Zaptel.conf

How to configure the low-level Zapata Interface Library for your hardware.

About this document

Most devices sold by Digium are members of the zaptel family of hardware devices. These devices share a common driver suite, called the Zapata Telephony Driver Suite (or zaptel, for short) and a common interface library.

Before using this document, you need to have sucessfully installed the zaptel, zapata, and asterisk software on your system. If you are using T1 or E1 hardware, you will also need to install libpri. See the 'Quick Install Guide for Asterisk and Zaptel Drivers,' available from www.asterisk.org in the Documentation section, for information on obtaining and installing those software packages.

This document currently covers the Wildcard family of products. As of this date, the complete list of devices is:

S100U, a single FXS(line-side) interface which connects to the USB port.
X100P, a single FX0(phone-side) interface which connects to the PCI bus.
T100P, a single-span T1 interface which connects to the PCI bus.
T400P, a four-span T1 interface which connects to the PCI bus.
E400P, like above, but designed for the E1 interface common outside the US.

In addition, any FXS, FXO, or T1/E1/PRI interface purchased from Digium is probably covered by this document.

Updated versions of this document will be made available immediately on the www.asterisk.org website.

About the Style of /etc/zaptel.conf

The zaptel.conf configuration is designed to be as easy to configure as possible. The format line by line is quite simple, similiar to the *.ini files used in older MS systems. Each line is like so:

parameter=value

Lines beginning with a hash(#) symbol are comments, and will be ignored by the software when the file is read. The default configuration file contains much of this information in these comment lines.

Configuring T1/E1 spans

The first lines of the config file (comments excluded) will be the span definitions for T1/E1

interfaces. If you are configuring FXO or FXS interfaces, you may skip this part and move down to section ___. If you are configuring T1/E1 interfaces, you will need some information about the device you are communicating with. You will find it helpful to have any documentation for the device you are configuring the T1 interfaces to speak to. The span definitions are slightly different from the other lines of the file, as they take several values.

A span definition is in this format:

span=(spannum),(timing),(LBO),(framing),(coding)

The exact values used here are dependent on the configuration of equipment at the other end of the line. Timing defines how you want to synchronize the timing of the devices. To not use this span as a sync source, use '0'. To make it the primary sync source, use '1', secondary is '2' and so forth.

0: 0 db (CSU) / 0-133 feet (DSX-1)
1: 133-266 feet (DSX-1)
2: 266-399 feet (DSX-1)
3: 399-533 feet (DSX-1)
4: 533-655 feet (DSX-1)
5: -7.5db (CSU)
6: -15db (CSU)7: -22.5db (CSU)

Line Build Out (LBO) is taken from the table above.

The next two values define how you will communicate with the hardware at the other end of the line.

For T1 - Framing is one of **d4** or **esf**. Coding is one of **ami** or **b8zs.**
For E1 – Framing is one of **cas** or **ccs**. Coding is one of **ami** or **hdb3.** E1's spans may also need to enable crc checking

A typical configuration for talking to a channel bank would look like this:

span=1,0,0,esf,b8zs

which would configure span 1 to NOT act as a sync source, set the line build out appropriate for 0-133 foot line run, and set the framing to esf, with b8zf encoding. Optionally a span definition can be followed by the keyword 'yellow' which will cause a yellow alarm signal to be transmitted when no channels are open.

A common E1 configuration might look like this:

span=1,0,0,cas,ami,crc

Leave off the ‘,crc’ if crc checking should not be enabled.

Configuring channels

If you are using only FXS or FXO direct devices, such as the X100P single port FXO or the S100U single FXS, you will start with this section, as the span definitions do not apply to these devices.

For the FXO and FXS interfaces, such as the X100P and S100U, channels will appear in the order that the drivers are loaded. For instance, if you have a single port FXO card and a USB single port FXS interface, you would load the drivers one by one. If you load the FXO card driver, followed by the USB FXS driver, the FXO would be channel 1, and the USB FXS would be channel 2.

For FXO and FXS, signaling is the reverse of the type of signaling the interface itself offers. FXS interfaces are signaled with FXO, and vice versa.

Configuring these interfaces is one-line simple. For our example of one FXO and one FXS, the config would look like this:

```
fxsks=1
fxoks=2
```

Note that we loaded the FXO device as channel one, and the FXS device as channel 2. Since we signal them with the reverse, we define the channels for fxsks signaling on channel 1, and fxoks signaling on channel 2.

The T1 and E1 cards have a more slightly complex configuration, since their signaling is dependent upon what is supported by the other end of the link. For the purpose of this document we will assume that the other end of the link is a channel bank.

Each T1 span carries 24 channels. E1 spans are 30 channels. Our example will be a T1 channel bank will be loaded with 8 FXO channels and 16 FXS channels.

Essentially identical to setting up the single interfaces, to set this up, we would use these lines to configure the devices.

```
fxsks=1-8
fxoks=9-24
```

Some Notes on the Order of Devices

Devices will appear as channels in the order they are loaded. For example, If an FXO and an FXS card are present in the system, one might load the FXO card driver, followed by the

FXS channel driver. The FXO device would be channel 1 (a multiport would be 1-(# of ports)). The FXS device would be channel 2 (or the first port after the FXO, in the case of multiport devices). Devices can generally be loaded in any order. The exception to this is T1/E1 interface cards. Their drivers must be loaded first, before the Station or USB based interface cards. Also, even if only one span is enabled, the first Station card in the device appear as the total number of channels **available** on the T1/E1 card plus one. If the T1 card driver is loaded, and an FXO driver is loaded afterwards, the FXO will be channel 97 (after the 96 channels available from the T400P) regardless of whether you are using all 96 channels.

Other hardware such as the X100P should have the drivers loaded afterwards, and therefore they will pick up in the config file as the first channel after the last span's channel definition. Therefore, if there is a single span T1 card, it would be configured as channel 1-24, with any internal or USB devices taking channels 25 and up, until all devices are configured.

Using T1/E1 cards for other applications (non-voice)

Zapata T1/E1 interfaces are most commonly used for voice applications, as interfaces between The Asterisk Private Branch Exchange Server and Channel Banks. However, the T1/E1 cards do support other modes of operation, most notably HDLC for data traffic. Alternate modes can be configured on a channel-by-channel basis. Signalling types available are:

"e&m" : Channel(s) are signalled using E&M signalling
(specific implementation, such as Immediate, Wink,
or Feature Group D are handled by the userspace library).

"fxsls" : Channel(s) are signalled using FXS Loopstart protocol.
"fxsgs" : Channel(s) are signalled using FXS Groundstart protocol.
"fxsks" : Channel(s) are signalled using FXS Koolstart protocol.
"fxols" : Channel(s) are signalled using FXO Loopstart protocol.
"fxogs" : Channel(s) are signalled using FXO Groundstart protocol.
"fxoks" : Channel(s) are signalled using FXO Koolstart protocol.
"unused" : No signalling is performed, each channel in the list remains idle
"clear" : Channel(s) are bundled into a single span. No conversion or
signalling is performed, and raw data is available on the master.
"indclear": Like "clear" except all channels are treated individually and
are not bundled. "bchan" is an alias for this.
"rawhdlc" : The zaptel driver performs HDLC encoding and decoding
on the bundle, and the resulting data is communicated via
the master device.

"fcshdlc" : The zapdel driver performs HDLC encoding and decoding on the
bundle and also performs incoming and outgoing FCS insertion
and verification. "dchan" is an alias for this.

"nethdlc" : The zaptel driver bundles the channels together into an
hdlc network device, which in turn can be configured with
sethdlc (available separately).

The most common non-voice application for our T1/E1 cards is PRI data modes.

Digium
6703 Odyssey Drive, Suite 104
Huntsville, Alabama, 35806
www.digium.com