

X100P SE PCI Card Installation Guide

Reference Number:	NVX-CP-USRG-01
Author:	Richard Spencer
Last Updated:	18 May 2008
Document Status:	Issue



Novavox are the sole authorised agents for X100P.com in Europe

Technical Support: This guide is freely available to anyone in the Open Source Community to help solve configuration and installation issues. However, email based technical support is only available to customers that have purchased Open Source telephony equipment directly from Novavox.

Updates: Due to the number of alternative Open Source IP PBX software packages available and new versions being regularly released, it is difficult to keep Open Source documentation fully up to date. If you find any out of date or inaccurate information within this document then please let us know by emailing us at feedback@novavox.co.uk and we will update the document as soon as possible.

Disclaimer: This document has been compiled based on experience resolving real world customer configuration issues as well as information available within the Open Source community. References to source material have not been included as it would make the document less easy to read. Also, keeping the references up to date would be very time consuming. However, if you discover information contained within this document for which you believe you are the original source and you would like to receive acknowledgement, then please let us know and we will add the appropriate reference.

Novavox Limited - www.novavox.co.uk

Registered in England - Number: 06363978
Registered office: 2nd Floor, 145-157 St John Street, London, EC1V 4PY
VAT Registration Number: GB 916422731

About Novavox

Novavox helps create reliable, affordable, integrated communication solutions to meet the needs of small businesses and medium sized enterprises. An office network connected to the Internet serves as the basis for the convergence of data, voice, video and mobile communications in a secure, integrated architecture. With an integrated communications solution, companies can save on costly phone bills, integrate customer information, and help make employees more mobile without sacrificing communication functionality.

To realise the benefits that integrated communications solutions can offer, a company needs an affordable, reliable solution tailored for the small/medium office and designed for ease of use. Novavox provides Open Source telephony equipment specifically designed for small businesses and medium size enterprises. Using Open Source solutions allows companies to benefit from protection against vendor lock-in and abandonment, a dramatic reduction in capital expenditure costs and an end to per user/feature licensing costs.

Novavox is dedicated to supplying reliable, affordable, feature rich Open Source computer telephony equipment to support businesses upgrading existing systems and those deploying new systems. Working in partnership with Novavox, small businesses and medium sized enterprises can deploy Open Source communication solutions to benefit from immediate return on investment (ROI) through dramatically reduced call charges and significant savings in equipment and infrastructure costs.

Product Range

Novavox supplies a range of alternative Open Source telephony products including:

- X100P Special Edition – The best of breed single port FXO card available
- OpenVox 4 port modular analogue PCI Express cards
- OpenVox 1-4 port ISDN BRI PCI cards
- OpenVox 1-2 port ISDN PRI/E1/T1/J1 PCI cards
- OctWare SoftEcho carrier grade echo cancellation software licences

For full details of our alternative products please visit our websites product section:

www.novavox.co.uk/products

Contact Details

Contact Sales

If you require additional product information or reseller pricing information, have an invoice/order query, or would like to request a quote for a communications solution to meet your company's specific needs then please contact us at sales@novavox.co.uk.

Contact Support

For assistance with equipment installation or for help with a device/service issue please visit our website support section in the first instance:

www.novavox.co.uk/support

Novavox customers only: If after following the guidance provided on the support page you are unable to resolve the issue then please contact us at support@novavox.co.uk.

Contents

1	Introduction	5
2	Scope	6
3	Pre-installation Requirements	7
	Physical Installation	8
4	Boot Process Card Detection	10
4.1	Command Line Boot Detection Verification	10
4.2	Hardware Detection/Configuration Tools	10
5	PCI Bus IRQ Allocation	13
5.1	Dual Core/Processor Considerations	14
6	Verifying the PSTN Connection	15
6.1	Successful PSTN Connection	15
6.2	Unsuccessful PSTN Connection	16
7	Understanding the Asterisk/Zaptel Configuration Files	18
7.1	zaptel.conf	18
7.2	zapata.conf	18
7.3	indications.conf	20
7.4	extensions.conf	20
8	Manual configuration of Zaptel/Asterisk Files	23
8.1	Loading the Zaptel Driver Modules	23
8.2	Starting Asterisk	23
8.3	Reloading zaptel.conf / zapata .conf	23
9	Automated Configuration	25
9.1	genzaptelconf – Trixbox CE 1.0 – 2.2	25
9.2	setup-pstn – Trixbox CE 2.4 and later	25
9.3	Trixbox Pro	26
9.4	AsteriskNOW zapscan	26
9.5	Checking the zaptel.conf Configuration File	27
9.6	Checking the zapata.conf Configuration File	27
10	Verifying Successful Card Configuration	29
10.1	Zaptel Zap Channels	29
10.2	Zap Channels in Asterisk	29
10.3	Tone Zone Settings	30
11	Global Line Standards Support	31
11.1	FCC Mode	31
11.2	CTR21 Mode	31
11.3	Full Global Line Standards Support	32
12	Configuring Trunks and Extensions	33
12.1	Connecting to FreePBX	33
12.2	Adding a trunk	33
12.3	Adding an Outbound Route	35
12.4	Adding an Extension	37
12.5	Setting up an IP Softphone	38
12.6	Add an Inbound Route	40

13	Further Assistance	42
14	Quick Reference.....	43
	Appendix A - Signalling methods	44
	Appendix B - Country codes / zones.....	45
	Appendix C - Zaptel.conf configuration parameters	46
	Appendix D - X100P SE PCI Card Information	48
	Appendix E - Acronyms.....	51

1 Introduction

This document describes how to install an X100P Special Edition (SE) FXO PCI card in an Open Source Linux based IP PBX system. Setting up an Open Source IP PBX for the first time can be a daunting and sometimes frustrating task, particularly if you are not familiar with Linux. There is online help available in the form of forums and “wikis”, and if you encounter a problem someone else has probably had the same problem and found a solution. However, searching through the forums and wikis for the relevant information and finding a solution can be very time consuming.

The main objective of the document is to make the card installation as easy as possible and to minimise the amount of time required. The guide provides detailed step-by-step installation instructions and background information explaining the purpose of each step. The information provided focuses on Asterisk® and Trixbox® but can be used to help install an X100P SE card in any Zaptel/Asterisk® based IP PBX system including AsteriskNOW®, Elastix and PBX in a Flash.

The X100P SE shown in the illustration below is a single port analogue computer telephony PCI card developed by X100P.com specifically for leading open source IP PBX software such as Digium® Asterisk® and Trixbox®.



The X100P SE provides a single, full featured FXO (Foreign Exchange Office) interface for connecting an Open Source IP PBX to the PSTN's (Public Switched Telephone Network's) CO (Central Office). The X100P SE works out of the box using the official Zaptel driver and is perfect for home and SOHO (Small Office Home Office) environments.

2 Scope

This installation guide for the X100P SE includes the following:

- Pre-installation setup/requirements
- Physical card installation and PSTN/telephone connectivity
- Zaptel/Asterisk configuration settings
- Configuration verification commands
- Basic trunk/extension configuration

Installation of the Open Source IP PBX software and tasks such as defining dial plans and configuring call handling/routing and IVR are outside the scope of this document.

3 Pre-installation Requirements

The X100P SE card must be installed in a Linux based IP PBX system that uses the Zapata Telephony (Zaptel) hardware drivers. Open source IP PBX software that use the Zaptel drivers includes Digium® Asterisk®, Trixbox®, Elastix® and PBX in a Flash. Installation of these software packages is outside the scope of this document. However, some links that may be useful are provided in Section 15:

The IP PBX system must have a spare PCI 2.2 slot, both 5v and 3.3v PCI buses are supported. Other system requirements are primarily dependant on the open source telephony software being used and on the number of simultaneous calls to be supported. Two examples are provided in the table below:

	 Asterisk (home use)	 Trixbox (8 calls)
Processor	Intel PIII 500MHz	Intel Celeron D 2.4 Ghz
RAM	128 MB	256 MB
Hard Disk Space	2 GB	80 GB EIDE

Before beginning the installation, the IP PBX should be powered down and the power supply switched off.

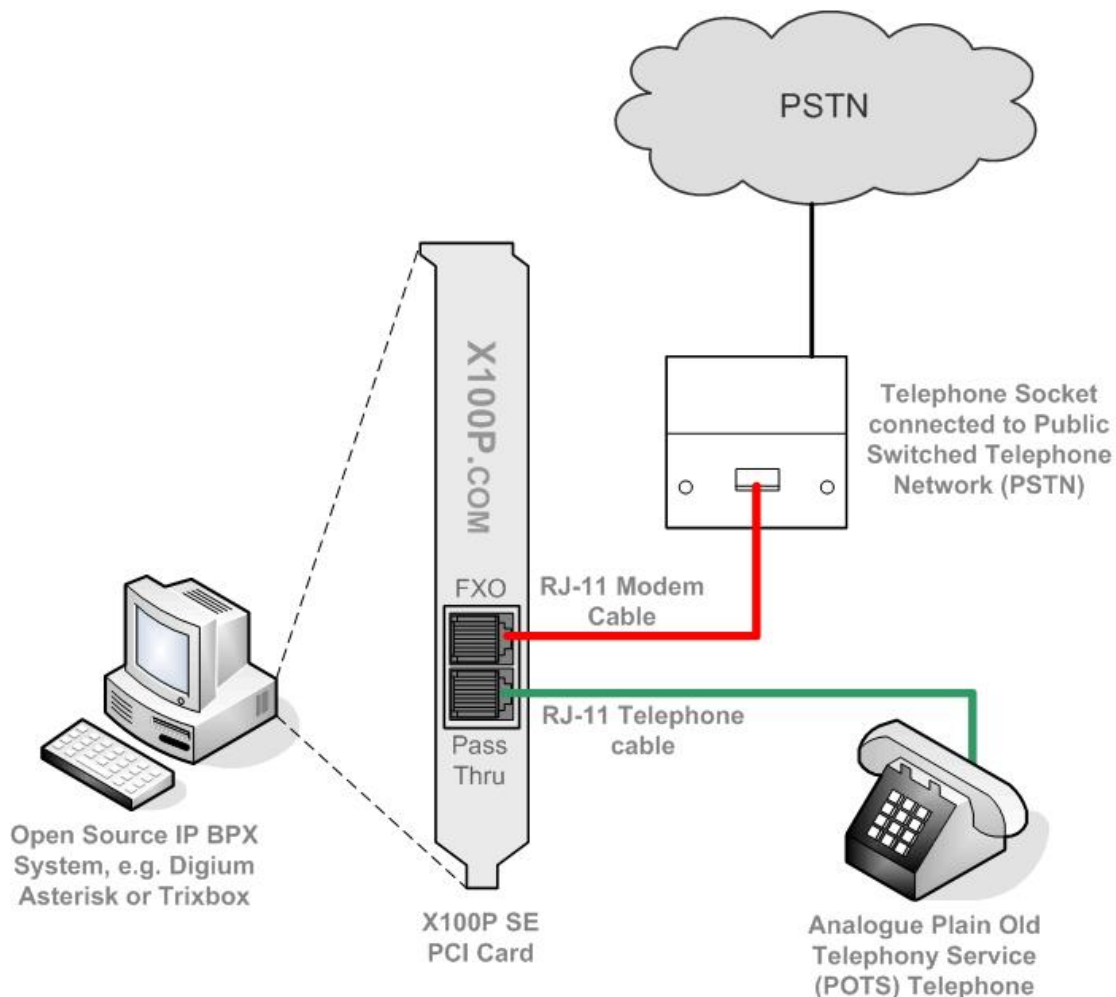
Disclaimer: Digium® (www.digium.com) and Asterisk® (www.asterisk.org) are registered trademarks of Digium, Inc. Trixbox® (www.trixbox.org) is a registered trademark of Fonality, Inc. Elastix® (www.elastix.org) is a registered of PaloSanto Solutions. Novavox Limited is not affiliated with, nor endorsed by any of the companies listed above.

Physical Installation

An anti-static wrist strap should be worn when installing the X100P SE card to avoid damage to electrically sensitive components on the card (or to IP PBX system components). It is also advisable to avoid installing the card in carpeted areas and to hold a metal part of the IP PBX system chassis with your bare hand before touching anything inside.

After switching off the power to the IP PBX system, the X100P SE should be held carefully by the edges (do not touch the electrical components or connectors) and inserted into a spare PCI slot. After inserting the card it should be secured with a screw.

The FXO port should be connected to the Public Switched Telephony Network (PSTN) telephone socket as shown in the diagram below (cable shown in red).



The cable used to connect the X100P SE to the PSTN telephone socket must be an RJ11 modem cable. A modem cable uses the centre pair of wires (3 and 4) whereas a telephone cable may use a different pair. For example, in the UK a telephone cable will often (not always) use wires 1 and 5. If the modem cable included with the X100P SE is being used for this connection, then a country specific adapter may be required for the telephone socket end. For example, in the UK a RJ11 female socket port to BT/UK male telephone socket plug adapter will be required.

Optionally, the Pass-Thru port may be connected to an analogue Plain Old Telephony Service (POTS) analogue telephone (cable shown in green in the diagram). Connecting an analogue telephone provides:

- 1) Telephony service backup: Provides a backup in case of IP PBX failure. It should be noted that to protect against power failure a phone that draws power from the PSTN line should be used rather than a phone that requires an external AC power adapter.
- 2) Troubleshooting test point: Useful for determining if an issue is due to the IP PBX or the PSTN line without having to disconnect anything.
- 3) In house call pickup: Allows calls to be picked up by users when in the house by configuring an appropriate pickup delay. Useful where an IP PBX is being used primarily as an answering machine.

4 Boot Process Card Detection

4.1 Command Line Boot Detection Verification

After installing the X100P SE card as described in the previous section and switching the IP PBX system power back on, the X100P SE card should be detected during the IP PBX system boot process. The `dmesg` (diagnostic message) command can be used to look at the Kernel's boot process message buffer and see if the card has been detected. Login to the IP PBX using the username/password you setup when installing the IP PBX software, then enter the following command:

```
#dmesg | more
```

If the card has successfully been detected, the `dmesg` output should show:

```
...  
wcfxo: DAA mode is 'FCC'  
Found a Wildcard FXO: Wildcard X100P  
...
```

If you cannot find the lines above in the output, then check that card has been inserted correctly and check that the PCI slot supports PCI 2.2.

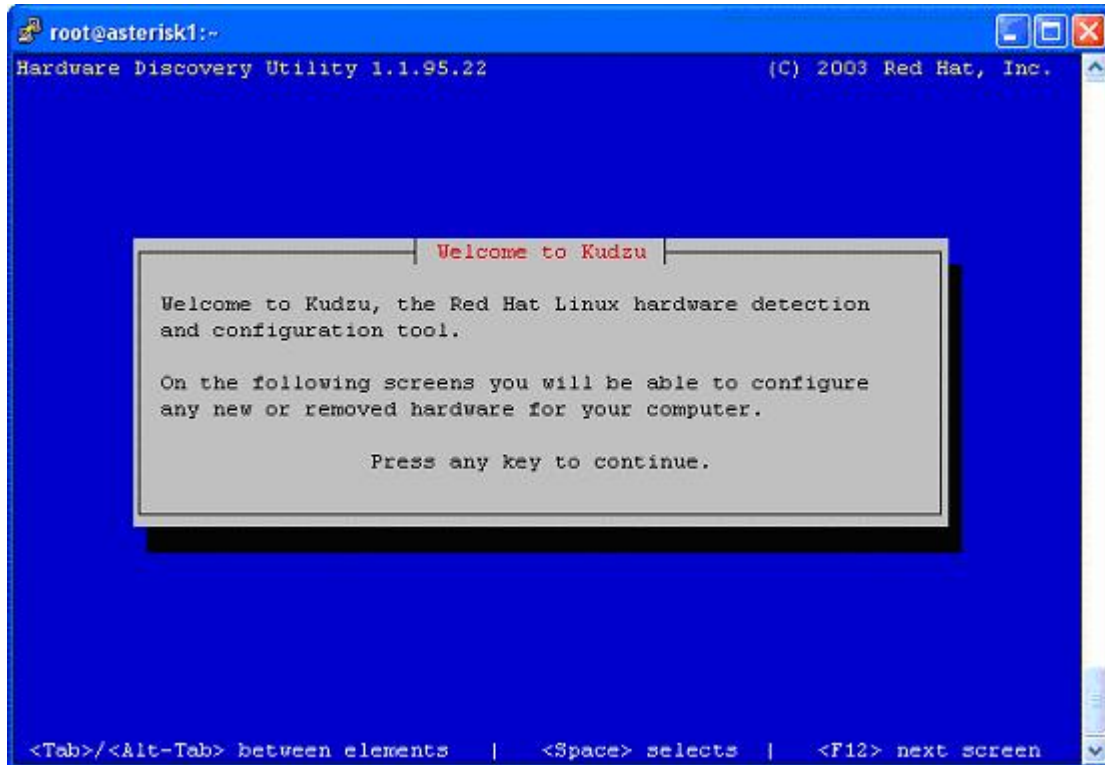
4.2 Hardware Detection/Configuration Tools

Depending on the IP PBX software package and Linux distribution version being used, the X100P SE card may be detected during the boot process by a hardware detection/configuration tool.

IP PBX systems that use Ubuntu and some other Debian Linux derivatives support a hardware detection/configuration tool called "Discover". CentOS based IP PBX systems such as Trixbox, Elastix and PBX in a Flash IP PBX support the Red Hat "Kudzu" hardware detection/configuration tool.

The information in this section describes how an X100P SE card is detected when installed in a CentOS based IP PBX that uses Kudzu. However, hardware detection/configuration tools used by other Linux OS distributions will work in a similar way.

After installing the X100P SE and switching the IP PBX power back on the following screen will appear during the boot process:



When the screen shown above appears, press **any key** to start the Kudzu hardware detection/configuration tool.

If you don't press a key then the Kudzu configuration tool welcome screen will time out and the boot process will continue as normal. If this happens, you can run the Kudzu tool after the bootup process has completed using the following command normally located in /usr/sbin/ (the Linux directory used for storing system administration/repair commands).

#kudzu

When the Kudzu hardware detection/configuration tool starts, the following screen will appear:



When this screen appears, press the **<Enter>** or **<F1>** key to configure the X100P SE card.

The Kudzu hardware detection/configuration tool will automatically add the configuration required for the X100P SE card and once complete the IP PBX system boot process will continue.

5 PCI Bus IRQ Allocation

The Peripheral Component Interconnect (PCI) bus provides communication between input/output (I/O) devices and a computer's processor. In order to communicate with the computer processor an I/O device must signal that it wishes to use the bus by sending an interrupt request (IRQ). I/O devices are normally located IRQ bus lines/channels (commonly referred to as IRQs) to allow them to signal an interrupt, which are identified by an index. In the case of Open Source Telephony cards, the line is an actual contact (finger) on the card, and it is a corresponding pin in the sockets on the bus connected by a trace on the motherboard. A card sends an interrupt to the motherboard's interrupt circuitry by changing the voltage level on the interrupt request line. This voltage change acts via interrupt controller circuitry to interrupt the processor to service the card needing the CPU's attention.

If there are more I/O devices than there are spare IRQs then two or more devices can share the same IRQ. For some I/O devices this does not cause any issues. However, the Zaptel driver generates 1000 interrupts a second for each telephony card that it controls. If the interrupts are not handled in time then this can cause significant issues for audio passing through the telephony cards and can even prevent the cards from working altogether. For this reason, it should be made certain that cards that use the Zaptel driver like the X100P SE have their own dedicated IRQ to minimise interrupt handling latency.

To determine if the X100P SE card has been detected, use the Linux utility `lspci` which displays information about devices connected to the PCI bus with the "-v" option to check what IRQ has been abnormally located by the Kernel:

```
#lspci -v
```

If the X100P SE card has been detected something similar to the following lines should be included in the output:

```
04:06.0 Communication controller: Motorola Wildcard X100P
Subsystem: Unknown device b100:0003
    Flags: bus master, medium devsel, latency 32, IRQ 217
    I/O ports at a800
    Memory at dfafe000 (32-bit, non-prefetchable)
    Capabilities: [40] Power Management version 2
```

The next step is to verify that the X100P SE card is recognised by the Linux Kernel and that it is using its abnormally located IRQ. This can be achieved using the following command:

```
#cat /proc/interrupts
```

If the X100P SE card is recognised and is being handled correctly by the Zaptel `wcfxo` driver then the output should include:

```
7:    25680730          XT-PIC  wcfxo
```

The above output is from a system using a standard XT Programmable Interrupt Controller (XT-PIC). If a system is using an Advanced Programmable Interrupt Controller (APIC) then the output will look like this instead:

```
217:   718680730        IO-APIC-level  wcfxo
```

In both cases, the Zaptel `wcfo` driver for the X100P SE must be abnormally located its own dedicated IRQ.

5.1 Dual Core/Processor Considerations

Dual core/processor IP PBX systems running a Symmetric Multi-Processing (SMP) kernel allow IRQs to be bound to single or multiple CPUs. It is advantageous to bind the X100P SE card IRQ to a separate processor used for other IRQs, particularly IRQs abnormally located to things like IDE drives. Using a separate CPU for the X100P SE card reduces interrupt latency and also improves cache coherency which reduces cache misses.

Before binding the X100P SE card IRQ to a specific processor first of all the irqbalance daemon should be disabled otherwise it will override the binding. This can be achieved using the following commands:

```
#service stop irqbalance
#chkconfig irqbalance --level 345 off
```

To bind the IRQs to a particular processor first of all check which IRQ has been assigned to each device using:

```
#cat /proc/interrupts
```

	CPU0		
0:	29421608	IO-APIC-edge	timer
1:	39	IO-APIC-edge	i8042
8:	3	IO-APIC-edge	rtc
9:	0	IO-APIC-level	acpi
14:	287820	IO-APIC-edge	ide0
205:	12333	IO-APIC-level	eth0
209:	8001	IO-APIC-level	ide1
217:	29406181	IO-APIC-level	wcfxo

Make a note of the IRQs for each peripheral device, any disk drives and your X100P SE card (which uses the Zaptel wcfxo driver). Next change the assignments using the following commands:

```
#echo 1 > /proc/irq/217/smp_affinity #wcfxo
#echo 2 > /proc/irq/14/smp_affinity #ide0
#echo 2 > /proc/irq/205/smp_affinity #eth0
#echo 2 > /proc/irq/209/smp_affinity #ide1
```

Using the commands above the X100P SE card IRQ has been assigned to processor 1, whilst all other relevant IRQs have been assigned to processor 2.

Once the IRQ binding configuration has been completed, each of the echo commands used should be added to the /etc/rc.d/rc.local file to ensure that the IRQ bindings will remain when the IP BPX server reboots, i.e.

```
#cat /etc/rc.d/rc.local
```

Should include:

```
echo 1 > /proc/irq/217/smp_affinity #wcfxo
echo 2 > /proc/irq/14/smp_affinity #ide0
echo 2 > /proc/irq/205/smp_affinity #eth0
echo 2 > /proc/irq/209/smp_affinity #ide1
```

Please note: If PCI Bus devices are removed or added to the system then the IRQ assigned may change. Therefore, the steps above should be repeated after removing/adding PCI Bus devices to ensure the IRQs remain assigned to the most appropriate processor.

6 Verifying the PSTN Connection

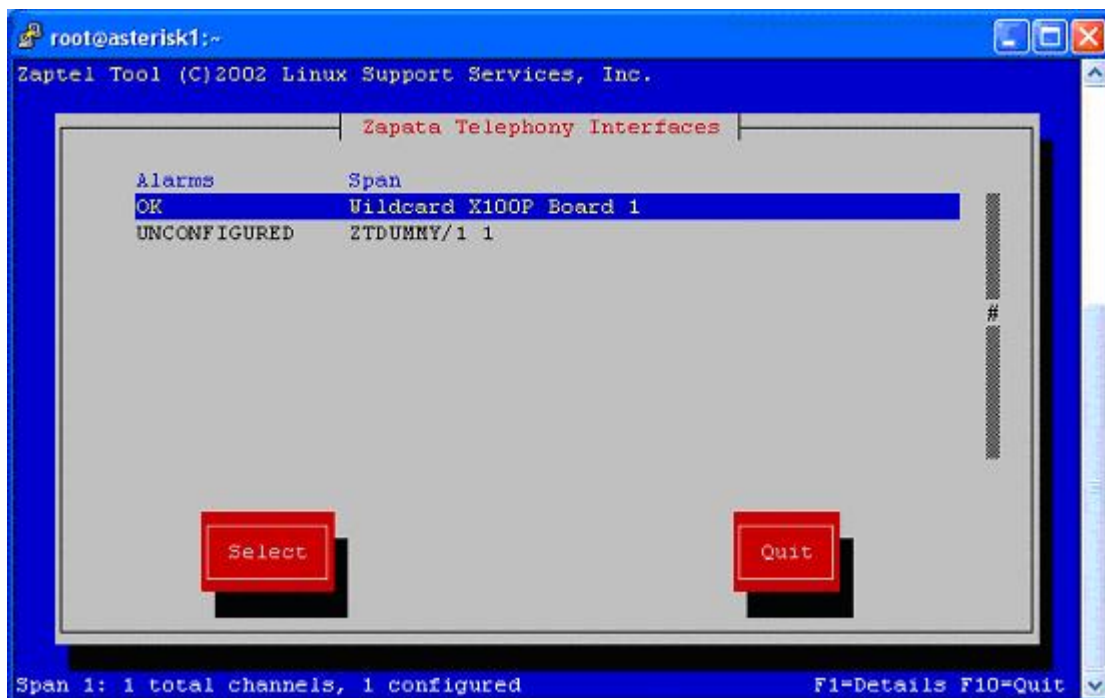
After logging into the IP PBX system, check that the FXO connection to the PSTN is working correctly using the command (normally located in /usr/sbin/):

```
#zttool
```

The Zaptel Tool (zttool) command shows the current status of any Zaptel supported interface cards installed in the IP PBX system and can also be used to run a loop test.

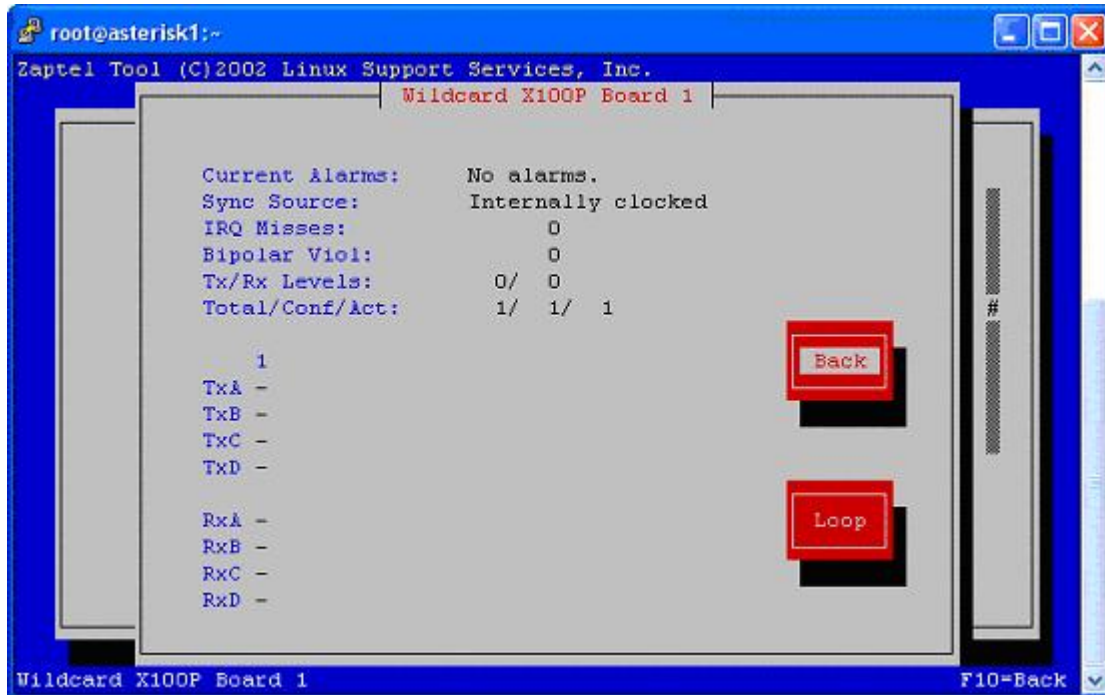
6.1 Successful PSTN Connection

When the zttool starts the following screen should appear:



The zttool output displayed above shows that a Wildcard X100P interface card has been detected and that there are no alarms. This means that the card is working correctly and has a good connection to the PSTN.

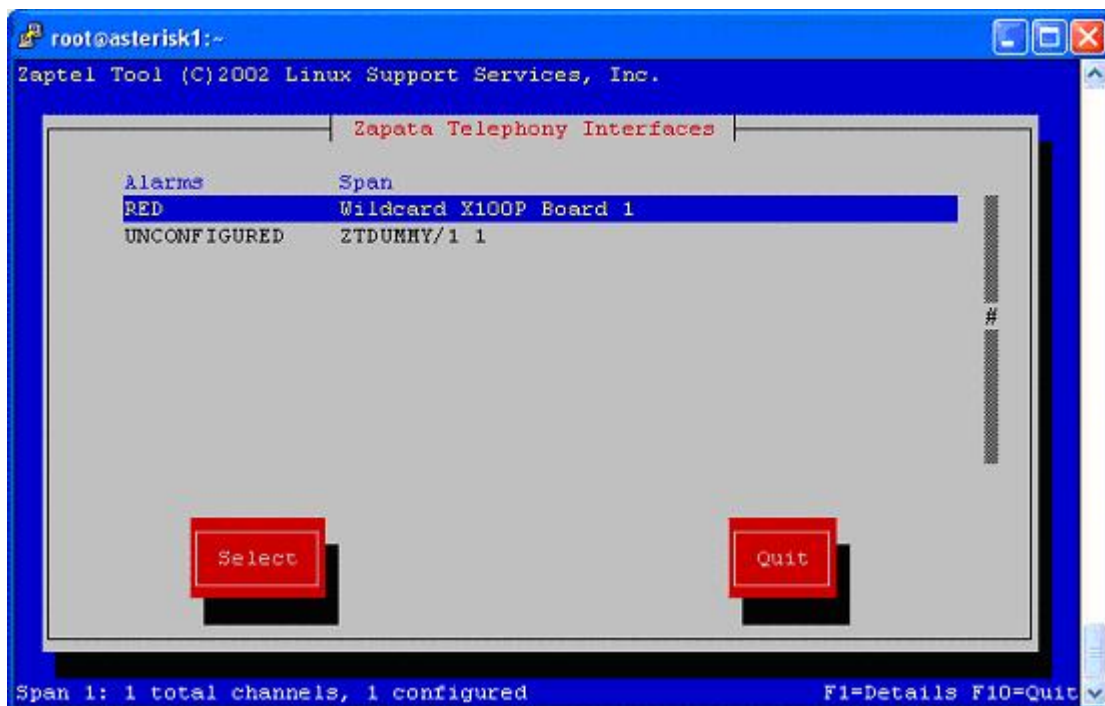
Pressing the <F1> key provides additional information on the X100P SE card as displayed in the following screenshot:



The zttool utility displays values such as current alarms, sync source, IRQ misses and TX/RX levels for each Zaptel interface. To go back to the previous screen press the <F10> key, and then press it again to exit from zttool.

6.2 Unsuccessful PSTN Connection

If there is an issue with the connection to the PSTN, then the zttool utility will show the following:



The red alarm indicates that there is an issue with the connection to the PSTN. The most common causes of a red alarm are i) the cable to the telephone socket not being connected properly at both ends, and ii) a telephone cable being used instead of a modem cable. Check X100P SE PCI Card Installation Guide

the cable/connectors, and if necessary check the telephone socket is working by connecting an analogue phone and listening for dial tone.

7 Understanding the Asterisk/Zaptel Configuration Files

This section describes the purpose of the main Zaptel/Asterisk configuration files. Information on how to configure the files can be found in the following sections.

7.1 zaptel.conf

The zaptel.conf configuration file is normally located in /etc/, which is the Linux directory used for storing critical startup/configuration files. The zaptel.conf file consists of keyword and value pairs and is used by the Zaptel driver to define the relevant parameters for zaptel supported hardware devices.

In order to view the zaptel.conf file the following command can be used:

```
#cat /etc/zaptel.conf
```

The zaptel.conf file should contain the following:

```
fxsks=1
loadzone = <country_code>
defaultzone = <country_code>
```

fxsks=1

The “fxs” in the configuration line fxsks=1 stands for Foreign Exchange Station/Subscriber and refers to the type of signalling used by the X100P SE card. This can be confusing as the X100P SE has an FXO (Foreign Exchange Office) interface, not an FXS interface. However, the reason for this is that FXO ports connect to the PSTN via the local exchange, and therefore must act like a subscriber/station (i.e. an analogue telephone/PBX). As such, the X100P SE port must be configured to use fxs signalling.

The “ks” in the configuration line fxsks=1 stands for Kewlstart, and the “=1” identifies that there is a single fxsks signalling channel. Multiple X100P SE cards can be configured using “=1-2” for two cards, “=1-3” for 3 cards, etc. Kewlstart is a type of signalling used by Asterisk, a description of the different signalling options is provided in Appendix A. Kewlstart is the preferred signalling protocol for analogue circuits in Asterisk and is the signalling protocol that should be used in the UK and the majority of European countries.

loadzone = <country_code>

Loadzone defines the set of tones that should be used to provide country specific indications such as call waiting, dial tone, busy tone, etc. For example, the country code for the UK is “uk”, A list of country codes for other countries with pre-defined tones is provided in Appendix B.

defaultzone = <country_code>

The defaultzone is used for all channels that do not have a specific zone configured.

7.2 zapata.conf

The information in the zaptel.conf configuration file is not visible to Asterisk, which also needs to know what the settings are for the hardware that has been installed. Therefore, Asterisk uses the zapata.conf configuration file (normally located in /etc/asterisk/) to record parameters relating to channels provided by the zaptel interface layer, e.g. echo cancellation, call waiting, pickup groups, etc.

Channels must be defined in this file before they can be used by Asterisk. Like the `zaptel.conf` file, the `zapata.conf` file consists of keyword and value pairs. Most of the keywords setting parameters are for the operation of channels. The keywords may be boolean (yes/no) or contain values specific to the keyword. Configured values remain in effect for all following channel definitions unless they are overridden.

In order to view the `zapata.conf` file the following command can be used:

```
#cat /etc/asterisk/zapata.conf
```

The `zapata.conf` file should contain at least the following:

```
signalling=fxs_ks
context=from-zaptel
group=0
channel => 1
```

signalling=fxs_ks

The signalling parameter tells Asterisk what signalling type should be used for the channel that the parameter applies to (channel 1 in this example). As explained in the `zaptel.conf` section `fxs` is the signalling mode used for FXO ports and `ks` means Kewlstart, which is the preferred signalling protocol for analogue circuits in Asterisk. A description of the alternative signalling types is provided in Appendix A.

context=from-zaptel

When a call comes in on an FXO interface Asterisk needs to perform some action. The action to be performed is configured inside a block of instructions called a context. The example here uses a context called `from-zaptel`. Incoming calls on the FXO interface are directed to the `from-zaptel` context with `context=from-zaptel`. The instructions to perform inside the context are defined in the file `extensions.conf`.

group=0

The group parameter allows multiple channels to be grouped together logically. This is useful if multiple FXO ports are installed into the same system as it allows the IP PBX to search for the next free channel in a particular group and use it to dial out.

channel => 1

The channel parameter assigns the all the parameters above it to a particular channel. The channel number needs to match the channel number in `zaptel.conf`. All the values (above `channel => 1`) become the default values for all other channels. This means that if you have multiple channels installed you only need to add configuration parameters for a particular channel if they differ from those above.

In addition to the `zapata.conf` configuration parameters listed above, there are a number of other parameters that can be configured to meet your specific needs. One of the most useful telephony features configured using `Zaptel.conf` options is Caller Identity (CID) or Calling Number Identification. CID is a telephony feature available on Public Switched Telephony lines (PSTN) lines, which sends a caller's number to the called party's telephony equipment. The CID options are listed in Appendix C along with the other key parameters.

Note: Due to an issue with the `Zaptel wcfxo` driver, CID cannot be detected in the usual way on UK BT PSTN lines using X100P cards. If you have a UK BT line and would like to enable CID then follow the instructions in our X100P SE UK Caller ID setup guide:

 [Novavox X100P SE UK Caller ID Setup Guide](#)

7.3 indications.conf

The indications.conf file (normally located in /etc/asterisk/) contains information about the sounds that a phone system in a particular country makes for various indications such as call waiting, dial tone, ringing, busy tone, etc. The correct tone zone must be loaded for the X100P channel so that it will imitate the indications tones for the relevant country. The correct country is set by configuring the following line in the indications.conf file:

```
country = <country_code>
```

The country codes are listed in Appendix B. One way to see what country zones are supported by your IP PBX is to use the command:

```
CLI>show indications
```

The output from this command should show a list of country codes that are supported by your IP PBX, e.g.:

```

Country Alias  Description
=====
at
au
br
be
ch
cl
cn
cz
de

etc...
```

7.4 extensions.conf

The core function of an IP PBX is to process inbound and outbound calls. The rules that determine how calls are handled are defined in a dial plan. Open Source IP PBX dial plans are fully customisable and Asterisk uses the extensions.conf file (normally located in /etc/asterisk/) to store dial plan configuration information.

The extensions.conf file contains a list of instructions called extensions. This can be confusing as extensions are normally thought of as phone extensions like SIP phones or analogue POTS phones. However, an extension is also an instruction that Asterisk will execute when triggered by an incoming call or by digits being dialled on a channel.

The dial plan extensions are divided up into separate sections called contexts. These prevent different parts of the dial plan from interacting with one another. Unless interaction is explicitly allowed, an extension that is defined in one context is completely isolated from extensions in any other context.

Extensions are made up of three main components:

- **Name/number** – The name/number that uniquely identifies the extension
- **Priority** – Extensions can contain multiple steps, the number of the step is called the priority
- **Application** – A command to perform a specific action, e.g. answer a call, hangup, playback a recording, etc.

The syntax for an extension is:

```
exten => name, priority, application()
```

Dial plan configuration is outside the scope of this document. However, a very simple example is provided below, which contains a single context with one extension.

```
[incoming]
```

```
exten => 1001,1,Dial(SIP/1001,10)  
exten => 1001,2,Playback(unavailable)  
exten => 1001,3,Hangup
```

```
[incoming]
```

Indicates the start of a context and gives the context a name (incoming).

```
exten =>
```

Indicates that one step of an extension is defined.

```
1001
```

The name/number of the Extension. If this number (1001) is called, the actions defined in the corresponding extension are executed.

```
1, 2 or 3
```

The priority of the step in the extension.

```
Dial()
```

An application/command that attempts to establish a new outgoing connection on a channel, and then link it to the existing input channel.

```
SIP/1001
```

SIP device/client number 1001.

```
10
```

Length of time in seconds that Asterisk will attempt to dial before moving onto the next step in the extension.

```
Playback()
```

An application/command that plays the specified sound file.

```
unavailable
```

The name of a sound file with the file extension omitted.

```
Hangup()
```

An application/command that unconditionally hangs up a channel.

In this example, when extension number 1001 is dialed the IP PBX will attempt to dial SIP device/client 1001, if the call is not answered after 10 seconds it will play the sound file "unavailable" and then hangup.

It should be noted that the extensions.conf file can quickly become very complex and therefore manual configuration is not recommended. It is advisable to use an IP PBX configuration tool/GUI like FreePBX which takes care of the complexity for you.

8 Manual configuration of Zaptel/Asterisk Files

One way to configure the IP PBX to support the X100P SE card is to manually configure the different Zaptel/Asterisk configuration files (for an automated method please see the next section). All the information required to set the correct parameters in each file is included in Section 7 and any text editor (e.g. vi, nano) can be used to edit the configuration files.

8.1 Loading the Zaptel Driver Modules

Once the zaptel.conf file has been configured the relevant Zaptel driver modules must be loaded into the Kernel. IP PBX packages like Trixbox automatically load the Zaptel modules during the boot process so they are already running when the IP PBX boots up. To check what modules are currently running the following command can be used:

```
#lsmod
```

The output should include the following:

Module	Size	Used by
...		
wcfxo	11424	2
zaptel	192932	5 zttranscode, ztdummy, wcfxo
...		

If the module listed above are not running then they can be added to the Kernel using:

```
#modprobe zaptel
#modprobe wcfxo
```

8.2 Starting Asterisk

Once the Zaptel modules have been loaded and the zapata.conf file has been configured Asterisk can be started. IP PBX packages like Trixbox automatically start Asterisk during the boot process, an easy way to find out if Asterisk is running is to type:

```
#asterisk -r
```

If Asterisk is running you will be connected to the CLI, to exit just use:

```
CLI>exit
```

If Asterisk is not running then you will receive the following message:

```
Unable to connect to remote asterisk (does /var/run/asterisk/
asteriskctl exist?)
```

If you receive this message you can start Asterisk using the command:

```
#ampportal start / #asterisk
```

8.3 Reloading zaptel.conf / zapata .conf

It is important to note that the configuration files need to be reloaded after making any changes in order for the changes to take effect. If the Zaptel modules are already running

when making changes to the zaptel.conf file then the ztcfg utility needs to be run reload the zaptel.conf configuration. This can be done using the non-verbose form of the command:

```
#ztcfg
```

If Asterisk is already running when making change so the zapata.conf file then Asterisk must be reloaded from the console. This can be achieved using the command:

```
CLI>reload
```

If any of the signalling methods in zapata.conf are changed then Asterisk must be restarted:

```
#ampportal stop / CLI> stop now  
#ampportal start / #asterisk
```


9 Automated Configuration

9.1 genzaptelconf – Trixbox CE 1.0 – 2.2

Genzaptelconf (normally located in /usr/local/sbin/) is a script supported by Trixbox (and other IP PBX packages like Elastix and PBX in a flash) that automatically detects hardware cards supported by the Zaptel driver and configures the appropriate parameters in the zaptel.conf. The genzaptelconf script also builds the zapata-auto.conf configuration file (normally located in /etc/asterisk/), which contains information about zap channels. Genzaptelconf performs the following actions:

1. Stops Asterisk and unloads the Zaptel drivers
2. Generates zaptel.conf and zapata-auto.conf configuration files
3. Loads the Zaptel driver modules, e.g. zaptel, wcfxo
4. Runs the ztcfg program which reads the zaptel.conf file and configures the hardware interface cards
5. Restarts Asterisk

In order to detect the X100P SE card and configure the appropriate parameters use the following command:

```
#genzaptelconf -c <country_code>
```

The country codes are listed in Appendix B.

When the script has completed it displays information about the Zaptel channels it has successfully configured. The following output shows that the X100P SE card has been successfully detected and configured:

```
...
Asterisk Started

STARTING FOP SERVER
FOP Server Started
  Chan Extension Context Language MusicOnHold
  pseudo from-zaptel en
  1 from-zaptel en
```

9.2 setup-pstn – Trixbox CE 2.4 and later

Newer versions of Trixbox support the setup-pstn command for automatically detecting Zaptel hardware and configuring the zaptel.conf and zapata-auto.conf configuration files. When setup-pstn runs it performs the following actions:

1. Stops Asterisk and unloads the Zaptel drivers
2. Generates zaptel.conf and zapata-auto.conf configuration files
3. Loads the Zaptel driver modules, e.g. zaptel, wcfxo
4. Runs the ztcfg program which reads the zaptel.conf file and configures the hardware interface cards
5. Restarts Asterisk

In order to detect the X100P SE card and configure the appropriate parameters use the following command:

```
#setup-pstn
```

When setup-pstn has completed it displays information about the Zaptel channels it has successfully configured. The following output shows that the X100P SE card has been successfully detected and configured:

```
...
STARTING ASTERISK
Asterisk Started

STARTING FOP SERVER
FOP Server Started
  Chan Extension Context Language MOH Interpret
  pseudo default en default
  1 from-zaptel en default
```

The setup-pstn command does not have an option for configuring country specific parameters. However, genzaptelconf can be used to provide this functionality. To set the country specific parameters use:

```
#genzaptelconf -c <country_code>
```

The country codes are listed in Appendix B. After using genzaptelconf to change the country parameters, the ztcfg utility needs to be run reload the zaptel.conf configuration. This can be done using the non-verbose form of the command:

```
#ztcfg
```

It should be noted that in Trixbox 2.6 the genzaptelconf command requires options to tell it what functions to perform, e.g. the option “-d” tells it to detect Zaptel hardware.

9.3 Trixbox Pro

Trixbox pro is aimed at resellers as it provides configuration automation and various paid support options. The X100P SE card can be used with Trixbox Pro, but the configuration files must be edited manually because the web configuration tool does not support X100P cards.

It should be noted that any manual configuration changes may be overwritten after making changes in the GUI. Further, Trixbox state that manual changes are not supported, and must be done at the customers own risk, i.e. you will not receive any assistance from Trixbox technical support if you experience any issues. Please see the following link for further details:

http://help.trixbox.com/index.php/Custom_Modifications

For the reasons identified above, we do not recommend using X100P SE cards with Trixbox Pro. If you wish to use Trixbox, we would suggest using Trixbox CE for the following reasons:

1. Trixbox CE provides some features that Trixbox PRO SE doesn't such as conference bridging
2. Trixbox CE uses FreePBX GUI for dial plan configuration which supports all open source telephony cards, not just those certified by Trixbox Pro.
3. There are significantly more users of Trixbox CE and therefore there is more help available from the Trixbox community for configuring/installing Trixbox CE

9.4 AsteriskNOW zapscan

Zaptel hardware configuration changes made using the AsteriskNOW GUI are saved in the files /etc/zaptel.conf.zapscan and /etc/asterisk/zapata.conf.zapscan. The AsteriskNOW GUI uses zapscan.conf (normally located in /etc/asterisk), which is detected at boot time by

zapskan run from /etc/init.d/zaptel (during the boot process the message "Running zapskan..." appears). When the IP PBX reboots, the zaptel.conf and zapata.conf files are updated with whatever is written in the zapata.conf.zapskan and zaptel.conf.zapskan files.

Therefore, if you wish to make changes to the zaptel.conf and zapata.conf files you should actually change the zapata.conf.zapskan and zaptel.conf.zapskan files so that the changes remain when the system is rebooted. It should be noted however that manual changes to the zapskan files may be lost if changes are made in the GUI or if the IP PBX server is updated.

9.5 Checking the zaptel.conf Configuration File

In order to check that the zaptel.conf file has been configured correctly use the following:

```
#cat /etc/zaptel.conf
```

Using the UK as an example, the output should include the following:

```
# Span 1: WCFXO/0 "Wildcard X100P Board 1" (MASTER)
fxsks=1

# Global data

loadzone      = uk
defaultzone   = uk
```

The configuration line that starts "# Span1: WCFXO/0..." identifies the configuration section for the X100P SE card and should read as shown above. It is possible that this line may have the word "RED" at the end of it. This means that there is an issue with PSTN connectivity, further information on this issue can be found in 7.2.

If the loadzone/defaultzone shows the default "us" rather than your country code it means that the genzaptelconf script does not support your country code. If so, follow the manual configuration steps described in section manually edit the zaptel.conf and indications.conf files so that they show the correct country code using a text editor (e.g. vi, nano).

9.6 Checking the zapata.conf Configuration File

Firstly, the zapata configuration changes made after running genzaptelconf or setup-pstn are saved in the zapata-auto.conf file. Therefore, the zapata.conf file needs to include a link to the zapata-auto.conf file. To check the link is there use:

```
#cat /etc/asterisk/zapata.conf
```

The line that needs to be included is:

```
#include zapata-auto.conf
```

In order to check that the zapata-auto.conf file has been configured correctly use the command:

```
#cat /etc/asterisk/zapata-auto.conf
```

The output should include the following:

```
; Span 1: WCFXO/0 "Wildcard X100P Board 1" (MASTER)
;;; line="1 WCFXO/0/0"
signalling=fxs_ks
callerid=asreceived
```

```
group=0  
context=from-zaptel  
channel => 1
```

10 Verifying Successful Card Configuration

10.1 Zaptel Zap Channels

In order to verify that the X100P SE card has been configured correctly in zaptel.conf the following tool can be used:

```
#ztcfg -vv
```

Ztcfg reads and loads the zaptel.conf file in order to configure the hardware interface cards installed in the IP PBX. The “-vv” sets the verbosity level of the information displayed about the Zaptel interface cards configured. Without the “-vv” the cards are still configured but there is no information displayed on the screen. With the verbosity set to “-vv”, information about the channels that have been configured and the signalling method in use is displayed. The output from the ztcfg tool for a correctly configured X100P SE card is shown below:

```
Zaptel Version: 1.4.9.2
Echo Cancellor: OSLEC
Configuration
=====

Channel map:

Channel 01: FXS Kewlstart (Default) (Slaves: 01)

1 channels to configure.
```

10.2 Zap Channels in Asterisk

In order to verify that the X100P SE card is correctly configured in zapata.conf the following commands can be used to check that the card has been detected:

```
#asterisk -r
CLI>zap show status
```

If zapata.conf is correctly configured then the following output should be shown:

Description	Alarms	IRQ	bpviol	CRC4
Wildcard X100P Board 1	OK	0	0	0

In order to verify the Zap channels recognised by Asterisk, the following command can be used:

```
CLI>zap show channels
```

If zapata.conf is correctly configured then the following output should be shown:

Chan	Extension	Context	Language	MOH Interpret
pseudo		default	en	default
1		from-zaptel	en	default

10.3 Tone Zone Settings

In order to verify that the tone zone configured in indications.conf is being used correctly the IP PBX should be rebooted (i.e. using `shutdown -r now`) and the dmesg output verified using:

```
#dmesg | more
```

If everything has been correctly configured, using the UK as an example the dmesg output should show:

```
...  
Found a Wildcard FXO: Wildcard X100P  
Registered tone zone 4 (United Kingdom)  
...
```

11 Global Line Standards Support

The Zaptel wcfxo driver has two user configurable modes of operation, FCC to support US line standards and CTR21 to support European line standards. The Silicon labs Si3012/Si3035 DAA chip used in the original Digium X100P card and low cost X100P clone cards only supports FCC/JATE line standards. However, the Si3014/Si3034 DAA chip used on the X100P SE supports global line standards.

There are two key differences between the CTR21 and FCC standards, which are concerned with AC/DC termination:

	FCC	CTR21
AC Termination	A real, nominal 600 Ω termination which satisfies the impedance requirements of FCC part 68, JATE, and other countries.	May use a complex impedance which satisfies the impedance requirements of European NET4 countries such as the UK and Germany OR 600 Ω termination.
DC Termination	Supports a transmit full scale level of -1 dBm at TIP and RING. This mode meets FCC requirements in addition to the requirements of many other countries.	Provides current limiting, while maintaining a transmit full scale level of -1 dBm at TIP and RING. In this mode, the dc termination will current limit before reaching 60 mA.

11.1 FCC Mode

FCC is the default mode for the Zaptel wcfxo driver, which supports line standards for the US and a number of other countries. If you are from one of the countries that the FCC mode supports then you do not need to make any changes to the default operation mode setting. Based on information from the Si3014/Si3034 DAA chip specification, the following countries should use FCC mode:

Argentina	El Salvador	Kuwait	Singapore
Bahrain	Guam	Macao	Slovakia
Canada	Hong Kong	Mexico	South Korea
Chile	Hungary	Peru	UAE
Colombia	India	Romania	USA
Ecuador	Indonesia	Saudi Arabia	Yemen

11.2 CTR21 Mode

The Silicon labs Si3014/Si3034 DAA chip used in the X100P SE supports 600 Ω impedance and complex impedance to meet CTR21 line standards. However, the Zaptel wcfxo driver only supports CTR21 mode with 600 Ω AC termination, which may or may not be the correct setting depending on the country and the phone system in use. For example, in the UK BT PSTN lines use complex impedance, whereas PSTN lines from some cable companies may use 600 Ω impedance.

Based on information from the Si3014/Si3034 DAA chip specification, CTR21 mode with 600 Ω impedance may work in the following countries depending on the PSTN service provider's network:

Austria	Greece	Norway
Belgium	Iceland	Portugal
Denmark	Ireland	Spain
Finland	Italy	Sweden
France	Luxembourg	Switzerland
Germany	Netherlands	UK

To change the mode to CTR21 with 600 Ω impedance, add the following lines to /etc/modprobe.conf:

```
options wcfxo opermode=1
install wcfxo /sbin/modprobe --ignore-install wcfxo && /sbin/ztcfg
```

Then shutdown and restart the IP PBX:

```
#shutdown -r now
```

To check that the correct mode is in operation use the following:

```
#dmesg | more
```

The output should include:

```
...
wcfxo: DAA mode is 'CTR21'
Found a Wildcard FXO: Wildcard X100P
...
```

Note: Debian based Linux distributions like Ubuntu do not have a /etc/modprobe.conf file. Instead modules can be loaded/unloaded by creating a configuration file with the relevant options and storing it in the /etc/modprobe.d/ directory. For example, you could create a file called modprobe.conf in /etc/modprobe.d/ and add the lines described above. After creating the file run `update-modules` to merge the changes with the system modules configuration.

11.3 Full Global Line Standards Support

To support global line standards we have put together a setup guide that uses a patch to provide additional operational modes to support line standard for all countries. If your country is not supported by the default FCC mode or CTR21 600 Ω impedance mode then use the instructions in the following guide to configure the appropriate settings for your country:

 [Novavox X100P SE Global Line Standards Setup Guide](#)

12 Configuring Trunks and Extensions

Guidance on configuring dial plans is outside the scope of this document. However, instructions are provided below to set up a very basic dial plan to test inbound/outbound calls using the X100P SE card.

As mentioned in Section 8.4, the easiest way to setup dial plans and trunks/extensions is to use a graphical based interface like FreePBX that comes with Trixbox and some other Asterisk based IP PBX systems. If you are using FreePBX then you should not edit `extensions.conf` as the file is generated by FreePBX and manual changes will get overwritten. FreePBX provides a front end to the mySQL configuration engine, in addition to generating `extensions.conf` it also creates its own files including:

```
/etc/asterisk/extensions_additional.conf  
/etc/asterisk/sip_additional.conf  
/etc/asterisk/iax_additional.conf
```

When you apply changes in the FreePBX GUI, it reads from the SQL database and writes to the relevant config files named "XXXX_additional.conf". As such, if you make any manual changes to these files the changes will be lost. However, FreePBX also allows you to create your own custom configurations using files that end in "-custom.conf", e.g. "extensions-custom.conf". These files do not get overwritten by FreePBX so can be modified manually as required.

The following instructions explain how to setup a very basic IP PBX using FreePBX. If you are using an IP PBX package that doesn't come with FreePBX then there is likely to be a similar alternate, e.g. AsteriskNOW has its own GUI.

12.1 Connecting to FreePBX

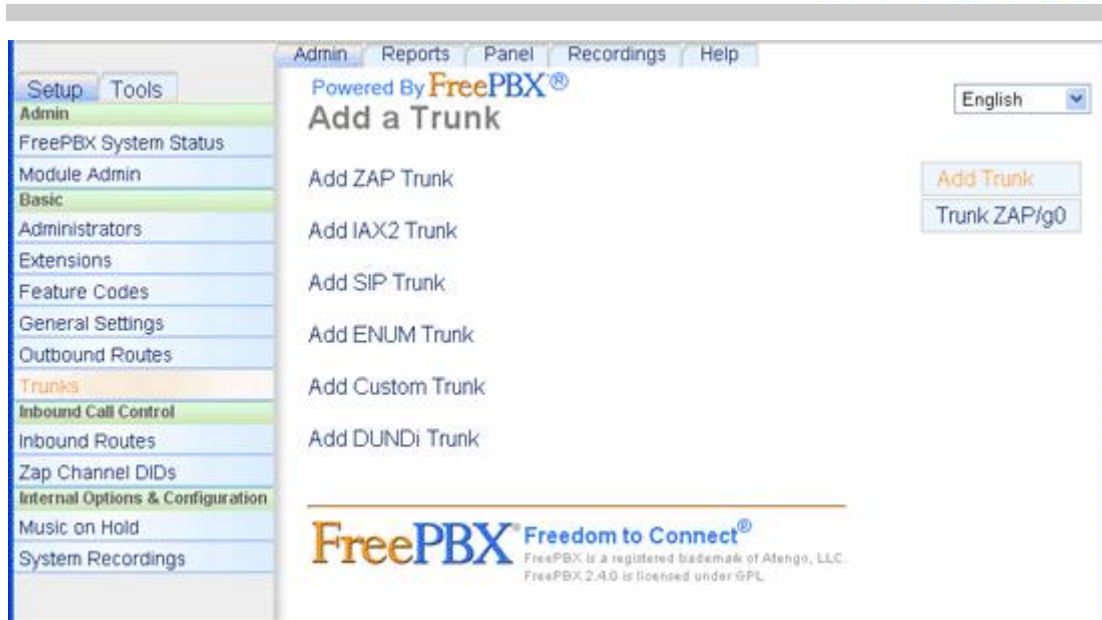
If you are using Trixbox open up a web browser and connect to your IP PBX using the IP address you configured when setting up the IP PBX, e.g. <http://192.168.10.100>. Switch to admin mode using your Trixbox username/password (the default is username: maint, password: password). From the menu options select **PBX > PBX Settings** and FreePBX will open up (either in an embedded window as with newer Trixbox versions or a separate Window).

If you aren't using Trixbox or want to access FreePBX directly you can do so by adding "/admin" to the URL when connecting to the IP PBX using a web browser, e.g. <http://192.168.10.100/admin>.

12.2 Adding a trunk

A trunk is the telephony service line that your IP PBX uses to make an external call, therefore you must have at least one trunk to make external calls. Trunks can be Session Initiation Protocol (SIP) lines for VoIP, analogue PSTN lines, ISDN lines or GSM channels. The X100P SE provides a single analogue trunk to the PSTN. Multiple trunks can be configured and you should normally always have a SIP trunk to take advantage of competitive SIP provider call rates. However, to test out the X100P SE card we just need a single analogue trunk to the PSTN.

To add a trunk in FreePBX select **Setup > Trunks**. You may already have a Trunk configured called **Trunk ZAP/g0** which is listed at the top right hand side under the **Add Trunk** option:

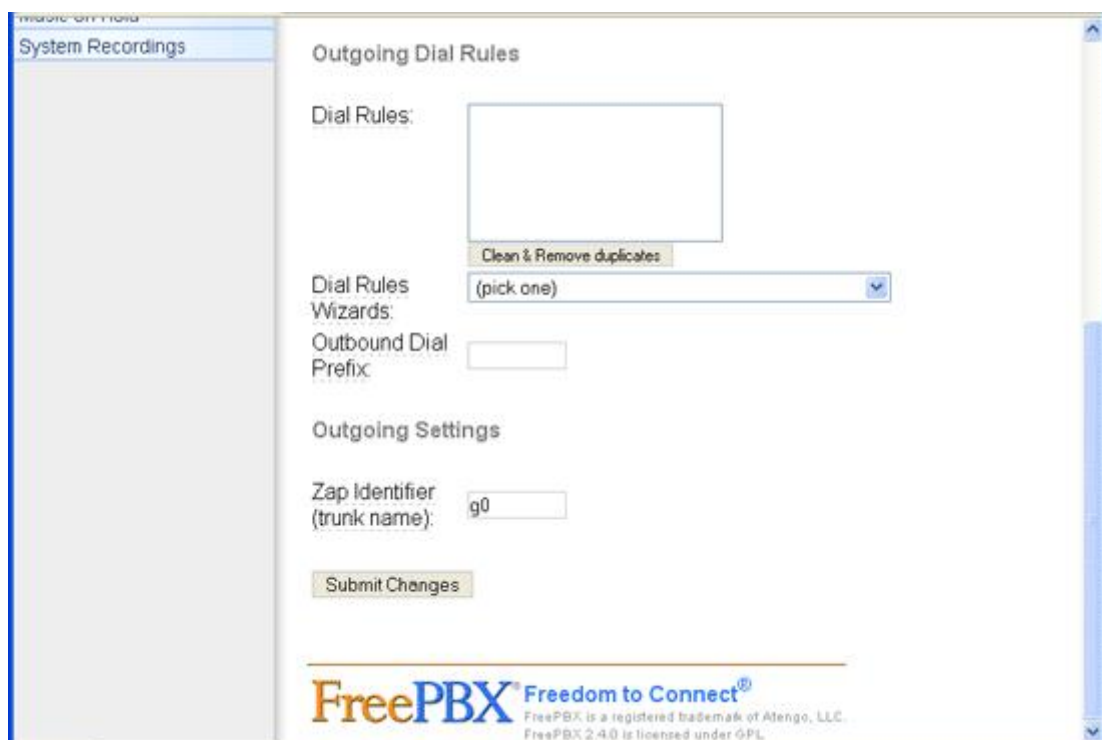


If the ZAP/g0 trunk is already setup then you can skip this step. If not then to add a new trunk for the X100P SE card select the **Add ZAP Trunk** option from the Trunks screen.

For the purposes of testing the X100P SE card, the only parameter that needs to be configured is:

Zap Identifier (trunk name): g0

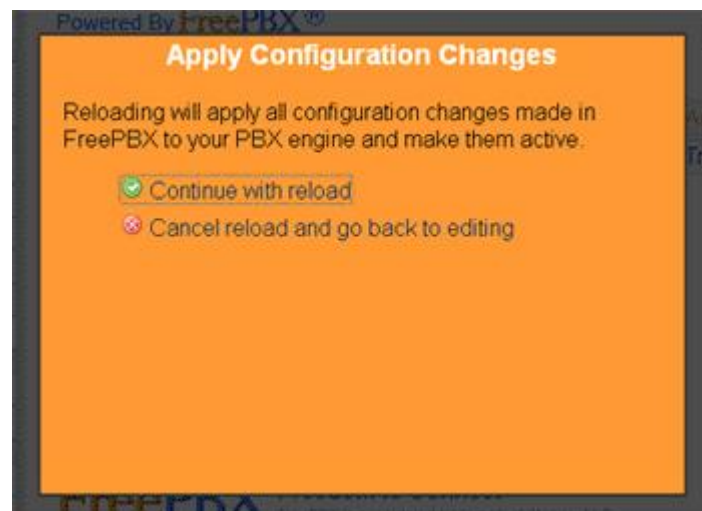
The “g0” corresponds to the group number configured in the etc/asterisk/zapata.conf configuration file for the X100P SE card. Set the Zap Identifier as g0 and then select **Submit Changes**.



After making changes to the configuration in the FreePBX GUI you need to select **Apply Configuration Changes** for the changes to be made to the IP PBX configuration.



After selecting **Apply Configuration Changes** a dialogue box will appear in newer FreePBX versions asking you to commit to the changes, select **Continue with reload**.

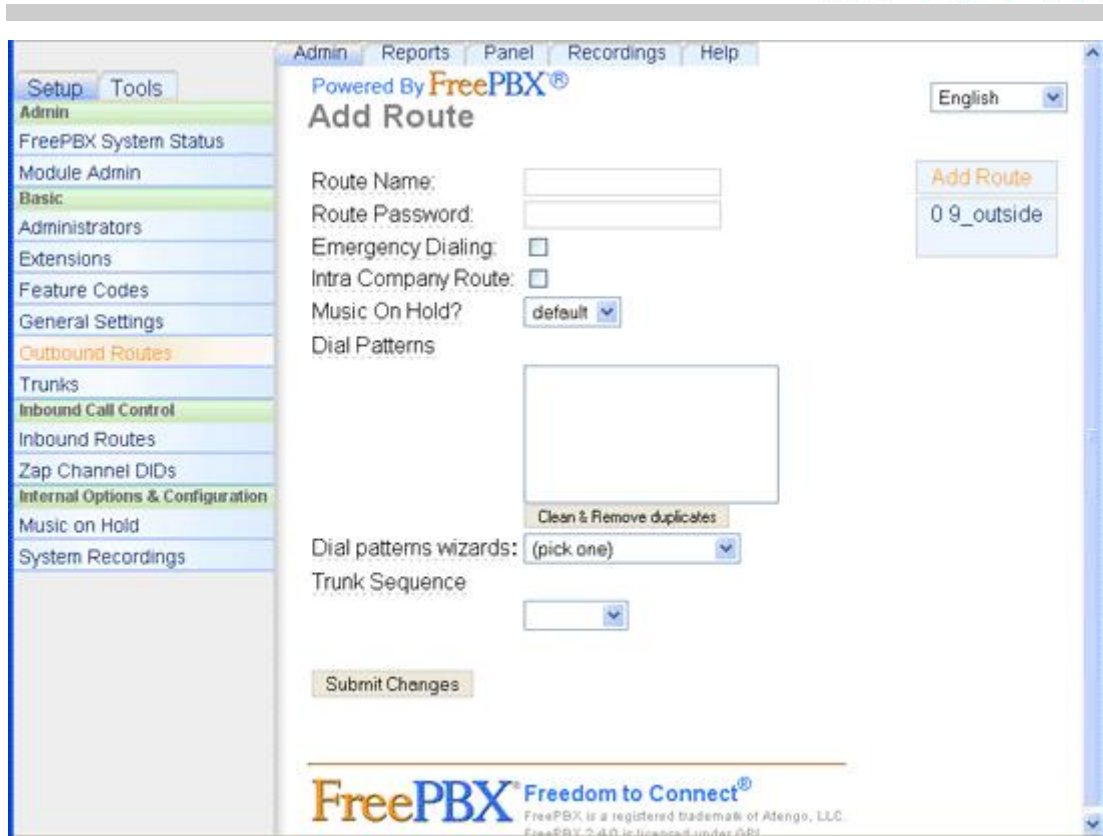


When making subsequent changes you can either select apply after each individual change (useful for testing purposes) or wait until you have finished making several changes and then select apply (faster).

12.3 Adding an Outbound Route

After setting up a trunk, in order to make outbound calls the IP PBX needs to be configured with an outbound route specifying which trunk outbound calls should be routed to. Outbound routes consist of a list of priorities/rules, which specify how to route a call based on the number dialled from an extension.

To add an outbound route in FreePBX select **Setup > Outbound Routes**. You may already have an Outbound route configured called **9_outside**, which is listed at the top right hand side under the **Add Route** option:



If the 9_outside route already exists, select it to check that the parameters are configured correctly. If not then select **Add Route**. The parameters that need to be configured are:

Route Name: 9_outside

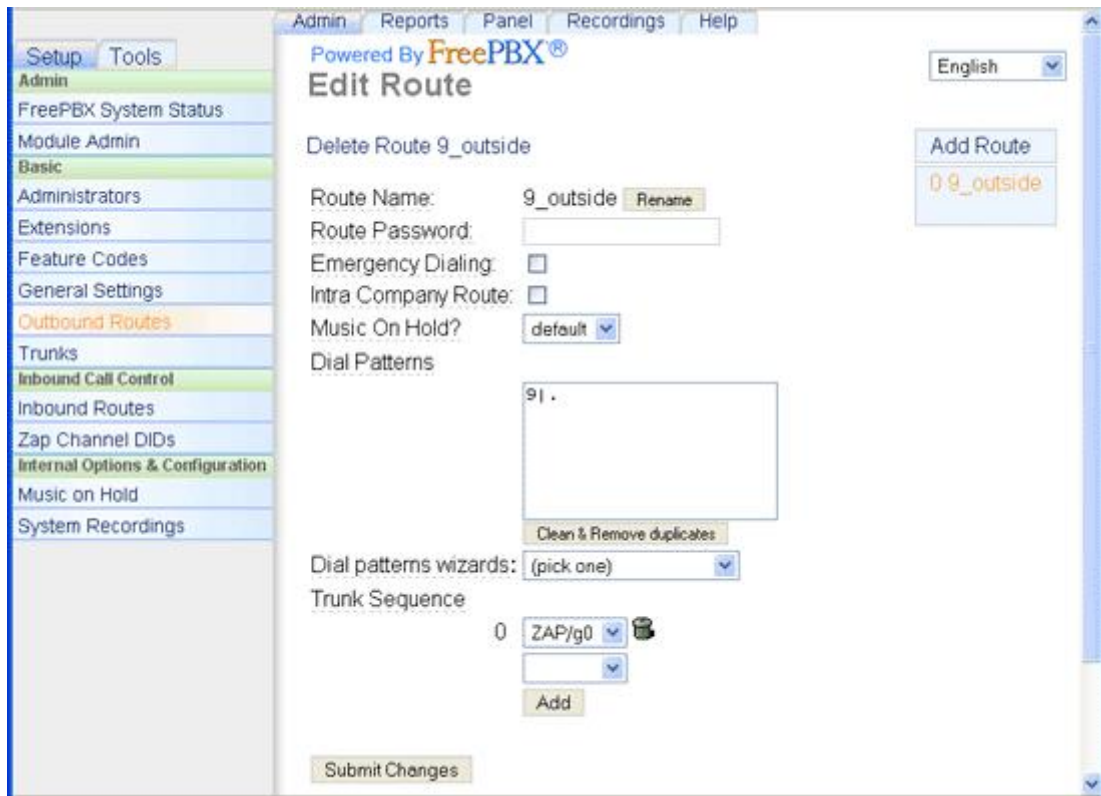
The route name is the name of the outbound route.

Dial Patterns: 9|.

The dial pattern “9|.” will route any calls made from an extension with a prefix of to the outbound trunk sequence configured in the outbound route. The IP PBX will remove the number 9 and use the remaining digits as the dialled number, e.g. 90113392371 will become 0113392371. The number 9 is required so that the IP PBX knows the number dialled on an extension is for an outbound call, not a call to an internal extension. Number 9 is the most common used but an alternative number can be used if required, e.g. 0.

Trunk Sequence: ZAP/g0

The trunk sequence lists the trunks (in order of priority) that should be used to route the outbound calls over. If the first trunk is in use then the next trunk in the list will be used. In this test scenario only a single trunk is configured which is the X100P SE analogue PSTN trunk, i.e. ZAP/g0.



After configuring the outbound route as described above select **Submit Changes**.

12.4 Adding an Extension

To make outbound calls the IP PBX needs to be configured with at least one telephone extension (and at least two to make internal calls). In this test scenario we will setup a single SIP extension to test inbound/outbound calls over the X100P SE analogue trunk.

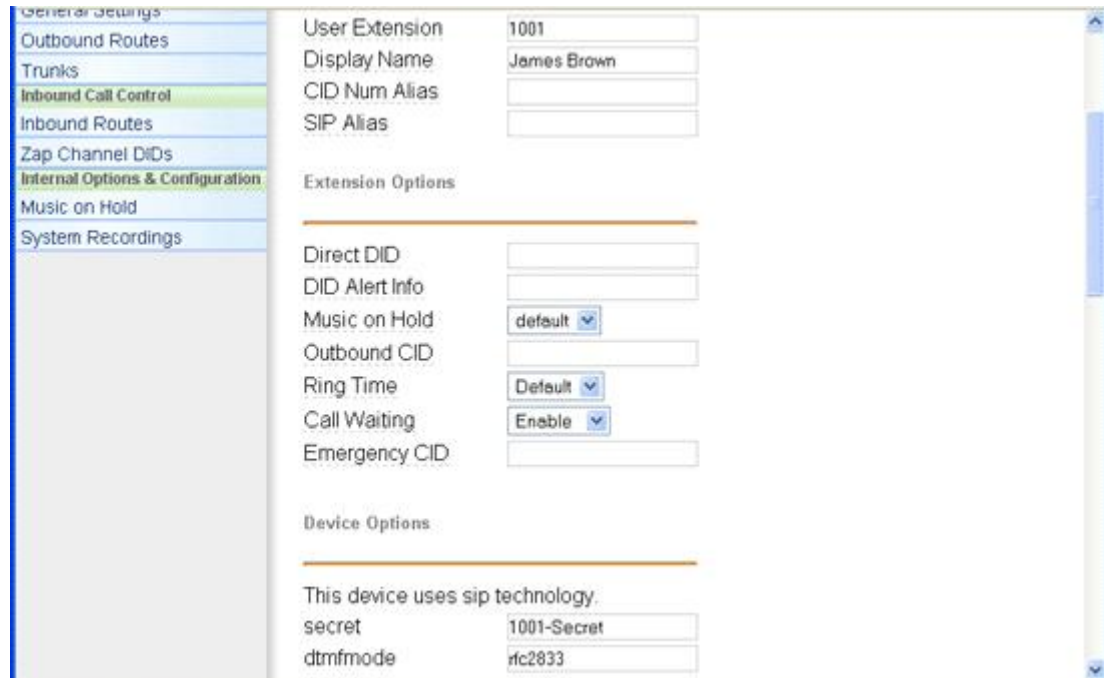
To add an extension in FreePBX select **Setup > Extensions**. Then select **Generic SIP Device** from the drop down menu and then select **Submit**.



The parameters that need to be configured are:

User Extension: 1001	The extension number to dial to reach the user
Display Name: James Brown	The CallerID name for calls from this user
Secret: 1001-Secret	Logon password for SIP device

All other parameters should be left unchanged with their default settings.



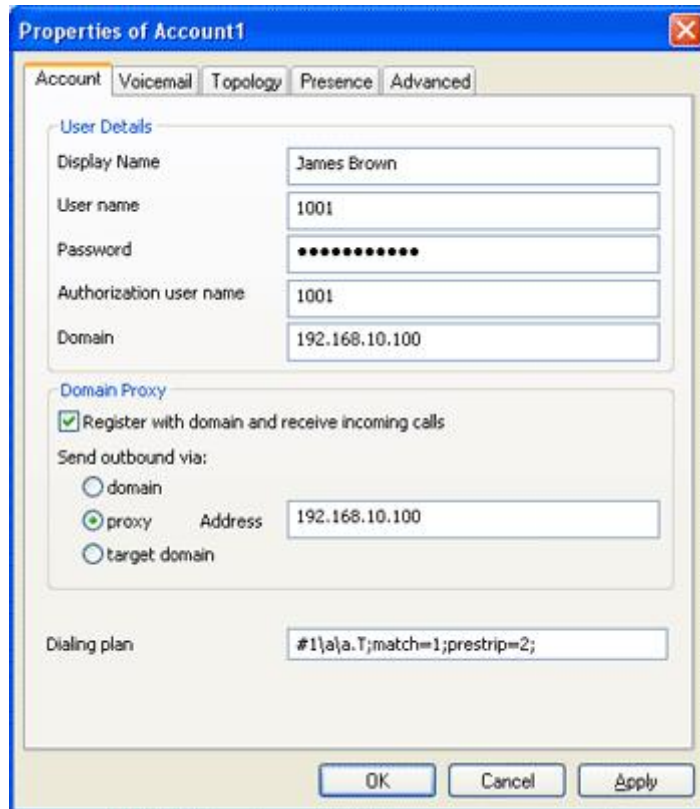
12.5 Setting up an IP Softphone

Unless you already have an IP phone, the easiest way to get started and make sure everything is working is to install an IP softphone. You can download a free softphone for Windows, Mac, or Linux called X-Lite from CounterPath (<http://www.counterpath.com>) although there a number of free softphones available.

Once you have installed and started the X-Lite IP Softphone, the X-Lite user interface will appear (with the message "No SIP accounts are enabled"). From the X-Lite drop down menu select **SIP Account Settings** and the SIP accounts information window will appear. From the SIP accounts screen select **Add**. Another window will appear to allow you to configure a SIP account. The parameters that need to be configured are as follows:

Display Name: James Brown	Same as extension CallerID
User Name: 1001	Extension number (will be displayed on user screen)
Password: 1001-Secret	SIP Login password
Authorization User Name: 1001:	Extension number
Domain: 192.168.10.100:	IP address of IP PBX server

Ensure the "Register with domain and receive incoming calls" checkbox is enabled, select the "Proxy" option and enter the IP address of IP PBX server in the Address box. Leave the other parameters with default values and select **OK**.



You will go back to the SIP accounts information window which should now show the SIP account that you have just setup. Select **Close** and the X-Lite client should connect to the IP PBX and show "Ready" along with "Your username is: 1001":



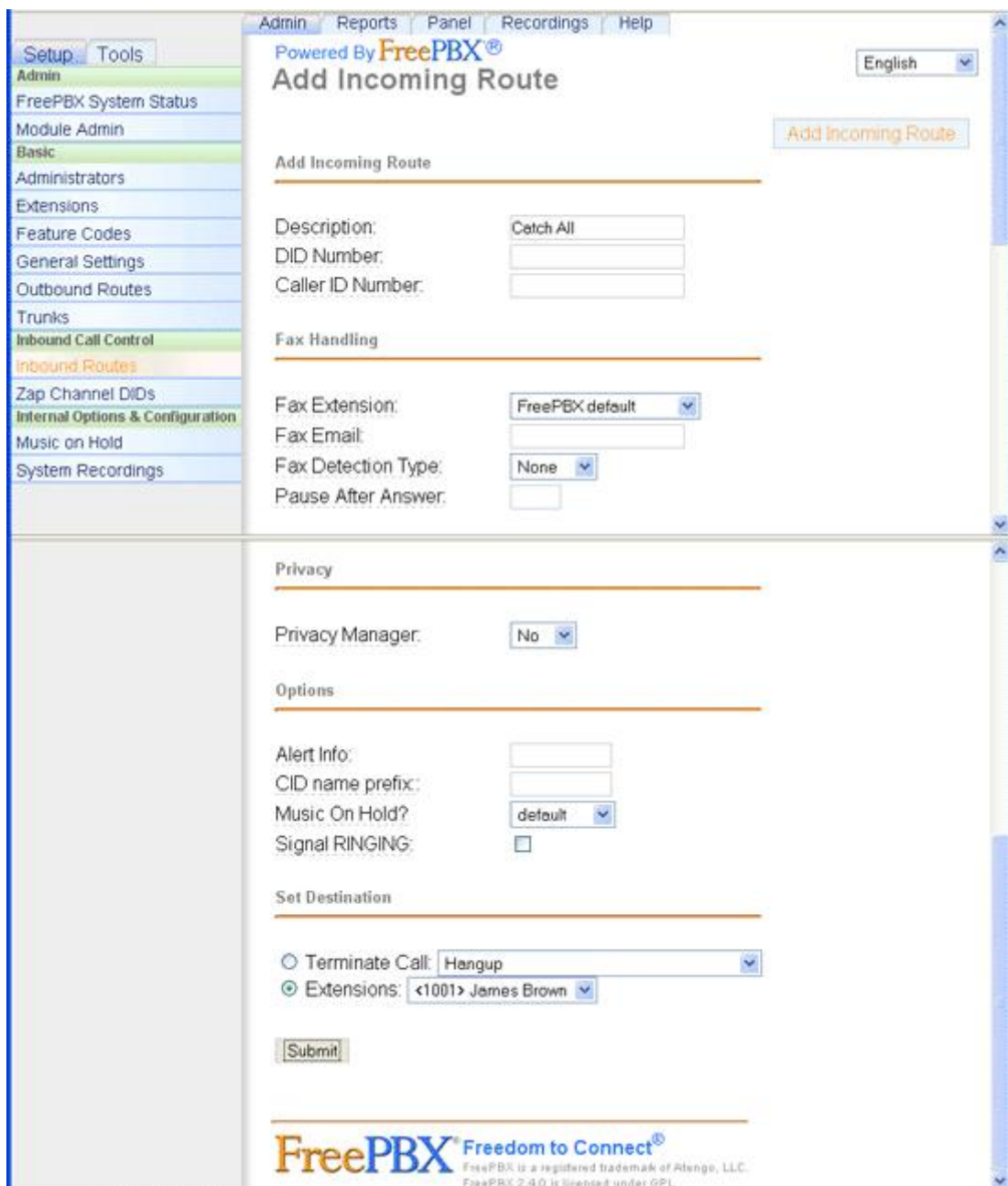
12.6 Add an Inbound Route

When an incoming call from PSTN or VoIP trunk is received, the IP PBX needs to know where to direct it and therefore at least one Inbound Route must be created to answer incoming. Incoming calls can be directed to a ring group, an individual extension, Digital Receptionist (IVR) or Queue. For this test scenario we will setup a single inbound, which will be a “Catch-All” route to direct all calls to SIP Extension 1001.

To add an extension in FreePBX select **Setup > Inbound Routes > Add Incoming Route**. The parameters that need to be configured are as follows:

Description: Catch All Description of the Incoming Route
Extensions: <100> James Brown Extension to route the incoming calls to

Leaving the “DID Number” and “Caller ID Number” blank means that all calls will be directed to SIP Extension 1001. Leave all the other parameters with default values and select **Submit**.



The screenshot shows the 'Add Incoming Route' configuration page in FreePBX. The page is titled 'Add Incoming Route' and is powered by FreePBX. The left sidebar contains navigation links such as 'Setup', 'Tools', 'Admin', 'FreePBX System Status', 'Module Admin', 'Basic', 'Administrators', 'Extensions', 'Feature Codes', 'General Settings', 'Outbound Routes', 'Trunks', 'Inbound Call Control', 'Inbound Routes', 'Zap Channel DIDs', 'Internal Options & Configuration', 'Music on Hold', and 'System Recordings'. The main content area is divided into sections: 'Add Incoming Route', 'Fax Handling', 'Privacy', 'Options', and 'Set Destination'. The 'Add Incoming Route' section includes fields for 'Description' (set to 'Catch All'), 'DID Number', and 'Caller ID Number'. The 'Fax Handling' section includes 'Fax Extension' (set to 'FreePBX default'), 'Fax Email', 'Fax Detection Type' (set to 'None'), and 'Pause After Answer'. The 'Privacy' section includes 'Privacy Manager' (set to 'No'). The 'Options' section includes 'Alert Info', 'CID name prefix', 'Music On Hold?' (set to 'default'), and 'Signal RINGING'. The 'Set Destination' section includes 'Terminate Call' (set to 'Hangup') and 'Extensions' (set to '<1001> James Brown'). A 'Submit' button is located at the bottom of the form. The footer of the page contains the FreePBX logo and the text 'Freedom to Connect', 'FreePBX is a registered trademark of Alango, LLC.', and 'FreePBX 2.4.0 is licensed under GPL'.

Configuration Complete

Your IP PBX is now fully configured to support your new X100P SE card and you should be able to make/receive calls.

13 Further Assistance

The following resources are useful for Open Source IP PBX server configuration:

Trixbox Without Tears - http://dumbme.voipeye.com.au/trixbox2/trixbox2_without_tears.pdf
Elastix Without Tears - http://www.elastixconnection.com/downloads/elastix_without_tears.pdf
Asterisk Installation Guide - <http://www.voip-info.org/wiki-Asterisk+installation+tips>
Tribox Installation Guide - <http://www.trixbox.org/wiki/trixbox-quick-install-guide>
PBX in a Flash Installation Guide - <http://pbxinaflash.net/docs/InstallPBXInAFlash-1.2.pdf>

Links to the online forums and wikis for assistance installing Asterisk/Trixbox are provided below:

Asterisk Wiki - <http://www.voip-info.org/wiki-Asterisk>
Asterisk Forum - <http://forums.digium.com>
Tribox Forum - <http://www.trixbox.org/forum>
Tribox Wiki - <http://help.trixbox.com>

For product installation/troubleshooting guides please visit our website support section:

www.novavox.co.uk/support

Novavox customers only: If after using the Open Source community resources available and following the guidance provided on our website you are unable to resolve an installation issue then please contact us at support@novavox.co.uk.

14 Quick Reference

Useful Commands

Command	Description
#amportal start / #asterisk	Start Asterisk
#amportal stop / CLI> stop now	Stop Asterisk
#asterisk -r	Connect to the Asterisk CLI
#cat /proc/interrupts	Lists PCI bus drivers/devices recognised by the Linux Kernel along with the abnormally located IRQ
#dmesg more	Diagnostic message command used to look at the Kernels boot process message buffer
#genzaptelconf -c <country_code>	Runs genzaptelconf script to detect/configure Zaptel hardware, "-c" option sets country specific parameters
#lsmod	Lists information about all loaded modules
#lspci -v	Linux utility that displays information about devices connected to the PCI bus, "-v" option displays IRQ allocation details
#modprobe	Adds a loadable module to the kernel, e.g. zaptel or wcfxo
#setup-pstn	Detects and configure Zaptel hardware
#ztcfg -vv	Reloads the zaptel.conf configuration, "-vv" option provides information on configured zap channels
#zttool	The Zaptel Tool (zttool) command shows the current status of any Zaptel supported interface cards
*CLI>exit	Exit from Asterisk CLI
*CLI>reload	Reload Asterisk
*CLI>zap show channels	Shows Zap channels recognised by Asterisk
*CLI>zap show status	Shows Zaptel devices recognised by Asterisk

Configuration File Descriptions

File	Description
/etc/zaptel.conf	Used by the Zaptel driver to define the relevant parameters for zaptel supported hardware devices
/etc/zaptel.conf.zapscan	Records Zaptel device configuration changes made in the AsteriskNOW GUI, used to update zaptel.conf
/etc/asterisk/zapata.conf	Used by Asterisk to store information about Zaptel devices and the features supported
/etc/asterisk/zapata-auto.conf	Automatically generated by genzaptelconf, contains Zaptel hardware configuration information used by Asterisk
/etc/asterisk/zapata.conf.zapscan	Records Zaptel device configuration changes made in the AsteriskNOW GUI, used to update zapata.conf
/etc/asterisk/extensions.conf	Contains dial plan information in the form of instructions called extensions, which are grouped by contexts
/etc/asterisk/indications.conf	Contains information about the sounds that a phone system in a particular country makes for various indications
/etc/sysconfig/zaptel	Zaptel service script used to load Zaptel modules

Appendix A - Signalling methods

Groundstart (gs)

Groundstart signalling also known as Earth Start and is not very common. The PBX/phone signals a call by connecting an earth to one of the wires and current flows from the PBX earth to the local exchange battery. The local exchange responds by sending dial tone to the PBX/phone. The local exchange signals a call to the PBX/phone by putting an earth on one of the wires. The PBX/phone detects the earth and looks for ringing voltage which the local exchange sends.

Loopstart (ls)

Loopstart signalling is supported by the majority of analogue phone lines. In Loopstart signalling the local exchange signals a call by sending ringing voltage to the phone/PBX. It allows a phone to indicate on hook/offhook (and a PBX to seize the line) by putting a loop across the wires. The local exchange responds by sending dial tone to the phone/PBX.

Kewlstart (ks)

Kewlstart signalling works in the same way as loopstart, but it allows the local exchange to drop battery/voltage momentarily on the phone line to indicate to the phone/PBX that the other end of the party has disconnected the call, i.e. to provide hang-up notification. This voltage drop is known as CPC (Calling Party Control) or Disconnect Clear Time (DCT) in the UK.

Appendix B - Country codes / zones

Code	Country / Zone
at	Austria
au	Australia
be	Belgium
br	Brazil
ch	Switzerland
cl	Chile
cn	China
cz	Czech Republic
de	Germany
dk	Denmark
ee	Estonia
es	Spain
fi	Finland
fr	France
gr	Greece
hu	Hungary
il	Israel
in	India
it	Italy
lt	Lithuania
mx	Mexico
nl	Netherlands
no	Norway
nz	New Zealand
pl	Poland
pt	Portugal
ru	Russia
se	Sweden
sg	Singapore
tw	Taiwan
uk	United Kingdom / Ireland
us	United States
us-o	United States - Old
ve	Venezuela
za	South Africa

Appendix C - Zaptel.conf configuration parameters

busycount - Busycount is the number of busy tones to detect, when **busydetect** is enabled. The lower the value the quicker the call will hang-up, but this increases the possibility of false hangups. [positive integer]

busydetect - With this option enabled, asterisk will listen for busy signals on the line, if your carrier sends you a busy tone on a hang-up, Asterisk should detect it. It should be noted that this option is less reliable than other hang-up detection methods such as polarity reversal and CPC (Calling Party Control) so should only be used to detect hang-ups if the other methods are not supported. [yes / no]

callerid - CallerID can be set to 'asreceived' or a specific number if you want to override it. Please note that asreceived only applies to trunk interfaces. [asreceived or a specific number]

callgroup, pickupgroup - Supports ring groups (a.k.a. call groups) and pickup groups. If a phone is ringing and it is a member of a group which is one of your pickup groups, then you can answer it by picking up using the pickup code. For simple offices, just make these both the same positive integer.

callprogress, progzone - This option is similar to the busydetect option, but will also listen for a ringing tone, congestion tone and will try to detect if the line got answered. However, this feature is experimental and can easily detect false answers and errors. Few zones are supported, but may be selected with progzone [yes / no, and two letter country code for progzone].

callreturn - Support call return. [yes / no]

callwaiting - Enable call waiting on FXO lines. [yes / no]

callwaitingcallerid - Sets whether to receive caller ID during call waiting indication. [yes / no]

canpark - Allow call parking. [yes / no]

cidsignalling - Type of caller ID signalling in use.

bell = bell202 (US)

v23 = v23 (UK)

dtmf = DTMF (Denmark, Sweden, & Netherlands)

cidstart - Identifies what signals the start of caller ID.

ring = a ring signals the start

polarity = polarity reversal signals the start

context - Defines the initial context for the channel. This will be the context available to the channel upon the initiation of a call. Note that contexts are an important part of maintaining site security. The initial context will govern the availability of extensions to a given channel. If an extension is placed in a different context from the initial context, that extension is unavailable to the caller.

echocancel - Enable echo cancellation. [yes / no, or a power of two from 32 to 256 to set the number of cancellation taps]

echocancelwhenbridged - Generally, it is not necessary (and in fact undesirable) to echo cancel when the circuit path is entirely TDM. You may, however, reverse this behaviour by enabling the echo cancel during pure TDM bridging. [yes / no]

echotraining - In some cases, the echo canceller doesn't train quickly enough and there is echo at the beginning of the call. Enabling echo training will cause Asterisk to briefly mute the channel, send an impulse, and use the impulse response to pre-train the echo canceller so it can start out with a much closer idea of the actual echo. [yes, no, or a number of milliseconds (10 – 2000) to delay before training]

faxdetect - Upon fax detection, routes fax to a fax extension. [Both, incoming, outgoing, or no.]

group - Allows a number of channels to be treated as one for the purpose of dialing. For dialing out, the channels will be called on a first available basis. For the purpose of ringing stations, all channels in the group will ring at once. Takes an integer from 0 to 63 and multiple groups can be specified.

hanguponpolarityswitch - In some countries, a polarity reversal is used to signal the disconnect of a phone line. If this option is enabled, the call will be considered disconnected on a polarity reversal. [yes / no]

hidecallerid - Sets whether to hide outgoing caller ID, defaults to no. [yes / no]

immediate - Specify whether the channel should be answered immediately or if the simple switch should provide dialtone, read digits, etc. [yes / no]

musiconhold - Select which class of music to use for music on hold. If not specified then the default will be used. The music class is defined in musiconhold.conf file. [default, loud, random]

pulsedial - Use pulse dial instead of DTMF for FXO (FXS signalled) devices. [yes / no]

relaxdtmf - If you have trouble with DTMF detection, you can relax the DTMF detection parameters. Relaxing them may make the DTMF detector more likely to have talk off where DTMF is detected. [yes / no]

rxgain - Adjusts receive gain. This can be used to raise or lower the incoming volume to compensate for hardware/line differences. [Positive or negative double, measured in dB]

sendcalleridafter - Some countries have ring tones with a set of cadences which differ from the default. This requires the callerid to be set with a delay, and not right after the first ring. [Positive integer]

threewaycalling - Support three-way calling. [yes / no]

transfer - Support flash-hook call transfer (requires three way calling). Also enables call parking (overrides the **canpark** parameter). [yes / no]

txgain - Adjusts transmit. This can be used to raise or lower the outgoing volume to compensate for hardware/line differences. [Positive or negative double, measured in dB]

usecallerid - Whether or not to use caller ID. [yes / no]

usedistinctiveringdetection - Indicates whether or not to allow distinctive ring detection on FXO lines. [yes / no]





















Appendix D - X100P SE PCI Card Information

Product Overview

The X100P Special Edition (SE) is a single port analogue computer telephony PCI card developed by X100P.com specifically for leading open source IP PBX software such as Digium Asterisk® and Trixbox®. The X100P SE provides a single, full featured FXO interface for connecting an Open Source IP PBX to the PSTN (Public Switched Telephone Network). The X100P SE works out of the box using the official Zaptel driver and is perfect for home and SOHO (Small Office Home Office) environments.

Product Comparison

The X100P SE is the best of breed X100P card available. The table below provides a comparison between the X100P SE and some of the X100 cards available from alternative suppliers.

	Authentic X100P SE	Motorola Wildcard	ATCOM AX-100P	Voxzone X100P	Additional Information
Global Line Standards Support					Supports complex impedance for UK/Europe, Australia and New Zealand
Built in Pass-Thru Port for Analogue Telephone					Built in telephony service backup in case of power or IP PBX system failure
3.3v and 5v PCI Bus Support					3.3v required for low power embedded appliances and some newer PCs
100% Surface Mounted Electrical Components					Most of the ATCOM components are surface mounted but not all
Low Profile PCI Bracket and RJ-45 Modem Cable					Low profile PCI bracket required for 2U high rack servers

Product Highlights

Global Line Standards and Caller ID Support

- The X100P SE uses Silicon Labs Si3014/Si3034 DAA chip which supports global line standards including Complex Impedance for UK/Europe, Australia and New Zealand
- Supports Caller ID including countries such as the UK using a Zaptel/Asterisk patch
- Full instructions provided for configuring Global line standards and UK Caller ID support for Trixbox® 2.6.07, instructions can be modified to work with any Asterisk based IP PBX system such as AsteriskNOW®, Elastix®, or PBX in a Flash

Built-in Pass-Thru Port for Connecting an Analogue Telephone

- Enables use of the phone line while not occupied by Asterisk
- Provides a telephony service backup capability in case of power or IP PBX system failure
- Intelligent Call Handling: Introducing a pickup delay allows users to pick up when in-house

Industrial Grade Surface Mounted Electronic Components

- Minimal Signal Interference: Due to significant reductions in component size and lead length
- Increased Reliability: Better mechanical performance under shake and vibration conditions
- Reduced Heat Transference: Reduces chance of damaging the circuit board
- Shipment Safe: Not a single protruding component (except for the RJ-11 phone line socket)



Authentic X100P Special Edition



Inferior, cheaper product

Multiple QC Tests Performed Under Official Asterisk Releases

- Every card Quality Tested/Assured at the X100P.com factory
- Voice Quality, Inbound and Outbound
- Voice Level, Inbound and Outbound
- Dual-tone multi-frequency (DTMF) Dialling, Outbound
- DTMF Detection, Inbound



```

[QC Test]
exten => s,1,wait,1
exten => s,2,answer
exten => s,3,digittimeout,5
exten => s,4,noop(-----)
exten => s,5,noop( )
exten => s,6,noop(CallerID: "${CALLERID}")
exten => s,7,noop( )
exten => s,8,noop(-----)
exten => s,9,wait,2
exten => s,10,record(testsound:rulaw,,4)
exten => s,11,playtones(record)
exten => s,12,wait,1
exten => s,13,background(testsound)
exten => s,14,waittexten(2)
exten => s,15,goto(10)
    
```

Excerpts from the Asterisk Dial plan used in the X100P.com QC Department

User Changeable Standard and Low Profile PCI Bracket Support

- Low Profile Bracket: Perfect for embedded appliance and Small Footprint server/PC Implementations
- Standard and Low Profile PCI Bracket Included with Every Purchase



Authentic X100P Special Edition
Low Profile Bracket Installed



Authentic X100P Special Edition
Standard Profile Bracket Installed

Enhanced PCI Bus Connectivity and Stability

- High Quality Printed Circuit Board (PCB) with Golden Pins
- Enables Stable PCI Connection
- No More Fatal 'PCI Master Abort'



Un-Retouched Photo Showing:
Authentic X100P Special Edition (Top)
Inferior, cheaper product (Bottom)

Technical Summary

- Half length PCI card compatible with nearly all commercially available PC motherboards
- Fully PCI 2.2 compliant: Auto-sense compatibility for both 5v and 3.3v PCI busses
- Full hardware and software compatibility: Compatible with most well known brands of open source computer telephony equipment
- Global telephony system support: Configurable line interface to meet international telephone line interface requirements
- Surface mounted industrial grade components: Minimal signal interference and increased reliability
- Solid tantalum capacitors: High volumetric efficiency/reliability, low Equivalent Series Resistance (ESR) and temperature stability over the range -55°C to +85°C
- Enhanced PCI Bus Connectivity and Stability
- Latest Revision of the Original DAA chipsets with numerous Bug Fixes
- Support for Digital Gain Control (Transmit and Receive)
- User Changeable Standard and Low Profile PCI Bracket Support
- Built-in Pass-Thru port: Enables use of the phone line while not occupied by Asterisk
- Dimensions excluding PCI bracket (H x L x W): 47mm x 120mm x 16mm / 1.8" x 4.7" x 0.6"
- Certificates: CE, FCC

Appendix E - Acronyms

Acronym	Description
ACPI	Advanced Configuration and Power Interface
APIC	Advanced Programmable Interrupt Controller
BT	British Telecom
CID	Caller Identity
CLI	Command Line Interface
CO	Central Office
CTR21	Common Technical Requirements directive 21
DTMF	Dual-Tone Multi-Frequency
FCC	Federal Communications Commission
FXO	Foreign Exchange Office
FXS	Foreign Exchange Station/Subscriber
GUI	Graphical User Interface
I/O	Input / Output
IDE	Intelligent/Integrated Drive Electronics
IP	Internet Protocol
IP PBX	IP Private Branch Exchange
IRQ	Interrupt Request
JATE	Japan Approvals Institute for Telecommunications Equipment
PCB	Printed Circuit Board
PCI	Peripheral Component Interconnect
POTS	Plain Old Telephony Service
PSTN	Public Switched Telephony Network
SCSI	Small Computer System Interface
SOHO	Small Office Home Office
SMP	Symmetric Multi-Processing
UK	United Kingdom
VoIP	Voice over IP