

Asterisk Manager and GUI Interfacing in Large Environments

Presented by Matt Florell

Chief Architect of the astGUIClient project



Manager API and GUI Topics

- ◆ Why use the manager API?
- ◆ Manager API capacity
- ◆ Multi-server Environments
- ◆ Central Queue vs. Individual Connections
- ◆ How a Central Queue Operates
- ◆ Why use MySQL for the Central Queue?
- ◆ Real-World Example of Queue in Action
- ◆ astGUIclient Design and Functionality
- ◆ Capacity Limits and Trade-offs
- ◆ The Future

Why Use the Manager API?

- ◆ Allows for actions that are either hard or impossible to program in a dialplan or AGI
 - ◆ Start/stop recording of existing call
 - ◆ Place calls from within meetme conferences
 - ◆ Originate call to an AGI and an outside number
 - ◆ Redirect existing calls to different channels
- ◆ Allows for Complex Client Applications
 - ◆ Windows and X-based GUI applications
 - ◆ Receptionist consoles
 - ◆ ACD/inbound CRM applications and popups
 - ◆ Outbound/predictive/auto-dialing call center applications
 - ◆ Web-based applications
 - ◆ Initiating conferences, remote worker telecommuting
 - ◆ Real-time usage reports

Manager API Capacity

- ◆ Greatly dependent upon Asterisk and server load
- ◆ At high loads data flow can “pause” on the manager interface for up to several seconds
- ◆ More manager connections means more chance for pauses no matter the load
- ◆ Customizing manager.conf for each connection's needs helps reduce unneeded data transfer and reduces strain on system

Multi-Server Environments

- ◆ Large installations that have high-capacity phone usage may need multiple Asterisk servers to run optimally, adding complexity to Manager API usage
- ◆ A high-end server running Asterisk can handle 50 concurrent SIP to Telco conversations reliably
- ◆ Options for server to server connections
 - ◆ IAX – Native Asterisk connection
 - ◆ PRI crossover – T1 to T1, no transcoding
 - ◆ SIP – harder to setup, wide VOIP standard

Central Queue vs. Individual Connections

- ◆ **Central Queue:**
 - ◆ Adds delay in execution (< 0.5 sec)
 - ◆ Creates a single point of failure
 - ◆ Easily handles interfacing with multiple Asterisk servers
 - ◆ Easier/more organized way of keeping track of calls
 - ◆ Only needs one local connection for output of all activity
 - ◆ Can use new connection for each initiation of a new action preventing action backlogs
- ◆ **Individual Connections:**
 - ◆ Can load server more
 - ◆ Not as fault tolerant
 - ◆ Must follow all call progresses for each connection meaning more work for client app
 - ◆ More prone to lock up or freeze

How a Central Queue Operates Server-side Operations

- ◆ Elements running on the Asterisk server
 - ◆ Channel state updater – This does nothing but ask for the list of live channels on the Asterisk server(Show Channels) a hundred times a minute to keep updated list of live channels
 - ◆ Action sender – Constantly checks for new Actions to be sent to Asterisk servers and starts child processes to send them to the Manager Interface
 - ◆ Action Listener – Looks for the Actions that were sent with the sender in the Manager output and updates their status in the Queue

How a Central Queue Operates

Client-side Operations

- ◆ Client app can grab the list of live channels at any time
- ◆ Client can initiate new actions and look for a response from Central Queue without ever interfacing with the Asterisk server
- ◆ Client does not need to keep active connection with the Central Queue like it would with the Manager Interface, making development platform options more flexible

How a Central Queue Operates

Call-Flow Example

- ◆ Here is a flow of how a call is initiated through a Central Queue
 - 1) Client connects to Queue and inserts parameters of the call to be placed
 - 2) Server sender app sees new Queue entry and initiates child process to connect to the Manager and send the new Action
 - 3) Server listener app sees output from Manager and matches it up to the Queue entry and updates that record's status
 - 4) Client looks at Queue record it sent and sees that the call went through

Why Use MySQL for the Central Queue?

- ◆ Speed – MySQL is a very fast system for information exchange
- ◆ Compatibility – Client libraries are widely available and are very light-weight, and server runs on UNIX and Win32
- ◆ Capacity – You can have several hundred concurrent connections on a single MySQL server
- ◆ Ease of use – MySQL is simple to Set up, Administrate and Write custom Queries for

Real-World Example of a Central Queue in Action

- ◆ Corporate and Call Center Environment with over 150 Employees
- ◆ Five Asterisk servers with 16 T1s connected
- ◆ One MySQL server acting as Central Queue
- ◆ 150 SIP telephone devices
- ◆ Inbound/Outbound telemarketing with custom GUI client apps for CRM and auto-dialing
- ◆ Local and Remote Customer service and Sales Agents using Web interface for call manipulation and information exchange
- ◆ Real-time stats and reports on system operation and agent performance
- ◆ Receptionist console and corporate console with click-to-record and click-to-conference

astGUIClient functionality

The screenshot displays the astGUIClient interface with the following components:

- Title Bar:** astGUIClient - 1.0.3
- Status Bar:** 2004/07/21 15:14:30
- Header:** your ID: SIP/cc100, VOICEMAIL, NEW 000, OLD 001, current channel:
- Live Extensions:** A list of extensions including 108joej - Joe Johnson, 118amy - TPP CS 1, 138pcom - Matt Florell, 152hr - Human Resources, and various cc100-136 call center extensions.
- Recording Section:** START RECORDING, STOP RECORDING, RECORDING MESSAGE: (text input), RECORDING FILENAME: (text input), RECORDING ID: (text input).
- Call Management Buttons:** INTRASYSTEM CALL, BLIND VOICEMAIL XFER, BLIND INTERNAL XFER, HELP, DIAL OUTSIDE NUMBER, BLIND EXTERNAL XFER, PARK THIS CALL, VIEW PARKED CALLS.
- Channels:** Busy: cc116, cc119, cc125, cc130 (highlighted), GALpark; Outside Lines: Zap/4-1_cc116, Zap/1-1_cc119, Zap/2-1_cc125, Zap/3-1_cc130, Zap/5-1_GALpark; SIP Lines: SIP/cc116-1c37_Zap/4, SIP/cc119-76a1_Zap/1, SIP/cc125-e0bf_Zap/2, SIP/cc130-5064_Zap/3.
- Other Features:** Monitor button, SWITCH YOUR ID button, RECENT: (dropdown), Parked: (text input), START CONFERENCE, STOP CONFERENCE.
- Footer:** Build 40716-1345, <astguIClient@eflo.net>

- List live channels
- Recording
- Blind Monitoring
- Voicemail Transfer
- Internal Transfer
- External Transfer
- Call Parking
- Forced Hangup
- Conferencing with up to six external channels
- List of recent calls out
- Click to check voicemail
- Inbound call popup

VICIDIAL screenshot

astVICIDIAL - 0.7

Time: 2004/09/20 12:01:09 Phone ID: SIP/cc100 Session ID: 8600099

User ID: 6666 Password: sales Campaign: TESTCAMP LOGOUT

STATUS: Auto-dialer resumed. Waiting for next call

PAUSE RESUME

RECORDING FILENAME: Length: 46 Zap: Zap/73-1 CustTime: SEP 20 12:01:09 AM

RECORDING ID: Customer Information:

START REC STOP REC Title: Mr First: Matt MI: Last: lead01

Address 1: 1234 Fake St. WEB FORM

Address 2:

Address 3:

City: Clearwater State: FL PostCode: 33780

Province: Vendor ID:

Phone: 7275551212 DialCode: 1 Alt Phone:

Show: surprise Email: test@test.com

Comments: comments go here

INTERNAL CLOSER

LOCAL CLOSER

Code:

LEAVE 3-WAY CALL

PARK CALL GRAB PARK

TRANSFER - CONF

CUSTOMER HUNGUP

HANGUP CUSTOMER

DTMF to send: UK1 UK2 MW2

Number to call: 8175551212 Length: Zap: ,7261,,,,1,,,4

DIAL WITH CUSTOMER PARK CUSTOMER DIAL DIAL BLIND TRANSFER SEND DTMF AUS1 AUS2 US1

GRAB PARK CUSTOMER HANGUP XFER LINE HANGUP BOTH LINES DIALPAD US2 MW

BUILD 40920-1146 <vicidial@eflo.net>

- Dial by list
- Call Recording
- Third party conference, transfer and drop
- DTMF macros
- Call Parking
- Custom on-hold music
- Click-to-dial or predictive dialing
- Web-based closers can be local or remote
- Time zone dialing restriction available

Administration screenshot

VICIDIAL ADMIN: Administration - Mozilla

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop http://10.10.10.15/vicidial/admin.php?ADD=318 Search Print

VICIDIAL ADMIN: Administration Wednesday July 21, 2004 16:07:07 PM

[WELCOME](#) | [LIST ALL USERS](#) | [ADD A NEW USER](#) | [SEARCH FOR A USER](#) | [ADD A CAMPAIGN](#) | [LIST ALL CAMPAIGNS](#)
[SHOW ALL LISTS](#) | [ADD A NEW LIST](#) | [SEARCH FOR A LEAD](#) | [GROUP HOURLY](#)
[SERVER STATS](#) | [PARK REPORT](#) | [YDAD REPORT](#)

MODIFY A CAMPAIGN'S RECORD: TESTCAMP

Campaign ID: TESTCAMP

Campaign Name: Test campaign for VICIDIAL

Active: Y

Park Extension: 8305 - classical

Web Form: http://astguiclient.sourceforge.net/test_VICIDIAL_output.php

Allow Closers: Y

Dial status 1: NEW

Dial status 2: NP

Dial status 3: N

Dial status 4: N

Dial status 5: DROP

List Order: DOWN COUNT

Hopper Level: 50

Force Reset of Hopper: N

Auto Dial Level: 1.5 (0 = off)

Next Agent Call: oldest_call_finish

submit

LISTS WITHIN THIS CAMPAIGN:

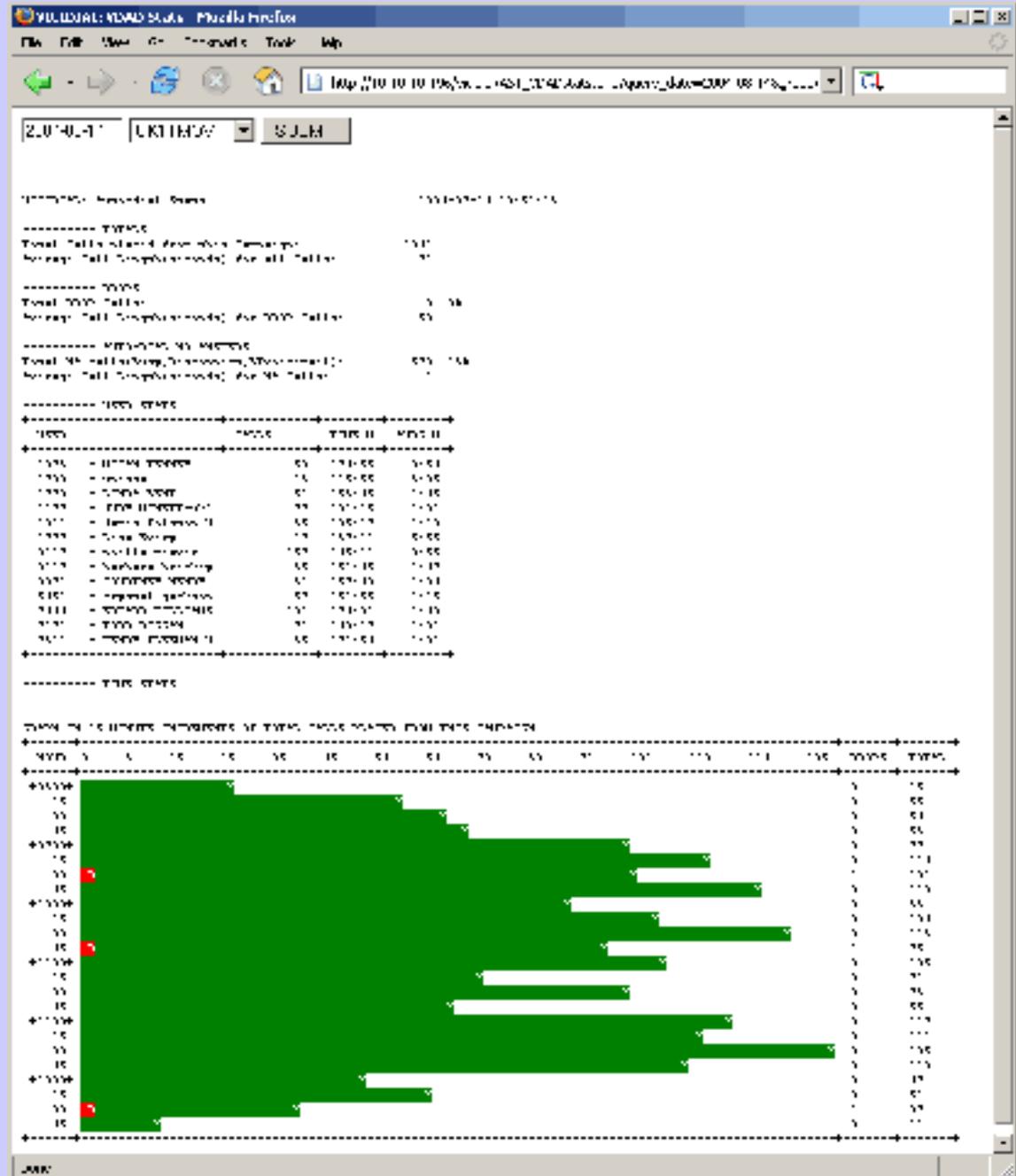
LIST ID	LIST NAME	ACTIVE
101	test domestic list	Y
102	Test UK Answering	N
103	Test list UK Busy	N
104	uk auto-answer	Y

This campaign has 2 active lists and 2 inactive lists

This campaign has 16 leads to be dialed in those lists

This campaign has 16 leads in the dial hopper

Reports screenshot



Capacity Limits and Trade-offs

- ◆ Example Central Queue handles:
 - ◆ Over 100,000 sent Manager actions daily
 - ◆ Over 30,000 calls in and out daily
 - ◆ Over 400 concurrent Asterisk channels
 - ◆ Five live Asterisk servers
 - ◆ 120 connected client applications
- ◆ Trade-offs of using Central Queue:
 - ◆ Only Manager traceable tag is “CallerID” which is a big negative especially for PRI users with customizable CallerID out
 - ◆ Central Queue is a single point of failure
 - ◆ Slight delay of action execution, not very noticeable

The Future

- ◆ Requested new definable tag for Manager Output “CallLabel” that would act just like CallerID to be able to use it for the reason it was meant to be used
- ◆ Research and testing into what causes Asterisk Manager Interface pauses at high loads
- ◆ Possibly a more robust Manager interface that allows for direct interaction with a channel, such as DTMF collection, from within the Manager API