

第二章 Asterisk 体系结构

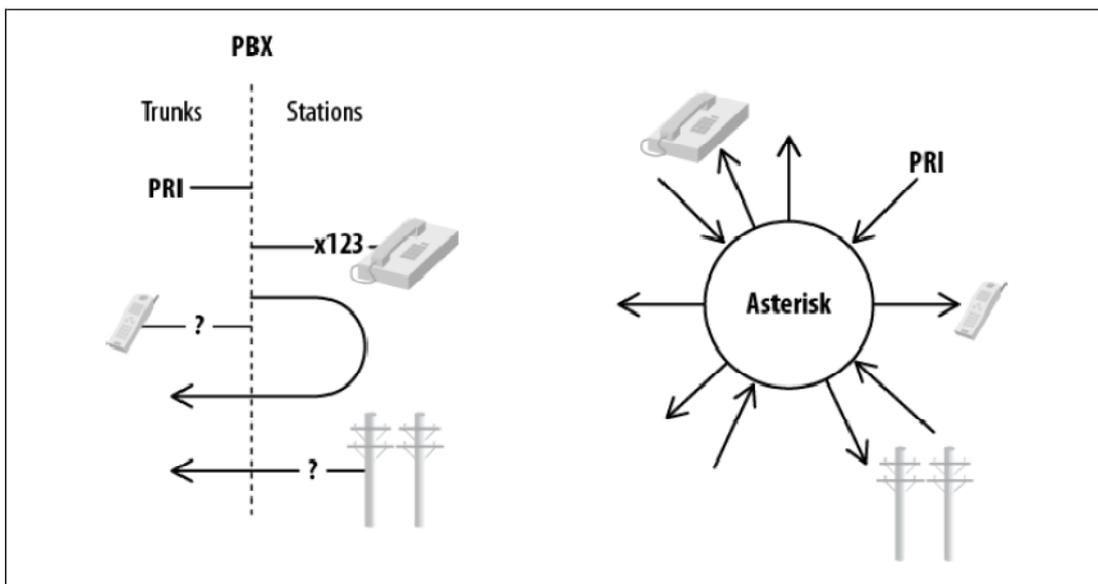
目录

2.1	Modules	2
2.1.1	Dialplan Applications.....	3
2.1.2	Bridging Modules	6
2.1.3	Call Detail Recording Modules	6
2.1.4	Channel event logging modules	7
2.1.5	Channel Drivers	7
2.1.6	Codec Translators	8
2.1.7	Format Interpreters.....	9
2.1.8	Dialplan Functions.....	10
2.1.9	PBX Modules	12
2.1.10	Resource Modules.....	12
2.1.11	Addon Modules.....	14
2.1.12	Test Modules	15
2.2	文件结构.....	15
2.2.1	Configuration files	15
2.2.2	Modules	15
2.2.3	The Resource Library	15
2.2.4	The Spool.....	16
2.2.5	Logging	16
2.3	The Dialplan	16
2.4	硬件.....	16
2.5	Asterisk 版本.....	17
2.5.1	以前的版本号机制.....	17
2.5.2	新的版本机制.....	18
2.6	结论.....	18

Asterisk 完全不同于其他的，更加传统的 PXB 系统。在 Asterisk 中，dialplan 基本上采用完全一样的方法处理所有呼入 channel。

在传统 PBX 中，“内线”和“外线”在逻辑上是分开的。举例来说，你不能把一个外部网关连接到 PBX 的内线分机端口上，用户也无法在不按下外线数字的情况下拨打外线（例如，很多的 IP PBX 拨打外线要加拨“0”）。

与此相反，在 Asterisk 中并没有严格的“内线”和“外线”的概念。在 Asterisk 中，所有进、出系统的数据都要通过某种类型的 Channel。Asterisk 中的 Channel 种类很多，但 Asterisk dialplan 都以类似的方法来处理这些 Channel。这就意味着，一个“内线”用户在物理上可以处在“外线”网络上（例如手机），并且被 dialplan 以处理内线分机一样的方法进行处理。除非你使用过传统 PBX，否则很难立即理解 Asterisk 的这一变化是何等的强大。



图表 2-1 Asterisk 与传统 PBX 的结构比较

2.1 Modules

Asterisk 是基于 Modules 构建的。Modules 是一种可加载的，实现特殊功能的组件，比如 channel driver（例如 chan_sip.so）；或者是一种可以用来访问其它技术的资源（例如 func_odbc.so）。Asterisk modules 是根据/etc/asterisk/modules.conf 中的配置来进行加载的。在本书中，我们将讨论很多的 modules。现在，我们先只介绍一些 modules 的基本概念，并让你对 modules 的类型有点感觉。

我们可以在不加载任何 modules 的情况下启动 Asterisk，尽管在这种状态下 Asterisk 什么也做不了。但这有助于我们理解和体会 Asterisk modules 的类型和结构。



你可以启动 Asterisk 而不加载任何默认 modules，然后再通过控制台手动加载，但这显然不是我们期望的产品特性。这种做法仅仅在我们调试系统性能时会比较有用，我们可以去除特定系统不需要的所有其它 modules。

Asterisk 中的 modules 类型包括：

- Dialplan Applications
- Bridging modules

- Call detail recording (CDR) modules
- Channel event logging (CEL) modules
- Channel drivers
- Codec translators
- Format interpreters
- Dialplan functions
- PBX modules
- Resource modules
- Addons modules
- Test modules

在本章后续部分，我们将按上述分类罗列出有效的 modules，简述其目的，并且给出我们对于每个 modules 相对流行性或重要性的看法（一些 modules 非常流行，而另一些则很少有人应用，维护它们仅仅是为了前向兼容）。某个特定 modules 的工作细节，将在本书的其它章节讨论。根据不同 modules 的类型和作用不同，有些 modules 会被详细讨论，而有些 modules 则根本不会讨论。

在后续表格中的 Popularity/Status 列，是按照我们观点对各个 modules 给出的评价，不同评价的含义如下：

Insignificant

这个 module 是过时的老古董，如果你要使用的话，你很可能是唯一的使用者。

Unreliable

这个 module 是新的或试验性的，不适合应用在产品中。

Useful

这个 module 是现行维护的流行版本，推荐使用。

New

这个 module 非常新，现在很难判断它的完成程度和受欢迎情况。

Deprecated

这个 module 已经被更优秀的其它 module 取代。

Limited

这个 module 是被限制使用的，这可能导致它不适合你的需求。

Essential

这个 module 是你不可或缺的。

现在，让我们根据类型分组来看看这些 modules。

2.1.1 Dialplan Applications

Dialplan applications 用于在 *extensions.conf* 中定义可作用于呼叫的各种操作。举例来说，*Dial()* 负责与外部资源建立联接。*Dial()* 是最重要的 Dialplan applications。所有有效的 Applications 参见表 2-1。

Table 2-1. Dialplan applications

Name	Purpose	Popularity/Status
app_adsiprog	Loads Analog Display Services Interface (ADSI) scripts into compatible analog phones	Insignificant
app_alarmreceiver	Supports receipt of reports from alarm equipment	Insignificant
app_amd	Detects answering machines	Unreliable
app_authenticate	Compares dual-tone multi-frequency (DTMF) input against a provided string (password)	Useful
app_cdr	Writes ad hoc record to CDR	Useful
app_celgenuserevent	Generates user-defined events for CEL	New
app_chanisavail	Checks the status of a channel	Unreliable
app_channelredirect	Forces another channel into a different part of the dialplan	Useful
app_chanspy	Allows a channel to listen to audio on another channel	Useful
app_confbridge	Provides conferencing (new version)	New—not fully featured yet
app_controlplayback	Plays back a prompt and offers fast forward and rewind functions	Useful
app_dahdibarge	Allows barging in on a DAHDI channel	Deprecated—see app_chanspy
app_dahdiras	Creates a RAS server over a DAHDI channel (no modem emulation)	Insignificant
app_db	Used to add/change/delete records in Asterisk's built-in Berkeley database	Deprecated—see func_db
app_dial	Used to connect channels together (i.e., make phone calls)	Essential
app_dictate	Plays back a recording and offers start/stop functions	Useful
app_directed_pickup	Answers a call for another extension	Useful
app_directory	Presents the list of names from <i>voicemail.conf</i>	Useful
app_disa	Provides dialtone and accepts DTMF input	Useful ^a
app_dumpchan	Dumps channel variables to Asterisk command-line interface (CLI)	Useful

Name	Purpose	Popularity/Status
app_read	Requests input of digits from callers and assigns input to a variable	Useful
app_readexten	Requests input of digits from callers and passes call to a designated extension and context	Usable
app_readfile	Loads contents of a text file into a channel variable	Deprecated—see the FILE() function in func_env
app_record	Records received audio to a file	Useful
app_rpt	Provides a method to interface with an audio board for the app_rpt project	Limited
app_sayunixtime	Plays back time in specified format	Useful
app_senddtmf	Transmits DTMF to calling party	Useful
app_sendtext	Sends a text string to compatible channels	Insignificant
app_setcallerid	Sets CallerID on a channel	Deprecated—see func_callerid
app_skel	Sample application for developers	Useful ^c
app_sms	Sends SMS message in compatible countries	Limited
app_softhangup	Requests hangup of channel	Useful
app_speech_utils	Provides utilities relating to speech recognition	Useful ^d
app_stack	Provides Gosub(), GoSubIf(), Return(), Stack Pop(), LOCAL(), and LOCAL_PEEK()	Essential
app_system	Executes commands in a Linux shell	Useful
app_talkdetect	Similar to app_background, but allows for any received audio to interrupt playback	Useful
app_test	Client/server testing application	Usable
app_transfer	Performs a transfer on the current channel	Useful
app_url	Passes a URI to the called channel	Limited
app_userevent	Generates a custom event in the Asterisk Manager Interface (AMI)	Useful
app_verbose	Generates a custom event in the Asterisk CLI	Useful
app_voicemail	Provides voicemail	Essential
app_waitforring	Waits for a RING signaling event (not to be confused with RINGING); most likely unnecessary, as only chan_dahdi with analog channels where ringing is received (such as an FXO port) generates the RING signaling event	Insignificant
app_waitforsilence	Includes WaitForSilence() and WaitForNoise(); listens to the incoming channel for a specified number of milliseconds of noise/silence	Useful
app_waituntil	Waits until current Linux epoch matches specified epoch	Useful

Name	Purpose	Popularity/Status
app_while	Includes While(), EndWhile(), ExitWhile(), and ContinueWhile(); provides while-loop functionality in the dialplan	Useful
app_zapateller	Generates SIT tone to discourage telemarketers	Usable

^a The use of (DISA) is considered to be a security risk.

^b Requires a suitable DSP engine to handle encoding/decoding of fax signaling (see Chapter 19).

^c If you are a developer.

^d Requires an external speech recognition application.

2.1.2 Bridging Modules

Bridging Modules 是在 Asterisk1.8 中新增的一组 Modules 。它们在新的 Bridging API 中负责实际的 channel 间的桥接工作。每个 Bridging Modules 提供不同的特性。有效的 Bridging Modules 列表在 Table2-2 中，需要注意的是，目前 Bridging Module 只能（也必须）应用在 app_confbridge 中。

Table 2-2. Bridging modules

Name	Purpose	Popularity/Status
bridge_builtin_features	Performs bridging when utilizing built-in user features (such as those found in <i>features.conf</i>).	New
bridge_multiplexed	Performs complex multiplexing, as would be required in a large conference room (multiple participants). Currently only used by app_confbridge.	New
bridge_simple	Performs simple channel-to-channel bridging.	New
bridge_softmix	Performs simple multiplexing, as would be required in a large conference room (multiple participants). Currently only used by app_confbridge.	New

2.1.3 Call Detail Recording Modules

在 Table2-3 中列出了有效的 CDR modules。它们被用来实现能够以尽量多的方法存储通话详单。你可以把通话详单（CDRs）存储在文件中（默认情况），存储在数据库存中，通过 RADIUS 存储，或者通过 syslog 存储。



通话详单并不适合用在复杂的计费系统中。如果你要求实现比计费和通话报告更多的管理，你还需去看 Channel event logging。我们认为 CRD 的优点只是它能够工作而已。

Table 2-3. Call detail recording modules

Name	Purpose	Popularity/Status
cdr_adaptive_odbc	Allows writing of CDRs through ODBC framework with ability to add custom fields	Useful
cdr_csv	Writes CDRs to disk as a comma-separated values file	Usable
cdr_custom	As above, but allows for the addition of custom fields	Useful
cdr_manager	Outputs CDRs to Asterisk Manager Interface (AMI)	Useful
cdr_odbc	Writes CDRs through ODBC framework	Usable
cdr_pgsq1	Writes CDRs to PostgreSQL	Useful
cdr_radius	Writes CDRs to RADIUS	Usable—does not support custom fields
cdr_sqlite	Writes CDRs to SQLite2 database	Deprecated—use sqlite3_custom
cdr_sqlite3_custom	Writes CDRs to SQLite3 with custom fields	Useful
cdr_syslog	Writes CDRs to <i>syslog</i>	Useful
cdr_tds	Writes CDRs to Microsoft SQL or Sybase	Usable—requires an old version of libtds

我们将在第 25 章讨论一些你可能希望和 CDR 一起使用的 Reporting packages。

2.1.4 Channel event logging modules

Channel event logging 提供了比呼叫报告强大的多的管理能力。出于同样原因，使用 CEL 需要谨慎的设计你的 Dialplan，否则不要期望它会自动工作正常。Asterisk 有效的 CEL modules 参见 Table2-4。

Table 2-4. Channel event logging modules

Name	Purpose	Popularity/Status
cel_custom	CEL to disk/file	Useful
cel_manager	CEL to AMI	Useful
cel_odbc	CEL to ODBC	Useful
cel_pgsq1	CEL to PostgreSQL	Useful
cel_radius	CEL to RADIUS	Usable—does not support custom fields
cel_sqlite3_custom	CEL to SQLite3	Useful
cel_tds	CEL to Microsoft SQL or Sybase	Usable—requires an old version of libtds

2.1.5 Channel Drivers

如果没有 channel drivers，Asterisk 根本无法实现任何呼叫。每一个 channel driver 都对应支持一个特定的协议或 channel 类型 (SIP, ISDN, 等等)。Channel modules 起着沟通 Asterisk 内核桥梁的作用。Asterisk 有效的 channel drivers 参见 Table2-5。

Table 2-5. Channel drivers

Name	Purpose	Popularity/Status
chan_agent	Provides agent channel for Queue()	Useful
chan_alsa	Provides connection to Advanced Linux Sound Architecture	Useful
chan_bridge	Used internally by the ConfBridge() application; should not be used directly	Essential ^a
chan_console	Provides connection to portaudio	New
chan_dahdi	Provides connection to PSTN cards that use DAHDI channel drivers	Useful
chan_gtalk	Provides connection to Google Talk	Usable
chan_h323	Provides connection to H.323 endpoints	Deprecated—see chan_ooh323 in Table 2-11
chan_iax2	Provides connection to IAX2 endpoints	Useful
chan_jingle	Provides connection to Jingle-enabled endpoints	Usable
chan_local	Provides a mechanism to treat a portion of the dialplan as a channel	Useful
chan_mgcp	Media Gateway Control Protocol channel driver	Usable
chan_misdn	Provides connection to mISDN supported ISDN cards	Limited
chan_multicast_rtp	Provides connection to multicast RTP streams	Useful
chan_nbs	Network Broadcast Sound channel driver	Insignificant
chan_oss	Open Sound System driver	Useful
chan_phone	Linux telephony interface driver (very old)	Insignificant
chan_sip	Session Initiation Protocol channel driver	Essential
chan_skinny	Cisco Skinny Client Control Protocol (SCCP) channel driver	Usable
chan_unistim	Nortel Unistim protocol channel driver	Usable
chan_usbradio	Channel driver for CM108 USB cards with radio interface	Usable
chan_vpb	Voicetronix channel driver	Insignificant ^b

^a If you are using the ConfBridge() application.

^b Some Voicetronix hardware is supported by Zaptel using an add-on Zaptel module distributed by Voicetronix. However, Zaptel is no longer supported by Asterisk and this driver has not been ported to DAHDI.

2.1.6 Codec Translators

Codec translators (Table 2-6) 允许 Asterisk 可以在不同的呼叫间转换音频流的数据格式。例如，如果从 PRI 上输入的一个呼叫（使用 G.711 编码）需要输出到压缩的 SIP Channel 上（例如，使用 G.729 编码），那么相关的 codec translator 就可以用来实现这个转换。



如果使用了算法复杂的编码（如 G.729），大量的编码转换工作会给 CPU 带来沉重的负担。此时最好使用诸如 Sangoma 或 Digium 这类硬件厂商提供的支持硬件编解码的板卡。

Table 2-6. Codec translators

Name	Purpose	Popularity/Status
codec_adpcm	Adaptive Differential Pulse Coded Modulation codec	Insignificant
codec_alaw	A-law PCM codec used all over the world (except Canada/USA) on the PSTN	Essential
codec_a_mu	A-law to mu-law direct converter	Useful
codec_dahdi	Utilizes proprietary Digium hardware transcoding card	Essential ^a
codec_g722	Wideband audio codec	Useful
codec_g726	Flavor of ADPCM	Insignificant
codec_gsm	Global System for Mobile Communications (GSM) codec	Useful
codec_ilbc	Internet Low Bitrate Codec	Insignificant
codec_lpc10	Linear Predictive Coding vocoder (extremely low bandwidth)	Insignificant
codec_resample	Resamples between 8-bit and 16-bit signed linear	Usable
codec_speex	Speex codec	Usable
codec_ulaw	Mu-law PCM codec used in Canada/USA on PSTN	Essential

^a If you are using a Digium codec transcoder card.

2.1.7 Format Interpreters

Format interpreters (Table 2-7) 也是实现编码格式转换，不过它针对的目标是文件，而不是 channel 中的媒体流。比如你录制了一个语音菜单并且以 GSM 格式保存，那么当你向任何不是使用 GSM 编码的 channel 播放这个语音菜单时都会用到 format interpreter。(注：我们并不推荐默认的 GSM 格式用于语音记录，WAV 格式占用更少的 CPU 资源而能提供优秀得多的声音效果。)

如果你用多种格式存储语音文件（例如 WAV，GSM，等等），当某个 channel 需要播放这个语音文件时，Asterisk 将会自动选用 CPU 消耗最小的格式。(注：一些编码会给 CPU 增加很大的计算负担，这样一来，本来在不做编码转换情况下能够支持数百个 channels 的系统，在做编码转换的情况下可能只能支持几十个 channels)。

Table 2-7. Format interpreters

Name	Plays files stored in	Popularity/Status
format_g723	G.723 .g723	Insignificant
format_g726	G.726 .g726	Insignificant
format_g729	G.729 .g729	Useful
format_gsm	RPE-LTP (original GSM codec) .gsm	Usable
format_h263	H.263—video .h263	Usable
format_h264	H.264—video .h264	Usable
format_ilbc	Internet Low Bitrate Codec .ilbc	Insignificant
format_jpeg	Graphic file .jpeg .jpg	Insignificant
format_ogg_vorbis	Ogg container .ogg	Usable
format_pcm	Various Pulse-Coded Modulation formats: .alaw, .al, .alw, .pcm, .ulaw, .ul, .mu, .ulw, .g722, .au	Useful
format_siren14	G.722.1 Annex C (14 kHz) .siren14	New
format_siren7	G.722.1 (7 kHz) .siren7	New
format_sln16	16-bit signed linear .sln16	New
format_sln	8-bit signed linear .sln .raw	Useful
format_vox	.vox	Insignificant
format_wav	.wav	Useful
format_wav_gsm	GSM audio in a WAV container .WAV, .wav49	Usable

2.1.8 Dialplan Functions

Dialplan Functions, 参见 Table 2-8, 用来实现 Dialplan Application (参见本章前面相关部分的介绍)。Dialplan Functions 提供了许多诸如字符串处理, 时间日期操作, ODBC 访问等方面的增强。

Table 2-8. Dialplan functions

Name	Purpose	Popularity/Status
func_aes	Encrypts/decrypts an AES string	Useful
func_audiohookinherit	Allows calls to be recorded after transfer	Useful
func_base64	Encodes/decodes a base-64 string	Usable
func_blacklist	Writes/reads blacklist in <i>astdb</i>	Useful

Name	Purpose	Popularity/Status
func_callcompletion	Gets/sets call completion configuration parameters for the channel	New
func_callerid	Gets/sets CallerID	Useful
func_cdr	Gets/sets CDR variable	Useful
func_channel	Gets/sets channel information	Useful
func_config	Includes AST_CONFIG(); reads variables from config file	Usable
func_connectedline	Changes connected line information on supported handsets	New
func_curl	Uses cURL to obtain data from a URI	Useful
func_cut	Slices and dices strings	Useful
func_db	Provides <i>astdb</i> functions	Useful
func_devstate	Gets state of device	Useful
func_dialgroup	Creates a group for simultaneous dialing	Useful
func_dialplan	Validates that designated target exists in dialplan	Useful
func_enum	Performs ENUM lookup	Useful
func_env	Includes FILE(), STAT(), and ENV(); performs operating system actions	Useful
func_extstate	Returns status of a hinted extension	Useful
func_global	Gets/sets global variables	Useful
func_groupcount	Gets/sets channel count for members of a group	Useful
func_iconv	Converts between character sets	Usable
func_lock	Includes LOCK(), UNLOCK(), and TRYLOCK(); sets a lock that can be used to avoid race conditions in the dialplan	Useful
func_logic	Includes ISNULL(), SET(), EXISTS(), IF(), IFTIME(), and IMPORT(); performs various logical functions	Useful
func_math	Includes MATH(), INC(), and DEC(); performs mathematical functions	Useful
func_md5	Converts supplied string to an MD5 hash	Useful
func_module	Checks to see if supplied module is loaded into memory	Usable
func_odbc	Allows dialplan integration with ODBC resources	Useful
func_pitchshift	Shifts the pitch of an audio stream	Useful
func_rand	Returns a random number within a given range	Useful
func_realtime	Performs lookups within the Asterisk Realtime Architecture (ARA)	Useful
func_redirecting	Provides access to information about where this call was redirected from	Useful
func_sha1	Converts supplied string to an SHA1 hash	Useful
func_shell	Performs Linux shell operations and returns results	Useful
func_speex	Reduces noise and performs dB gain/loss on an audio stream	Useful

Name	Purpose	Popularity/Status
func_sprintf	Performs string format functions similar to C function of same name	Useful
func_srv	Perform SRV lookups in the dialplan	Useful
func_strings	Includes over a dozen string manipulation functions	Useful
func_sysinfo	Gets system information such as RAM, swap, load average, etc.	Useful
func_timeout	Gets/sets timeouts on channel	Useful
func_uri	Converts strings to URI-safe encoding	Useful
func_version	Returns Asterisk version information	Usable
func_vmcount	Returns count of messages in a voicemail folder for a particular user	Useful
func_volume	Sets volume on a channel	Useful

2.1.9 PBX Modules

PBX modules 是一个提供了增强管理和配置机制的外围 modules。例如，**pbx_config** 提供了对传统 Asterisk Dialplan 的加载功能。目前有效的 PBX modules 参见 Table 2-9。

Table 2-9. PBX modules

Name	Purpose	Popularity/Status
pbx_ael	Asterisk Extension Logic (AEL) offers a dialplan scripting language that looks like a modern programming language.	Usable ^a
pbx_config	This is the traditional, and most popular, dialplan language for Asterisk. Without this module, Asterisk cannot read <i>extensions.conf</i> .	Useful
pbx_dundi	Performs data lookups on remote Asterisk systems.	Useful
pbx_loopback	Performs something similar to a dialplan include, but in a deprecated manner.	Insignificant ^b
pbx_lua	Allows creation of a dialplan using the Lua scripting language.	Useful
pbx_realtime	Provides functionality related to the Asterisk Realtime Architecture.	Useful
pbx_spool	Provides outgoing spool support relating to Asterisk call files.	Useful

^a We have not found too many people using AEL. We suspect this is because most developers will tend to use AGI/AMI if they do not want to use traditional dialplans.

^b We've never heard of this being used in production.

2.1.10 Resource Modules

Resource modules 用于将 Asterisk 与外部资源集成。例如，**res_odbc** 可以实现 Asterisk 与 ODBC 数据库连接的交互。当前有效的 resource modules 参见 Table 2-10。

Table 2-10. Resource modules

Name	Purpose	Popularity/Status
res_adi	Provides ADIS	Essential ^a
res_ael_share	Provides shared routines for use with pbx_ael	Essential if you're using AEL
res_agi	Provides Asterisk Gateway Interface	Useful
res_ais	Provides distributed message waiting indication (MWI) and device state notifications via an implementation of the AIS standard, such as OpenAIS	Useful
res_calendar	Enables base integration to calendaring systems	Useful
res_calendar_caldav	Provides CalDAV-specific capabilities	Useful
res_calendar_exchange	Provides MS Exchange capabilities	Useful
res_calendar_icalendar	Provides Apple/Google iCalendar capabilities	Useful
res_clialiases	Creates CLI aliases	Useful
res_clioriginate	Originates a call from the CLI	Usable
res_config_curl	Pulls configuration information using cURL	Useful
res_config_ldap	Pulls configuration information using LDAP	Usable
res_config_odbc	Pulls configuration information using ODBC	Useful
res_config_pgsql	Pulls configuration information using PostgreSQL	Usable
res_config_sqlite	Pulls configuration information using SQLite	Usable
res_convert	Uses the CLI to perform file conversions	Usable
res_crypto	Provides cryptographic capabilities	Useful
res_curl	Provides common subroutines for other cURL modules	Useful
res_fax	Provides common subroutines for other fax modules	Useful
res_fax_spandsp	Plug-in for fax using the spandsp package	Useful
res_http_post	Provides POST upload support for the Asterisk HTTP server	Usable
res_jabber	Provides Jabber/XMPP resources	Useful
res_limit	Enables adjusting of system limits on the Asterisk process	Usable
res_monitor	Provides call recording resources	Useful
res_musiconhold	Provides music on hold (MOH) resources	Essential
res_mutestream	Allows muting/unmuting of audio streams	New
res_odbc	Provides common subroutines for other ODBC modules	Useful
res_phoneprov	Provisions phones from Asterisk HTTP server	New
res_pktccops	Provides PacketCable COPS resources	New
res_realtime	Provides CLI commands for the Asterisk Realtime Architecture (ARA)	Useful
res_rtp_asterisk	Provides RTP	Essential
res_rtp_multicast	Provides multicast-RTP	New

Name	Purpose	Popularity/Status
res_security_log	Enables security logging	New
res_smdi	Provides voicemail notification using the SMDI protocol	Limited
res_snmp	Provides system status information to an SNMP-managed network	Usable
res_speech	Generic speech recognition API	Limited ^b
res_timing_dahdi	Provides timing using the DAHDI kernel interface	Useful
res_timing_kqueue	Provides timing using a kernel feature in some BSDs, including Mac OS X	New
res_timing_pthread	Provides timing using only parts of the standard pthread API; less efficient but more portable than other timing modules.	Useful
res_timing_timerfd	Provides timing using the timerfd API provided by newer versions of the Linux kernel	Useful

^a While most of the ADSI functionality in Asterisk is never used, the voicemail application uses this resource.

^b Requires a separately licensed product in order to be used.

2.1.11 Addon Modules

Addon modules 是开源社区开发的 modules，它们与 Asterisk 主要代码有不同的用途及版权。它们被放置在独立的目录下，并且不会默认被编译及安装。要启用这些 modules，可以使用 *menuselect* 编译配置工具。当前有效的 addon modules 参见 Table 2-11。

Table 2-11. Addon modules

Name	Purpose	Popularity/Status
app_mysql	Executes MySQL queries with a dialplan application	Deprecated—see func_odbc
app_saycountpl	Says Polish counting words	Deprecated—now integrated in <i>say.conf</i>
cdr_mysql	Logs call detail records to a MySQL database	Usable—we recommend <i>cdr_adaptive_odbc</i> instead
chan_mobile	Enables making and receiving phone calls using cell phones over Bluetooth	Limited ^a
chan_ooh323	Enables making and receiving VoIP calls using the H.323 protocol	Usable
format_mp3	Allows Asterisk to play MP3 files	Usable
res_config_mysql	Uses a MySQL database as a real-time configuration backend	Useful

^a While *chan_mobile* works great with many phones, problems have been reported with some models. When a problem does occur, it is very difficult for developers to solve unless they have a phone of the same model to test with.

2.1.12 Test Modules

Test modules 是 Asterisk 开发组用于验证代码的 modules。它们总是在不断的修改和增加中，并且对于一般用户而言这组 modules 并没有用，除非你打算参与开发 Asterisk 的开发。

如果你是一名 Asterisk 软件开发者，那么，你会对 Asterisk 测试套件非常感兴趣。你可以用 test modules 构建自动测试工具，并且它可以运行在不同的操作系统和不同的机器类型上。正是通过对测试工作的持续投入，Asterisk 项目才得以避免代码的退步。当你向项目组提交你的测试结果时，也也会对未来的升级更有信心。

更多的关于安装 Asterisk 测试套件的信息可以参考这个博客：

<http://blogs.asterisk.org/2010/04/29/installing-the-asterisk-test-suite/>。更多的关于创建测试的信息请参见文档 <http://svn.asterisk.org/svn/testsuite/asterisk/trunk/README.txt>，或者你可以加入 Freenode IRC network 的 #asterisk-testing 频道。

2.2 文件结构

Asterisk 是一个由许多资源构成的复杂系统。这些资源以不同的方法来操作文件系统。由于 Linux 在这一点上是如此灵活，因此理解什么数据被存储了是非常有用的。以便你可以了解特定的数据存储在哪里（例如语音邮件消息或日志文件）。

2.2.1 Configuration files

Asterisk 的配置文件包括 extensions.conf, sip.conf, modules.conf，以及其它许多为不同的信道、资源和可能用到的功能定义参数的文件。

这些文件可以在 /etc/asterisk 目录下找到。当你配置和管理你的 Asterisk 系统时，你将经常与这个目录打交道。

2.2.2 Modules

Asterisk modules 一般被安装在 /usr/lib/asterisk/modules 目录下。你通常并不会访问这个文件夹，当然，了解这个文件夹有助于我们了解这些 modules 都安装在哪里。举例来说，如果你升级了 Asterisk 并且在做 *menuselect* 时选择了不同的 modules，那么从前一个版本继承下来的旧的（不兼容的）modules 需要被删除掉，否则你会在安装新版本时看到一个警告信息。删除这些旧的文件就需要从 modules 文件夹下删除。你可以选择手工删除，或者通过“uninstall”删除（*make uninstall*）。

2.2.3 The Resource Library

有些资源（resource）需要外部的数据源。例如，Music on hold（MOH）必须事先准备好需要播放的音乐文件。系统提示（System prompts）同样需要在硬件驱动中存储一些东西。目录 /var/lib/asterisk 用于 system prompts, AGI scripts, music on hold，以及其它一些资源文件的存放。

2.2.4 The Spool

Spool 是 Linux 用于存储需要频繁修改的, 或者稍后需要其它处理器处理的文件的地方。例如, 在 Linux 系统下, 打印作业或待处理的邮件 (emails) 一般是存储在 Spool 中, 直到它们被处理。

对 Asterisk 来说, spool 被用来存储临时项目, 诸如语音留言 (voice messages), 呼叫记录 (call recording, 注意不是 CRDs, 而是指由 **MixMonitor()** 相关应用产生呼叫的语音记录), 呼叫文件 (call file) 等。

Asterisk spool 位于 `/var/spool/asterisk` 目录下。

2.2.5 Logging

Asterisk 能够生成几种不同类型的日志文件 (log files)。这些日志都存储在 `/var/log/asterisk` 目录下, 例如呼叫详单 (CDRs), CEL, 调试日志 (debug logs), 队列日志 (queue logs), 消息 (messages), 错误 (errors), 以及一些其它输出记录。

这个目录对于任何故障排除工作都绝对非常重要。我们将在第 24 章进一步讨论如何利用 Asterisk 的日志文件。

2.3 The Dialplan

Dialplan 是 Asterisk 的核心。所有到达的 channel 都需要经过 dialplan 的处理, 在 dialplan 中, 包含有定义了如何处理一个来电的呼叫流程脚本 (call-flow script)。

Dialplan 可以有以下三种编辑方法:

- 采用传统的 Asterisk Dialplan 语法编辑 `/etc/asterisk/extensions.conf`
- 采用 Asterisk Extension Logic (AEL) 编辑 `/etc/asterisk/extensions.ael`
- 采用 LUA 编辑 `/etc/asterisk/extensions.lua`

在本书后续章节, 会花费几章来讨论如何用传统的 Asterisk Dialplan 语法来编辑 dialplan (这也是目前最普及的做法)。一旦你学会了种方法, 如果你想改用 AEL 或 LUA, 将是非常容易的。

2.4 硬件

Asterisk 有能力与许多不同的技术通信。一般来说, 这些通信是通过网络对接的。然而, 如果希望与传统通信技术对接, 如与 PSTN 对接, 则需要专门的硬件。

流行的 Asterisk 板卡一般被设计为支持 DAHDI 接口, 即 Digium Asterisk Hardware Device Interface。不同的板卡一般有不同的安装要求及不同的安装目录。

我们将在第七章进一步讨论 DAHDI 接口。然而, 本书仅限于讨论 DAHDI 本身。对于你所安装板卡的细节信息, 你还是需要参考厂商提供的文档。

2.5 Asterisk 版本

Asterisk 的版本发布机制在过去几年中出现过一些反复。因此，我们编写本章以帮助你了解 Asterisk 的版本号到底是什么意思。特别要进行讨论的是 1.6.x 系列的版本号，与其它任何 Asterisk 版本（1.0 到 1.8 直到今后的版本）比较，1.6.x 系列版本遵循了完全不同的编号逻辑。

2.5.1 以前的版本号机制

当我们只有 Asterisk 的 1.2 和 1.4 版本时，所有的新特性开发都是在主干版本上进行的（今天也依然是这样），只有 BUG 修复会在 1.2 及 1.4 的分支版本上进行。Asterisk 1.2 版本现在已经停止维护了（EOL, End of Life），Asterisk 开发组不会再对它进行 BUG 修复及安全升级工作。除了 1.6.x 分支的版本，Asterisk 开发组只对主干版本和 1.4 分支版本进行 BUG 修复。（译者注：1.4 版本也已经在 2011 年 EOL 了）

在 1.6 分支版本出现之前，由于所有的新特性开发只在主干版本上进行，用户是无法访问这些新特性和新功能的。这并不是说这些新特性和新功能不可用，而是由于它们都只发布在主干版本上。而要安装及运行一个基于主干版本的 Asterisk，需要系统管理员对 Asterisk 和 C 语言都有非常深入的理解。

为了缓解给系统管理员带来的压力，同时又能让人们更快的应用这些新特性（从时间上说，是每隔几个月更新，而不是几年），Asterisk 开发组在 1.6 分支版本上采用了一种新的机制。1.6 分支版本实际上被标识为 1.6.0, 1.6.1, 1.6.2, 等等，每当新的特性被增加进来时，第三个数字就会增加。这样做的目的是每 3 到 4 个月就可以提供一个增加了新特性的版本（这个版本是从主干版本分出来的）。如果你需要新的版本，你只需要等待几个月并升级到下一个分支版本就可以了。

在 1.6.x 版本机制下的版本标签看起来是这样的：

- 1.6.0.1 ~ 1.6.0.2 ~ 1.6.0.3 ~ 1.6.0.4 ~ etc
- 1.6.1.1 ~ 1.6.1.2 ~ 1.6.1.3 ~ 1.6.1.4 ~ etc
- 1.6.2.1 ~ 1.6.2.2 ~ 1.6.2.3 ~ 1.6.2.4 ~ etc

Figure 2-2 更清楚的表示除了这种版本关系。

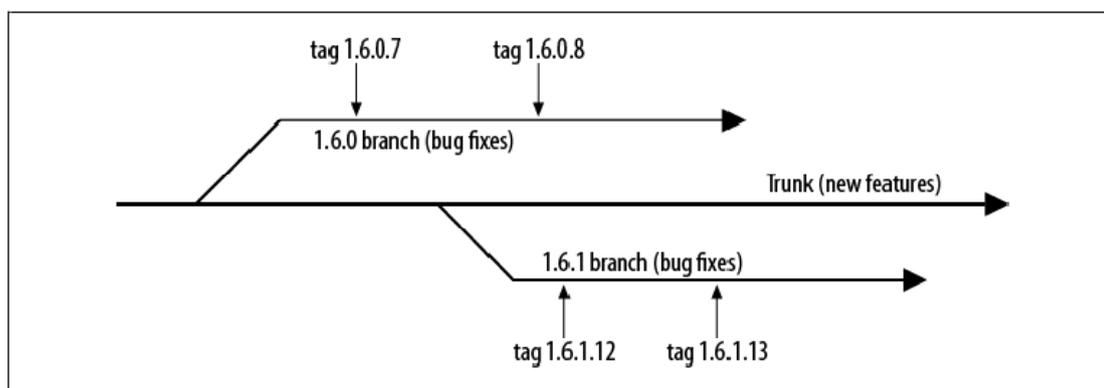


Figure 2-2. The Asterisk 1.6.x release process

所以，迄今为止，Asterisk 的分支版本包括 1.2, 1.4, 1.6.0, 1.6.1, 和 1.6.2（没有 1.6 分支）。在每一个分支上，我们创建标签 tags(releases)就得到了发行版本，它们看起来像 1.2.14, 1.4.30, 1.6.0.12, 1.6.2.15。（译者注：原文如此，到 2011 年 11 月译者翻译时，显然

1.8 版本早已发布，并且 Digium 再次更改了版本编号规则，发布了 10 版本)。

不幸的是，Asterisk 开发组在 1.6.x 上的这种努力并没有实现每 3 到 4 个月创建一个新的分支版本的目标：开发流程导致最少也要 6 到 8 个月才能实现一个分支版本。不仅如此，1.6.x 的编号方式本身也带来了麻烦。人们被 1.6.0, 1.6.1, 1.6.2 是完全不同的分支并独立升级的作法搞糊涂了，不知道该选择哪个版本。当你从 1.2 到 1.4，再到 1.8 时，你可以看到很明显的分支版本变化，但是 1.6.0, 1.6.1, 1.6.2 则看起来没那么明显。

2.5.2 新的版本机制

Asterisk 开发组在 1.6.x 的版本管理上学习到了很多经验。1.6.x 版本发布机制的想法是很好的，但在实践上并不成功。所以，Asterisk 1.8 的版本管理机制再次做了调整，新的方法看起来更像 1.2 和 1.4 的版本管理。

Asterisk 开发组仍旧希望以更合适的周期（目标是 12 个月）提供新的版本，同时他们也认识到为一个稳定、流行的版本提供长时间的支持是个很好的方法。Asterisk 1.4 分支版本就可以认为是这样一个长时间支持（LTS, long-term support）版本。而 1.6.0, 1.6.1 和 1.6.2 分支版本则可以看作一个特性版本，虽然在发布后也会对它们进行 BUG 维护，但只会支持相对较短的一段时间（一年左右）。目前最新的 LTS 版本是 1.8，本书也是依据这个版本写作的。Digium 的 Asterisk 开发组会为 Asterisk 1.8 提供 5 年的支持，包括 4 年的 BUG 维护以及再附加的 1 年安全维护。

在 Asterisk 1.8 的 LTS 周期中，也会半正式发布一些特性分支版本。这些版本会被标记为 1.10, 1.12, 和 1.14。这些版本在进入 EOL 状态前，也都会得到一年的 BUG 维护及再附加的一年安全维护。

所有 Asterisk 分支版本的当前状态，如它们何时进入仅提供安全维护的状态，以及什么时候结束产品的生命周期（进入 EOL）等信息，都可以在如下 wiki 中找到：

<https://wiki.asterisk.org/wiki/display/AST/Asterisk+Versions>

2.6 结论

Asterisk 是由许多不同的技术组成的，而且大部分这些技术都很复杂。因此，理解 Asterisk 的体系结构就变得非常重要。实际上，Asterisk 不仅设计良好，而且按照我们的观点，也非常好的平衡了灵活性和复杂性。