CALLMaster VOIP Configuration and Settings

CALLMaster IP supports both the H.323 and SIP VOIP protocols. In addition it can be set to use eight audio coder formats, three DTMF formats and T.38 Fax. By default CALLMaster will run in standalone mode and listen on port 5060 for incoming VOIP calls. CALLMaster can also be configured to register with a Gateway or Registrar.

There are two sets of VOIP parameters that need to be configured for proper CALLMaster operation. These parameters are contained in the CMVOIP.cfg file which is located in the \Program Files\CALLMaster\Data\ folder. A sample listing of the file's content is listed at the end of this document.

The First Set of Parameters is the CALLMaster Registration Parameters.
These need to be set when CALLMaster will work in conjunction with a Gateway or Registrar. These parameters, for example user name and password, basically allow CALLMaster to pass login information to the Gateway or Registrar server. A Gateway provides the interface for calls that go between the PSTN - Public Switched Telephone Network and VOIP - Voice Over Internet Protocol. A Registrar is a server that provides directory and address lookup services for VOIP devices, usually in a corporate environment.

If CALLMaster is to use a Gateway for outbound calls but does not need to register with a Gateway for inbound calls then set the GWAcctName to NOLOGIN and the GWIP to <IP address of the outbound Gateway>. If CALLMaster will not register with a Gateway or Registrar for either inbound or outbound calls then set the GWIP to 0.0.0.0.

The parameters that need to be set for Gateway or Registrar registration are:

Parameter	Value	Description
GWAcctName	= <username>	login to VOIP GateWay server; set to NOLOGIN to use a Gateway without registering
GWAcctPW	= <password>	password for login to VOIP GateWay server
GWAcctRealm	= <realm>	Realm of VOIP GateWay, provided by GateWay administrator
GWIP	= <IP address>	Dot notation of VOIP GateWay server with optional port number e.g. 206.80.70.54:5060
LocalAlias	= <IP address>	Dot notation of CALLMaster

The Second Set of Parameters is for Call Setup and Audio Coder.
The parameters that need to be set for call setup are:

Parameter	Value	Desription
IP Protocol	= H323|SIP|both	The VOIP Signaling Protocol, possible options: H323, SIP, both
DTMFmode	= OutOfBand|inband|rfc2833	The Touch-Tone signaling mode, possible options: OutOfBand, inband, rfc2833

The "IP Protocol" parameter set the VOIP Protocol supported for incoming calls. Valid settings are: H323, SIP or both.
The "DTMF mode" determines the way Touch Tone signals are supported. Valid settings are: H323, SIP or both.

The parameters for audio coder are:
For each Audio Coder used both Tx - transmit and Rx - Receive values must be set.
Audio Coder settings:

TxType	g711Mulaw|g711Mulaw|gsm|gsmEFR|g723_5_3k|g723_6_3k|g729a|g729ab|t38UDPFax	- Coder algorithm
TxFramesPerPkt	20 | 30	- Frames per packet
TxVAD	0 | 1	- Voice activation Detection
RxType	= g711Mulaw|g711Mulaw|gsm|gsmEFR|g723_5_3k|g723_6_3k|g729a|g729ab|t38UDPFax	- Coder algorithm
RxFramesPerPkt	= 20 | 30	- Frames per packet
RxVAD	= 0 | 1	- Voice activation Detection

Audio coders convert audio signals like a human voice into digital format for transmission over the internet.

Coder setting	Description
Type	- Represents the algorithms and data rates. The higher the data rate the better the sound quality.
FramesPerPkt	- Determines how many bytes of data or frames are in each data packet sent.
VAD	- Voice Activation Detection enables ports to send data only when someone is talking and

These three settings: Type, Vad and FramesPerPacket for both Transmitting and Receiving, are configured for each Coder supported by including them in capability blocks as demonstrated below.

You should only configure the audio coder that you will support since each coder definition consumes resources (memory) from your system.

For each coder the system will support add a block of parameters to the Capability section of the CMVOIP.cfg file. For example if you wanted to support a g.711 coder with no VAD for both sending and receiving you would add the following bock of parameters in the Capability section of the file.

```
  Capability
  {
   TxType = g711Mulaw
    TxFramesPerPkt = 20
    TxVAD = 0
    RxType = g711Mulaw
    RxFramesPerPkt = 20
    RxVAD = 0
  }
```

QoS Monitoring and Alarm Settings

CALLMaster also provides for monitoring/alarms for a specific QoS - Quality of Service as determined by the amount of Lost Data Packets and the Delay of Data Packets. The default settings are useful in most environments and will send log data to the CALLMaster log file if the QoS falls below these defined values. If the QoS is below the desired level then the Audio signal will be choppy or non-existent. Poor QoS will need to be addressed by the network administrator and these alarms are for monitoring purpose only. These parameters are in the MediaAlarmLostPackets and MediaAlarmJitter section of the CMVOIP.cfg file.

See the sample file below for reference of the contents of the CMVOIP.cfg file.

Lines in the CMVOIP.cfg file that start with a '#' character are ignored as comments.

```
#################################################################################################################
#
#           CALLMaster parameters for registering with a Gateway or Registrar
#
# GWAcctName   - <username> GateWay account username for login to VOIP Gateway server
# GWAcctPW     - <password> GateWay account password for login to VOIP Gateway server
# GWAcctRealm  - <realm> of VOIP Gateway
# GWIP         - <IP address> Dot notation of VOIP Gateway server with optional port number e.g. 206.80.70.54:5060. Set to
0.0.0.0 for no registration
# LocalAlias   - <IP address> Dot notation of CALLMaster
#

GWAcctName = AccountName
GWAcctPW = password
GWAcctRealm = asterisk
GWIP = 192.168.70.54:5060
GWLocalAlias = 192.168.5.37


#################################################################################################################
# HMP parameters
#
# IP Protocol:
#   The IP Protocol used for opening the IP Line devices, possible options: H323, SIP, both
#
# DTMFmode:
#           possible options: OutOfBand, inband, rfc2833
#
# Coder Types:
#   g711Alaw
#   g711Mulaw
#   gsm
#   gsmEFR
#   g723_5_3k
#   g723_6_3k
#   g729a
#   g729ab
#   t38UDPFax
#
#   Note: if you want to run the demo with coder g729 use:
#   g729a for running with VAD disable
#   and 729ab for running with VAD enable
#
# Caution:
#    If capability is g711Alaw /Mulaw  ==> FramesPerPkt = 10,20,30.
#                          G711 frame per packet defines the packet size in milliseconds
#    If capability is g723_5_3k / 6_3k ==> FramesPerPkt = 1, 2, 3 .
#                         FrameSize isn't needed, default= 30ms.
#    If capability is gsm          ==> FramesPerPkt = 1, 2, 3 .
#                          FrameSize isn't needed, default= 20ms.
#    If capability is gsmEFR         ==> FramesPerPkt = 1, 2, 3 .
#                          FrameSize isn't needed, default= 20ms.
#    If capability is g729a          ==> FramesPerPkt = 3, 4 .
#                          FrameSize isn't needed, default= 10ms.
#                          VAD disable, the VAD parameter is ignored
#    If capability is g729ab          ==>FramesPerPkt = 3, 4 .
#                          FrameSize isn't needed, default= 10ms.
#                          VAD enable, the VAD parameter is ignored
#
##################################################################################################

ipProtocolName = SIP
DTMFmode = rfc2833
Channel = 1-120
{
  Capability
  {
    TxType                  = g711Mulaw
    TxFramesPerPkt          = 30
    TxVAD                   = 0
    RxType                  = g711Mulaw
    RxFramesPerPkt          = 30
    RxVAD                   = 0

# TxType = t38UDPFax
  }

  MediaAlarmLostPackets
  {
     Threshold              = 20              # Threshold value
     DebounceOn             = 10000           # Threshold debounce ON
     DebounceOff            = 10000           # Threshold debounce OFF
     Interval               = 1000            # Threshold Time Interval (ms)
     PercentSuccess         = 60              # Threshold Success Percent
     PercentFail            = 40              # Threshold Fail Percent
  }

  MediaAlarmJitter
  {
     Threshold              = 60              # Threshold value
     DebounceOn             = 20000           # Threshold debounce ON
     DebounceOff            = 60000           # Threshold debounce OFF
     Interval               = 5000            # Threshold Time Interval (ms)
     PercentSuccess         = 60              # Threshold Success Percent
     PercentFail            = 40              # Threshold Fail Percent
  }

# MediaAlarmResetAlarmState       = 0

}
```