Elastix Application Note #201201111:

Elastix Basic Fault Finding Information

Title	Elastix Basic Fault Finding Information
Author	Bob Fryer
Date Written	11th January 2011
Revision	1.0
Replaces Document	Elastix – Basic Fault Finding & Techniques
Tested on Elastix Version	2.2
Backward Compatible	2.0+ (most do apply to earlier versions of Elastix)
Elastix Level	Beginner to Experienced
Linux Level	Beginner to Experienced
Network Level	Beginner to Experienced
Latest Document Source available from	www.elastixconnection.com
Credits	N/A
Licence	GNU FDL

Foreword

These application notes are intended to be a guide to implement features or extend the features of the Elastix IP PBX system.

Whilst many (but not all) guides available are basically a random collection of notes, usually while someone is implementing a feature for themselves, these guides are meant to be more definitive guide that has been tested in a lab with specific equipment, and particular versions of Elastix.

Finding information on the Internet can be haphazard due to the lack of document version control, lack of attention to software versions, and in some cases they are wrong. Then you have the cross pollination issues, where a guide has been done for another distribution, which may or may not be applicable to your Elastix system.

You will note on the front page of every Application note written in this way, will be an easy to read summary, regarding the Elastix system it was tested on, when the document was written, whether it is backward compatible, and the level of expertise needed to accomplish the implementation.

These application notes are written up and tested in a lab that has been specially setup to write these notes. This includes

- 5 x Elastix IP PBX Hardware with a mixture of SIP only, Digium, Sangoma, OpenVox Cards
- 1 x WAN Simulator (including latency, jitter, random disconnects, random packet drop)
- 8 x Consumer / Business routers, including Drayteks, Cisco 1842, Cisco 877, Linksys WRT54GL
- 2 x IBM XSeries servers running VMware with 8 images of various versions of Elastix IP PBX
- 1 x Standard Microsoft SBS Network providing DHCP and DNS and Mail system
- 2 x Linux Servers

The Elastix IP PBX systems, both hardware and Virtual based have image systems to refresh the systems to limit infection from other testing. Combined with a range of Phones, which include Aastra, Linksys, Cisco, Yealink, it provides a reasonable cross section of typical systems currently in the field.

These application notes are not just done in isolation either. Behind them is over 6-7 years of commercial implementation of IP PBX systems, utilising these methods and concepts. The Lab is just used to reconfirm the implementation in a less production like environment.

How you use these application notes is entirely up to you. However, it is highly recommended that in the first instance, that you follow the notes and configurations in their entirety (except for IP addresses) of course. If you follow it exactly, then it will be easier for others to assist you when you do have an issue.

Trademarks used in these guides

Asterisk® is a *registered trademark* of DIGIUM, Inc
FreePBX is a Registered Trademark of Bandwidth.com
Elastix is a registered Trademark of Palosanto Solutions

Contents

Introduction

So by now, you have installed Elastix....Congratulations!!!

For many, it is probably their first introduction to a Linux based server or you may have previously installed a Linux desktop system and have a base understanding of what Linux is.

For those that are not aware, Linux is generally split up into two main implementations:

- Workstation
- Server

Realistically there is actually no real difference between the two implementations. Some of you may have come from a Windows Environment, where you have a separate product for Windows Server and Windows Workstation, but basically, at least at the lower levels they are generally the same product. Microsoft choose what applications that they add to the base OS, as well as limiting some of the DLL's for licencing e.g. limiting 20 connections and 16000 open files on Windows 7. The same goes for the Server. They limit how many users can connect based on your Licence.

So with Linux, you will find on the popular distributions, that both the workstation and the Server install is done from the same disk (although some distributions are trying to make it easier for users by differentiating).

Your Elastix system is basically a Linux Server. It does not contain a desktop component, but does include a Web Server component which is how you configure the system via the Elastix GUI or FreePBX GUI. These GUI's basically read, manage and write out the configuration files. Otherwise you would have to write and manage these files yourself, increasing the complexity of what are now simple tasks with a GUI.

However, one of the issues with a GUI interface is that 90% of users never see what is happening underneath and therefore don't understand what is happening underneath. This becomes an issue when you need to do something different, or something you have installed, is causing the system to fail, or there is a bug in the system.

The purpose of this guide is to take you through some basic fault finding techniques including where to find some of the information that you need. Some of these techniques are not unique to Linux systems, but every once in a while you need to be reminded how to diagnose and correct faults. Some have the aptitude and experience and do it naturally. Some do not and no matter how much you tell them to break it down in to simple blocks, they race ahead missing vital clues.

Responding to Forum Posts for Information

One of the most common places that you will use to try and resolve any problems with your Elastix system is the Community Forums on www.elastix.org. It's the right place to go.

However, one of the common mistakes is not providing enough relevant information about your issue, which can result in your post being skipped over or ignored. No one deliberately wants to ignore your post, but the more relevant information that you post has, the more likely you are going to engage someone to assist you.

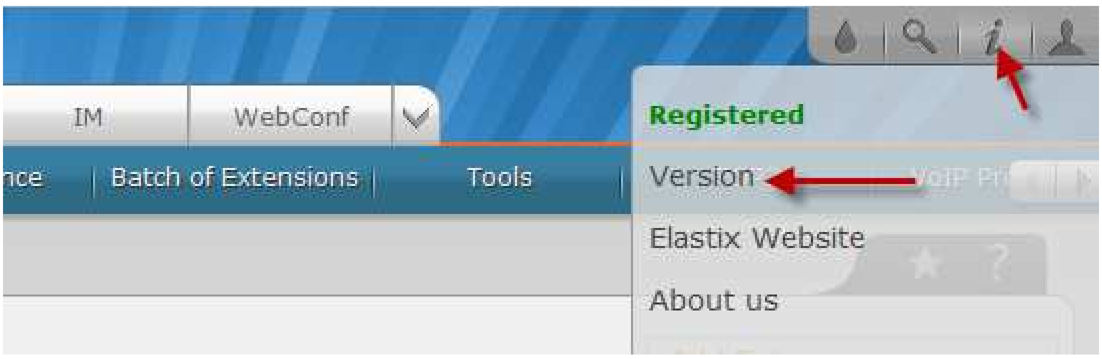
So this chapter will take you through some of the common requests that someone will ask you to provide, usually for your second post, but if you can provide this information in your first post, the more likely you are to engage someone to assist.

Versions

One of the first things to provide is your Elastix version. Roughly 80% of the posts do not have a single line about what version you are running. Just for your information, of all the versions that are probably still in use today, there are 13 (including a couple beta's), then add to the mix the possibly updates in between, you have a large number of possibilities. So as a minimum, at least provide a couple of critical lines such as

Elastix 2.2.0-21
Freepbx 2.8.1.9
Dahdi 2.4.1.2-5
Asterisk 1.8.7.0

These can easily be obtained by moving your mouse over the "i", and the menu will pop down, and then selecting version like the screenshot below.



You will then see the following screen

Details of package versions

(Text Mode)

Name	Package Name	Version	Release
Kernel			
	Linux(i386)	2.6.18	238.12.1.el5
Name	Package Name	Version	Release
Elastix			
	elastix	2.2.0	21
	elastix-vtigercrm	5.1.0	8
	elastix-asterisk-sounds	1.2.3	1
	elastix-firstboot	2.2.0	8
	elastix-agenda	2.2.0	7
	elastix-email_admin	2.2.0	11
	elastix-my_extension	2.2.0	6
	elastix-extras	2.2.0	1
	elastix-red5	1.0.0	1.rc1.svn4199
	elastix-fax	2.2.0	6
	elastix-security	2.2.0	9
	elastix-reports	2.2.0	8
	elastix-pbx	2.2.0	18
	elastix-addons	2.2.0	6
	elastix-im	2.2.0	2
	elastix-conferenceroom	2.2.0	1
	elastix-framework	2.2.0	22
	elastix-system	2.2.0	16
	elastix-a2billing	1.9.4	1
Name	Package Name	Version	Release
RounCubeMail			
	RoundCubeMail	0.3.1	11
Name	Package Name	Version	Release
Mail			
	postfix	2.3.3	2.3.el5_6
	cyrus-imapd	2.3.7	12.el5_7.2
Name	Package Name	Version	Release
IM			
	openfire	3.7.1	1
Name	Package Name	Version	Release
FreePBX			
	freePBX	2.8.1	9
Name	Package Name	Version	Release
Asterisk			
	asterisk	1.8.7.0	0
	asterisk-perl	0.10	2
	asterisk-addons	1.8.7.0	0
Name	Package Name	Version	Release
FAX			
	hylafax	4.3.10	2rhel5
	iaxmodem	1.2.0	1.1
Name	Package Name	Version	Release
DRIVERS			
	dahdi	2.4.1.2	5
	rhino	0.99.4	2.rc1
	wanpipe-util	3.5.23	1

CLOSE ✕

So you can see that the elastix version is 2.2.0 – 21, and the Freepbx version is 2.8.1-9, and the Dahdi driver is 2.4.1.2-5.

If your issue is about faxing, you might also include the Hylafax and IAXModem versions as well as the ones mentioned previous.

You may not have an understanding of what component is at fault, so like many, they provide the full list of versions in their post. From the previous screenshot, you will note that there is a text mode hyperlink in blue. Click on this and you will have a text list of versions that you can copy and paste into your post.

Just remember, post this towards the end of your post. Describe your issue first, and readers will look at your versions once they understand what you are asking.

Posting Asterisk Logs

One of the other common requests is that someone will ask you to post your Asterisk logs or Asterisk Full logs. Note the request for full logs doesn't mean the whole log. It's the name of the log.

The Asterisk log is found at **/var/log/asterisk** and the name of the file is **full.** There is also rotated logs named **full.x** usually for the last few days.

Typically, someone will ask you to show your log relating to your issue. Now this does not mean post the whole full log. No one will bother reading this.

Ideally, if at all possible, it is best if you can provide a clean log. What this means is by providing a log of only that one process. Asterisk logs are realtime log. As the event happens it writes it to the log. If you have a busy office, and 2 or more calls happening at once, all of these events are written to the log at the same time. As you can imagine, trying to review a log with 2 or 3 calls going at once, the log lines are intermixed and painful to work through. Again, your ability to do this, could result in a better response to your post.

Try to locate the relevant part of the call log, in other words the area that you perceive to be the issue. Don't worry if you get it wrong, someone will normally tell you to expand the excerpt or provide more either before or after. The fact you have gone to the trouble, shows that you know what the log is for, and you will understand the request for more of the log.

These lines below are a perfect example of the most relevant part of the log. Ideally you might post say 10 lines before and maybe 10 lines after.

[Jan 11 14:29:40] WARNING[12690] app_dial.c: Unable to create channel of type 'SIP' (cause 20 - Unknown)
[Jan 11 14:29:40] VERBOSE[12690] app_dial.c: == Everyone is busy/congested at this time (1:0/0/1)

Always look at the whole call process in the log, making sure that you don't just paste the most obvious error, as sometimes, the cause is a smaller error 15 lines previous.

Posting Asterisk CLI Output

Sometimes you will be asked to post the Asterisk CLI (Command Line Interface) output.

This requires you to enter the Asterisk CLI which you do from the Linux Prompt with the command

asterisk –r –vvvvvvvvvvvvvvvv

The number of v's is not critical, it just increases the verbosity level.

The trick is in capturing this output. The best way if you using SSH (e.g. Putty), most applications have the ability to turn on logging of the session.

Turn this on, provide a log file name, and then enter the Asterisk CLI and perform the action that creates the failure or issue.

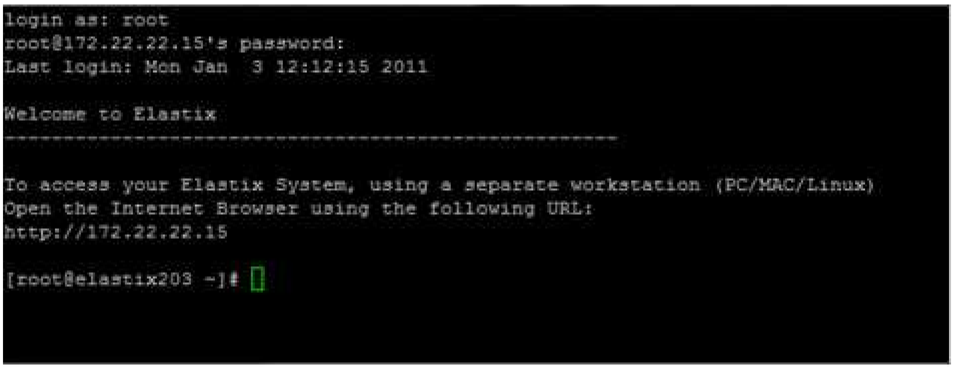
Once you are sure you have caught the issue, look at the log file you created (which will be on your local workstation), copy and paste the relevant lines, and post back on the forum.

See the chapter further in this document call Asterisk CLI for more information on the Asterisk CLI.

The Linux Console

The Linux console is something you are going to be dealing with a lot. The console is visible from the monitor you have attached to the Elastix system. When you first turn on the Elastix system, after the boot up process, you are presented with a login prompt like the following

At this point it is expecting you to type in **root** as your login and then it will ask for a password (which you set during the Elastix installation. Just remember the password will not show any characters or even * on the screen. Once you have successfully logged in, you will see the following (or very similar)



We won't go any further with the Linux Console with the Monitor attached as it is limited in its capabilities due to the screen resolution and the number of characters that can be seen across the screen. If you ever try and read the log files using this method, it is almost impossible to read due to the line wrapping. It is also more comfortable to do it in a comfortable chair, on a system you are very familiar with, with your mouse, cut and paste tools and access to the Internet Browser. However, it is still an essential tool should you not be able to get remote access to your system

Furthermore, it is sometimes useful to have on when you are performing other work on your system remotely, as it shows you what is happening at the lower hardware levels such as detecting your telephony hardware. These "warning/informational" messages are only sent to the /dev/console (the monitor directly attached to your Elastix system) but can also be seen in the logs, but there is nothing better than real-time. You can also use ALT-F1, ALT-F2 and so on both during installation and after installation to access other consoles. During installation some of these console screens are written to, and also after installation you may find further information written to these console screens. Have a look around, you may find something useful.

The Linux Console via SSH

By default the Elastix system is configured to accept SSH connections from any IP address. This can be changed so that it can only be accessed by select addresses, but for the moment this paper is about fault finding. Security will be covered by another paper.

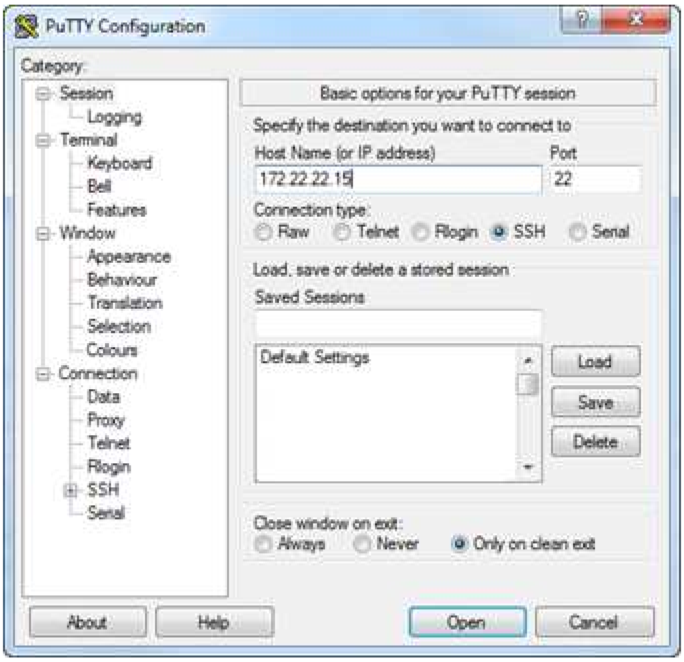
So what do you use to access your Elastix system remotely? Basically any SSH Terminal Client that supports the SSH protocol. These are available for almost any platform, whether it be a Windows system, Linux, Macintosh, Unix, the list goes on.

You may find that some include a FTP style program as part of the SSH Terminal Client, others provide two separate applications (usually from the same authors). You may even find that you have an SSH Terminal Client as part of your operating system applications (e.g. Linux Workstation).

One of the popular ones for most Windows users is **Putty.** It is a free download from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

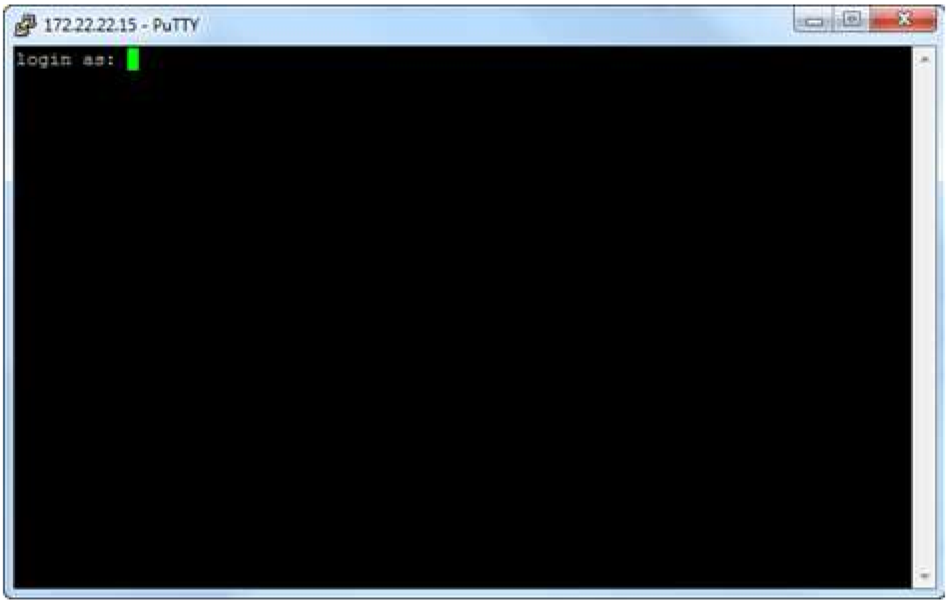
There is a lot more to this product than just a SSH Terminal Client, but for the purposes of this paper, we will restrict our discussion to using it as a Terminal Client only.

When you first start Putty, you will find a screen similar to the one below.

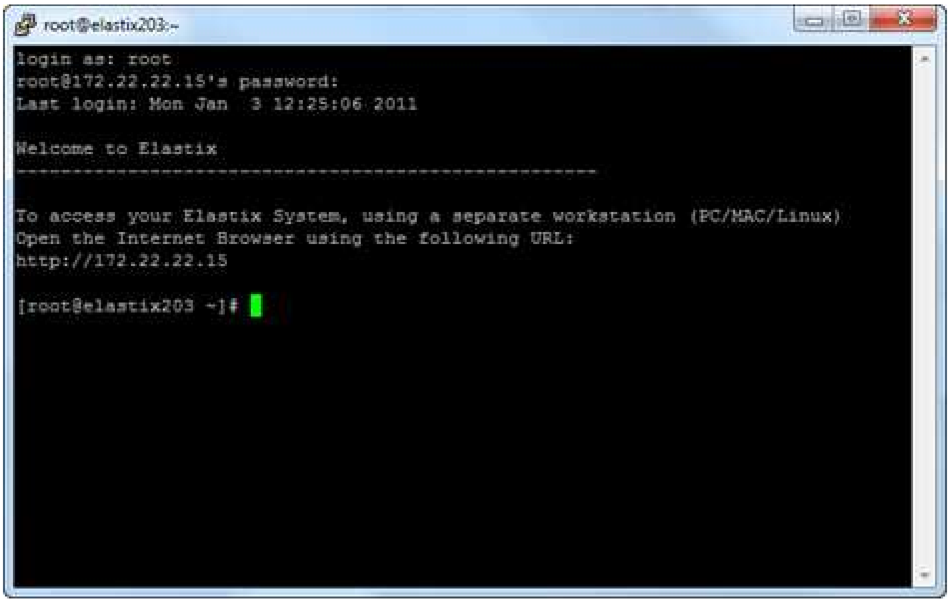


Most other basic SSH Terminal clients have a similar interface or at least follow the same concept. Like many other Terminal Clients, they also support Telnet, Rlogin and Serial, but we are interested in SSH.

So set your terminal client to SSH and enter the IP address of your Elastix system. You may also have the opportunity to enter the port number as well. In Putty's case it defaults to the SSH default which is port 22 which is the way that Elastix has SSH set (on port 22). You can now click OPEN and the following screen will pop up.



As you can see it looks identical to the console we spoke about in the previous section. Login as root again as we did in the previous section and we will end up on the following screen



As you can see, it is basically identical to logging in at the Elastix Console. Other than not seeing the “warning/informational” messages, everything is the same, except now you can use cut/paste, and hopefully sitting in a more comfortable chair.

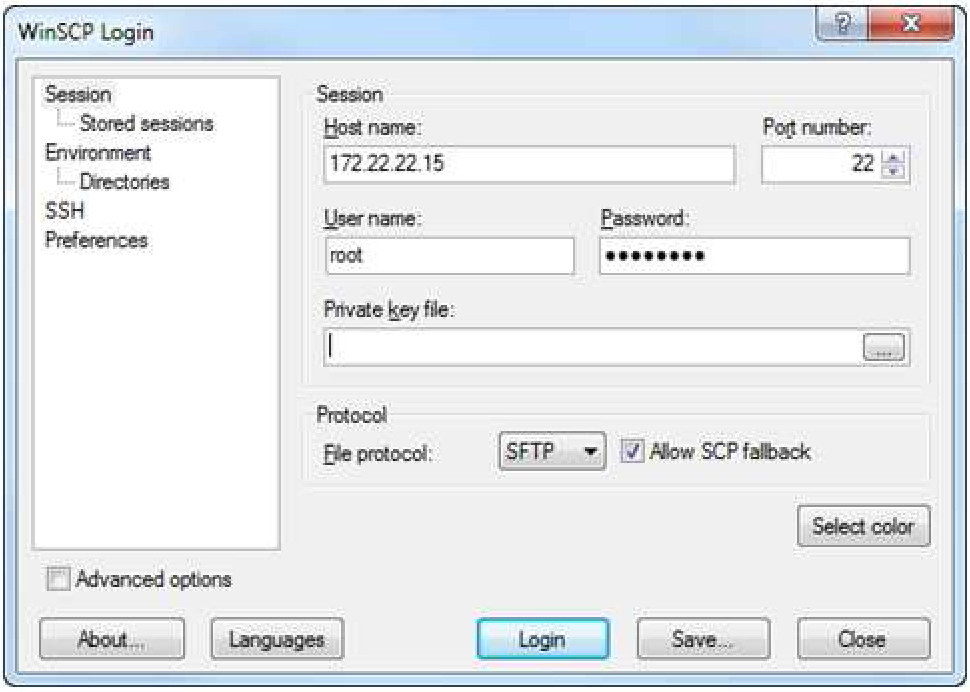
Before we move on, a couple more items on using an SSH Terminal client to access your Elastix system. There is no reason why you cannot open multiple SSH Terminal windows. This is useful if you are watching the Asterisk CLI (more on this later) on one Terminal Window and making changes on another. Using an SSH Terminal program will allow you to cut and paste both from and to the SSH Terminal Window. This is important so that you can document your changes easily either via Cut/Paste or via Screen Captures.

Transferring files to your Elastix system

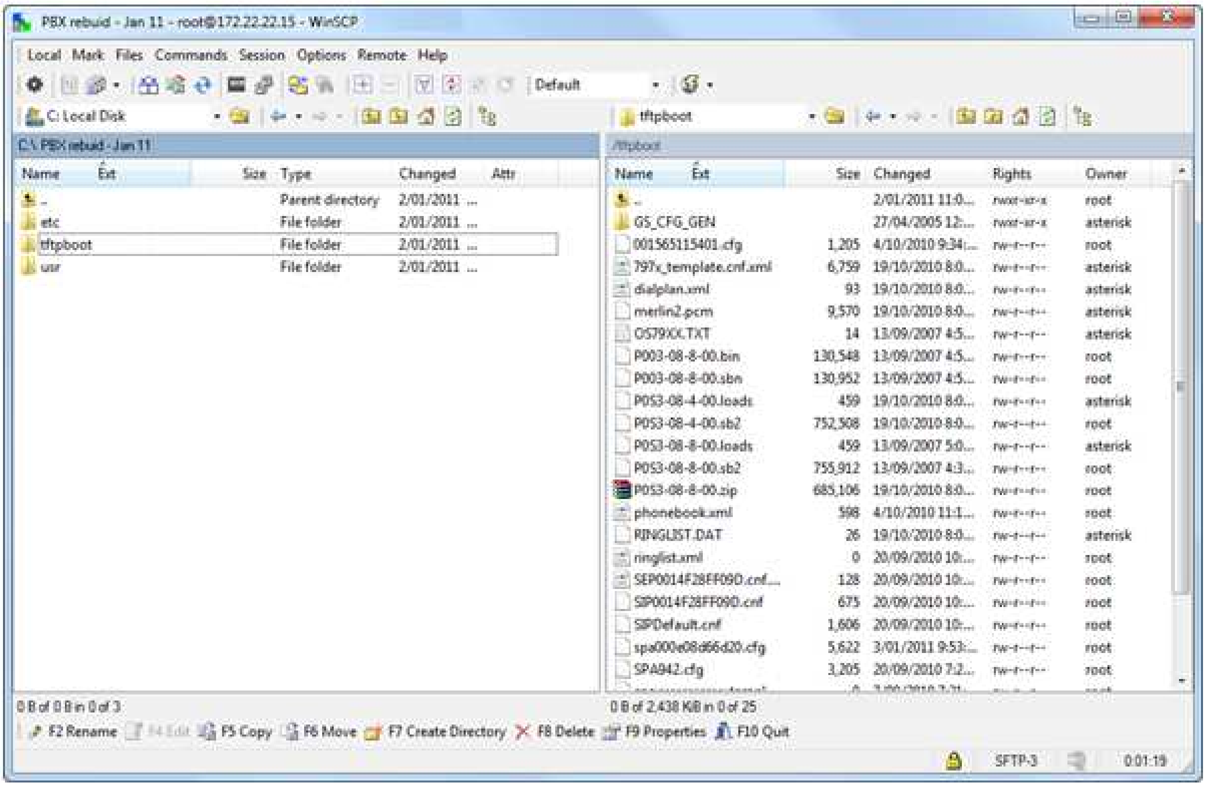
Elastix, which utilises Linux as its operating system, does not support FTP unless you setup an FTP Server. Instead it uses Secure FTP which uses SSH Protocol. Again this tool is available across a wide range of systems such as Windows, Linux and Macintosh amongst many others.

For Windows, the most popular program is WINSCP. It can provides an Explorer Style interface like many FTP programs, allowing to perform normal drag and drop operations from your workstation to the Elastix system and vice versa.

With a very similar configuration screen to Putty, you just enter your Elastix IP address, use the root login name and the password you used for your root login and click on login



You should now see the following screen if you have logged in correctly.



You can transfer files backwards and forwards, and even perform remote editing of files if you prefer. Other functions include creation on directories and setting of CHMOD rights (which you will read in the next section).

Caveat

One thing to keep in mind when transferring files from your Elastix system is that it is possible (usually by accident) to create a file of the same name on a Linux system (e.g. SIPdefault.conf and sipdefault.conf are treated as two differently named files). Quite often this occurs when you are editing a file and you realise that it had a capital Letter at the start. In a rush to get it finished, you create a new file of the correct name, and copy the contents across, and leave the incorrectly named file sitting there.

A few months later, you transfer the files from a particular directory from your Elastix system to your Windows system, and WINSCP will warn you, you are about to overwrite a file, and in your haste, you just accept. This is because those two files that Linux sees as two separate named files, Windows sees them as one file (because Windows does not differentiate upper and lower case characters as different in file names).

So be wary, don’t just accept the file over-write, as you may be over-writing the file that you wish to keep.

Basic Linux Commands

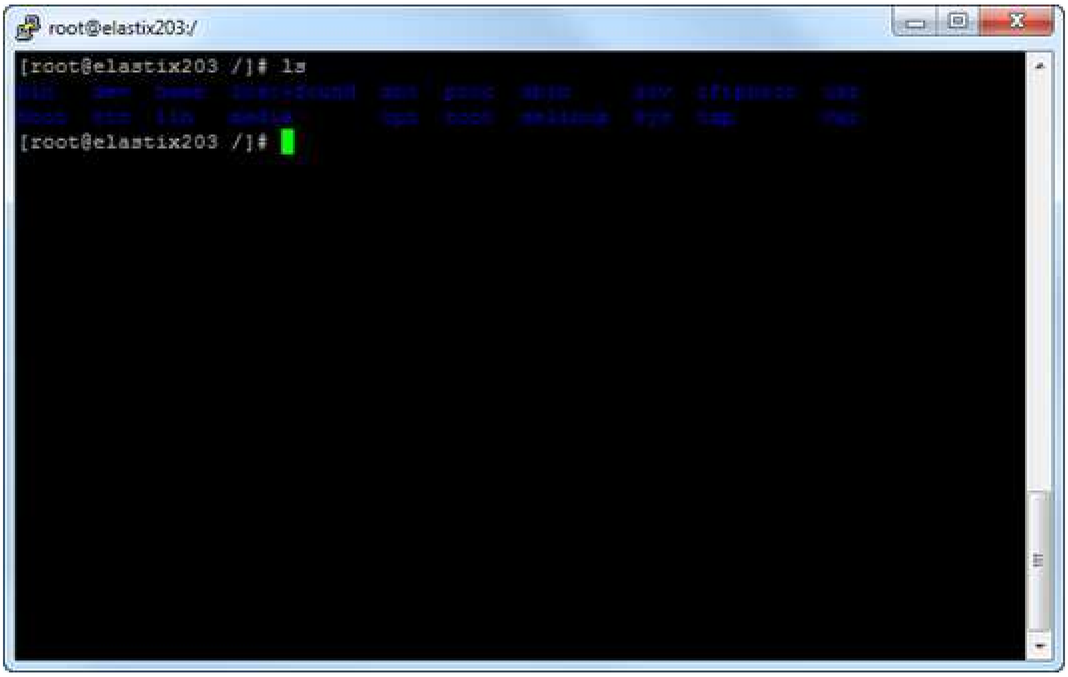
The purpose of this guide is not to take you through Linux commands in detail as there are hundreds of books and thousands of Websites covering this subject.

An important point which is sometimes painful or hard to grasp coming from or predominately working in a Windows environment is that uppercase and lowercase matters in Linux.

Here are a few basics we will cover that will make your life easier and relate to the majority of the commands that you might use with Elastix.

ls

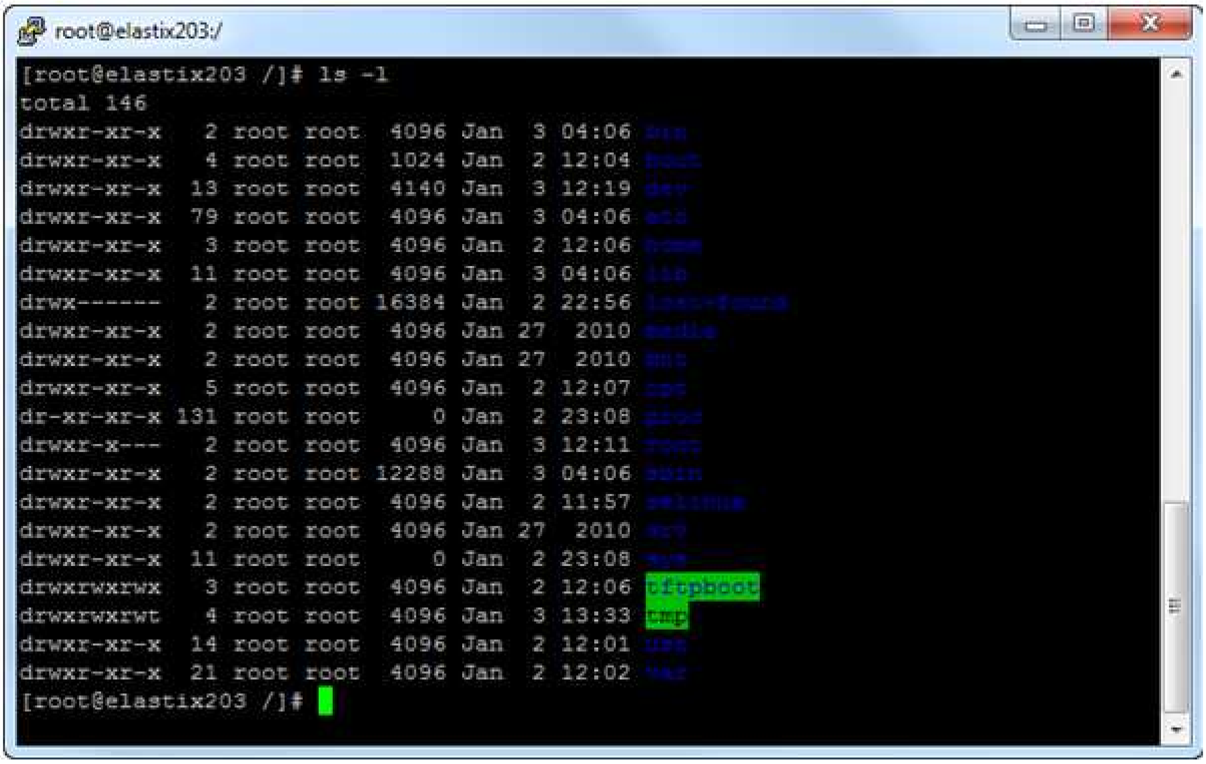
This will list the files and directories in the current directory



As you can see, just the names, useful to check a name of a file or directory and that it exists.

ls –l

This will list the files and directories with extended information including the file name, date information and size, file owner and group of each file and directory rights that are set for owner, group and public. (more on this later)



cd

By itself it does nothing except return to you to the directory called /**root.** This is normally seen on the console as **~.**

One thing to be aware of is that this directory called **/root** is not to be confused with the root directory. This **/root** directory is actually the home directory of the root user, not the root directory which most documentation refer to which is at the top of the directory tree.

cd example : **cd /**

This command as you will see actually returns you to the Linux root (the top of the directory tree), and we performed an **ls** command to confirm that we are at the top of the directory tree

cd example : **cd /var/log/asterisk**

This is an example of a Change Directory command that changes to the **/var/log/asterisk** directory. You will probably be using this command a lot, so hence the example.

cp example : **cp /var/log/asterisk/full /var/log/asterisk/templogfile**

This is an example of the copy command. The colours mean nothing except to show the arguments clearly, in this case where the file is to be copied from and where it is to be copied to. Ignore the example command as the chances you need to perform this exact command is minimal; it is just to show how to use the **cp** command

mv example : **mv /var/log/asterisk/templogfile /var**

This is an example of the move command. It copies the file from one directory to another, and deletes the original.

mkdir example : **mkdir /usr/src/asterisk1.6**

This command creates a directory. The example is something you may come across if you need to make a directory for some source code or need to access some documentation.

less	example : **less /var/log/asterisk/full**

This command allows you to view the file. You can scroll up and down, as well as search for words or patterns. The example shown is probably something you would type often to view the asterisk logs. One of the common things that you will want to do is perform a search on the Asterisk logs. It is worth reading more on what you can do with LESS, but as a start to perform a search for a particular expression such as a time stamp, or an extension, or phone number called, you move to the part of the log that you want to search from (e.g. top, middle, particular time), and press the **"/"** key. Such as

/12:11:13
/0418555555
/SIP/201

This command will find the next occurrence and highlight all occurrences making it easier to skim though the log file.

nano	example : **nano /etc/dahdi/system.conf**

This command utilises the editor that has been installed. There are a lot of editors including the stock standard VI Editor that comes with most distributions of Linux and Unix. Nano was only included in the standard Elastix distribution recently Elastix 2.0 (possibly earlier). If you don't have it, you can add it using YUM (discussed further in this document).

> DISCLAIMER : Some of the commands, especially CHOWN and CHMOD can have major security implications for your Elastix system. It is strongly recommended that you learn what these commands do &the security impacts they can have, before you start using

chmod	example : **chmod 777 \var\templogfile**

This command (normally executed under the user called root) will allow you to change permissions on files or directories. The number represents the rights you are going to apply to the file for each level of user. The first number is for the file owner, and 7 provides read/write/execute rights to the owner of the file. The second number is for the group, and again 7 provides read/write/execute rights, and the final 7 provides the world (think of it as everyone if you are from the windows world).

This is represented when you do the ls –l command as

-rwxrwxrwx : which is also the same as the binary notation which is as follows
111 | 111 |111 : which if you know your binary, translates to the numbers
7 | 7 | 7 : which as you can see is the notation we used in the example.
Owner | Group | Everyone : which each of the digits represent

This is not meant to be a full tutorial on CHMOD, it goes much deeper than that with many more options, twists and turns, but it is a start. I recommend reading many more articles on CHMOD before you experiment, and like always don't experiment on a production system

chown	example : **chown asterisk:asterisk \var\templogfile**

This command allows you to set the owner of the file or directory and/or the group associated with the file or directory. It is worth getting to know this command and again the example show above us quite simple and there are many more options available. As you can probably guess, this command works hand in hand with CHMOD.

Basic Fault Finding Rules

Now we have gone through a few basics, now we can move onto some basic fault finding rules.

Rule 1: Always have a backup or at least a rollback plan

This is a rule that is universal, not just with Elastix but just about any IT related system.

You have a few choices, one being the Elastix Backup System, the other being the Freepbx Backup system, and if you are fortunate enough, a disk image system application installed or external software that does the same thing.

As a minimum, I always perform a FreePBX backup. Partly due to the fact that this backup takes less than a minute or two, and if everything goes wayward, it takes about 20 minutes to re-install your Elastix system, and about 3-4 minutes to reload your Freepbx backup. Sure it means that you don't have all your peripheral products like Elastix Fax, Openfire, Sugar, Vtiger, Email settings restored, but you have the basic Elastix PBX back to the way it was before you changed anything. For systems that are using the basic PBX functions, this restoration is reliable. This is also a good technique for production systems. Being able to bring a system back online after a complete failure with 30 minutes, at least with basic phone calls, can remove a lot of the pressure.

If you have a feature laden system, then using the Elastix backup will provide a little more confidence that you bring all the systems back online.

Rule 2: Break your complex issue down into smaller parts

Take a piece at a time and understand its operation. Yes this means doing some research and it may take some time. In a complex system such as Elastix, just changing things on a whim is in 90% of cases not going to correct an issue. In fact, it generally makes things worse.

Rule 3: YUM is not the answer to every problem

One of the features of Elastix is the use of the Centos Distribution which uses YUM as an update mechanism removing the need to understand installation of RPM's and worry about dependencies. Whilst this is a great feature, it can also become your worst enemy.

YUM is useful if you know what your issue is and that there is a fix in the repository to fix it. And even then, you issue a yum update for that component. There is more information on YUM updates further in this paper.

Rule 4: There is no such thing as magic (attributed to Richard White – an ex-boss)

One of the better lessons drilled into me over 20 years ago. When we are fault finding, we are always looking for the quick fix. Generally if we fixed it and we didn't understand how we fixed it, then there is a great chance that we fixed nothing and the problem is probably going to come back. Quite often the environment changed whilst you were fault finding, which appears to have resolved the issue. Agreed you might have had a problem with your SIP provider and they have fixed things on their end, that's great, but how can you be so sure, when you couldn't diagnose the fault in the first place. So what can you do if you have one of these issues that seems to fix itself....next rule

Rule 5: Test, test, test, test, test

This rule comes into play whether you had a scenario like the one described in Rule 3 or whether you have made some changes to your system. This also applies if you make changes to the environment around the Elastix system such as the Internet router, network switches, IP address changes. Even the simple changes as they are always the ones that will bring you undone.

This means not a quick simple test of the handset, and make a call, but in most cases restarting your Elastix system, restarting the phones, possibly restarting the router and switches, basically simulating a complete power failure. You and/or your client need to be confident that the system is going to restart itself. This doesn't just mean hardware, but many of the modules that Asterisk is made of. You can have a small error in a SIP configuration file that can cause the SIP module not to load in Asterisk. Asterisk looks like it is up, the IAX trunks are working, but your system completely fails on SIP communications. After about 10 mins, the phones start showing offline; the users start to panic as there are no phones working. Your job has now become twice or three times as hard to resolve as you have pressure to deal with.

If you are using NAT on a router, this is another part in the mix. NAT can take 10-20 minutes to breakdown. You might believe everything is working, you leave the site, and 20 minutes later, they call to say that they are not getting calls in.

Yes it takes time to do this testing, but it is necessary!!! In many cases, that fault you thought just magically disappeared, usually will reappear.

Rule 6: Check the logs regularly and understand them

The logs are really one of the best tools for finding faults on your system. Up to 80% of the problems that you encounter can be found in the logs.

Rule 7: Perform the research

The more you understand the system and how it works, the easier it is to find any faults. No matter how good you are, or you think you are, there is a lot more you can learn about the system. Every release of Elastix, every upgrade to Asterisk, every upgrade of FreePBX, every update to components of Elastix, there is something new to learn.

Take the time to read the release notes. If there is something in there that you don't understand, then perform the research. If you note that Asterisk now uses XMPP and don't understand what XMPP is, then take the time to at least learn the basics.

Even the components already built into Elastix, take the time to understand how they have been implemented. Learn how FreePBX works with Asterisk, and what its role is. Learn how the macros have been designed and why the system works. Learn the difference between the original way of writing Asterisk configuration files and why the FreePBX written files work.

The more you learn, the more you will understand when something goes wrong.

The Asterisk CLI (Command Line Interface)

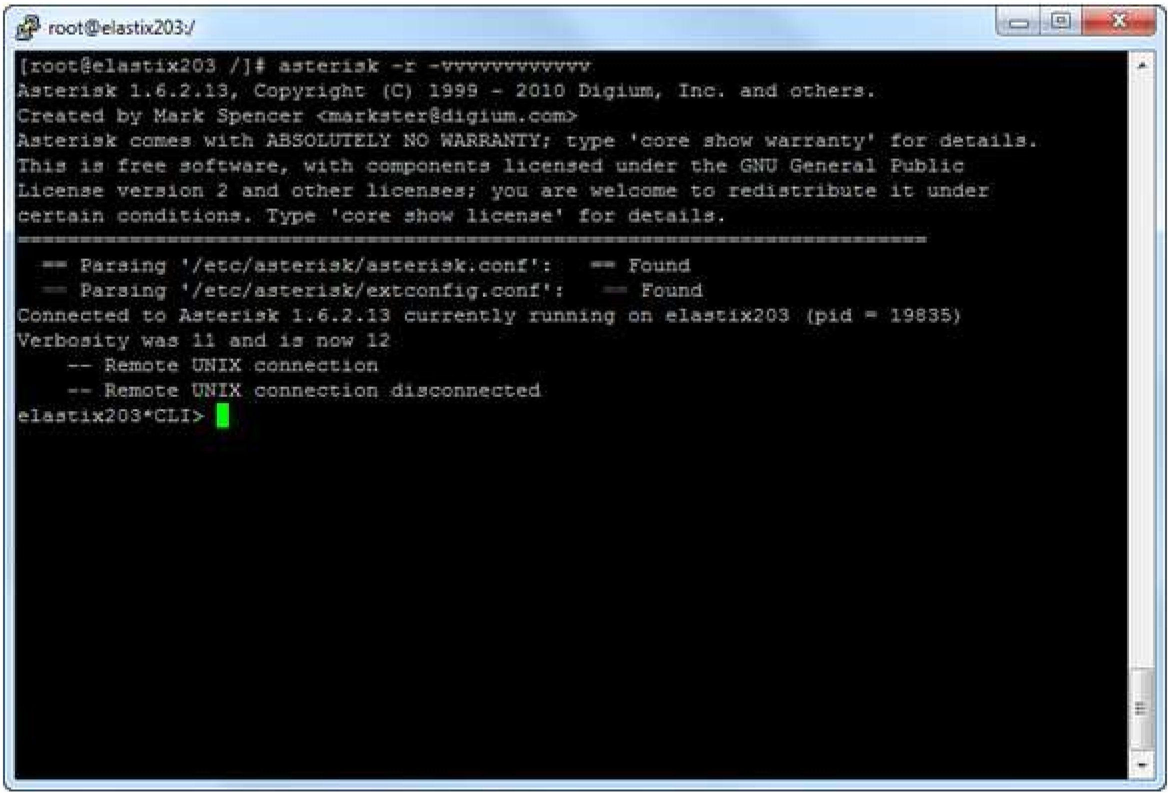
This is probably one of the most powerful tools for diagnosing problems with your system, particularly if your system is failing to perform a particular function, or your Voice Provider is rejecting your calls, or your system is rejecting calls from your Voice Provider.

From the Linux prompt, all you need to type is

asterisk –r –vvvvvvvvvvvvv

Don’t panic with the number of v’s in your command line, it relates to the verbosity or amount of detail that the CLI shows.

You should see a screen similar to the following




```
[root@elastix203 /]# asterisk -r -vvvvvvvvvvvv
Asterisk 1.6.2.13, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=========================================================================
  == Parsing '/etc/asterisk/asterisk.conf':   == Found
  == Parsing '/etc/asterisk/extconfig.conf':   == Found
Connected to Asterisk 1.6.2.13 currently running on elastix203 (pid = 19835)
Verbosity was 11 and is now 12
    -- Remote UNIX connection
    -- Remote UNIX connection disconnected
elastix203*CLI>
```

We are now looking at and are inside the Asterisk CLI. If you are on a production system, your screen will be scrolling with text. If you are on a test system, you may see the occasional line pop up, but pick up a phone and dial, you will see a flurry of activity.

There are many guides on the Web on what commands you can use in the Asterisk CLI, so again this article will not be a definitive guide to the Asterisk CLI. It will however give you an start.

One thing that requires mentioning is that as of Asterisk 1.6 (Elastix 2.0), there was a reorganisation of the command structure in the CLI to provide a more intuitive hierarchal style command set. So be aware that some guides on the web may be a little outdated.

A guide to the commands can be found here http://www.voip-info.org/wiki/view/Asterisk+CLI but as mentioned even this is out of date.

What we will do however, in line with the level and focus of this paper is run through some of the common commands that are useful for troubleshooting.

It should also be noted that like many CLI systems (including Cisco's), you can commence typing the command such as **sip show** and you can then press space bar and then a question mark and it will show you the next possible arguments of the command. The following shows the result of using this method.

```
root@elastix203:~
[root@elastix203 ~]# asterisk -r -vvvvvvvvvvvv
Asterisk 1.6.2.13, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=========================================================================
  == Parsing '/etc/asterisk/asterisk.conf':   == Found
  == Parsing '/etc/asterisk/extconfig.conf':   == Found
Connected to Asterisk 1.6.2.13 currently running on elastix203 (pid = 19835)
Verbosity is at least 12
elastix203*CLI> sip show
channel         channels        channelstats    domains         history         inuse
mwi             objects         peer            peers           registry        sched
settings        subscriptions   tcp             users           user
elastix203*CLI> sip show
```

As you can see, we can see the possible arguments. So now you might type **sip show peers.** If you try the space bar and question mark again, it will probably beep at you, which means you have no further command arguments available, just press enter. If we use the command **sip show peer** it will also beep at us, but in this case it is waiting for further input such as a peer number or other argument e.g. **sip show peer 201**

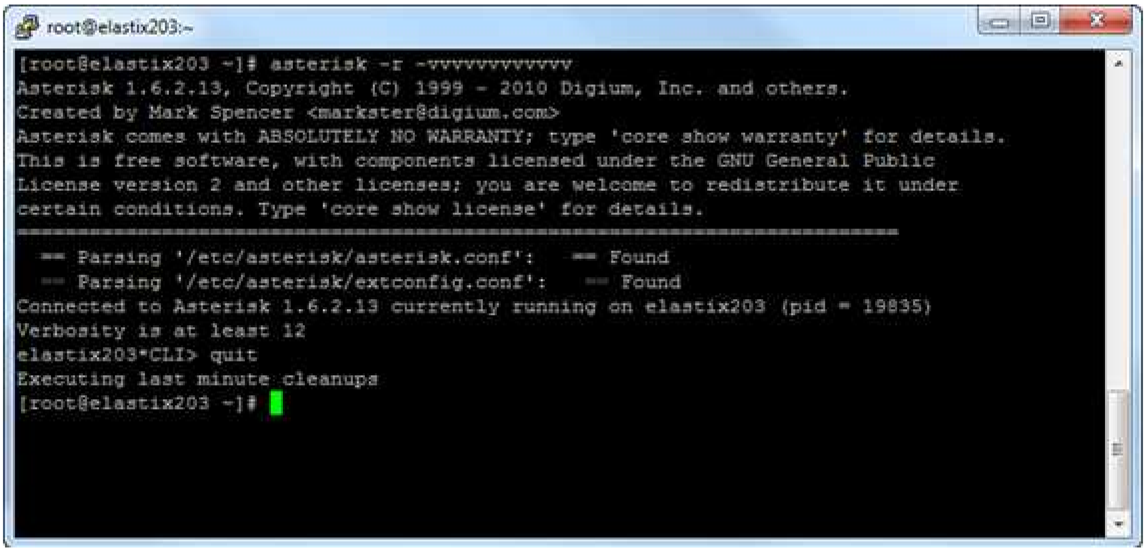
You can also hit the question mark when you first enter the Asterisk CLI, and this will provide you with a list of commands that are at the top of the tree.

Through this section you may see iax show peer 201 or iax show registry. SIP and IAX are two method protocols used for devices, trunks and so on. Whilst the most popular and default is sip, iax has its benefits as well. For the purpose of this guide, generally (but not always), if you can perform a sip show peer 201, then you will find that iax has a matching command iax2 show peer 201 (if the extension was using IAX instead of SIP).

One other thing, if you notice that you cannot use a particular command e.g. sip show peer 201, and in fact none of the SIP commands work, then you will find that SIP has not loaded. Likewise the same for the IAX commands or the PRI commands. The Asterisk & the Asterisk CLI is a dynamic environment. If the SIP module does not load as part of the Asterisk system, likewise the commands for the CLI will not be available either. Quite often this issue comes down a configuration problem and Asterisk will continue to load, all looks good, except the failed module will not load.

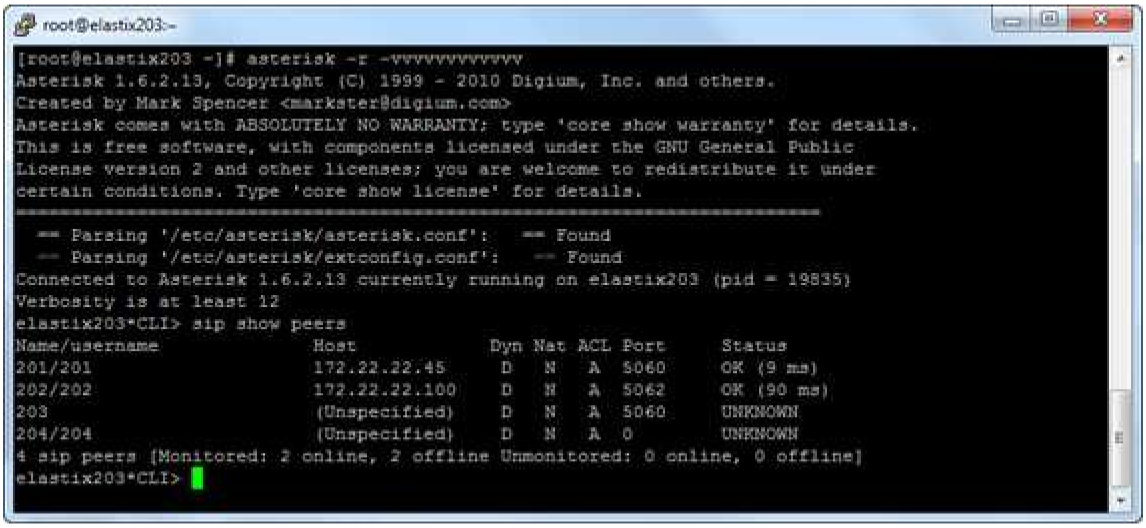
So now onto the commands that you might use regularly

quit



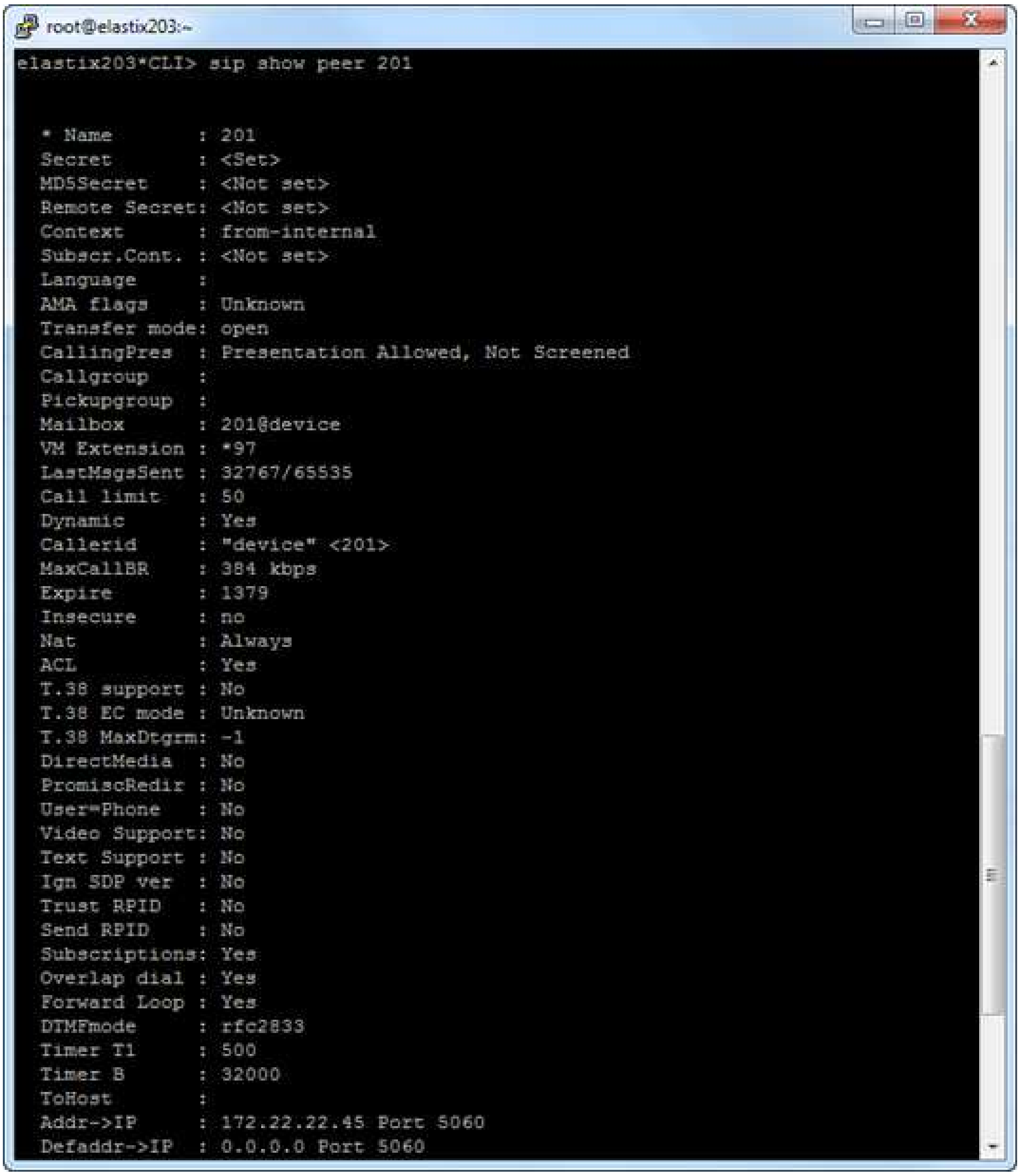
Probably the first command you need to know is the quit command. As shown above, it basically just allows you to get out of the Asterisk CLI back to the Linux prompt. It does not shutdown Asterisk, just the Asterisk CLI shell.

sip show peers



This is probably one of the most used commands in the Asterisk CLI. It shows you what SIP devices you have defined in your Elastix system, and whether they have registered or not. In the example shown above it shows that extension 201 and 202 are defined and a SIP device (in this case IP Phones) are registered. It also shows that extension 203 and 204 have not registered.

sip show peer 201

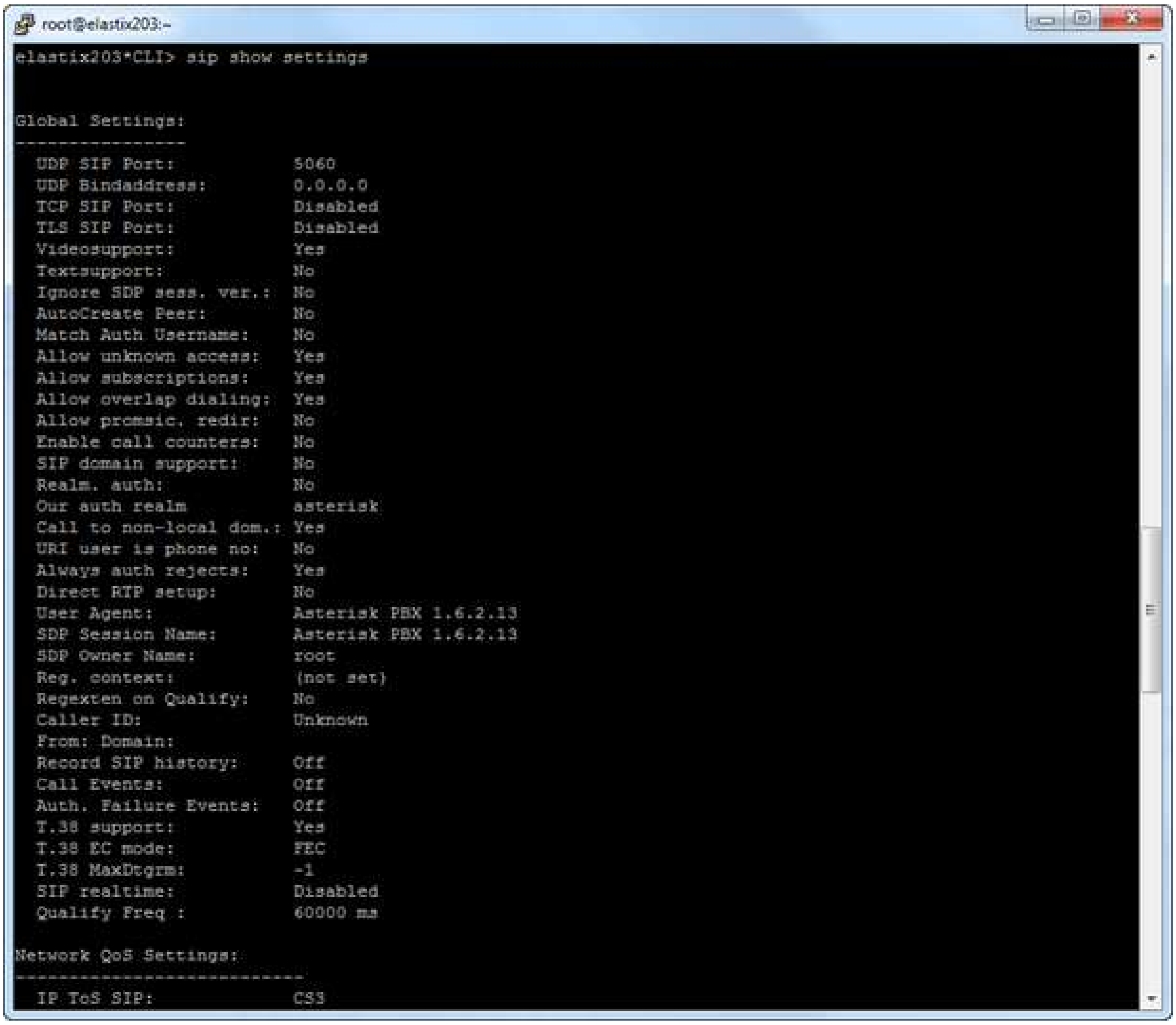


```
root@elastix203:~
elastix203*CLI> sip show peer 201


  * Name       : 201
  Secret       : <Set>
  MD5Secret    : <Not set>
  Remote Secret: <Not set>
  Context      : from-internal
  Subscr.Cont. : <Not set>
  Language     :
  AMA flags    : Unknown
  Transfer mode: open
  CallingPres  : Presentation Allowed, Not Screened
  Callgroup    :
  Pickupgroup  :
  Mailbox      : 201@device
  VM Extension : *97
  LastMsgsSent : 32767/65535
  Call limit   : 50
  Dynamic      : Yes
  Callerid     : "device" <201>
  MaxCallBR    : 384 kbps
  Expire       : 1379
  Insecure     : no
  Nat          : Always
  ACL          : Yes
  T.38 support : No
  T.38 EC mode : Unknown
  T.38 MaxDtgrm: -1
  DirectMedia  : No
  PromiscRedir : No
  User=Phone   : No
  Video Support: No
  Text Support : No
  Ign SDP ver  : No
  Trust RPID   : No
  Send RPID    : No
  Subscriptions: Yes
  Overlap dial : Yes
  Forward Loop : Yes
  DTMFmode     : rfc2833
  Timer T1     : 500
  Timer B      : 32000
  ToHost       :
  Addr->IP     : 172.22.22.45 Port 5060
  Defaddr->IP  : 0.0.0.0 Port 5060
```

This provides more detailed information about the peer, in this case we chose 201. This is useful to confirm details including codecs supported by the device and also the codec negotiation order, as well as the useragent (so we can confirm that the right device is connected and also useful to confirm firmware versions of the device).

sip show settings



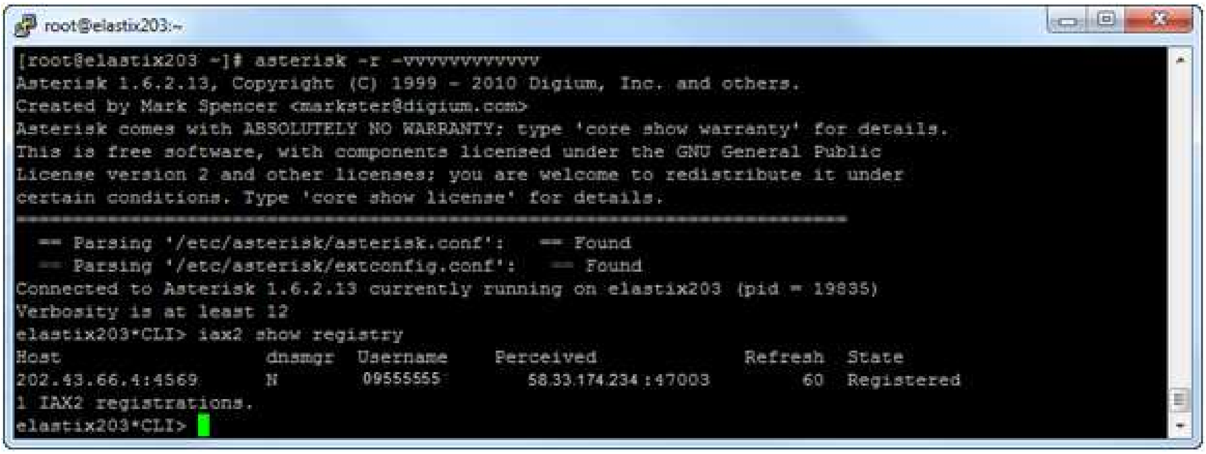
```
elastix203*CLI> sip show settings


Global Settings:
----------------
  UDP SIP Port:            5060
  UDP Bindaddress:         0.0.0.0
  TCP SIP Port:            Disabled
  TLS SIP Port:            Disabled
  Videosupport:            Yes
  Textsupport:             No
  Ignore SDP sess. ver.:   No
  AutoCreate Peer:         No
  Match Auth Username:     No
  Allow unknown access:    Yes
  Allow subscriptions:     Yes
  Allow overlap dialing:   Yes
  Allow promsic. redir:    No
  Enable call counters:    No
  SIP domain support:      No
  Realm. auth:             No
  Our auth realm           asterisk
  Call to non-local dom.:  Yes
  URI user is phone no:    No
  Always auth rejects:     Yes
  Direct RTP setup:        No
  User Agent:              Asterisk PBX 1.6.2.13
  SDP Session Name:        Asterisk PBX 1.6.2.13
  SDP Owner Name:          root
  Reg. context:            (not set)
  Regexten on Qualify:     No
  Caller ID:               Unknown
  From: Domain:
  Record SIP history:      Off
  Call Events:             Off
  Auth. Failure Events:    Off
  T.38 support:            Yes
  T.38 EC mode:            FEC
  T.38 MaxDtgrm:           -1
  SIP realtime:            Disabled
  Qualify Freq :           60000 ms

Network QoS Settings:
---------------------------
  IP ToS SIP:              CS3
```

This will produce a couple of pages (not all shown here). This command is useful to check all the SIP global settings. This includes almost anything to do with SIP such as registration expiry timeouts, QOS settings, whether your externIP and localnet variables are setup correctly, what Codecs are offered to SIP devices for negotiation, whether you have video support enabled, what transports (UDP/TCP) are being used, the standard DTMF settings being used and much more. This command is useful as the source of truth, no matter what the GUI is showing you, what line you manually put into the configuration files, this command is showing you what settings Asterisk is using based on the last parse of the configuration files.

iax2 show registry



This command will show what iax2 registrations are in place and what the registration state is. This particular system uses an IAX2 trunk, and it shows it as healthy with its state being registered. You may say see the state on a system with an issue as “trying” or blank...

pri show span 1

This command is useful for a system with a E1 or T1 Card. This will show you the status of the span

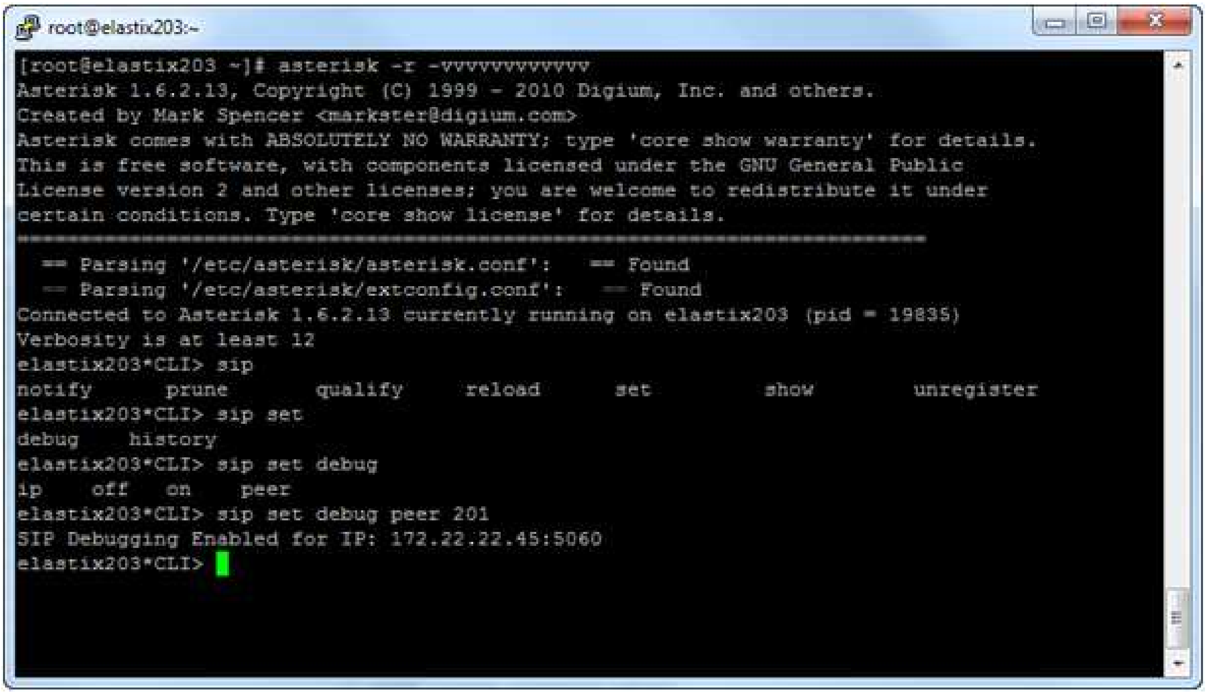
SIP and IAX Debug commands

One of the most powerful area of the Asterisk CLI is the ability to turn SIP and IAX debugging on. This is extremely useful if you are trying to setup a new Trunk with a provider that just wont register, or trying to find out why a phone won't register.

It is recommended that you learn and understand how SIP and IAX call flows work as again, this is not the purpose of this document. Even if you don't have an issue at the moment, it is well worth looking at the debug logs to see how a healthy system communicates, so that when you do have an issue that you want to resolve, you know what looks right and what doesn't look right

Again these commands will generate large amounts of material on the console and in most cases will be useless trying to read on the console. The good thing is that it also writes these logs to the standard Asterisk logs, so you can review them in more detail. This means you should also turn off debugging when you finish as it will generate large amounts of data in the logs (especially if you turn on global SIP debugging). So, instead of your logs rotating on a daily basis, they will rotate on size which means you may lose archived log data earlier than you expect.

sip set debug peer 201



```
root@elastix203:~
[root@elastix203 ~]# asterisk -r -vvvvvvvvvvvvv
Asterisk 1.6.2.13, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=========================================================================
  == Parsing '/etc/asterisk/asterisk.conf':   == Found
  == Parsing '/etc/asterisk/extconfig.conf':   == Found
Connected to Asterisk 1.6.2.13 currently running on elastix203 (pid = 19835)
Verbosity is at least 12
elastix203*CLI> sip
notify      prune       qualify     reload      set         show        unregister
elastix203*CLI> sip set
debug    history
elastix203*CLI> sip set debug
ip    off    on    peer
elastix203*CLI> sip set debug peer 201
SIP Debugging Enabled for IP: 172.22.22.45:5060
elastix203*CLI>
```

This command will only show you SIP debugging messages that relate to SIP device 201. This is useful if you are only trying to work out why extension 201 is failing registration. Here is a couple of typical lines with SIP debugging turned on

```
<--- SIP read from UDP:172.22.22.45:5060 --->
SIP/2.0 200 OK
To: <sip:201@172.22.22.45:5060>;tag=ea8941104ede692bi0
From: "Unknown" <sip:Unknown@172.22.22.15>;tag=as041e5409
Call-ID: 311b72c6157906ce6a0b4598040be019@172.22.22.15
CSeq: 102 OPTIONS
Via: SIP/2.0/UDP 172.22.22.15:5060;branch=z9hG4bK6705d5e7
Server: Linksys/SPA942-6.1.3(a)
Content-Length: 0
Allow: ACK, BYE, CANCEL, INFO, INVITE, NOTIFY, OPTIONS, REFER
Supported: replaces
```

It is worth taking the time to read the many SIP tutorials on the Web to understand what this packet is about, but in a nutshell, this packet is typical of the messages between the IP Phone and to the PBX.

Other debug commands that you can use are as follows

sip set debug peer on

This turns on SIP debugging globally showing all SIP traffic to and from the Elastix system. The amount of data is large, so you would only use this if you are really interested in SIP messages from all devices or if you are looking at the total interaction of a conversation from your SIP phone through Elastix to your Voice Provider

sip set debug ip 202.223.233.122

This allows you to debug only messages to and from a particular IP address such as your Voice Provider, so you can isolate traffic between your Elastix system and your Voice Provider only. This can also be used for instance to look at SIP traffic from a phone as well, especially if your phone/device is not registering and doesn't have a SIP extension. It is commonly used to see if the phone is communicating at all with the Elastix system

iax2 set debug ip 203.33.42.123

Basically the same as the sip debug command above but for IAX. As mentioned before, many of the debug commands for SIP are also available for IAX as well.

sip set debug off

Turns off all SIP debugging. You should get into the routine of making sure this is off once you have finished performing SIP debugging

Checking Basic Network Issues

One of the most misunderstood issues with IP Telephony is the network. IP Telephony relies on a network being reliable.

The fact that your workstations connect to the server and can see the internet is absolutely no indication that your network is working reliably.

Most of your workstation and server applications, the internet browser, and almost every other application use the TCP protocol. What this means is that every packet of information from one device to another is guaranteed of delivery even with substandard networks. This is because every packet is acknowledged on receipt. If for instance the sending computer does not receive acknowledgement, it resends it. If the network is that bad, normally the O/S and generally the sending application will throw up an error.

IP Telephony generally uses the UDP protocol, for both signalling and also the voice data. UDP does not have this acknowledgement overhead (necessary as voice cannot handle this overhead), and therefore, it has to rely on the network working reliably.

I have seen a network, where the business had run for several years with a faulty network. The reason for the fault was a “jabbering” Ethernet connection. However it was not identified until they implemented an Elastix system. The symptoms ranged from the following

- Lower call volume and/or static on the line
- BLF Lights not updating intermittently
- Calls not hanging up
- Phones going offline

All of these are related to packets being lost on the network.

Your Elastix system running on Centos has one of the simplest measures for performing quick testing. The ping test from the Linux console

Usually the best method is pinging each of the phones from the Linux console

In the screenshot below, I have pinged a phone on 172.22.22.45 using the ping command

ping –s 160 172.22.22.45

You might be wondering why the –s 160 option. Just something I have always done simulating a typical G711 packet size, as sometimes the issue may only show itself up on larger packet sizes.

```
[root@elastix22 asterisk]# ping -s 160 172.22.22.45
PING 172.22.22.45 (172.22.22.45) 160(188) bytes of data.
168 bytes from 172.22.22.45: icmp_seq=1 ttl=250 time=0.484 ms
168 bytes from 172.22.22.45: icmp_seq=2 ttl=250 time=0.477 ms
168 bytes from 172.22.22.45: icmp_seq=3 ttl=250 time=0.473 ms
168 bytes from 172.22.22.45: icmp_seq=4 ttl=250 time=0.484 ms
168 bytes from 172.22.22.45: icmp_seq=5 ttl=250 time=0.475 ms
168 bytes from 172.22.22.45: icmp_seq=6 ttl=250 time=0.479 ms
168 bytes from 172.22.22.45: icmp_seq=7 ttl=250 time=1.13 ms
168 bytes from 172.22.22.45: icmp_seq=8 ttl=250 time=1.16 ms
168 bytes from 172.22.22.45: icmp_seq=9 ttl=250 time=0.497 ms
168 bytes from 172.22.22.45: icmp_seq=10 ttl=250 time=0.499 ms
168 bytes from 172.22.22.45: icmp_seq=11 ttl=250 time=0.497 ms
168 bytes from 172.22.22.45: icmp_seq=12 ttl=250 time=0.477 ms
168 bytes from 172.22.22.45: icmp_seq=13 ttl=250 time=0.472 ms
168 bytes from 172.22.22.45: icmp_seq=14 ttl=250 time=0.474 ms
168 bytes from 172.22.22.45: icmp_seq=15 ttl=250 time=0.473 ms
168 bytes from 172.22.22.45: icmp_seq=16 ttl=250 time=0.478 ms
168 bytes from 172.22.22.45: icmp_seq=17 ttl=250 time=0.474 ms

--- 172.22.22.45 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16007ms
rtt min/avg/max/mdev = 0.472/0.559/1.163/0.216 ms
[root@elastix22 asterisk]#
```

Nothing special here, as the network is working well.

On a network that has some major issues, you will notice things like icmp_seq numbers missing out of the sequence. Other good indications are large fluctuations in round trip times.

Generally I let it run for at least 100 pings to get a good indication.

And this is not just for Local Area Networks, there is no reason why you cannot perform the same command and ping your service provider or the remote office or the remote phone via the VPN.

This is such a quick and simple test, but could save you hours of looking for problems.

Common files that you may want to access

Asterisk Logs **/var/log/asterisk/full**

These are the primary Asterisk logs. They contain a log of each action that occurs in the Asterisk system. This is one of the first places you want to look at when you are looking for a fault or problem. On a production system these logs can go forever, so it is a smarter method to have an idea of what you are looking for, such as a time stamp, a phone number or and extension. The logs are rotated daily or in some instances by size.

Security Logs **/var/log/secure**

Any tiem you login to the Linux system, or perform some sort of authentication (fail or pass) to the system it is logged here. This file is useful to see if someone is trying to gain access to your system via SSH that shouldn't be. Normally it will list the IP address the login or attempt to login came from.

Linux Messages Logs **/var/log/messages**

We mentioned about the warning or informational messages that come up on the main Linux Console (with monitor attached). These messages are logged to this file (amongst many others). You will find messages from the Kernel, NTP Daemon, Xinet Daemon, TFTP Daemon, and is worth reviewing if you are having issues with TFTP or similar

Mail logs **/var/log/maillog**

One of the major functions in the Elastix system is the mail function, this is used when you setup voicemail to email, faxing (which emails the fax) etc. The maillog file is the file that will tell you what is happening to your mail. You may find that your mail is queuing because of authentication issue, or DNS failure.

There are many log files in this **/var/log** directory and it is worth talking the time to review what logs you may be interested in

Dahdi configuration file **/etc/dahdi/system.conf**

This file holds the base configuration for your hardware telephony cards. It is automatically generated when you use the hardware detector in the Elastix GUI, and is refreshed every time you use the hardware detector. The reason why you may want to change this file is to replace the loadzone and defaultzone settings for your particular country. You may also wish to edit this file, to remove the echocanceller line if you are using that particular line for your fax which may or may not need it turned off.

Chan_dahdi.conf file **/etc/asterisk/chan_dahdi.conf**

This file may need to be modified, if you want to change the context or items like busydetect and busycount. If you modify this file, remember it may be overwritten if you select overwrite chan_dahdi in the Hardware Detection

Dahdi-channels.conf **/etc/asterisk/dahdi-channels.conf**

Mainly for advanced users, but PSTN users may need to change the context if they want to use the ZAP to DID channel Mapping in FreePBX. Again may be overwritten if you use Hardware Detection.

Common directory locations

/etc/asterisk

This directory holds almost all the standard Asterisk configuration files.

/etc/dahdi

This directory holds configuration files that are used to initialise telephony hardware

/var/log

This directory is used to hold all the Linux logs

/var/log/asterisk

This directory holds the logs for Asterisk

/usr/lib/asterisk/modules

This directory holds the codec files and other modules loaded by Asterisk

/var/lib/asterisk/mohmp3

This directory holds the music files for Music On Hold (MOH)

/var/lib/asterisk/backups

This directory is created by FreePBX after you have created your first backup file. It holds the backup files created by FreePBX (note this is different from the Elastix Backup system)

/var/lib/asterisk/sounds

This is the root directory for the sound files used by Asterisk.

/var/lib/asterisk/sounds/en

This is the directory for all Asterisk prompts for the English language. You will find other language directories with similar directory names such as **es** for Spanish and **fr** for French

/var/lib/asterisk/sounds/custom

This directory is used to hold your custom recordings that you either upload or record via the System Recordings in FreePBX.

/var/lib/asterisk/keys

This directory is used to hold keys. This may include licencing keys for Digium purchased products, but can also be used for RSA encryption keys used by Trunks.

/var/www/backup/
This is the directory where Elastix Backups reside.