
```

General operating settings

Date / Time server settings and formatting

Asterisk server address and port settings

Alternant servers and port settings

```

<meetMeServiceURI>x-cisco-serviceuri-meetme</meetMeServiceURI>
<abbreviatedDialURI>x-cisco-serviceuri-abbrdial</abbreviatedDialURI>
<rfc2543Hold>false</rfc2543Hold>
<callHoldRingback>2</callHoldRingback>
<localCfwdEnable>true</localCfwdEnable>
<semiAttendedTransfer>true</semiAttendedTransfer>
<anonymousCallBlock>2</anonymousCallBlock>
<callerIdBlocking>2</callerIdBlocking>
<dndControl>0</dndControl>
<remoteCcEnable>true</remoteCcEnable>
</sipCallFeatures>
<sipStack>
  <sipInviteRetx>6</sipInviteRetx>
  <sipRetx>10</sipRetx>
  <timerInviteExpires>180</timerInviteExpires>
  <timerRegisterExpires>3600</timerRegisterExpires>
  <timerRegisterDelta>5</timerRegisterDelta>
  <timerKeepAliveExpires>120</timerKeepAliveExpires>
  <timerSubscribeExpires>120</timerSubscribeExpires>
  <timerSubscribeDelta>5</timerSubscribeDelta>
  <timerT1>500</timerT1>
  <timerT2>4000</timerT2>
  <maxRedirects>70</maxRedirects>
  <remotePartyID>false</remotePartyID>
  <userInfo>None</userInfo>
</sipStack>
<autoAnswerTimer>1</autoAnswerTimer>
<autoAnswerAltBehavior>false</autoAnswerAltBehavior>
<autoAnswerOverride>true</autoAnswerOverride>
<transferOnhookEnabled>false</transferOnhookEnabled>
<enableVad>false</enableVad>
<dtmfAvPayload>101</dtmfAvPayload>
<dtmfDbLevel>3</dtmfDbLevel>
<dtmfOutOfBand>avt</dtmfOutOfBand>
<alwaysUsePrimeLine>false</alwaysUsePrimeLine>
<alwaysUsePrimeLineVoiceMail>false</alwaysUsePrimeLineVoiceMail>
<kpm1>3</kpm1>
<phoneLabel>PHONE TITLE</phoneLabel>
<stutterMsgWaiting>1</stutterMsgWaiting>
<callStats>false</callStats>
<silentPeriodBetweenCallWaitingBursts>10</silentPeriodBetweenCallWaitingBursts>
<disableLocalSpeedDialConfig>false</disableLocalSpeedDialConfig>
<siplines>
  <line button="1">
    <featureID>9</featureID>
    <featureLabel>DISPLAYED ON SCREEN</featureLabel>
    <proxy>xxx.xxx.xxx.xxx</proxy>
    <port>5060</port>
    <name>EXT NUMBER</name>
    <displayName>DISPLAY NAME</displayName>
    <autoAnswer>
      <autoAnswerEnabled>2</autoAnswerEnabled>
    </autoAnswer>
    <callWaiting>3</callWaiting>
    <authName>EXT NUMBER</authName>
    <authPassword>EXT PASSWORD</authPassword>
    <sharedLine>false</sharedLine>
    <messageWaitingLampPolicy>1</messageWaitingLampPolicy>
    <messagesNumber>*99</messagesNumber>
    <ringSettingIdle>4</ringSettingIdle>
    <ringSettingActive>5</ringSettingActive>
    <contact>EXT NUMBER</contact>
    <forwardCallInfoDisplay>
      <callerName>true</callerName>
      <callerNumber>false</callerNumber>
      <redirectedNumber>false</redirectedNumber>
      <dialNumber>true</dialNumber>
    </forwardCallInfoDisplay>
  </line>
</siplines>

```

Calling features settings

SIP protocol settings

Device preferences
(auto-answer, call stats, etc.)

Phone label

Line settings

NOTE: lines can be programmed for other functions when the correct featureID is provided

```

<line button="2">
  <featureID>20</featureID>
  <featureLabel>Menu</featureLabel>
  <serviceURI>http://example.domain.ext/services/menu.xml</serviceURI>
</line>
</sipLines>
<voipControlPort>5060</voipControlPort>
<startMediaPort>16348</startMediaPort>
<stopMediaPort>20134</stopMediaPort>
<dscpForAudio>184</dscpForAudio>
<ringSettingBusyStationPolicy>0</ringSettingBusyStationPolicy>
<dialTemplate>dialplan.xml</dialTemplate>
<softKeyFile></softKeyFile>
</sipProfile>
<commonProfile>
  <phonePassword></phonePassword>
  <backgroundImageAccess>true</backgroundImageAccess>
  <callLogBlfEnabled>2</callLogBlfEnabled>
</commonProfile>
<loadInformation>SIP45.8-4-2S</loadInformation>
<vendorConfig>
  <disableSpeaker>false</disableSpeaker>
  <disableSpeakerAndHeadset>false</disableSpeakerAndHeadset>
  <pcPort>0</pcPort>
  <settingsAccess>1</settingsAccess>
  <garp>0</garp>
  <voiceVlanAccess>0</voiceVlanAccess>
  <videoCapability>0</videoCapability>
  <autoSelectLineEnable>0</autoSelectLineEnable>
  <webAccess>0</webAccess>
  <daysDisplayNotActive>1,2,3,4,5,6,7</daysDisplayNotActive>
  <displayOnTime>00:00</displayOnTime>
  <displayOnDuration>00:00</displayOnDuration>
  <displayIdleTimeout>00:00</displayIdleTimeout>
  <spanToPCPort>1</spanToPCPort>
  <loggingDisplay>1</loggingDisplay>
  <loadServer></loadServer>
</vendorConfig>
<userLocale>
  <name></name>
  <uid></uid>
  <langCode>en_US</langCode>
  <version>1.0.0.0-1</version>
  <winCharSet>iso-8859-1</winCharSet>
</userLocale>
<networkLocale></networkLocale>
<networkLocaleInfo>
  <name></name>
  <uid></uid>
  <version>1.0.0.0-1</version>
</networkLocaleInfo>
<deviceSecurityMode>1</deviceSecurityMode>
<authenticationURL>http://example.domain.ext/services/authenticate.php</authenticationURL>
<directoryURL>http://example.domain.ext/services/directory.php</directoryURL>
<servicesURL>http://example.domain.ext/services/menu.xml</servicesURL>
<idleURL></idleURL>
<informationURL></informationURL>
<messagesURL></messagesURL>
<proxyServerURL></proxyServerURL>
<dscpForSCCPPhoneConfig>96</dscpForSCCPPhoneConfig>
<dscpForSCCPPhoneServices>0</dscpForSCCPPhoneServices>
<dscpForCm2Dvce>96</dscpForCm2Dvce>
<transportLayerProtocol>4</transportLayerProtocol>
<capfAuthMode>0</capfAuthMode>
<capfList>
  <capf>
    <phonePort>3804</phonePort>
  </capf>
</capfList>
<certHash></certHash>

```

Additional port settings

Dialplan and softkey specification

Vendor specific settings

Locale preferences

Additional function keys and logo destinations

Additional quality of service (QoS) settings

```
<encrConfig>>false</encrConfig>
</device>
```

Obviously these files are very complex in terms of the settings being specified. In this guide only the necessary settings will be discussed. More information is available on all the settings and their specifics, most reliably from Cisco's documentation on their website or from any number of other helpful VoIP support sites.

The first important section to modify is the date and time settings. If desired, the format of the date display can be altered however, this is not necessary. The time zone does need to be modified to reflect the local time zone in the `<timeZone>...</timeZone>` tags. A complete list of acceptable entries is available in the Cisco documentation. The other important item to change in this section is the location of the date / time server to use to maintain the phone's clock. This is done in the `<name>...</name>` tags under the `<ntp>...</ntp>` subsection. If a server is maintained locally this can be used, otherwise a number of publicly available servers can be found on the internet.

The `<callManagerGroup>` section contains information about the server that manages the call handling. In this case that will be the Asterisk server that is being used. The IP address of the server should be entered into `<processNodeName>...</processNodeName>`. If the device is going to connect to the server on a non-standard port, those settings should also be changed. Additionally, the `<sipProxies>` section allows for more advanced call handling proxies to be specified. These can be left blank if not being used.

The next two groupings of settings, `<sipCallFeatures>` and `<sipStack>`, can usually be left unchanged, unless some specific advanced functionality is being setup. If these settings are being modified, be sure to check the Cisco documentation for more details on them.

`<phoneLabel>...</phoneLabel>` is the next item in the configuration file that could use modifying. This text is used to as the display in the top right corner of the screen on the phone.

Moving down the configuration to the `<sipLines>` group, this is where the provisioning of the phone lines will be done. This is where the configuration gets more useful and interesting. The newer model Cisco phones that are configured using the SEP.cnf.xml files have the ability to have the lines programmed for various different functions. Each type of functionality that can be assigned has a corresponding `<featureID></featureID>`. For an ordinary SIP extension, `<featureID>9</featureID>` would need to be used. To assign the line to function like the XML services button, `<featureID>20</featureID>` would be needed. The `<featureID></featureID>` is the primary part of the line configuration. All other

defined items for the line will be determined by the `<featureID></featureID>` is given (Currently an appendix of these featureID's and their subsequent requirements is being compiled).

Following the configuration of the line buttons, some additional ports can be specified. In this section, it is recommended that the `<startMediaPort></startMediaPort>` and `<stopMediaPort></stopMediaPort>` be modified to match with the settings being used by the Asterisk server.

The remainder of the configuration file does not usually require modification with the exception of the various service URLs that can be provided (if desired). These are optional items that would be linked to XML generating scripts or XML files (in the Cisco Services XML Standard).

5) Upload And Test

After preparing the needed configuration files, upload all of them to the TFTP server created earlier. Once all files have been uploaded connect the phone to a local network port and power. The phone will run through its boot sequence. If everything has been done correctly, the phone will look for the firmware it has been directed to use and download if necessary, request its configuration files and apply the downloaded configurations.

As the phone boots it is often useful to watch the output of the Asterisk server as well as the network packets received by the server using the “*tshark*” application mentioned earlier. These two collections of information should be of great assistance for any troubleshooting that needs to be done after.

NOTES:

Many of the Cisco phone units have different slightly different model numbers based on slight manufacturing differences (eg. Information pertaining to the 7960 also applies to 7961, 7960G, and 7961G). This is the same for most of the Cisco phone model series.

LINKS:

Asterisk: <http://www.asterisk.org>

AsteriskNOW: <http://www.asterisknow.org>

FreePBX: <http://www.freepbx.org>

VoIP-info: <http://voip-info.org>

Wireshark: <http://www.wireshark.org/>

Cisco Systems: <http://www.cisco.com/>

MORE INFORMATION:

If there are any questions or comments about this document or some assistance with the materials application to your system, please contact me via the website: <http://minded.ca/>

Tyler Winfield

Minded Software Solutions