**Asterisk Manager Interface (AMI)**

The Asterisk Manager Interface (AMI) allows a client program to connect to an Asterisk instance and issue commands or read events over a TCP/IP stream. Integrators will find this particularly useful when trying to track the state of a telephony client inside Asterisk, and directing that client based on custom (and possibly dynamic) rules.

**Protocol Behavior**

The protocol has the following characteristics:

* Before issuing commands to Asterisk, you must establish a manager session (see below).
* Packets may be transmitted in either direction at any time after authentication.
* The first line of a packet will have a key of "Action" when sent from the client to Asterisk, but "Event" or "Response" when sent from Asterisk to the client.
* The order of lines within a packet is insignificant, so you may use your favorite programming language's native unordered dictionary type to efficiently store a single packet.
* CR/LF is used to delimit each line and a blank line (two CR/LF in a row) indicates the end of the command which Asterisk is now expected to process.

**Packet Types**

The type of a packet is determined by the existence of one of the following keys:

* **Action**: A packet sent by the connected client to Asterisk, requesting a particular Action be performed. There are a finite (but extendable) set of actions available to the client, determined by the modules presently loaded in the Asterisk engine. Only one action may be outstanding at a time. The Action packet contains the name of the operation to be performed as well as all required parameters.
* **Response**: the response sent by Asterisk to the last action sent by the client.
* **Event**: data pertaining to an event generated from within the Asterisk core or an extension module.

Generally the client sends Action packets to the Asterisk server, the Asterisk server performs the requested operation and returns the result (often only success or failure) in a Response packet. As there is no guarantee regarding the order of Response packets the client usually includes an ActionID parameter in every Action packet that is sent back by Asterisk in the corresponding Response packet. That way the client can easily match Action and Response packets while sending Actions at any desired rate without having to wait for outstanding Response packets before sending the next action.  
  
Event packets are used in two different contexts: On the one hand they inform clients about state changes in Asterisk (like new channels being created and hung up or agents being logged in and out) on the other hand they are used to transport the response payload for actions that return a list of data (event generating actions). When a client sends an event generating action Asterisk sends a Response packed indicating success and containing a "Response: Follows" line. Then it sends zero or more events that contain the actual payload and finally an action complete event indicating that all data has been sent. The events sent in response to an event generating action and the action complete event contain the ActionID of the Action packet that triggered them, so you can easily match them the same way as Response packets. An example of an event generating action is the Status action that triggers Status events for each active channel. When all Status events have been sent a terminating a StatusComplete event is sent.

**Opening a Manager Session and Authenticating as a User**

In order to access the Asterisk Manager functionality a user needs to establish a session by opening a TCP/IP connection to the listening port (usually 5038) of the Asterisk instance and logging into the manager using the 'Login' action. This requires a previously established user account on the Asterisk server. User accounts are configured in [**/etc/asterisk/manager.conf**](http://www.voip-info.org/wiki/view/Asterisk+config+manager.conf). A user account consists of a set of permitted IP hosts, an authentication secret (password), and a list of granted permissions.  
  
There is a finite set of permissions, each may be granted with either "read", "write", or "read/write" granularity. If a client is granted the ability to read a given class, Asterisk will send it events of that class. If a client is granted the ability to write a given class, it may send actions of that class.  
  
To login and authenticate to the manager, you must send a "*Login*" action, with your user name and secret (password) as parameters. Here is an example:  
  
Action: login  
Username: admin  
Secret: god  
  
If you do not need to subscribe to events being generated by Asterisk, you may also include the "Events: off" parameter, which will prevent event packets being sent to your connection. This is the equivalent of calling the "Events" action. Example:  
  
Action: login  
Username: admin  
Secret: god  
Events: off

**Action Packets**

When sending Asterisk an action, extra keys may be provided to further direct execution, for example, you may wish to specify a number to call, a channel to disconnect. Additionally, if your action causes Asterisk to execute an entry in the dialplan, you may wish to pass variables to the dialplan (available as of [bug 1268](http://bugs.digium.com/bug_view_page.php?bug_id=0001268)). This is done exactly the same way you would send keys.  
  
To send Asterisk an action, follow this simple format:  
  
Action: <*action type*><*CRLF*>  
<*Key 1*>: <*Value 1*><*CRLF*>  
<*Key 2*>: <*Value 2*><*CRLF*>  
...  
Variable: <*Variable 1*>=<*Value 1*><*CRLF*>  
Variable: <*Variable 2*>=<*Value 2*><*CRLF*>  
...  
<*CRLF*>

**Manager Actions**

Output from the [CLI](http://www.voip-info.org/wiki/view/Asterisk+CLI) command **manager show commands**:  
(For Asterisk 1.2 and earlier, use **show manager commands**)

* [AbsoluteTimeout](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+AbsoluteTimeout): Set Absolute Timeout **(privilege: call,all)**
* [ChangeMonitor](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ChangeMonitor): Change monitoring filename of a channel **(privilege: call,all)**
* [Command](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Command): Execute Command **(privilege: command,all)**
* [Events](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Events): Control Event Flow
* [ExtensionState](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ExtensionState): Check Extension Status **(privilege: call,all)**
* [GetVar](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+GetVar): Gets a Channel Variable **(privilege: call,all)**
* [Hangup](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Hangup): Hangup Channel \_\_(privilege: call,all)
* [IAXpeers](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+IAXpeers): List IAX Peers **(privilege: system,all)**
* [ListCommands](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ListCommands): List available manager commands
* [Logoff](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Logoff): Logoff Manager
* [MailboxCount](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+MailboxCount): Check Mailbox Message Count **(privilege: call,all)**
* [MailboxStatus](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+MailboxStatus): Check Mailbox **(privilege: call,all)**
* [Monitor](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Monitor): Monitor a channel **(privilege: call,all)**
* [Originate](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Originate): Originate Call **(privilege: call,all)** NOTE: starting from 1.6: originate,all
* [ParkedCalls](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ParkedCalls): List parked calls
* [Ping](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Ping): Ping
* [QueueAdd](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+QueueAdd): Queues **(privilege: agent,all)**
* [QueueRemove](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+QueueRemove): Queues **(privilege: agent,all)**
* [Queues](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Queues): Queues
* [QueueStatus](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+QueueStatus): Queue Status
* [Redirect](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Redirect): Redirect **(privilege: call,all)**
* [SetCDRUserField](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+SetCDRUserField): Set the CDR UserField **(privilege: call,all)**
* [SetVar](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+SetVar): Set Channel Variable **(privilege: call,all)**
* [SIPpeers](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+SIPpeers): List SIP Peers (chan\_sip2 only. Not available in chan\_sip as of 9/20/2004) **(privilege: system,all)**
* [Status](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Status): Status **(privilege: call,all)**
* [StopMonitor](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+StopMonitor): Stop monitoring a channel **(privilege: call,all)**
* [ZapDialOffhook](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ZapDialOffhook): Dial over Zap channel while offhook
* [ZapDNDoff](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ZapDNDoff): Toggle Zap channel Do Not Disturb status OFF
* [ZapDNDon](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ZapDNDon): Toggle Zap channel Do Not Disturb status ON
* [ZapHangup](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ZapHangup): Hangup Zap Channel
* [ZapTransfer](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ZapTransfer): Transfer Zap Channel
* [ZapShowChannels](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ZapShowChannels): Show Zap Channels

(New?) in Asterisk 1.2.1 (was "CVS HEAD") (Taken from the output of [CLI](http://www.voip-info.org/wiki/view/Asterisk+CLI) command **show manager commands**):

* [AgentCallbackLogin](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+AgentCallBackLogin): Sets an agent as logged in by callback **(Privilege: agent,all)**
* [AgentLogoff](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+AgentLogoff): Sets an agent as no longer logged in **(Privilege: agent,all)**
* Agents: Lists agents and their status **(Privilege: agent,all)**
* [DBGet](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+DBGet): Get DB Entry **(Privilege: system,all)**
* [DBPut](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+DBPut): Put DB Entry **(Privilege: system,all)**
* [QueuePause](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+QueuePause): Makes a queue member temporarily unavailable **(Privilege: agent,all)**
* [SIPshowPeer](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+SIPshowPeer): Show SIP peer (text format) **(Privilege: system,all)**

New in Asterisk 1.4.0

* [GetConfig](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+GetConfig): Display a configuration file, used mainly by [AJAM](http://www.voip-info.org/wiki/view/Aynchronous+Javascript+Asterisk+Manager+%28AJAM%29)/Asterisk-gui. **(Privilege: config,all)**
* [PlayDTMF](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+PlayDTMF): Play DTMF signal on a specific channel. **(Privilege: call,all)**
* [UpdateConfig](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+UpdateConfig): Updates a configuration file, used mainly by [AJAM](http://www.voip-info.org/wiki/view/Aynchronous+Javascript+Asterisk+Manager+%28AJAM%29)/Asterisk-gui. **(Privilege: config,all)**

Available in Asterisk 1.6.0

* [AbsoluteTimeout](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+AbsoluteTimeout): Set Absolute Timeout **(Priv: system,call,all)**
* AgentLogoff: Sets an agent as no longer logged in **(Priv: agent,all)**
* Agents: Lists agents and their status **(Priv: agent,all)**
* AGI: Add an AGI command to execute by Async AGI **(Priv: call,all)**
* Bridge: Bridge two channels already in the PBX **(Priv: call,all)**
* Challenge: Generate Challenge for MD5 Auth **(Priv: <none>)**
* [ChangeMonitor](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ChangeMonitor): Change monitoring filename of a channel **(Priv: call,all)**
* [Command](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Command): Execute Asterisk CLI Command **(Priv: command,all)**
* CoreSettings: Show PBX core settings (version etc) **(Priv: system,reporting,all)**
* CoreShowChannels: List currently active channels **(Priv: system,reporting,all)**
* CoreStatus: Show PBX core status variables **(Priv: system,reporting,all)**
* CreateConfig: Creates an empty file in the configuration directory **(Priv: config,all)**
* DAHDIDialOffhook: Dial over DAHDI channel while offhook **(Priv: <none>)**
* DAHDIDNDoff: Toggle DAHDI channel Do Not Disturb status OFF **(Priv: <none>)**
* DAHDIDNDon: Toggle DAHDI channel Do Not Disturb status ON **(Priv: <none>)**
* DAHDIHangup: Hangup DAHDI Channel **(Priv: <none>)**
* DAHDIRestart: Fully Restart DAHDI channels (terminates calls) **(Priv: <none>)**
* DAHDIShowChannels: Show status dahdi channels **(Priv: <none>)**
* DAHDITransfer: Transfer DAHDI Channel **(Priv: <none>)**
* DBDel: Delete DB Entry **(Priv: system,all)**
* DBDelTree: Delete DB Tree **(Priv: system,all)**
* [DBGet](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+DBGet): Get DB Entry **(Priv: system,reporting,all)**
* [DBPut](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+DBPut): Put DB Entry **(Priv: system,all)**
* [Events](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Events): Control Event Flow **(Priv: <none>)**
* [ExtensionState](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ExtensionState): Check Extension Status **(Priv: call,reporting,all)**
* GetConfigJSON: Retrieve configuration (JSON format) **(Priv: system,config,all)**
* [GetConfig](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+GetConfig): Retrieve configuration **(Priv: system,config,all)**
* [Getvar](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+GetVar): Gets a Channel Variable **(Priv: call,reporting,all)**
* [Hangup](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Hangup): Hangup Channel **(Priv: system,call,all)**
* IAXnetstats: Show IAX Netstats **(Priv: system,reporting,all)**
* IAXpeerlist: List IAX Peers **(Priv: system,reporting,all)**
* [IAXpeers](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+IAXpeers): List IAX Peers **(Priv: system,reporting,all)**
* ListCategories: List categories in configuration file **(Priv: config,all)**
* [ListCommands](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ListCommands): List available manager commands **(Priv: <none>)**
* [Login](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Login): Login Manager **(Priv: <none>)**
* [Logoff](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Logoff): Logoff Manager **(Priv: <none>)**
* [MailboxCount](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+MailboxCount): Check Mailbox Message Count **(Priv: call,reporting,all)**
* [MailboxStatus](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+MailboxStatus): Check Mailbox **(Priv: call,reporting,all)**
* MeetmeMute: Mute a Meetme user **(Priv: call,all)**
* MeetmeUnmute: Unmute a Meetme user **(Priv: call,all)**
* ModuleCheck: Check if module is loaded **(Priv: system,all)**
* ModuleLoad: Module management **(Priv: system,all)**
* [Monitor](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Monitor): Monitor a channel **(Priv: call,all)**
* [Originate](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Originate): Originate Call **(Priv: originate,all)**
* [ParkedCalls](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+ParkedCalls): List parked calls **(Priv: <none>)**
* Park: Park a channel **(Priv: call,all)**
* PauseMonitor: Pause monitoring of a channel **(Priv: call,all)**
* [Ping](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Ping): Keepalive command **(Priv: <none>)**
* [PlayDTMF](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+PlayDTMF): Play DTMF signal on a specific channel. **(Priv: call,all)**
* [QueueAdd](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+QueueAdd): Add interface to queue. **(Priv: agent,all)**
* QueueLog: Adds custom entry in queue\_log **(Priv: agent,all)**
* [QueuePause](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+QueuePause): Makes a queue member temporarily unavailable **(Priv: agent,all)**
* QueuePenalty: Set the penalty for a queue member **(Priv: agent,all)**
* [QueueRemove](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+QueueRemove): Remove interface from queue. **(Priv: agent,all)**
* QueueRule: Queue Rules **(Priv: <none>)**
* [Queues](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Queues): Queues **(Priv: <none>)**
* [QueueStatus](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+QueueStatus): Queue Status **(Priv: <none>)**
* [QueueSummary](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+QueueSummary): Queue Status **(Priv: <none>)**
* Redirect: Redirect (transfer) a call **(Priv: call,all)**
* [Reload](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Reload): Send a reload event **(Priv: system,config,all)**
* SendText: Send text message to channel **(Priv: call,all)**
* [Setvar](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+SetVar): Set Channel Variable **(Priv: call,all)**
* ShowDialPlan: List dialplan **(Priv: config,reporting,all)**
* [SIPpeers](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+SIPpeers): List SIP peers (text format) **(Priv: system,reporting,all)**
* [SIPshowpeer](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+SIPshowPeer): Show SIP peer (text format) **(Priv: system,reporting,all)**
* SIPshowregistry: Show SIP registrations (text format) **(Priv: system,reporting,all)**
* [Status](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+Status): Lists channel status **(Priv: system,call,reporting,all)**
* [StopMonitor](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+StopMonitor): Stop monitoring a channel **(Priv: call,all)**
* UnpauseMonitor: Unpause monitoring of a channel **(Priv: call,all)**
* [UpdateConfig](http://www.voip-info.org/wiki/view/Asterisk+Manager+API+Action+UpdateConfig): Update basic configuration **(Priv: config,all)**
* UserEvent: Send an arbitrary event **(Priv: user,all)**
* VoicemailUsersList: List All Voicemail User Information **(Priv: call,reporting,all)**
* WaitEvent: Wait for an event to occur **(Priv: <none>).**

# Asterisk Manager: Events

**Agent Status Events**

'Agentcallbacklogin' Event

**Description**:  
[derived from chan\_agent.c]  
  
**Data Sample**:  
Event: Agentcallbacklogin  
Agent: <agent>  
Loginchan: <loginchan>  
Uniqueid: <uniqueid>

'Agentcallbacklogoff' Event

**Description**:  
[derived from chan\_agent.c]  
  
**Data Sample**:  
Event: Agentcallbacklogoff  
Agent: <agent>  
Loginchan: <loginchan>  
Logintime: <logintime>  
Reason: Autologoff  
Uniqueid: <uniqueid>  
  
Event: Agentcallbacklogoff  
Agent: <agent>  
Loginchan: <loginchan>  
Logintime: <logintime>  
Uniqueid: <uniqueid>

'AgentCalled' Event

**Description**:  
[derived from app\_queue.c]  
  
**Data Sample**:  
Event: AgentCalled  
AgentCalled: <channel>  
ChannelCalling: <channel>  
CallerID: <callerid>  
Context: <context>  
Extension: <extension>  
Priority: <priority>

'AgentComplete' Event

**Description**:  
[derived from app\_queue.c]  
  
**Data Sample**:  
Event: AgentComplete  
Queue: <queue>  
Uniqueid: <uniqueid>  
Channel: <channel>  
Member: <member>  
MemberName: <membername>  
HoldTime: <holdtime>  
TalkTime: <talktime>  
Reason: <reason>

'AgentConnect' Event

**Description**:  
[derived from app\_queue.c]  
  
**Data Sample**:  
Event: AgentConnect  
Queue: <queue>  
Uniqueid: <uniqueid>  
Channel: <channel>  
Member: <member>  
MemberName: <membername>  
Holdtime: <holdtime>  
BridgedChannel: <bridgedchannel>

'AgentDump' Event

**Description**:  
[derived from app\_queue.c]  
  
**Data Sample**:  
Event: AgentDump  
Queue: <queue>  
Uniqueid: <uniqueid>  
Channel: <channel>  
Member: <member>  
MemberName: <membername>

'Agentlogin' Event

**Description**:  
[derived from chan\_agent.c]  
  
**Data Sample**:  
Event: Agentlogin  
Agent: <agent>  
Channel: <channel>  
Uniqueid: <uniqueid>

'Agentlogoff' Event

**Description**:  
[derived from chan\_agent.c]  
  
**Data Sample**:  
Event: Agentlogoff  
Agent: <agent>  
Logintime: <logintime>  
Uniqueid: <uniqueid>

'QueueMemberAdded' Event

**Description**:

1. Sent on Action QueueAdd

[derived from app\_queue.c]  
  
**Data Sample**:  
Queue: testing  
Location: Agent/AgentId  
Membership: dynamic  
Penalty: 0  
CallsTaken: 0  
LastCall: 0  
Status: 4  
Paused: 1

'QueueMemberPaused' Event

**Description**:

1. Sent on Action: QueuePause

[derived from app\_queue.c]  
  
**Data Sample**:  
Event: QueueMemberPaused  
Location: <location>  
MemberName: <membername>  
Paused: <paused>

'QueueMemberStatus' Event

**Description**:  
[derived from app\_queue.c]  
  
  
As far as I know Possible values are:  
/\*! Device is valid but channel didn't know state \*/

1. define AST\_DEVICE\_UNKNOWN 0

/\*! Device is not used \*/

1. define AST\_DEVICE\_NOT\_INUSE 1

/\*! Device is in use \*/

1. define AST\_DEVICE\_INUSE 2

/\*! Device is busy \*/

1. define AST\_DEVICE\_BUSY 3

/\*! Device is invalid \*/

1. define AST\_DEVICE\_INVALID 4

/\*! Device is unavailable \*/

1. define AST\_DEVICE\_UNAVAILABLE 5

/\*! Device is ringing \*/

1. define AST\_DEVICE\_RINGING 6

/\*! Device is ringing \*and\* in use \*/

1. define AST\_DEVICE\_RINGINUSE 7

/\*! Device is on hold \*/

1. define AST\_DEVICE\_ONHOLD 8

fernando.berretta@voipexperts.com.ar  
  
**Data Sample**:  
Event: QueueMemberStatus  
Queue: <queue>  
Location: <location>  
MemberName: <membername>  
Membership: <membership>  
Penalty: <penalty>  
CallsTaken: <callstaken>  
LastCall: <lastcall>  
Status: <status>  
Paused: <paused>

**Command Status Events**

**Call Status Events**

'Cdr' Event

**Description**:  
[derived from cdr\_manager.c]  
  
Must be enabled in cdr\_manager.conf  
  
[general]  
enabled = yes  
  
**Data Sample**:  
Event: Cdr  
AccountCode:  
Source:  
Destination:  
DestinationContext:  
CallerID:  
Channel:  
DestinationChannel:  
LastApplication:  
LastData:  
StartTime:  
AnswerTime:  
EndTime:  
Duration:  
BillableSeconds:  
Disposition:  
AMAFlags:  
UniqueID:  
UserField:

'Dial' Event

**Description**:  
[derived from app\_dial.c]  
  
**Data Sample**:  
Event: Dial  
Privilege: call,all  
Source: Local/900@default-2dbf,2  
Destination: SIP/900-4c21  
CallerID: <unknown>  
CallerIDName: default  
SrcUniqueID: 1149161705.2  
DestUniqueID: 1149161705.4

'ExtensionStatus' Event

**Description**:  
[derived from manager.c]  
  
**Data Sample**:  
Event: ExtensionStatus  
Exten: <ext>  
Context: <context>  
Status: <state>

'Hangup' Event

**Description**:  
[derived from channel.c]  
  
**Data Sample**:  
Event: Hangup  
Channel: SIP/101-3f3f  
Uniqueid: 1094154427.10  
Cause: 0  
  
**Cause Codes**

* UNALLOCATED = 1
* NO ROUTE TRANSIT NET = 2
* NO\_ROUTE\_DESTINATION = 3
* CHANNEL\_UNACCEPTABLE = 6
* CALL\_AWARDED\_DELIVERED = 7
* NORMAL\_CLEARING = 16
* USER\_BUSY = 17
* NO USER RESPONSE = 18
* NO ANSWER = 19
* CALL REJECTED = 21
* NUMBER CHANGED = 22
* DESTINATION OUT OF ORDER = 27
* INVALID NUMBER FORMAT = 28
* FACILITY REJECTED = 29
* RESPONSE TO STATUS ENQUIRY = 30
* NORMAL UNSPECIFIED = 31
* NORMAL CIRCUIT CONGESTION = 34
* NETWORK OUT OF ORDER = 38
* NORMAL TEMPORARY FAILURE = 41
* SWITCH CONGESTION = 42
* ACCESS INFO DISCARDED = 43
* REQUESTED CHAN UNAVAIL = 44
* PRE EMPTED = 45
* FACILITY NOT SUBSCRIBED = 50
* OUTGOING CALL BARRED = 52
* INCOMING CALL BARRED = 54
* BEARERCAPABILITY NOTAUTH = 57
* BEARERCAPABILITY NOTAVAIL = 58
* BEARERCAPABILITY NOTIMPL = 65
* CHAN NOT IMPLEMENTED = 66
* FACILITY NOT IMPLEMENTED = 69
* INVALID CALL REFERENCE = 81
* INCOMPATIBLE DESTINATION = 88
* INVALID MSG UNSPECIFIED = 95
* MANDATORY IE MISSING = 96
* MESSAGE TYPE NONEXIST = 97
* WRONG MESSAGE = 98
* IE NONEXIST = 99
* INVALID IE CONTENTS = 100
* WRONG CALL STATE = 101
* RECOVERY ON TIMER EXPIRE = 102
* MANDATORY IE LENGTH ERROR = 103
* PROTOCOL ERROR = 111
* INTERWORKING = 127
* NOT DEFINED = 0

'MusicOnHold' Event

**Description**:  
1. Occurs when a channel is placed on hold/unhold and music is played to the caller.  
  
**Data Sample**:  
  
Event: MusicOnHold  
Channel: <Channel ID>  
State: <Start/Stop>  
Uniqueid: <Unique ID>

'Join' Event

**Description**:  
[derived from app\_queue.c]  
  
**Data Sample**:  
Event: Join  
Channel: <channel>  
CallerID: <callerid|unknown>  
Queue: <queuename>  
Position: <entryposition>  
Count: <queuemembercount>

'Leave' Event

**Description**:  
[derived from app\_queue.c]  
  
**Data Sample**:  
Event: Leave  
Channel: <channel>  
Queue: <queuename>  
Count: <queuemembercount>

'Link' Event

**Description**:

1. Fired when two voice channels are linked together and voice data exchange commences.

**Notes**:

1. Several *Link* events may be seen for a single call. This can occur when Asterisk fails to setup a *native bridge* for the call. As far as I can tell, this is when Asterisk must sit between two telephones and perform CODEC conversion on their behalf.

**Data Sample**:  
  
Event: Link  
Channel1: SIP/101-3f3f  
Channel2: Zap/2-1  
Uniqueid1: 1094154427.10  
Uniqueid2: 1094154427.11  
  
Note: in current version it replaced by Bridge event, example is  
Channel2: SIP/1-1.1.1.1-0002ef99  
Bridgestate: Link  
Event: Bridge  
Privilege: call,all  
Uniqueid2: 1393028418.530941  
Channel1: SIP/peer-local-0002ef98,  
Bridgetype: core  
Uniqueid1: 1393028346.530940  
Timestamp: 1393028428.618726  
CallerID1: 323187134981  
CallerID2: 323187134981

'MeetmeJoin' Event

**Description**:  
[derived from app\_meetme.c]  
  
**Data Sample**:  
Event: MeetmeJoin  
Channel: <channel>  
Uniqueid: <uniqueid>  
Meetme: <meetme>  
Usernum: <usernum>

'MeetmeLeave' Event

**Description**:  
[derived from app\_meetme.c]  
  
**Data Sample**:  
Event: MeetmeLeave  
Channel: <channel>  
Uniqueid: <uniqueid>  
Meetme: <meetme>  
Usernum: <usernum>

'MeetmeStopTalking' Event

**Description**:  
[derived from app\_meetme.c]  
  
**Notes**:

1. This requires the T option on the meetme application

**Data Sample**:  
  
Event: MeetmeStopTalking  
Privilege: call,all  
Channel: SIP/200-ABC1  
Uniqueid: 1234567890.1  
Meetme: 400  
Usernum: 2

'MeetmeTalking' Event

**Description**:  
[derived from app\_meetme.c]  
  
**Notes**:

1. This requires the T option on the meetme application

**Data Sample**:  
  
Event: MeetmeTalking  
Privilege: call,all  
Channel: SIP/200-ABC1  
Uniqueid: 1234567890.1  
Meetme: 400  
Usernum: 2

'MessageWaiting' Event

**Description**:  
[derived from app\_voicemail.c]  
  
**Data Sample**:  
Event: MessageWaiting  
Mailbox: <mailbox>@<context>  
Waiting: <count>  
New: <number>  
Old: <number>  
  
Event: MessageWaiting  
Mailbox: <context>  
Waiting: <count>

'Newcallerid' Event

**Description**:  
[derived from channel.c]  
  
**Data Sample**:  
Event: Newcallerid  
Channel: <channel>  
Callerid: <callerid>  
Uniqueid: <uniqueid>

'Newchannel' Event

**Description**:  
[derived from channel.c]  
  
**Data Sample**:  
Event: Newchannel  
Channel: Zap/2-1  
State: Rsrvd  
Callerid: <unknown>  
Uniqueid: 1094154427.11  
  
Event: Newchannel  
Channel: SIP/101-3f3f  
State: Ring  
Callerid: 101  
Uniqueid: 1094154427.10

'Newexten' Event

**Description**:

1. Fired whenever a pbx function (such as execution of dialplan) occurs

**Data Sample**:  
Event: Newexten  
Channel: SIP/101-00c7  
Context: macro-ext  
Extension: s  
Priority: 3  
Application: Goto  
AppData: s-BUSY  
Uniqueid: 1094154321.8  
  
Event: Newexten  
Channel: SIP/101-3f3f  
Context: local\_extensions  
Extension: 917070  
Priority: 1  
Application: AGI  
AppData: /etc/asterisk/agi/ks\_doorman\_pickup.py|channel\_up  
Uniqueid: 1094154427.10  
  
Event: Newexten  
Channel: SIP/101-3f3f  
Context: local\_extensions  
Extension: 917070  
Priority: 2  
Application: Dial  
AppData: Zap/G1/17070  
Uniqueid: 1094154427.10

'ParkedCall' Event

**Description**:  
[derived from res\_features.c]  
  
**Data Sample**:  
Event: ParkedCall  
Exten: <parkexten>  
Channel: <channel>  
From: <from>  
Timeout: <timeout>  
CallerID: <callerid>

'Rename' Event

**Description**:  
[derived from channel.c: channel 'rename' event]  
  
**Data Sample**:  
Event: Rename  
Oldname: <oldname>  
Newname: <newname>  
Uniqueid: <uniqueid>

'SetCDRUserField' Event

**Description**:  
[derived from app\_setcdruserfield.c]  
  
**Data Sample**:

'Unlink' Event

**Description**:

1. Fired when a link between two voice channels is discontinued, for example, just before call completion.

**Notes**:

1. Several *Unlink* events may be seen for a single call. This can occur when Asterisk fails to setup a *native bridge* for the call. As far as I can tell, this is when Asterisk must sit between two telephones and perform CODEC conversion on their behalf.

**Data Sample**:  
  
Event: Unlink  
Channel1: SIP/101-3f3f  
Channel2: Zap/2-1  
Uniqueid1: 1094154427.10  
Uniqueid2: 1094154427.11  
Note: in current version it replaced by Bridge event, example is  
Channel2: SIP/1-1.1.1.1-0002ef99  
Bridgestate: Unlink  
Event: Bridge  
Privilege: call,all  
Uniqueid2: 1393028418.530941  
Channel1: SIP/peer-local-0002ef98,  
Bridgetype: core  
Uniqueid1: 1393028346.530940  
Timestamp: 1393028428.618726  
CallerID1: 323187134981  
CallerID2: 323187134981

'UnParkedCall' Event

**Description**:  
[derived from res\_features.c]  
  
**Data Sample**:

**Log Status Events**

**System Status Events**

'Alarm' Event:

**Description**:  
[derived from chan\_zap.c]  
  
**Data Sample**:  
Event: Alarm  
Alarm: <(Red|Yellow|Blue|No|Unknown) Alarm|Recovering|Loopback|Not Open|None>  
Channel: <channel>

'AlarmClear' Event:

**Description**:  
[derived from chan\_zap.c]  
  
**Data Sample**:  
Event: AlarmClear  
Channel: <channel>

'DNDState' Event:

**Description**:  
[derived from chan\_dahdi.c]  
  
**Data Sample**:  
Event: DNDState  
Channel: Zap/1  
Status: <enabled|disabled>

'LogChannel' Event

**Description**:  
[derived from logger.c]  
  
**Data Sample**:  
Event: LogChannel  
Channel: /var/log/asterisk/messages  
Enabled: Yes  
  
Event: LogChannel  
Channel: /var/log/asterisk/messages  
Enabled: No  
Reason: 13 - Permission denied

'PeerStatus' Event

**Description**:

1. Fired when a peer registers/unregisters with Asterisk

[derived from chan\_sip.c, chan\_iax2.c]  
  
**Data Sample**:  
Event: PeerStatus  
Peer: SIP/2005  
PeerStatus: Registered  
  
Event: PeerStatus  
Peer: SIP/2005  
PeerStatus: Unregistered  
*Cause: Expired*  
  
Event: PeerStatus  
Peer: IAX2/2007  
PeerStatus: <Lagged|Reachable|Unreachable>  
Time: 1000

'Registry' Event

**Description**:

1. Fired when Asterisk registers with a peer

[derived from chan\_sip.c, chan\_iax2.c]  
  
**Notes**:  
For an entry like:  
register => username:password:authname@sip.domain:port/local\_contact  
*Domain* would reflect the value of *sip.domain*  
  
**Data Sample**:  
Event: Registry  
Channel: SIP  
Domain: *sip.domain*  
Status: Registered

'Reload' Event

**Description**:

1. Fired when the "RELOAD" console command is executed.

[derived from manager.c]  
  
**Data Sample**:  
Event: Reload  
Message: Reload Requested

'Shutdown' Event

**Description**:  
[derived from asterisk.c]  
  
**Data Sample**:  
Event: Shutdown  
Shutdown: <Uncleanly|Cleanly>  
Restart: <True|False>

**User Status Events**

'UserEvent' Event

**Description**:  
[derived from app\_userevent.c]  
  
**Data Sample**:  
Event: <event>  
Channel: <channel>  
Uniqueid: <uniqueid>  
  
Event: <event>  
Channel: <channel>  
Uniqueid: <uniqueid>  
<body>

**Verbose Status Events**

**Unformatted and Undocumented**

Newstate:

Event: Newstate

Channel: Zap/2-1

State: Dialing

Callerid: 101

Uniqueid: 1094154427.11

Event: Newstate

Channel: Zap/2-1

State: Up

Callerid: 101

Uniqueid: 1094154427.11

ParkedCallsComplete:

[sent following an Action: ParkedCalls]

Event: ParkedCallsComplete

QueueParams:

[sent following an Action: Queues]

Event: QueueParams

Queue: sales

Max: 0

Calls: 0

Holdtime: 0

Completed: 0

Abandoned: 0

ServiceLevel: 0

ServicelevelPerf: 0.0

QueueMember:

[sent following an Action: Queues if a queue has members]

Event: QueueMember

Queue: sales

Location: SIP/101

Membership: dynamic

Penalty: 0

CallsTaken: 0

LastCall: 0

QueueStatusEnd:

[sent following an Action: Queues to signify end of output]

Event: QueueStatusEnd

Status:

Event: Status

Channel: Zap/2-1

CallerID: 101

Account:

State: Up

Link: SIP/101-5cf0

Uniqueid: 1094166088.26

Event: Status

Channel: SIP/101-5cf0

CallerID: 101

Account:

State: Up

Context: local\_extensions

Extension: 917070

Priority: 2

Seconds: 11

Link: Zap/2-1

Uniqueid: 1094166088.25

StatusComplete:

[sent on end of Status events after Action: status]

Event: StatusComplete

ZapShowChannels:

[sent on Action: ZapShowChannels]

Event: ZapShowChannels

Channel: 2

Signalling: FXS Kewlstart

Context: pstn\_menu

Alarm: No Alarm

ZapShowChannelsComplete:

[send on Action: ZapShowChannels end]

Event: ZapShowChannelsComplete.

# Asterisk Manager API Action List Commands:

Action: ListCommands  
Parameters: ActionID  
  
As of Asterisk ver. 1.0.9

Response: Success  
ActionID: SIP/x7062618529-99a0  
AbsoluteTimeout: Set Absolute Timeout  
ChangeMonitor: Change monitoring filename of a channel  
Command: Execute Command  
Events: Contol Event Flow  
ExtensionState: Check Extension Status  
Getvar: Gets a Channel Variable  
Hangup: Hangup Channel  
IAXpeers: List IAX Peers  
ListCommands: List available manager commands  
Logoff: Logoff Manager  
MailboxCount: Check Mailbox Message Count  
MailboxStatus: Check Mailbox  
Monitor: Monitor a channel  
Originate: Originate Call  
ParkedCalls: List parked calls  
Ping: Ping  
QueueAdd: Add interface to queue.  
QueueRemove: Remove interface from queue.  
Queues: Queues  
QueueStatus: Queue Status  
Redirect: Redirect  
SetCDRUserField: Set the CDR UserField  
Setvar: Set Channel Variable  
Status: Status  
StopMonitor: Stop monitoring a channel  
ZapDialOffhook: Dial over Zap channel while offhook  
ZapDNDoff: Toggle Zap channel Do Not Disturb status OFF  
ZapDNDon: Toggle Zap channel Do Not Disturb status ON  
ZapHangup: Hangup Zap Channel  
ZapShowChannels: Show status zapata channels  
ZapTransfer: Transfer Zap Channel