



AMI 1.1 Changes

Created by Malcolm Davenport, last modified by Vlad Povorozniuc on Jun 30, 2011

Changes to manager version 1.1:

SYNTAX CLEANUPS

- Response: headers are now either
 - "Success" - Action OK, this message contains response
 - "Error" - Action failed, reason in Message: header
 - "Follows" - Action OK, response follows in following Events.

- Manager version changed to 1.1

CHANGED EVENTS AND ACTIONS

- The Hold/Unhold events
 - Both are now "Hold" events
 - For hold, there's a "Status: On" header, for unhold, status is off
 - Modules chan_sip/chan_iax2
- The Ping Action
 - Now use Response: success
 - New header "Ping: pong" :-)
- The Events action
 - Now use Response: Success
 - The new status is reported as "Events: On" or "Events: Off"
- The JabberSend action
 - The Response: header is now the first header in the response
 - now sends "Response: Error" instead of "Failure"
- Newstate and Newchannel events
 - these have changed headers

"State"	-> ChannelStateDesc	Text based channel state
	-> ChannelState	Numeric channel state
 - The events does not send "<unknown>" for unknown caller IDs just an empty field
- Newchannel event
 - Now includes "AccountCode"
- Newstate event
 - Now has "CalleridNum" for numeric caller id, like Newchannel
 - The event does not send "<unknown>" for unknown caller IDs just an empty field
- Newexten and VarSet events
 - Now are part of the new Dialplan privilege class, instead of the Call class
- Dial event
 - Event Dial has new headers, to comply with other events
 - Source -> Channel Channel name (caller)
 - SrcUniqueID -> UniqueID Uniqueid
 - (new) -> Dialstring Dialstring in app data
- Link and Unlink events
 - The "Link" and "Unlink" bridge events in channel.c are now renamed to "Bridge"
 - The link state is in the bridgestate: header as "Link" or "Unlink"
 - For channel.c bridges, "Bridgetype: core" is added. This opens up for bridge events in rtp.c
 - The RTP channel also reports Bridge: events with bridgetypes
 - rtp-native RTP native bridge
 - rtp-direct RTP peer-2-peer bridge (NAT support only)
 - rtp-remote Remote (re-invite) bridge. (Not reported yet)
- The "Rename" manager event has a renamed header, to use the same terminology for the current channel as other events
 - Oldname -> Channel
- The "NewCallerID" manager event has a renamed header
 - CallerID -> CallerIDnum
 - The event does not send "<unknown>" for unknown caller IDs just an empty field

- Reload event
 - The "Reload" event sent at manager reload now has a new header and is now implemented in more modules than manager to alert a reload. For channels, there's a CHANNELRELOAD event to use.
 - (new) -> Module: manager | CDR | DNSmgr | RTP | ENUM
 - (new) -> Status: enabled | disabled
 - To support reload events from other modules too
 - cdr module added

- Status action replies (Event: Status)
 - Header changes
 - link -> BridgedChannel
 - Account -> AccountCode
 - (new) -> BridgedUniqueid

- StatusComplete Event
 - New header
 - (new) -> Items Number of channels reported

- The ExtensionStatus manager command now has a "StatusDesc" field with text description of the sta
- The Registry and Peerstatus events in chan_sip and chan_iax now use "ChannelType" instead of "Cha
- The Response to Action: IAXpeers now have a Response: Success header
- The MeetmeJoin now has caller ID name and Caller ID number fields (like MeetMeLeave)

- Action DAHDIShowChannels
 - Header changes
 - Channel: -> DAHDIChannel
 - For active channels, the Channel: and Uniqueid: headers are added
 - You can now add a "DAHDIChannel: " argument to DAHDIShowchannels actions to only get information about one channel.

- Event DAHDIShowChannelsComplete
 - New header
 - (new) -> Items: Reports number of channels reported

- Action VoicemailUsersList
 - Added new headers for SayEnvelope, SayCID, AttachMessage, CanReview and CallOperator voicemail configuration settings.

- Action Originate
 - Now requires the new Originate privilege.
 - If you call out to a subshell in Originate with the Application parameter, you now also need the System privilege.

- Event QueueEntry now also returns the Uniqueid field like other events from app_queue.

- Action IAXpeerlist
 - Now includes if the IAX link is a trunk or not

- Action IAXpeers
 - Now includes if the IAX link is a trunk or not

- Action Ping
 - Response now includes a timestamp

- Action SIPshowpeer
Response now includes the configured parkinglot
- Action SKINNYshowline
Response now includes the configured parkinglot

NEW ACTIONS

- Action: DataGet
 - Modules: data.c
 - Purpose:
 - To be able to retrieve the asterisk data tree.
 - Variables:
 - ActionID: <id> Action ID for this transaction. Will be returned.
 - Path: <data path> The path to the callback node to retrieve.
 - Filter: <filter> Which nodes to retrieve.
 - Search: <search> Search condition.

- Action: IAXregistry
 - Modules: chan_iax2
 - Purpose:
 - To list all IAX2 peers in the IAX registry with their registration status.
 - Variables:
 - ActionID: <id> Action ID for this transaction. Will be returned.

- Action: ModuleLoad
 - Modules: loader.c
 - Purpose:
 - To be able to unload, reload and unload modules from AMI.
 - Variables:
 - ActionID: <id> Action ID for this transaction. Will be returned.
 - Module: <name> Asterisk module name (including .so extension) or subsystem identifier:
 - cdr, enum, dnsmgr, extconfig, manager, rtp, http
 - LoadType: load | unload | reload
 - The operation to be done on module
 - If no module is specified for a reload loadtype, all modules are reloaded

- Action: ModuleCheck
 - Modules: loader.c
 - Purpose:
 - To check version of a module - if it's loaded
 - Variables:
 - ActionID: <id> Action ID for this transaction. Will be returned.
 - Module: <name> Asterisk module name (not including extension)
 - Returns:
 - If module is loaded, returns version number of the module
 - Note: This will have to change. I don't like sending Response: failure on both command not found (trying this command in earlier versions of Asterisk) and module not found.
 - Also, check if other manager actions behave that way.

- Action: QueueSummary
 - Modules: app_queue
 - Purpose:
 - To request that the manager send a QueueSummary event (see the NEW EVENTS section for more details).
 - Variables:
 - ActionID: <id> Action ID for this transaction. Will be returned.
 - Queue: <name> Queue for which the summary is desired

- Action: QueuePenalty
 - Modules: app_queue
 - Purpose:

To change the penalty of a queue member from AMI

Variables:

Interface: <tech/name> The interface of the member whose penalty you wish to change
Penalty: <number> The new penalty for the member. Must be nonnegative.
Queue: <name> If specified, only set the penalty for the member for this queue.
Otherwise, set the penalty for the member in all queues to which he belongs.

- Action: QueueRule

Modules: app_queue

Purpose:

To list queue rules defined in queuerules.conf

Variables:

ActionID: <id> Action ID for this transaction. Will be returned.
Rule: <name> The name of the rule whose contents you wish to list. If the rule is not present, all rules in queuerules.conf will be listed.

- Action: Atxfer

Modules: none

Purpose:

Initiate an attended transfer

Variables:

Channel: The transferer channel's name
Exten: The extension to transfer to
Priority: The priority to transfer to
Context: The context to transfer to

- Action: SipShowRegistry

Modules: chan_sip

Purpose:

To request that the manager send a list of RegistryEntry events.

Variables:

ActionID: <id> Action ID for this transaction. Will be returned.

- Action: QueueReload

Modules: app_queue

Purpose:

To reload queue rules, a queue's members, a queue's parameters, or all of the above.

Variable:

ActionID: <id>
Queue: <name> The name of the queue to take action on.
If no queue name is specified, then all queues are affected.
Rules: <yes or no> Whether to reload queuerules.conf
Members: <yes or no> Whether to reload the queue's members
Parameters: <yes or no> Whether to reload the other queue options

- Action: QueueReset

Modules: app_queue

Purpose:

Reset the statistics for a queue

Variables:

ActionID: <id>
Queue: <name> The name of the queue on which to reset statistics

- Action: SKINNYdevices

Modules: chan_skinny

Purpose:

To list all SKINNY devices configured.

- Variables:
ActionId: <id> Action ID for this transaction. Will be returned.
- Action: SKINNYlines
Modules: chan_skinny
Purpose:
To list all SKINNY lines configured.
Variables:
ActionId: <id> Action ID for this transaction. Will be returned.
 - Action SKINNYshowdevice
Modules: chan_skinny
Purpose:
To list the information about a specific SKINNY device.
Variables:
Device: <device> Device to show information about.
 - Action SKINNYshowline
Modules: chan_skinny
Purpose:
To list the information about a specific SKINNY line.
Variables:
Line: <line> Line to show information about.
 - Action: CoreSettings
Modules: manager.c
Purpose: To report core settings, like AMI and Asterisk version,
maxcalls and maxload settings.
* Integrated in SVN trunk as of May 4th, 2007
Example:
Response: Success
ActionID: 1681692777
AMIVersion: 1.1
AsteriskVersion: SVN-oej-moremanager-r61756M
SystemName: EDVINA-node-a
CoreMaxCalls: 120
CoreMaxLoadAvg: 0.000000
CoreRunUser: edvina
CoreRunGroup: edvina
 - Action: CoreStatus
Modules: manager.c
Purpose: To report current PBX core status flags, like
number of concurrent calls, startup and reload time.
* Integrated in SVN trunk as of May 4th, 2007
Example:
Response: Success
ActionID: 1649760492
CoreStartupTime: 22:35:17
CoreReloadTime: 22:35:17
CoreCurrentCalls: 20
 - Action: MixMonitorMute
Modules: app_mixmonitor.c
Purpose:
Mute / unMute a Mixmonitor recording.
Variables:
ActionId: <id> Action ID for this transaction. Will be returned.

Channel: the channel MixMonitor is running on
Direction: Which part of the recording to mute: read, write or both (from
channel, to channel or both channels).
State: Turn mute on or off : 1 to turn on, 0 to turn off.

NEW EVENTS

- Event: FullyBooted

Modules: loader.c

Purpose:

It is handy to have a single event notification for when all Asterisk modules have been loaded--especially for situations like running automated tests. This event will fire 1) immediately upon all modules loading or 2) upon connection to the AMI interface if the modules have already finished loading before the connection was made. This ensures that a user will never miss getting a FullyBooted event. In vary rare circumstances, it might be possible to get two copies of the message if the AMI connection is made right as the modules finish loading.

Example:

```
Event: FullyBooted
Privilege: system,all
Status: Fully Booted
```

- Event: Transfer

Modules: res_features, chan_sip

Purpose:

Inform about call transfer, linking transferer with transfer target. You should be able to trace the call flow with this missing piece of information. If it works out well, the "Transfer" event should be followed by a "Bridge" event. The transfermethod: header informs if this is a pbx core transfer or something done on channel driver level. For SIP, check the example:

Example:

```
Event: Transfer
Privilege: call,all
TransferMethod: SIP
TransferType: Blind
Channel: SIP/device1-01849800
SIP-Callid: 091386f505842c87016c4d93195ec67d@127.0.0.1
TargetChannel: SIP/device2-01841200
TransferExten: 100
TransferContext: default
```

- Event: ChannelUpdate

Modules: chan_sip.c, chan_iax2.c

Purpose:

Updates channel information with ID of PVT in channel driver, to be able to link events on channel driver level.
* Integrated in SVN trunk as of May 4th, 2007

Example:

```
Event: ChannelUpdate
Privilege: system,all
Uniqueid: 1177271625.27
Channel: SIP/olle-01843c00
Channeltype: SIP
SIPcallid: NTQzYWFi0WM4NmE0MWRkZjExMzU2YzQ3OWQwNzg3ZmI.
SIPfullcontact: sip:olle@127.0.0.1:49054
```

- Event: NewAccountCode

Modules: cdr.c

Purpose: To report a change in account code for a live channel

Example:

```
Event: NewAccountCode
Privilege: call,all
Channel: SIP/olle-01844600
Uniqueid: 1177530895.2
AccountCode: Stinas account 1234848484
OldAccountCode: OllesAccount 12345
```

- Event: ModuleLoadReport

Modules: loader.c

Purpose: To report that module loading is complete. Some aggressive clients connect very quickly to AMI and needs to know when all manager events embedded in modules are loaded
Also, if this does not happen, something is seriously wrong.
This could happen to chan_sip and other modules using DNS.

Example:

```
Event: ModuleLoad
ModuleLoadStatus: Done
ModuleSelection: All
ModuleCount: 24
```

- Event: QueueSummary

Modules: app_queue

Purpose: To report a summary of queue information. This event is generated by issuing a QueueSummary AMI action.

Example:

```
Event: QueueSummary
Queue: Sales
LoggedIn: 12
Available: 5
Callers: 10
HoldTime: 47
```

If an actionID was specified for the QueueSummary action, it will be appended as the last line of the QueueSummary event.

- Event: AgentRingNoAnswer

Modules: app_queue

Purpose: Reports when a queue member was rung but there was no answer.

Example:

```
Event: AgentRingNoAnswer
Queue: Support
Uniqueid: 1177530895.2
Channel: SIP/1000-53aee458
Member: SIP/1000
MemberName: Thaddeus McClintock
Ringtime: 10
```

- Event: RegistryEntry

Modules: chan_sip

Purpose: Reports the state of the SIP registrations. This event is generated by issuing a QueueSummary AMI action.
The RegistrationTime header is expressed as epoch.

Example:

```
Event: RegistryEntry
Host: sip.myvoipprovider.com
Port: 5060
Username: guestuser
Refresh: 105
```

State: Registered

RegistrationTime: 1219161830

If an actionID was specified for the SipShowRegistry action, it will be appended as the last line of the RegistrationsComplete event.

- Event: ChanSpyStart

Modules: app_chanspy

Purpose: Reports when an active channel starts to be monitored by someone.

Example:

Event: ChanSpyStart

SpyerChannel: SIP/4321-13bba124

SpyeeChannel: SIP/1234-56ecc098

- Event: ChanSpyStop

Modules: app_chanspy

Purpose: Reports when an active channel stops to be monitored by someone.

Example:

Event: ChanSpyStop

SpyeeChannel: SIP/1234-56ecc098

TODO

...