Unusual Uses: What you didn't know your Asterisk system could do!

...or, how I learned to love the 1.6 branch
About Me!

• Co-author of Asterisk: The Future of Telephony with Jim van Meggelen and Jared Smith  
  (http://astbook.asteriskdocs.org)
• Asterisk bug tracker marshal and release manager
• Consultant with more than 5 years experience specializing in database integration and clustering

http://www.leifmadsen.com
Covered in this presentation

• Cool new(ish) features in (or almost in) Asterisk 1.6:
  – IMAP voicemail integration (with greeting storage)
  – New ODBC features (adaptive_cdr_odbc)
  – Calendar Integration
  – CURL
  – XMPP (Jabber) Integration
  – (This presentation based on CentOS 5.x)
IMAP Integration

• Allows you to store your voicemails and emails in the same location
• Repurpose existing IMAP (MS Exchange) infrastructure
• Get to start touting a new (old?) buzzword; Converged!
Adaptive CDR ODBC

- Allows you to store additional call information to the database simply by adding a new column to the database (and writing to it from the dialplan)
- Will automatically create additional columns that the system needs (if the database allows for it)
Calendar Integration

- Allows you to hook your Asterisk system to things like Google Calendar, Exchange, or Zimbra to get status from a calendar
- Perform routing logic based on your calendars status
- Redirect calls to voicemail automatically when you're listed as in a meeting
CURL

• Existed in Asterisk 1.4, but not widely used
• Allows you to get information from a web page and use that information in your dialplan
• Has been used for things like looking up route costs that can be easily managed outside of Asterisk
XMPP Integration

• Can use the XMPP protocol (used by Jabber) to get information to and from Asterisk

• Send a text message from the dialplan to someone
  – Use as a simple way of getting a pop-up on your machine for incoming calls
IMAP
IMAP Integration

• IMAP first appeared in Asterisk 1.4
• Allows us to store voicemail in the same location as our email; Unified Communications! </buzz_word>
• In the 1.6.x branches, we now have the ability to also store greetings in IMAP, and not just on the local file system
Building IMAP Integration

• Need `OpenSSL-devel` and `pam-devel` packages
• On CentOS
  – 64-bit
    • `yum install openssl-devel.x86_64 pam-devel.x86_64`
  – 32-bit
    • `yum install openssl-devel.i386 pam-devel.i386`
Building IMAP Integration

• We also need to build the \textit{c-client} libraries from University of Washington
  
  \begin{itemize}
    \item \texttt{wget ftp://ftp.cac.washington.edu/mail/imap.tar.Z}
    \item Extract it and run:
      \begin{itemize}
        \item 64-bit
          \begin{itemize}
            \item \texttt{make lr5 EXTRACTFLAGS=-fPIC IP6=4}
          \end{itemize}
        \item 32-bit
          \begin{itemize}
            \item \texttt{make lr5 IP6=4}
          \end{itemize}
      \end{itemize}
  \end{itemize}
Install Dovecot

• Then we need to install the IMAP server; Dovecot
• On CentOS
  – 64-bit
    • yum install dovecot.x86_64
  – 32-bit
    • yum install dovecot.i386
Configure Dovecot

- useradd phonesys
- passwd phonesys
- mkdir /var/mail/asterisk
- mkdir /var/mail/asterisk/phonesys
- chown phonesys:phonesys /var/mail/asterisk/phonesys
Configure Dovecot

- `vim /etc/dovecot.conf`

  ```
  mail_location = maildir:/var/mail/asterisk/phonesys/%u
  protocol imap {
  }
  auth default {
    mechanisms = plain
    passdb pam {
    }
    passdb passwd-file {
      args = /etc/dovecot.masterusers
      master = yes
    }
  userdb static {
    args = uid=500 gid=500
  }
  }
  ```
Configure Dovecot

• Need to allow Asterisk to authenticate for other users
  • touch /etc/dovecot.masterusers
  • Then add to the file
    phonesys: {PLAIN}phonesys
  • Then you can restart the Dovecot service
    service dovecot restart
Configure Asterisk with IMAP Support

• Next we get to compile Asterisk with IMAP support
  
  ./configure –with-
imap=/usr/src/libraries/imap/imap-2007e

• Then select the IMAP_STORAGE option from Voicemail Build Options in menuselect

• Now we can reinstall Asterisk after building
  
  make install
Configure voicemail.conf

Next we need to configure our voicemail.conf file to tell Asterisk to connect to the IMAP server

imapserver=localhost
imapflags=notls
imapgreetings=yes    ; <-- new!
authuser=phonesys
authpassword=phonesys
expungeonhangup=yes
Configure Voicemail Users

• And then in `voicemail.conf`, we can configure which mailbox our voicemails should be stored in.

• We can also use the `imapsecret` option if we needed to authenticate with the server as our peer (not necessary in our case).

```
[imapvoicemail]
100 => 1234,Sue's Mailbox,,,imapuser=sue@example.tld
101 => 5555,Bob's Mailbox,,,imapuser=bob@example.tld
```
Sorry, nothing fancy here :)  

• Once you have everything setup and running, your Voicemail() and VoicemailMain() applications just work the same as before!  
• (I promise some dialplan and such coming up!)
Adaptive ODBC
Adaptive ODBC

• Allows Asterisk to 'adapt' to table layouts
• Can add columns it expects and needs
• Lets you create new columns, and access them from the dialplan (such as adding a custom value to your CDRs)
• Minimizes the amount of work required to get tables setup for the Asterisk Realtime Architecture (ARA)
Building Adaptive Capabilities

- Need the `unixODBC-devel` and `libtool-ltdl-devel` packages
- On CentOS run
  - 64-bit
    - `yum install unixODBC-devel.x86_64 libtool-ltdl-devel.x86_64`
  - 32-bit
    - `yum install unixODBC-devel.i386 libtool-ltdl-devel.i386`
MySQL ODBC

• If you want to use MySQL with ODBC, then you will need to also install the `mysql-connector-odbc` package
  – `yum install mysql-connector-odbc`

• If you want to use `res_mysql`, then you need to install `asterisk-addons` and the `mysql-devel` development headers
  – `yum install mysql-devel`
CDR Adaptive ODBC

- The start of the adaptive realtime engine
- Allows you to omit data you don't want to log by not including the column in your table
- Create aliases for column names in cdr_adaptive_odbcc.conf
- Now you can adapt Asterisk to your own table layouts!
Going Beyond CDRs

• With the advantages the adaptive engine provided to CDRs, it was taken a step further
• With the ARA (realtime), it would fail previously if you were missing a column
• Now Asterisk will warn you about the missing column, and adapt the SELECT, UPDATE, and INSERT queries to the current table layout
No More Broken Realtime!

• If your table layout wasn't exactly what Asterisk expected, it just wouldn't work
• If the developers wanted to add a new column for a new feature and you updated, that new column would cause your existing realtime install to stop working
Doing The Work For You

• If you use the `res_config_pgsql` or `res_config_mysql` modules, Asterisk can even create the missing columns for you

• `res_pgsql.conf` (stock) and `res_mysql.conf` (addons) gives you the `requirements` option
Doing The Work For You

• warn: provide a warning about missing columns, types, or lengths
• createchar: create column as a CHAR with appropriate length
• createclose: create column as appropriate type and length
• On occasion may even widen a column for you (if necessary)
Configuring res_mysql.conf

- This is the file where we define our connection to the database

```ini
[asterisk]
dbhost = 127.0.0.1
dbname = asterisk
dbuser = asterisk
dbpass = asterisk
dbport = 3306
dbsock = /tmp/mysql.sock
requirements=warn ; or createclose or createchar
```
Configuring Realtime

• Then in `extconfig.conf` we can configure our SIP registrations (and other realtime things) to store and read data from our MySQL connection

```
[settings]
;iaxusers => odbc,asterisk
;iaxpeers => odbc,asterisk
;sipusers => odbc,asterisk
;sippeers => odbc,asterisk
;sipregs => mysql,asterisk
;voicemail => odbc,asterisk
;extensions => odbc,asterisk
;meetme => mysql,conferences
;queues => odbc,asterisk
;queue_members => odbc,asterisk
;musiconhold => mysql,asterisk
;queue_log => mysql,asterisk
```
• Once we've configured res_mysql.conf then we get warned that we're missing the table to store our SIP registrations into

• Now the administrator knows what is missing (this is new!)
And Then There Were Columns!

- Enable `createclose` in `res_mysql.conf`, create your table, and start Asterisk

```sql
mysql> CREATE TABLE sipregs (  
  -> id int NOT NULL AUTO_INCREMENT,  
  -> PRIMARY KEY (id)  
  -> );
Query OK, 0 rows affected (0.00 sec)
```

- Before...

```sql
mysql> describe sipregs;
+-------+-------+------|---|--------+-------------------+
<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Key</th>
<th>Default</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>int(11)</td>
<td>NO</td>
<td>PRI</td>
<td>NULL</td>
<td>auto_increment</td>
</tr>
</tbody>
</table>
+-------+-------+------|-----|---------+-------------------+
1 row in set (0.00 sec)
```
...And After!

- Columns automatically created for us!
Calendar Integration
Calendar Integration

• Currently in a branch and can be tracked at

• Works with MS Exchange, Zimbra, and Google Calendar

• Currently 'Ready for Testing'

• My examples will be with Google Calendar
Calendar Integration

• We can perform call routing decisions based on calendar status; for example, send calls to Voicemail() when you're busy

• Automatically call participants for a conference when you schedule it

• Usage of functions may still change prior to release; additional functions may be necessary
Building Calendar Integration

• Depends on *libical-devel* package from EPEL

• EPEL installation RPM available at [http://fedoraproject.org/wiki/EPEL](http://fedoraproject.org/wiki/EPEL)

• On CentOS with EPEL repo installed:
  – 64-bit
    • `yum install libical-devel.x86_64`
  – 32-bit
    • `yum install libical-devel.i386`
Configuring for Google Calendar

• Default configuration file `calendar.conf` contains examples for MS Exchange, Zimbra, and Google Calendar

```
[asterisk-gcal]
type = caldav ; type of calendar--currently supported: ical, caldav, or exchange
; Main GMail calendar (the trailing slash is significant!)
url = https://www.google.com/calendar/dav/leif@leifmadsen.com/events/
user = leif@leifmadsen.com ; username
secret = welcome ; password
refresh = 60 ; refresh calendar every n seconds
timeframe = 60 ; number of minutes of calendar data to pull for each refresh period
; should always be >= refresh / 60
```
Configuring for Google Calendar

• We can show the events for the calendar modules after we reload it.
Routing Calls When Busy

- We can create a simple dialplan that will first check our status to determine if we're busy, and if so, to route calls to Voicemail() instead of ringing our devices.

```
-- Executing [100@phones:1] Verbose("SIP/1madsen-lmentinc-b409f150", "2, Checking if extension 100 is free") in new stack
== Checking if extension 100 is free
-- Executing [100@phones:2] Set("SIP/1madsen-lmentinc-b409f150", "myCalendarStatus=1") in new stack
-- Executing [100@phones:3] GotoIf("SIP/1madsen-lmentinc-b409f150", "1?voicemail") in new stack
-- Goto (phones,100,6)
-- Executing [100@phones:6] VoiceMail("SIP/1madsen-lmentinc-b409f150", "100@lmentinc.b") in new stack
-- <SIP/1madsen-lmentinc-b409f150> Playing '/var/spool/asterisk/voicemail/lmentinc/100/busy.slin' (language 'en')
```
Routing Calls When Busy

- Use the CALENDAR_BUSY() function to get a '1' or '0' when busy, or not busy
- Go right to Voicemail() with busy status if we're not available currently
Automatically Call Meeting Participants

- With some clever tricks, we can automatically call people we want to participate in our conference call – and connect them to the conference room!
- We configure `calendar.conf` to call a Local channel, then use the Originate() dialplan function.
Automatically Call Meeting Participants

- Remember this part in the Description field?  
  Description: x100\,p6474483075\,dSIP/leif@leifmadsen.com

- First character tells us what we're calling:
  - x: local extension
  - d: local device
  - p: phone number
Configure Auto Dial

• In the `calendar.conf` file we can configure it to connect to the dialplan when it encounters a new busy status
• From there we can get information from the calendar, such as data in the description and location fields
• We use `CALENDER_EVENT()` for this
exten => tryCall,1,Verbose(2, Calendar is looking to call someone)
exten => tryCall,n,Set(DESCRIPTION=${CALENDAR_EVENT(description)})
exten => tryCall,n,Set(CONFERENCE=${CALENDAR_EVENT(location)})
exten => tryCall,n,Set(AUTOCALL=${CUT(DESCRIPTION,-,1)})
exten => tryCall,n,GotoIf(&&!${ISNULL(DESCRIPTION)}?exit,1)
exten => tryCall,n,Set(AUTOCALL=${CUT(DESCRIPTION,-,1)})
exten => tryCall,n,GotoIf(!"${AUTOCALL}" = "AUTOCALL"?autocall,1:exit,1)

exten => autocall,1,Verbose(2, Attempting to call people in description)
exten => autocall,n,Set(OFFSET=2)
exten => autocall,n,Set(WHO=${CUT(DESCRIPTION,-,OFFSET)})

exten => autocall,n,While($[WHO] "=" ""
  autocall,n,Set(METHOD=${WHO:0:1})
  autocall,n,GotoIf($[METHOD] = "x"?extension)
  autocall,n,GotoIf($[METHOD] = "p"?phone)
  autocall,n,GotoIf($[METHOD] = "d"?device)
  autocall,n,Goto(offset)
  autocall,n(extension),NoOp()
  autocall,n,Set(EXTENSION=${WHO:1})
  autocall,n,Origininate(${DB(phones/${EXTENSION}/tech})/${DB(phones/${EXTENSION}/username}),app,MeetMe,${CONFERENCE}d)
  autocall,n,Verbose(2, Fall-through)
  autocall,n,Goto(offset)
  autocall,n(phone),NoOp()
  autocall,n,Set(PHONE=${WHO:1})
  autocall,n,Origininate(${GLOBAL(G_PRIM_ITSP)}/${PHONE},app,MeetMe,${CONFERENCE}d)
  autocall,n,Goto(offset)

  autocall,n(device),NoOp()
  autocall,n,Set(DEVICE=${WHO:1})
  autocall,n,Origininate(${DEVICE},app,MeetMe,${CONFERENCE}d)
  autocall,n,Goto(offset)

  autocall,n(offset),Set(OFFSET=${OFFSET}+1)
  autocall,n,Set(WHO=${CUT(DESCRIPTION,-,OFFSET)})
  autocall,n,EndWhile()
  autocall,n,Goto(exit,1)

exit => exit,1,Verbose(2, Done with this calendar event)
exit => exit,n,Hangup()
Some Kinks…

- Unfortunately the previous dialplan doesn't currently work
- The Originate() application should fall through, but doesn't seem to when used inside a Local channel
- Currently working with a developer to resolve this somehow... such is the life of a tester!
CURL
Using CURL for call rate tracking

- Lookup rate for international / national calling and track cost for each call
- Uses a simple webpage lookup to get the rate for the call
- Allows you to simply update the rate table on the website side, and not have to change anything in Asterisk
- Could be expanded to become a Lease Cost Routing engine
CURL

• I created a PHP script (with some Internet help) to parse and search a CSV file (http://www.leifmadsen.com/presentations/IT360/20080408/curl-example.php)

• Asterisk then passes the number being dialed to the website

• The CURL() function then retrieves the data and places it into a variable in the dialplan
Building CURL

• To build the 'res_config_curl', 'res_curl', and 'func_curl' functions, you need to install the CURL development libraries for your system

• On CentOS/RHEL:
  – 64-bit
    • `yum install curl-devel.x86_64`
  – 32-bit
    • `yum install curl-devel.i386`
Format of CURL()

core show function CURL

|-- Info about function 'CURL' |--

[Synopsis]
Retrieves the contents of a URL

[Description]
    url           - URL to retrieve
    post-data    - Optional data to send as a POST
                   (GET is default action)

[Syntax]
CURL(url[,post-data])
Setting options for CURL()

Syntax: CURLOPT(<option>)

- **cookie** - Send cookie with request
- **conntimeout** - Number of seconds to wait for connection
- **dnstimeout** - Number of seconds to wait for DNS response
- **ftptext** - For FTP, force a text transfer (boolean)
- **ftptimeout** - For FTP, the server response timeout
- **header** - Retrieve header information (boolean)
- **httptimeout** - Number of seconds to wait for HTTP response
- **maxredirs** - Maximum number of redirects to follow
- **proxy** - Hostname or IP to use as a proxy
- **proxytype** - http, socks4, or socks5
- **proxyport** - Port number of the proxy
- **proxyuserpwd** - A <user>:<pass> to use for authentication
- **referer** - Referer URL to use for the request
- **useragent** - UserAgent string to use
- **userpwd** - A <user>:<pass> to use for authentication
- **hashcompat** - Result data will be compatible for use with HASH()
Website Output

• URL:

• Result:
  – CANADA-647,647,0.011,0.88807702064514,2.1482679843903
Dialplan

- The above is the “trick” that we're using to get the data from the website, and then writing the values into separate variables.
Dialplan

```
exten => _NXXNXXXX,1,Verbose(2,CURL Test)
exten => _NXXNXXXX,n,Set(toDial=${EXTEN})
exten => _NXXNXXXX,n,Set(RES=${CURL(http://192.168.128.50/index.php?number=${toDial})})
exten => _NXXNXXXX,n,GotoIf(${RES} "No rate found."?no_rate,1)
exten => _NXXNXXXX,n,Set(ARRAY(country,location,rate,haystack_time,search_time)=$RES)
exten => _NXXNXXXX,n,GotoIf(${rate} ""?no_rate,1)
exten => _NXXNXXXX,n,Dial(${GLOBAL(G_PRIM_ITSP)}/${toDial},30)
exten => _NXXNXXXX,n,Hangup()
exten => _NXXNXXXX,n,Playback(No rate found)
exten => _NXXNXXXX,n,Hangup()
exten => _NXXNXXXX,n,Hangup()
exten => `h,1,Verbose(2,Call cleanup)
exten => h,n,Set(BILLYSEC=${CDR(billsec)})
exten => h,n,Set(MINUTES=[${BILLSEC} / 60])
exten => h,n,ExecIf(${rate} ""?Set(CALL_COST=${MINUTES} * ${rate}))
exten => h,n,Verbose(2,Call cleanup)
```
XMPP (Jabber)
XMPP (Jabber) Integration

• Currently have JabberSend() app; first appeared in Asterisk 1.4
• Not widely used; perhaps no one knows about it?
• Branch currently being worked on to give us JabberReceive() (Ready for Testing!)
Building XMPP

- Need to install some dependencies
- On CentOS, need to install EPEL repository
- Depends on `iksemel-devel` and can use `openssl-devel` (for secure connections)
- EPEL installation RPM available at http://fedoraproject.org/wiki/EPEL
Building XMPP

• On CentOS with EPEL repository installed:
  – 64-bit
    • `yum install iksemel-devel.x86_64 openssl-devel.x86_64`
  – 32-bit
    • `yum install iksemel-devel.i386 openssl-devel.i386`
Configuring jabber.conf

- It's pretty easy!
- Use your Google talk login, or you can use your company email if using Google apps

``` ini
[general]
debug=no
autoregister=yes

[asterisk]
type=client
serverhost=talk.google.com
username=asterisk@leifmadsen.com
secret=welcome
priority=1
port=5222
usetls=yes
usesasl=yes
status=available
statusmessage="I am available"
```
Using JabberSend()

• We can create a simple incoming caller pop-up
• Whenever someone calls my extension, an XMPP message pops up to tell me who is calling
• Useful in dark situations because my Polycom IP501 doesn't have a back light
JabberSend() Pop-Up

-- Executing [100@lmentinc_in:1] **Verbose**("SIP/4164790259-ae8f9ad0", "2,"LEIF MADSEN" <6474483075> is requesting to speak to extension 100") in new stack
== "LEIF MADSEN" <6474483075> is requesting to speak to extension 100
-- Executing [100@lmentinc_in:2] **JabberSend**("SIP/4164790259-ae8f9ad0", "asterisk,leif.madsen@gmail.com,Incoming caller from "LEIF MADSEN" <6474483075>") in new stack
-- Executing [100@lmentinc_in:3] **Dial**("SIP/4164790259-ae8f9ad0", "SIP/lmadsen-lmentinc,30,0") in new stack
-- Called lmadsen-lmentinc
-- No one is available to answer at this time (1:8/0/0)
-- Executing [100@lmentinc_in:4] **Playback**("SIP/4164790259-ae8f9ad0", "silence/1") in new stack
-- <SIP/4164790259-ae8f9ad0> Playing 'silence/1.ulaw' (language 'en')
-- Executing [100@lmentinc_in:5] **Voicemail**("SIP/4164790259-ae8f9ad0", "100@lmentinc,u") in new stack
-- <SIP/4164790259-ae8f9ad0> Playing '/var/spool/asterisk/voicemail/lmentinc/100/unavail.slin' (language 'en')
== Spawn extension (lmentinc_in, 100, 5) exited non-zero on 'SIP/4164790259-ae8f9ad0'

* Asterisk PBX

From: leif.madsen@gmail.com
7:05

Incoming caller from "LEIF MADSEN" <6474483075>
JABBER_RECEIVE()

- Currently in a branch and ready for testing
- Will go into a future 1.6.x branch (most likely 1.6.3, or potentially 1.6.4)
- Bug tracker location
- Allows us to receive text from a client and act on it in the dialplan
Call Control via Jabber

• With the JABBER_RECEIVE() function, we can control call flow by sending Asterisk messages

• My example will use JABBER_RECEIVE() and Local channels to control call rejection and forwarding
Call Control via Jabber

• When a call rings my extension, it rings my desk phone, while sending me a message with options

Incoming call from “LEIF MADSEN” <6474483075>
Press 1 to send call to voicemail
Press 2 to send call to cell
Call Control via Jabber

- Call comes into the server and dials extension 100
- Hits the Dial() application and simultaneously calls two contexts via the Local channel
Call Control via Jabber

- While calling desk phone, we send options to the Jabber client
- We receive option '2' back and set to the RES channel variable
Call Control via Jabber

- Since option '2' is send call to cell, we do a Goto() and call out the provider to a cell phone
Making it all work

• Caller dials extension 100, which calls two local extensions via Local channels
• If we come back with no Answer() after 30 seconds, we fall over to Voicemail()
Making it all work

```
[dial-phone]
exten => start,1,Dial(SIP/leif.madsen@imentinc.com,0)

[receive-jabber]
exten => start,1,Verbose(2,Trying to get data back from Jabber)
exten => start,n,JabberSend(asterisk,leif.madsen@gmail.com,Incoming caller from ${CALLERID(all)})
exten => start,n,JabberSend(asterisk,leif.madsen@gmail.com,Press 1 to send to Voicemail)
exten => start,n,JabberSend(asterisk,leif.madsen@gmail.com,Press 2 to send to Cell)
exten => start,n,Set(RES=${JABBER_RECEIVE(asterisk,leif.madsen@gmail.com,20)})
exten => start,n,ExecIf(${ISNULL($RES))}?Hangup():NoOp())
exten => start,n,Verbose(2,Answering call because we got data back)
exten => start,n,Answer()
exten => start,n,Gotolf("$RES" = "1"?voicemail,1)
exten => start,n,Gotolf("$RES" = "2"?cell,1)
exten => start,n,Goto(voicemail,1)

exten => voicemail,1,NoOp()
exten => voicemail,n,Playback(silence/1)
exten => voicemail,n,Voicemail(100@imentinc.com)
exten => voicemail,n,Hangup()

exten => cell,1,NoOp()
exten => cell,n,Dial(SIP/4164790259/6474483075)
exten => cell,n,Hangup()
```
Webliography

• Read CSV file into multidimensional array
  – http://www.defproc.co.uk/php/magic_csv_load_a_CSV_file_into_an_associative_array

• Search multidimensional array
Webliography

• International wholesale rates from Unlimitel.ca
  – http://www.unlimitel.ca/temp/support/voip_support/international_call_codes.php
Contact Information
Leif Madsen
http://www.leifmadsen.com

twitter: leif_madsen
email: leif@leifmadsen.com