Incorporating SIP Resources Into a Telex Radio Dispatch System

For C-Soft versions 5.0, 5.2 & 5.3

Notes

Incorporating SIP Resources into a Telex Radio Dispatch System

By Dave Grant

February 22, 2011

Table of Contents

Introduction

I am going to assume that anyone using this document is familiar with C-Soft Designer 5.0 or later.

You are most likely a radio person that has been asked about adding VoIP or SIP telephone services to your Telex Radio Dispatch System.

So you ask yourself "What is SIP"?

This document discusses what SIP is and how to add SIP resources to a C-Soft Console.

We will touch on the following topics

- SIP and what it requires to exist on its own
- SIP integration into a Telex RoIP Dispatch System
- Standard and Extended SIP Client in C-Soft
- Building a test C-Soft Console and use the SIP features

Definitions

- **Telephony** (pronounced `ta-lef-a-nee') - Is the general service of providing voice over distance.
- **PSTN** – Public Switched Telephone Network, the phone company at large
- **SIP** - Session Initiation Protocol, a VoIP standard for processing voice calls
- **RTP** – Real Time Protocol, an Ethernet standard used to transport voice
- **POTS** – Plain Old Telephone Service, a standard dial tone line
- **ISDN** – Integrated Services Digital Network, a digital telephone service
- **Ethernet** – A de facto standard for transporting packet data, including TCP/IP
- **Asterisk** – An Open Source Telephony Switching Platform
- **Codec** – Used to Code and Decode voice information
- **VoIP** – Voice over Internet Protocol, using IP Networks to make voice calls
- **PBX** – Private Branch Exchange, an office phone system
- **H.323** – A protocol used to transport multimedia content, including voice

References

- **LIT000082000_CSoft.pdf,** C-Soft Software Console Administrators Guide, 10/2010 Bosch Security Systems http://www.telexradiodispatch.com/
- **LIT000539000.pdf,** SIP Quick Reference Guide, 10/2010 Bosch Security Systems http://www.telexradiodispatch.com/
- **http://www.asterisk.org/**
- **http://www.voip-info.org/**
- **http://opensourcemadness.blogspot.com/2007/09/build-your-own-linuxasteriskopenserdhcp.html**

What is SIP?

SIP (Session Initiation Protocol) is an agreement.

When someone says they use SIP, what they mean is they agree to use a certain set of Ethernet protocols to place telephone calls to other users.

What does SIP require?

For the purposes of this document, the SIP Server being used is Asterisk.

- **Server**
  by definition SIP will use a server. A program know as "Asterisk" is widely used in the Telephony industry to provide SIP Services. As illustrated in figure 1, Asterisk can bridge many different Telephony related services, SIP is just one.

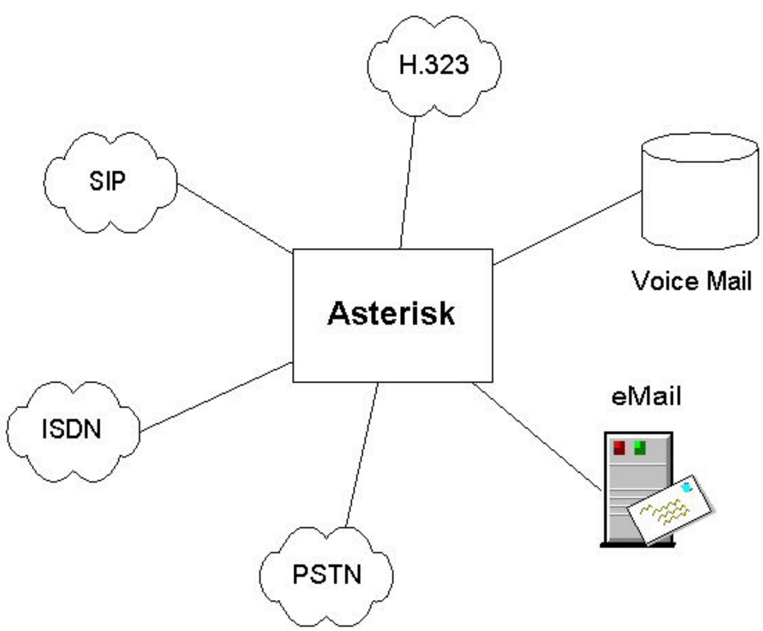


Fig.1 – Asterisk can connect many types of calls together.

- **Client**
  There are many clients that support SIP, both hard (deskset) and soft (a program). There is a soft client built into C-Soft. It has been available since version 5.0. C-Soft 5.3 introduced an Enhanced SIP Client. We will discuss both standard and enhanced versions.

- **Ethernet**
  SIP requires Ethernet to operate. Depending on the size and scope of your network, you may only have to deal with IP Switching. A larger network may have IP Routing requirements. There may be requirements for Firewall adjustments as well.

- **Protocols**
  Protocols can include RTP, RTCP, STP, SIP, H.323, G.711, G.726, G.729, etc. The actual protocols used depend on the client and server used in the call.

What does Asterisk provide?

Full featured Telephony Switching Services

Caller ID / Call Waiting
Call Hold / Call Transfer
Call Park / Call Conference
Call Forward on Hold / Busy / Variable
Automated Attendant / Voice Mail
Email Notification / Message Waiting
Advanced DB Access Features
Etc.

Hardware independence

Asterisk exists in the Public Domain, and is available to run on many different platforms. It has prompted many hardware vendors to package an Asterisk Server on a small Linux Appliance as a SIP Server. Just about any version of Linux, and most any Windows Platform from Windows 2000 on, will run Asterisk. This makes almost any computer an Asterisk SIP Server.

How does Asterisk do it?

In a simple VoIP telephone system, as shown in figure 2, the SIP telephone talks to the Asterisk SIP Server via the Ethernet Network. There is a short exchange of messages that setup the type of call, where the call is destined and the audio codec to use between the telephones. The SIP Server then calls the destination SIP telephone and connects the call together. Once the call is done, Asterisk tears the call down and clears up the connection.

The client or the server can initiate the call, but the server is always in the middle. It is responsible for audio path and format, Signaling, Call Routing, Voicemail and Notifications of Message Waiting.

A local SIP Server can link to a remote SIP Server to allow for access to remote telephones or other resources.

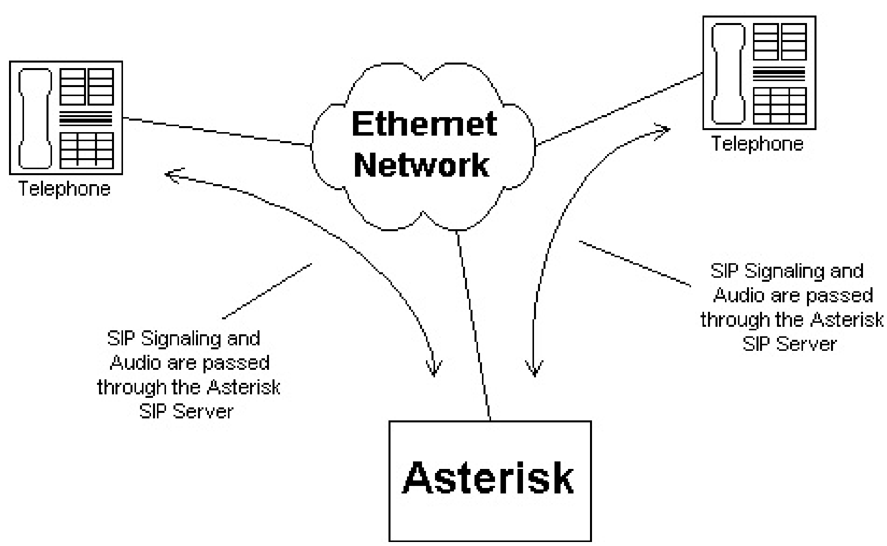


Fig. 2 – 2 SIP telephones connected on a simple system require a SIP Server.

There has been a Paradigm shift regarding telephones

Up till now the paradigm in the Telex Radio Dispatch System has been "a single telephone resource can be accessed by all consoles". With the introduction of SIP, the way that telephone resources are allocated to the C-Soft Console has changed. SIP gives each C-Soft Console a separate telephone line that is not shared with any other console. In fact there can be up to six (6) SIP lines on a single C-Soft Console. This is in addition to any traditional shared POTS resources located on the network.

Figure 3 illustrates the traditional method that a C-Soft Console uses to access a telephone resource. As shown, there is a single access point to the PSTN and only one conversation can occur at one time.

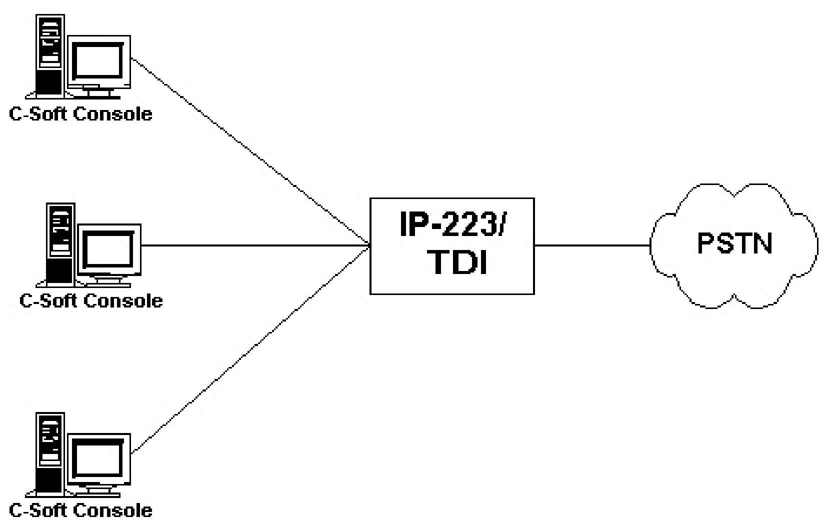


Fig. 3 – C-Soft Consoles traditionally share a single access point to PSTN.

Where does SIP fit in a Telex Radio Dispatch System?

In a traditional Telex Dispatch System, access to a telephone has been through a POTS line connected to a C6200 console or a TDI via an IP-223. All of the telephone lines have shared access and can be programmed across multiple consoles.

With SIP, the telephone exists inside C-Soft. The SIP client is configured in the C-Soft Designer and is added to a console as a button. In fact there can be multiple SIP Buttons on a C-Soft Console. What this means is the C-Soft SIP configurations can not be changed on a running console.

As illustrated in figure 4, an Asterisk SIP Server is connected to the local Telex Radio Dispatch Network and acts as a buffer between the dispatch console and the Public Switched Telephone Network. In this case a C-Soft console with a SIP client has replaced one of the telephones in our network and a remote SIP server allows access to the Public Switch Telephone Network. This also insures that the C-Soft Console is never directly exposed to any outside network.

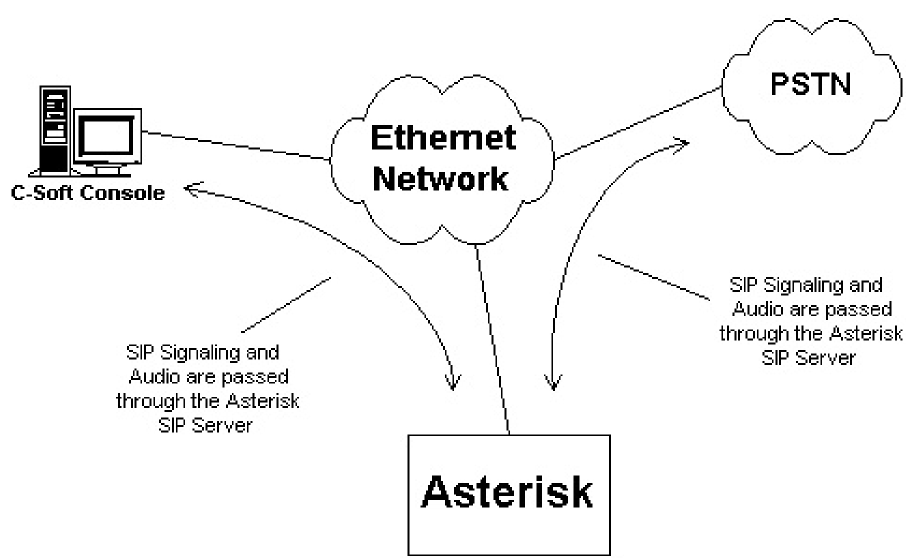


Fig. 4 – C-Soft Console uses SIP Server to place and receive calls.

SIP Changes our thinking about console configuration

When we add a SIP client to the C-Soft Console we include a direct line or lines to each console position. The number of SIP lines that can be used on a C-Soft Console will depend on the license key or keys that are installed on the console.

Currently any C-Soft Key with 24 lines has 2 *Standard SIP* resources. Any C-Soft Key with 50 or more lines has 6 *Standard SIP* resources available for use on the C-Soft Console.

One thing to keep in mind, the *Standard* SIP resources take the place of the existing C-Soft line resources and are not in addition to the 24 or 50 plus lines on the key. So a 24 line key with 2 SIP resources enabled, will really only support 22 lines.

Standard SIP Client

Key Type	Available SIP resources	Available Telex resources
C-Soft 24 line	2	22
C-Soft 50 line	6	44
C-Soft 100 line	6	94
C-Soft 150 line	6	144
C-Soft 200 line	6	194

Enhanced SIP Client

There is an additional *Enhanced* SIP *Client* C-Soft Key that is available in 2 or 6 SIP resource sizes. This is a second key that is connected to the C-Soft console. The ADHB4 can also be used to host *Enhanced SIP Client* Keys as an option to a second USB or PTR Key on a C-Soft Console.

The *Standard* and the *Enhanced* SIP *Clients* will be described later in this document.

To illustrate what has changed, figure 5 shows the addition of SIP access to the PSTN. This can be used along with the traditional method of sharing analog lines.

However the SIP access is completely controlled by the Asterisk SIP Server. As far as the C-Soft Console is concerned each line is an extension on a SIP Server and is only allowed to do certain tasks. So extension by extension, the Asterisk SIP Server controls whether they can receive calls, place calls to other extensions (consoles), or to the PSTN, etc.

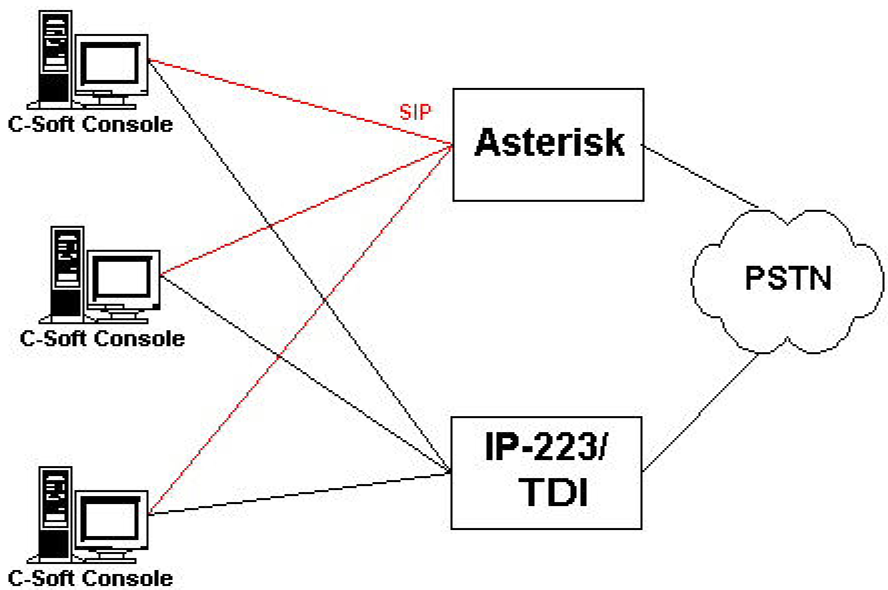
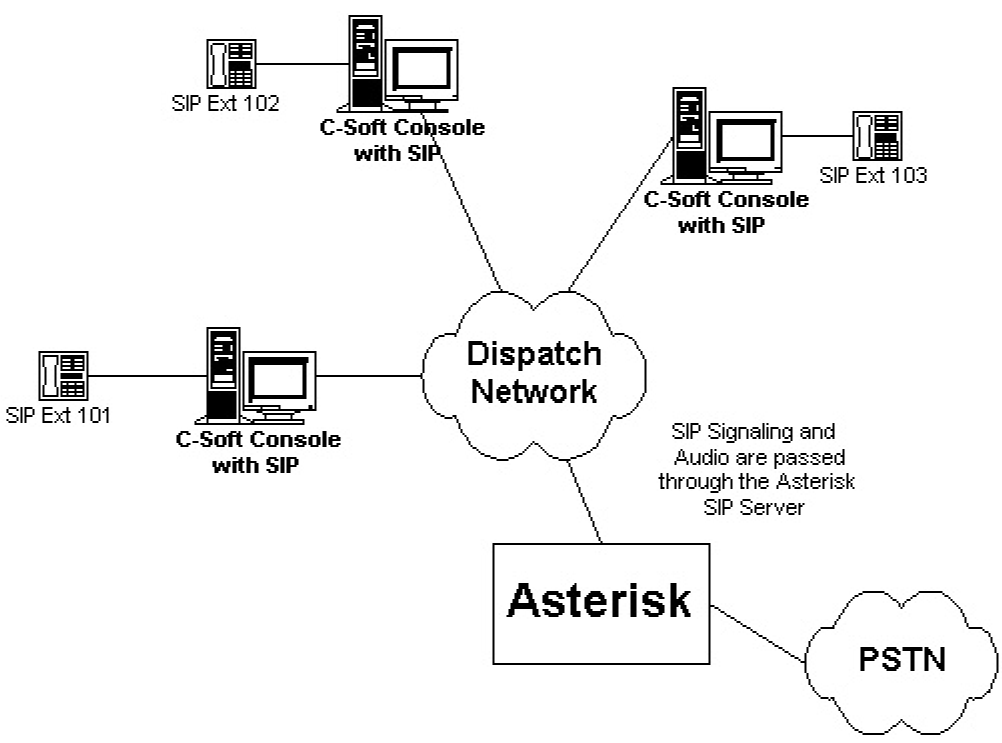


Fig. 5 – C-Soft Consoles with SIP, have private access points to PSTN.

C-Soft SIP client has multiple extensions

When adding SIP resources to a C-Soft Console, we are adding logical extensions to the actual console. So if you were in a big office with a lot of desksets, each of those desksets is an extension on a PBX. The SIP client on the C-Soft Console can have up to six (6) extensions registered on it at once. That means that a single dispatcher could have up to six SIP calls active at the same time. Planning is very important in the integration of SIP resources. Illustrated in figure 6, each console has a SIP license and has a single SIP extension on the Asterisk SIP Server.

Remember that the SIP extension is a logical thing and isn't an actual telephone deskset connected to the console.



Fig, 6 – Simple dispatch system with a SIP extension for each console.

In contrast, you may want to have a specific console handle all SIP traffic and distribute calls to other consoles via a traditional POTS line from the Asterisk SIP Server to a TDI resource.

Figure 7 illustrates a configuration where all SIP traffic is handled by one console and all other consoles have the traditional POTS line access via an IP223/TDI resource. In our example, the Asterisk SIP Server has a POTS interface that is configured as extension 107. This configuration allows the console with SIP resources to transfer a SIP call to a console that does not have SIP via a POTS resource. In addition, because we have a SIP Server, we can route incoming SIP calls to the C-Soft Consoles without SIP by using the POTS extension 107. This type of inbound call would bypass the SIP C-Soft Console completely and would be answered by one of the non SIP C-Soft Consoles.

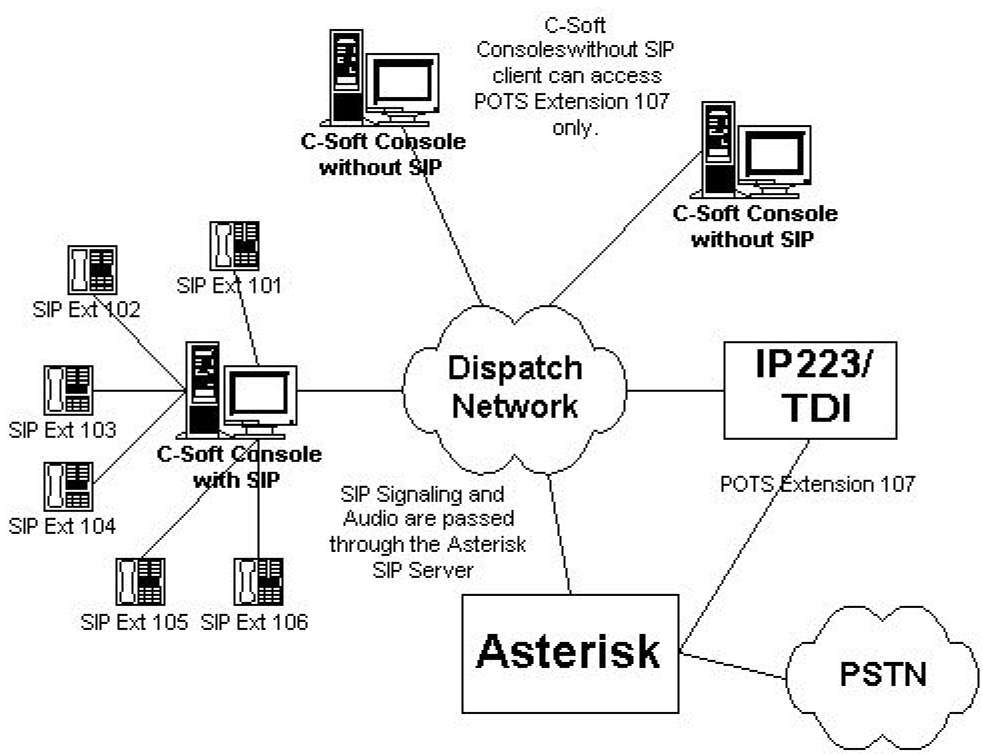


Fig. 7 – Single console with SIP resources can transfer a call via a POTS line.

Adding a SIP Server to a Telex Radio Dispatch System

In order to use any of the SIP resources in the C-Soft Console you must have access to a SIP Server.

If you don’t have a SIP Server, find someone that can provide you a SIP Server. Configure it to have 2 SIP extensions (3000, 3001) that can call each other and have voicemail. From the C-Soft Console programming point of view, only an extension number and the IP Address of the SIP Server are required to use SIP resources.

I use Asterisk for the testing I do when writing these articles. Asterisk is a set of Open Source Software, and there is almost unlimited information on the Internet concerning it’s use and configuration. The best thing about Asterisk is it’s free.

My Asterisk SIP Server runs on an HP T5720 Thin Client, running Debian Etch Linux. The T5720 (shown below) has more than enough horsepower to provide a fully featured SIP Server. Including Voicemail, email alerts, time of day forwarding, call intercept, call hold, etc.

Fig. 8 – The T5720 case is roughly 4.5"x9.5"x10.5".

This unit has no hard drive, no fan, no moving parts. If you are interested in building an Asterisk Server from a Thin Client, here is a link to the article I followed in building mine.

Asterisk will run on just about any hardware/software platform, including Windows, and is available from several manufactures as “Appliances” that are built and ready to go. All that is required is to program the extensions and features.

Asterisk SIP Server setup

There is not enough time to show how to install and configure an Asterisk SIP Server in this article, but I will show the minimum configuration settings that I used to provision my test system. This section assumes that you are familiar with Asterisk configuration. The location of configuration files, the commands to reload and restart the SIP Server, show connected peers, etc.

A SIP Server is fairly simple, a client registers with the server by exchanging a few messages, similar to an email exchange, that verify that the extension exists and is enabled to handle calls. In an Asterisk SIP Server there are three configuration files that need to be adjusted to allow the C-Soft SIP client to register with the SIP Server and place or receive calls. In essence the only information the client needs is the extension and IP Address of the SIP Server.

In order to provision our Asterisk SIP Server to allow the C-Soft client to register, we must adjust three Asterisk configuration files. The first is the “sip.conf” file that defines the SIP extensions that are allowed to connect to the server. In “sip.conf” we need to add the following information to enable extension 3000 and 3001 on our Asterisk SIP Server.

```
[3000]
type=friend
host=dynamic
context=3000
mailbox=3000
dtmfmode=rfc2833
disallow=all
allow=ulaw

[3001]
type=friend
host=dynamic
context=3001
mailbox=3001
dtmfmode=rfc2833
disallow=all
allow=ulaw
```

This configuration defines SIP extensions 3000 and 3001 that have voicemail accounts, it uses inband DTMF RTP payload and only uses the G.711 ulaw codec for audio.

The second file is "extensions.conf" which defines the actions or dialplans that this extension can access. Again for our extensions 3000 and 3001 we would add the following information to the "extensions.conf" file on our Asterisk SIP Server.

```
[3000]  ;voip client on C-Soft Console
include => retrieve-voicemail
include => local-dial

[3001]  ;voip client on C-Soft Console
include => retrieve-voicemail
include => local-dial

[local-dial]
exten => _3XXX,1,Answer(500)
exten => _3XXX,n,Ringing
exten => _3XXX,n,Dial(SIP/${EXTEN},25,d)
exten => _3XXX,n,Hangup

[retrieve-voicemail]
exten => _8000,1,VoicemailMain()
```

This configuration gives the C-Soft client the minimum services of registering with the Asterisk SIP Server. And once registered, the client is allowed to call other extensions that begin with 3xxx and retrieve voicemail from extension 8000.

The third file is "voicemail.conf" and it defines the users voicemail features. Adding the following entries to the "[default]" section of voicemail.conf will allow the users to have voicemail accounts with the user names "Dave" and "Joe", and will be alerted at email account "VMxxx". For these users we don't want to attach a wave file of the voicemail to the notification email.

```
[default]
3000 =>3000,Dave,VMDave@foo.org,pager.com,tz=pacific|attach=no
3001 =>3001,Joe,VMJoe@foo.org,pager.com,tz=pacific|attach=no
```

C-Soft SIP Client types

As of C-Soft 5.3 there are two (2) SIP clients available to the console. The first is the *Standard SIP Client*, and the second is the *Enhanced SIP Client*.

Standard SIP Client

The *Standard SIP Client* is limited to placing and receiving Calls only. As illustrated in figure 9, the client user interface has a small Call list and Call history. Calls on the *Standard SIP Client* have no advanced telephony features like Call Hold or Call Transfer. If the SIP Server provided voicemail services, the *Standard SIP Client* could access by calling a local extension.

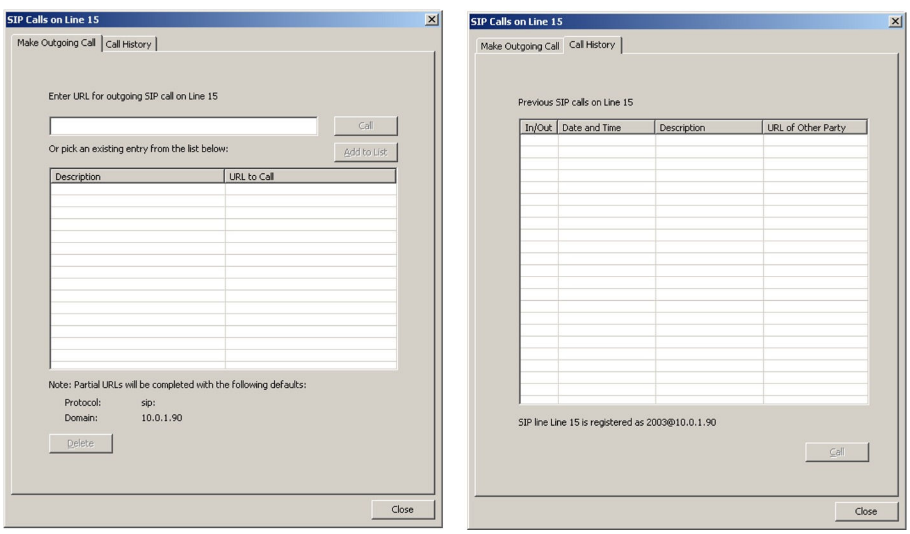


Fig. 9 – The *Standard SIP Client* can place and receive Calls with History.

Enhanced SIP Client

The *Enhanced SIP Client* has many additional features beyond that of the standard client. This includes the ability to place and receive Calls, Call Hold, Call Transfer, Call Conference, Mute, Flash, Keypad, etc. Refer to the C-Soft Software Administrators Guide and the SIP Quick Reference Guide provided by the factory for complete details on SIP client functionality. More information on these guides can be found at http://www.telexradiodispatch.com/.

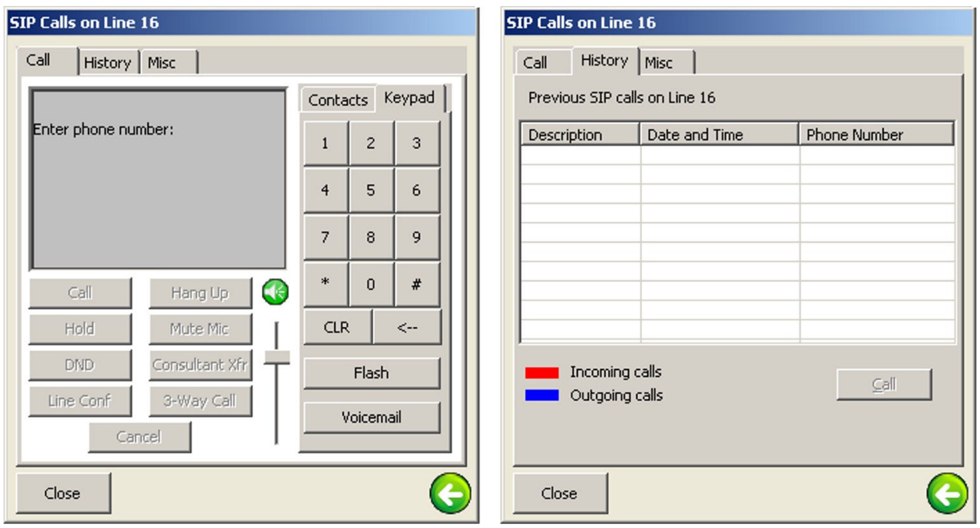


Fig. 10 – The *Enhanced SIP Client* has many additional Call Features.

There is a reason that I showed the *Standard SIP Client* and the *Enhanced SIP Client* user interface pages. Because until you get your SIP client programmed correctly and registered on the SIP Server, you will never see these pages. When the C-Soft Console is started, the SIP client(s), one for each active SIP Button, will attempt to register with the SIP Server that it is programmed to use. If that registration fails, you get an error message in the Global Call History telling you that the registration failed. You also get an inactive SIP Button on your console that does nothing.

C-Soft Designer SIP Setup

There is no difference in the programming between the Standard and Extended SIP Client as far as the C-Soft Designer is concerned.

The only information that is needed to program a C-Soft SIP Client is the SIP Server IP Address and the extension number assigned to the SIP client.

Global SIP Configurations

There are two places in the C-Soft Designer that we configure SIP settings. The first is in the Edit Menu, where you select Setup SIP Phone.

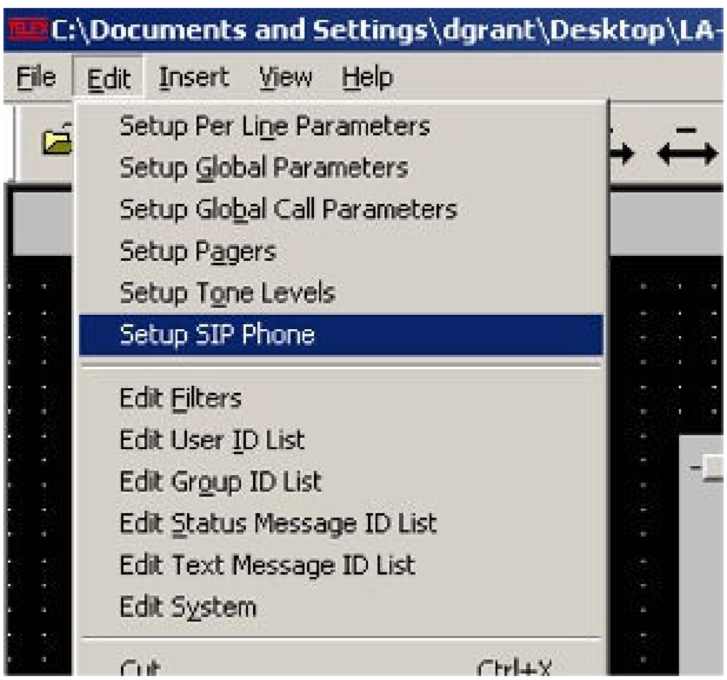


Fig. 11 C-Soft Designer Edit Menu, selecting Setup SIP Phone.

There are two tabs in Global SIP Configuration. The SIP Network Tab will generally just have the User and Display Name entered here.

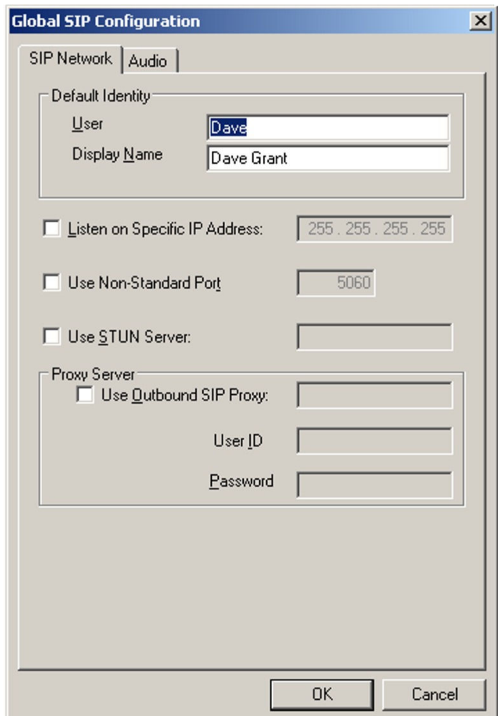


Fig. 12 Entering the Default User and Display Name in Global SIP Configuration.

The use of a SIP Proxy, non standard SIP addressing and port assignment, or using STUN is not within the scope of this document. All of these features require special programming on the SIP Server and special coordination with your IT Administrator.

Using these settings may prevent your SIP client from registering with the SIP Server.

The Audio Tab is where you set the LAM level and LAM time for use during a crosspatch. The default settings are shown below and are generally sufficient. The LAM Setup is used in a crosspatch, so that C-Soft knows when to start or stop transmitting to the other resources.

The Silence Detection settings have to do with the voice audio path silence detection between two SIP clients, which is used to conserve bandwidth in some SIP servers.

The last section is for adjusting the Jitter Buffer, again this will depend on the bandwidth available on the network and what the load variations do to latency.

The settings on the Audio Tab in Global SIP Configuration will generally not need to be adjusted. A typical implementation will be done on a private network where delay is less of a factor, and network bandwidth is completely within your control.

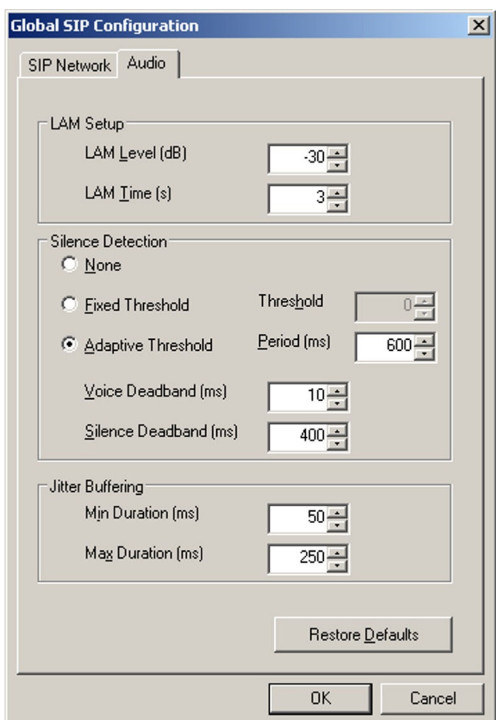


Fig. 13 Global SIP Audio Configuration settings.

Per Line SIP Configurations

The second place we go to configure SIP parameters is in the C-Soft Designer Edit Menu, where you select Setup Per Line Parameters.

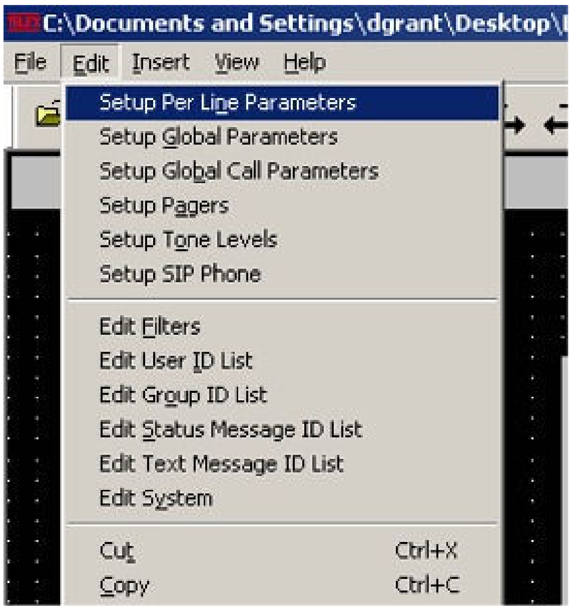


Fig. 14 C-Soft Designer Edit Menu, Setup Per Line Parameters for SIP.

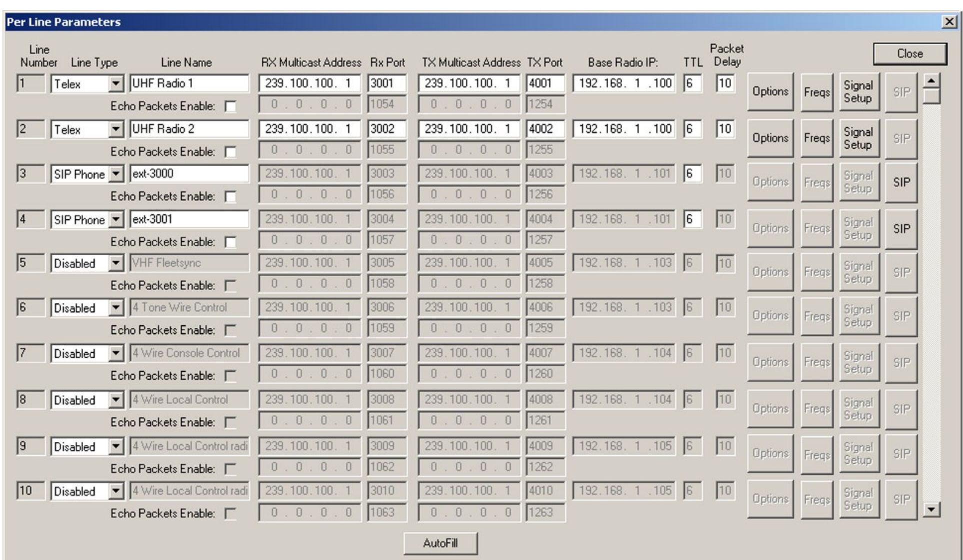


Fig. 15 Per Line Parameters page for our test console.

On the Per Line Parameters page, the first thing to do is set the Line Type to SIP Phone. Use the Line Name box to reference the extension that this Line uses on the SIP Server. Illustrated in figure 16 we set the SIP Phone Line Type and entered Line Names of ext-3000 for Line 3 and ext-3001 for Line 4.

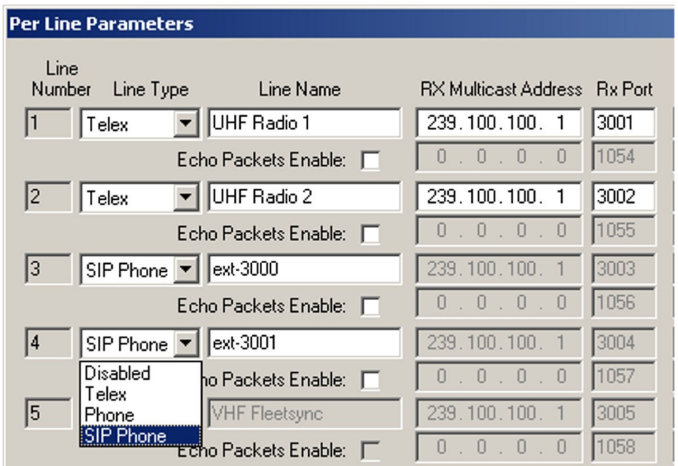


Fig. 16 Setting Line Type to SIP Phone.

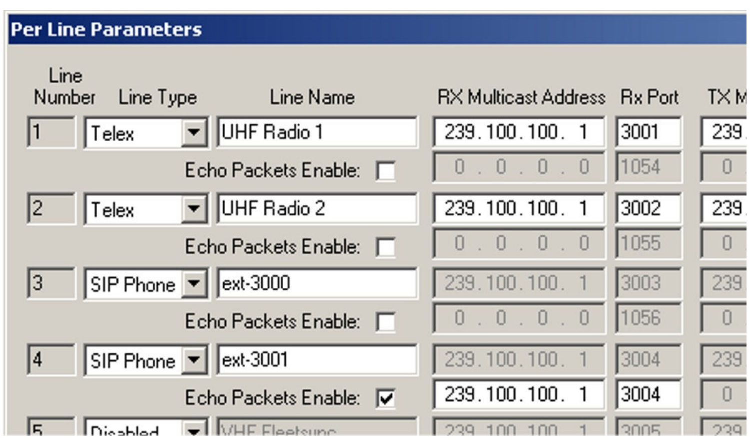


Fig. 17 Enable SIP resource recording with Echo Packets.

If you plan on recording SIP Calls, you will need to enable Echo Packets by checking the box and supplying a Multicast Address and unique port number. This is required if you plan to record directly to a Telex Network Recorder or via an IP-223 to an analog voice recorder. In figure 17, I have chosen to only record Line 4, (or extension 3001) by checking the checkbox *Echo Packets Enable* and providing a Multicast Address and a unique port number.

Once the Line Type is set to SIP Phone, the SIP Button at the far right of the page will turn active and have SIP in black lettering on it. Click on the SIP Button, this is where we assign the extension and IP Address of our SIP Server.

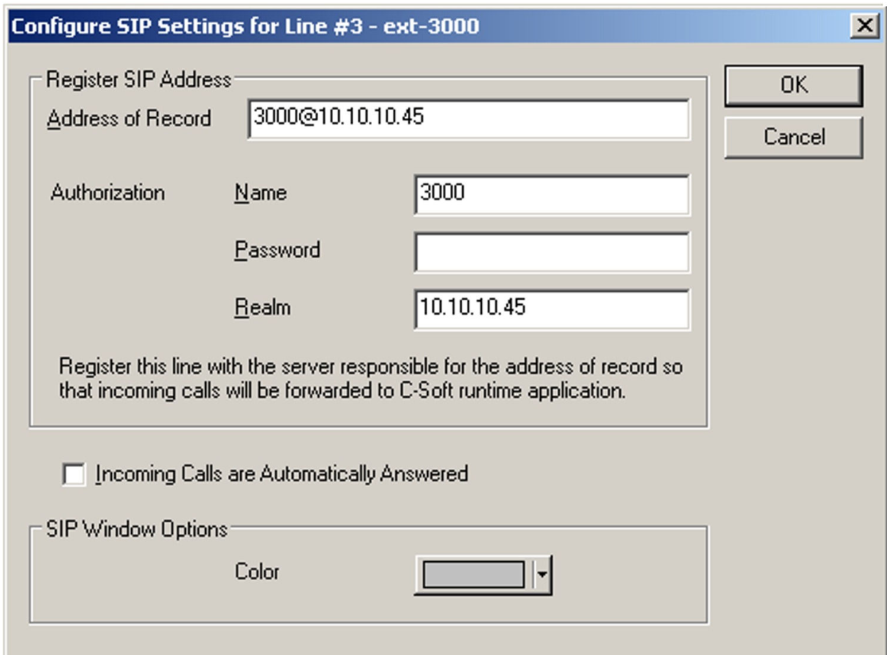


Fig. 18 - Settings for SIP line (or Button) ext-3000.

Address of Record will be set as follows, *extension@SIP-Server-Name*. So what we are looking at in figure 18, in the Address of Record field is an extension on Asterisk that I configured earlier, followed by an Ampersand, followed by the IP Address of the SIP Server. 3000@10.10.10.45. Looks a lot like an email address, huh. The 10.10.10.45 happens to the IP Address of my test SIP Server.

Name is the extension that has been assigned in the SIP Server for this Line. There will be one of these dialogs filled in for each SIP Button that is on your console. In figure 18 it is 3000 for this Line, it will be 3001 for Line 4.

Password is not used in my test configuration. But it would just need to configured in the SIP Server to be required in this configuration.

Realm is the address for our SIP Server. It can be any valid host address, in a Dispatch Network environment it would likely be a static IP Address, as I have configured here.

Incoming Calls will be answered automatically if the checkbox *Incoming calls are Automatically Answered* is checked.

You will need to supply the same information for Line 4 by clicking OK and then clicking the SIP Button at the far right side of Line 4. figure 19 shows the settings made in Line 4.

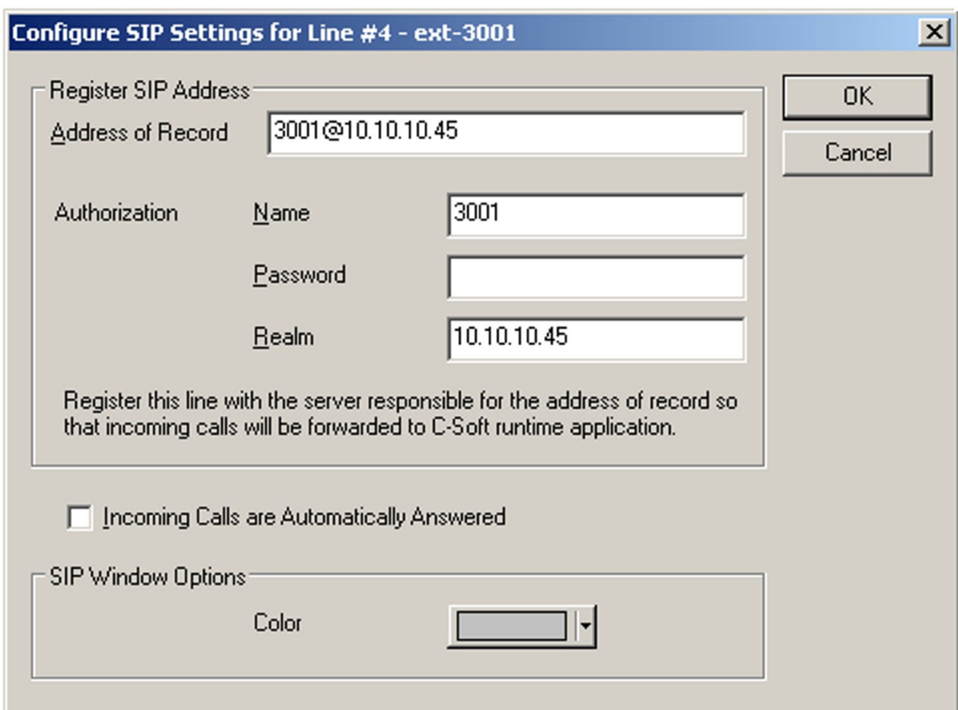


Fig. 18 - Settings for SIP line (or Button) ext-3001.

Click OK when your done. And then close the per line settings so that we can build our console.

Now that we have the C-Soft SIP configurations set to match that of our SIP Server, we can build a console that has a SIP resource on it.

Adding a SIP Button in C-Soft Designer

As I had stated in the Introduction, I am assuming that you have built a C-Soft Console before. So I am not going to show how to add UI Buttons in C-Soft. What I am going to do is build a simple console that has the basic elements of a console, including 2 SIP Call Control Buttons.

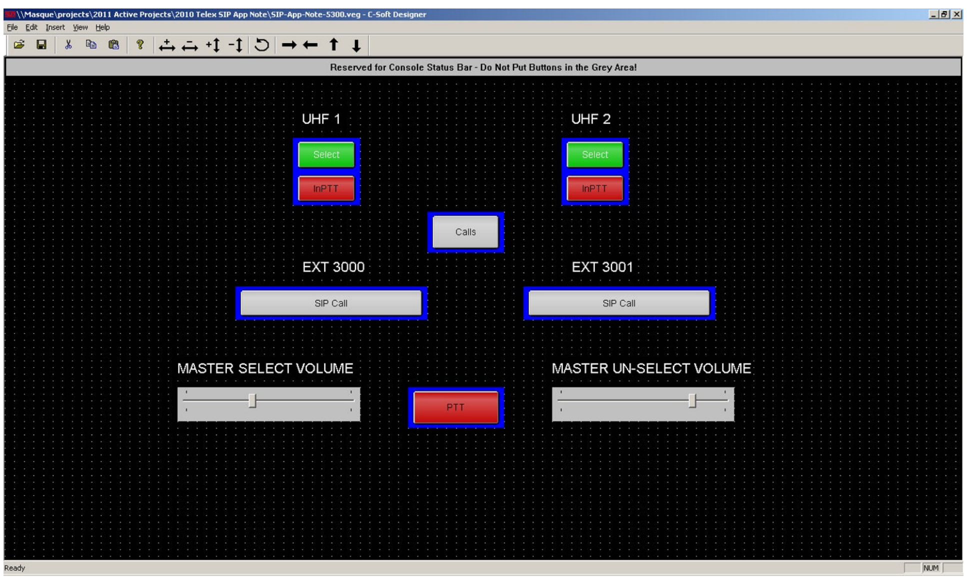


Fig. 19 C-Soft Designer Console with 2 SIP resources and 2 radio resources.

Figure 19 shows the C-Soft Designer screen of my test console. I have 2 standard radio Line resources, a Global Call History, 2 SIP Call resources, Select and Un-Select volume controls and a PTT All Button.

SIP Call Control resource

The SIP Call Control is programmed like any other button in C-Soft Designer. So I have added 2 standard UI Buttons and have changed their UI Element Function to a SIP Call Control. Once the UI Function has been set to SIP Call Control, the *Line to Associate Function With* dropdown box must be set. In our test console that is "ext-3000" for one of the SIP Buttons and "ext-3001" for the other. The settings are shown in figure 20 for the extension 3000 SIP Button.

The SIP Call Control Button is added as a standard UI Button and has the following settings.

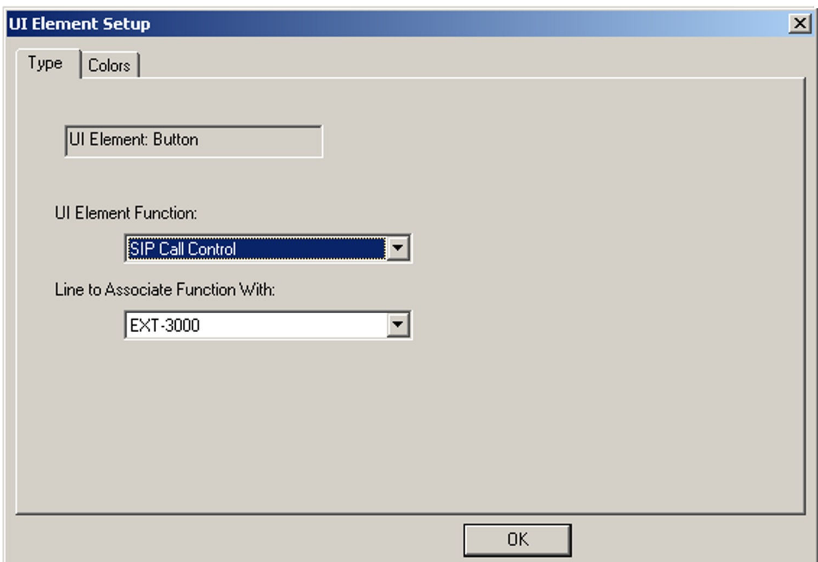


Fig. 20 Setting UI Element Function and Line Association for extension 3000.

Once OK has been clicked the Button face will change to "SIP Call" by default. We will need to set the other SIP Button to associate to extension 3001 before we continue. So here is your first test, add a new SIP Call Control Button and associate it to extension 3001.

Global Call History

I have also added a Global Call History Button on my test console. If we are using SIP on a C-Soft Console we should have a Global Call History Window on the console. The Global Call History will show the registration results for all of the SIP resources on the console. And can be a way to determine if your SIP clients are registered.

Making a SIP Call

Once you have entered the information for the SIP extension 3001 on the second SIP Call Control Button, we should be able to save our console. If the console saves without error, and the SIP Server is configured and available on the network, then we should see the C-Soft Console shown in figure 20 when opening the .veg file.

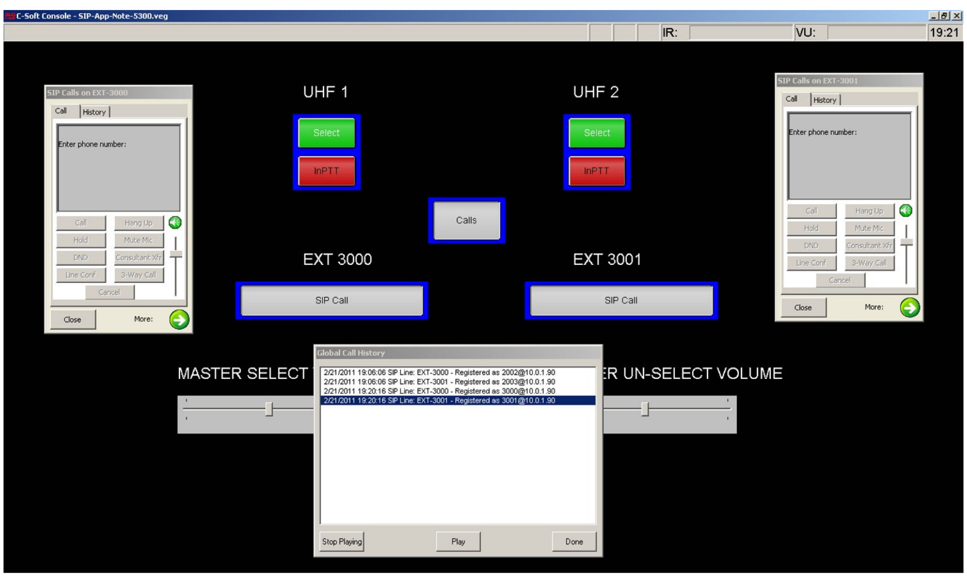


Fig. 20 Test C-Soft Console with 2 SIP resources active.

From here it is just a matter of entering a number and clicking on the Call Button. You can do only what the SIP Server allows.

If you are using the *Standard SIP Client*, the only difference would be the SIP Call dialog box that appears will be different.

C-Soft Designer / Runtime versions and SIP

This C-Soft console can be built on any 5.x version of C-Soft Designer. The SIP feature is simply enabled with a key.

To be the most flexible I have built my base console on C-Soft Designer 5.0. This allows me to use the C-Soft Designer versions 5.2 and 5.3 to open and copy the 5.0 veg file and then just re-save as a 5.2 or 5.3 C-Soft veg file with no problem. I generally save the veg file with 5200 or 5300 in the file name so that I know which is which.

The 5.0 and 5.2 C-Soft Designers can't open a 5.3 veg file. So once you build a console in 5.3 it won't work with 5.0 or 5.2 C-Soft Designer or Runtime. Currently the only recourse is to rebuild the console.

In versions 5.0 and 5.2 of C-Soft, any SIP key will be interpreted as a *Standard SIP Client*. The *Enhanced SIP Client* is only available in C-Soft 5.3.

Good Luck.

Phase 4 Design, Inc. is located in the Pacific Northwest, near Seattle, Washington. We have been providing solutions to complex technical problems since 1989. Our nearly 40 years of combined experience in the areas of Two-way Radio / Public Safety / Paging, Telephony, LAN / WAN Network Design and Software System Design, makes Phase 4 Design, Inc. uniquely qualified to support Telex Radio Dispatch System Design, Implementation and Maintenance. Services are available Worldwide.

For more information contact Dave Grant at dgrant@hostnw.net or call 425.402.7308. Visit our website at www.phase4.org.

