

Upgrading from Zaptel to DAHDI

Revision History

Revision	Date	By	Notes
0.1	December 12, 2008	Glenn Jenkins	Final Draft
0.2	December 15, 2008	Bill Savage	Review
0.3	January 6, 2009	Bill Savage	Final Draft



Upgrading from Zaptel to DAHDI

As announced in early 2008, Digium has renamed the Zaptel telephony interface to DAHDI in order to accommodate the desires of the owner of the Zaptel trademark for telephony purposes. The Digium Asterisk Hardware Device Interface (DAHDI) is a collection of drivers and utilities for connecting your Asterisk installation to the Public Switched Telephone Network (PSTN). dahdi-linux is the package that contains the Linux kernel mode drivers, while dahdi-tools contain the user mode scripts and utilities for working with dahdi-linux.

DAHDI is supported by Asterisk versions 1.4.22 and greater as well as Asterisk versions 1.6.0 and greater. DAHDI 2.1.0 is the current release of the DAHDI Linux kernel modules package, which replaces the kernel modules from Zaptel. The primary purpose of the 2.0.0 release was to rename the package from Zaptel. The primary purpose of the 2.1.0 release was to incorporate support for BRI. A short list of bug fixes as well as new features is provided below:

- * Echo cancellers can now be applied per channel and selected at configuration time.
- * Channel memory allocation changed from one large block into smaller blocks in order to reduce out of memory errors on a system that has been running for some time.
- * Layout changes to support binary packaging.
- * Neon MWI support added to the wctdm24xpx driver.
- * Dropped support for Linux Kernel 2.4 as well as the torisa and wusb drivers.
- * Native driver for the [B410P](#) four port BRI module

Important General Upgrade Notes

- * Zaptel 1.2 will continue to be released in new versions as bugs are found and fixed.
- * Asterisk 1.2 will be unaffected by these changes.
- * Now that DAHDI 2.0.0 has been released there will be no more releases of Zaptel 1.4.
- * Asterisk 1.4 will continue to have support for Zaptel, although it will be enhanced to also transparently support DAHDI instead, and the documentation (and default configuration files) will encourage new users to use DAHDI instead of Zaptel.
- * Asterisk 1.4 releases later than 1.4.21, and all releases of Asterisk 1.6, will automatically use DAHDI in preference to Zaptel, even if Zaptel is still installed on the system.
- * Asterisk 1.6.0 will use DAHDI exclusively (it will not have support for Zaptel), although this will be done in a highly backwards-compatible way so that disruption to existing dialplans and configurations will be minimized.



DAHDI 2.1.0

The Zaptel package, which included both kernel modules and user space tools for configuring and managing the modules, has been split into two packages:

The list of packages released includes:

- * dahdi-linux 2.1.0 : kernel modules
- * dahdi-tools 2.1.0 : user space tools
 - o The dahdi-linux and dahdi-tools packages have **separate** version numbers; they will not be released 'in sync', and it is perfectly acceptable to use (for example) dahdi-tools 2.0.6 with dahdi-linux 2.0.11.
- * dahdi-linux-complete 2.1.0+2.1.0
 - o The dahdi-linux-complete package version number will always include **both** of these version numbers so that you will know what is included in it.

Both dahdi-linux and dahdi-tools are required to enable DAHDI support in your system. You will need to install dahdi-linux first and then dahdi-tools, and finally you can configure and make Asterisk. dahdi-linux-complete is both dahdi-linux and dahdi-tools combined into one download as a convenience. You still need libpri for PRI support with Asterisk if you are using DAHDI.

DAHDI is supported in both Asterisk 1.4.22 and Asterisk 1.6.0.

Notes about the DAHDI-LINUX Package

The primary kernel modules have changed names; the new names are:

- * zaptel.ko -> dahdi.ko
- * ztd-eth.ko -> dahdi_dynamic_eth.ko
- * ztd-loc.ko -> dahdi_dynamic_loc.ko
- * ztdummy.ko -> dahdi_dummy.ko
- * ztdynamic.ko -> dahdi_dynamic.ko
- * zttranscode.ko -> dahdi_transcode.ko



The kernel modules for card drivers have *not* changed names, although the wusb and torisa drivers are no longer included.

This package no longer includes the 'menuselect' utility for choosing which modules to build; all modules that can be built are built automatically.

It is no longer possible to select a software echo canceller at compile time to build into dahdi.ko. All four included echo cancellers (MG2, KB1, SEC and SEC2) are built as loadable modules, and if the Digium HPEC binary object file has been placed into the proper directory the HPEC module will be built as well. Any or all of these modules can be loaded at the same time, and the echo canceller to be used on the system's channels can be configured using the dahdi_cfg tool from the dahdi-tools package.

Note: It is **mandatory** to configure an echo canceller for the system's channels using dahdi_cfg unless the interface cards in use have echo canceller modules available and enabled. There is **no** default software echo canceller with DAHDI.

Notes about the DAHDI-Tools Package

Many tool names have changed:

- ztcfg -> dahdi_cfg
- ztmonitor -> dahdi_monitor
- ztscan -> dahdi_scan
- ztspeed -> dahdi_speed
- zttest -> dahdi_test
- zttool -> dahdi_tool
- zapconf -> dahdi_genconf (deprecates genzaptelconf)

The system configuration file has moved from /etc/zaptel.conf to /etc/dahdi/system.conf.

The dahdi_cfg tool can now be used to select an echo canceller on a channel-by-channel basis in the system configuration file. See the system.conf sample for examples of how to do this.

Asterisk 1.4 DAHDI Notes

This version of Asterisk can be built using either Zaptel or DAHDI, and has many changes to make the use of DAHDI as easy as possible for existing users with dialplans, CDR parsers, AMI applications, and others that expect Zaptel to be in use.

First, the modules that directly use services from Zaptel/DAHDI have been renamed; the new names are:

- `chan_zap.so` -> `chan_dahdi.so`
- `app_zapbarge.so` -> `app_dahdibarge.so`
- `app_zapras.so` -> `app_dahdiras.so`
- `app_zapscan.so` -> `app_dahdiscan.so`
- `codec_zap.so` -> `codec_dahdi.so`

However, in spite of the file name changes, the channels and applications provided by these modules can still be used with 'Zap' style names; see below for more information.

Second, there have been a number of efforts made to ensure that existing systems will not require any major configuration changes solely because Asterisk was built against DAHDI instead of Zaptel. This includes:

chan_dahdi.so:

This module will determine which channel name ('Zap' or 'DAHDI') should be used for incoming and outgoing channels based on the build-time choice of telephony drivers. However, if you wish to continue using the 'Zap' channel name even though you built Asterisk against the DAHDI drivers, you can add the following line to the [options] section of your `/etc/asterisk/asterisk.conf` file.

dahdichannname = no

- All CLI commands that begin with 'zap' are now available as 'dahdi' commands. The 'zap' variants will report that they are deprecated the first time you use each one in an Asterisk instance. They will otherwise operate just as they did in previous versions.
- All Asterisk Manager Interface (AMI) actions that begin with 'Zap' are also available with 'DAHDI' prefixes.



- The ZapSendKeypadFacility dialplan application is now available as DAHDISendKeypadFacility. The Zap variant will report a deprecation warning but will otherwise operate as it did in previous versions.
- The configuration for the channel driver will be read from `/etc/asterisk/chan_dahdi.conf` unless 'dahdichannname' has been set to 'no' in `asterisk.conf`. If that is done, then the configuration will be read from `/etc/asterisk/zapata.conf`, just as it was in previous versions.

app_dahdibarge.so

The ZapBarge application is now available as DAHDIBarge. The ZapBarge variant will report a deprecation warning when used, but will otherwise operate as it did in previous versions. Regardless of which application name is used, the application will restrict itself to channels of the proper type, based on the 'dahdichannname' setting in `asterisk.conf`.

app_dahdiras.so

The ZapRAS application is now available as DAHDIRAS. The ZapRAS variant will report a deprecation warning when used, but will otherwise operate as it did in previous versions. Regardless of which application name is used, the application will restrict itself to channels of the proper type, based on the 'dahdichannname' setting in `asterisk.conf`.

app_dahdiscan.so

The ZapScan application is now available as DAHDIScan. The ZapScan variant will report a deprecation warning when used, but will otherwise operate as it did in previous versions. Regardless of which application name is used, the application will restrict itself to channels of the proper type, based on the 'dahdichannname' setting in `asterisk.conf`.

app_flash.so

This application has not had any name changes, but will report its usage (via 'show application flash') as being for either DAHDI or Zaptel channels based on the 'dahdichannname' setting in `asterisk.conf`.

app_chanspy.so

This application will transparently create 'DAHDI' or 'Zap' channels as needed, based on the 'dahdichannname' setting in `asterisk.conf`.



app_meetme.so

This application will transparently create 'DAHDI' or 'Zap' channels as needed, based on the 'dahdichannname' setting in asterisk.conf.

Asterisk 1.6 to DAHDI Notes

This version of Asterisk can only be built using DAHDI, and as a result there are number of changes that will be visible to previous users of Asterisk with ZapTEL.

First, the modules that directly use services from DAHDI have been renamed; the new names are:

- chan_zap.so -> chan_dahdi.so
- app_zapbarge.so -> app_dahdibarge.so
- app_zapras.so -> app_dahdiras.so
- app_zapscan.so -> app_dahdiscan.so
- codec_zap.so -> codec_dahdi.so

Second, the behavior of many modules has changed due to the switch to DAHDI. The changes are listed below.

chan_dahdi.so:

- Incoming and outgoing channels managed by this module will be 'DAHDI' channels instead of 'Zap' channels.
- All CLI commands that began with 'zap' have been renamed to 'dahdi' commands.
- All Asterisk Manager Interface (AMI) actions that began with 'Zap' have changed to 'DAHDI' prefixes.
- The ZapSendKeypadFacility dialplan application has been renamed to DAHDISendKeypadFacility.
- The configuration for the channel driver will be read from /etc/asterisk/chan_dahdi.conf instead of /etc/asterisk/zapata.conf.



app_dahdibarge.so:

The ZapBarge application has been renamed to DAHDIBarge.

app_dahdiras.so:

The ZapRAS application has been renamed to DAHDIRAS.

app_dahdiscan.so:

The ZapScan application has been renamed to DAHDIScan.

