VoiceGenie 7.2

MRCP Proxy

System Reference Guide

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel, is 100% focused on software for call centers. Genesys recognizes that better interactions drive better business and build company reputations. Customer service solutions from Genesys deliver on this promise for Global 2000 enterprises, government organizations, and telecommunications service providers across 80 countries, directing more than 100 million customer interactions every day. Sophisticated routing and reporting across voice, e-mail, and Web channels ensure that customers are quickly connected to the best available resource—the first time. Genesys offers solutions for customer service, help desks, order desks, collections, outbound telesales and service, and workforce management. Visit www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library CD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North and Latin America	+888-369-5555 or +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-118-974-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-5649-6871	support@genesyslab.co.jp

Prior to contacting technical support, please refer to the Genesys Technical Support Guide for complete contact information and procedures.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the Genesys 7 Licensing Guide.

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 09-2006

Table of Contents

Table of Contents

Chapter

1 Introduction

This document contains all reference information for the MRCP Proxy (also known as the SRM Proxy), including configuration parameters, logging/metric entries, directory structures and alarm codes (and their associated recommended responses).

Chapter

2 Directory Structure

This section describes the directory structure for the MRCP Proxy running on Linux; for the Windows version, simply replace the root install directory of the Linux version (`/usr/local/srm-proxy/`) with this root install directory (`C:\VoiceGenie\srm-proxy\`).

The following directories are used in for the MRCP Proxy:

Directory	Description
/usr/local/srm-proxy/bin	This directory contains the executable binaries for the MRCP Proxy
/usr/local/srm-proxy/config	This directory contains the configuration files to be used for the MRCP Proxy
/usr/local/srm-proxy/logs	This directory is where the logs from the MRCP Proxy

Chapter

3 Configuration Parameters

3.1 MRCP Proxy Resource Provisioning entry parameters

The following table lists all possible resource provisioning entry parameter names and explains their usages:

Parameter Name	Description
vrmproxy.proxy_uri	This is the RTSP uri for a MRCP client to request a specific resource. The MRCP proxy groups resources that have the same value of this parameter into the same resource pool Example: `rtsp://10.0.0.72/nuance_recognizer`
vrmproxy.resource_uri	This is the RTSP uri for MRCP proxy to establish a session with a MRCP server The value relies on the hosted resource configuration Example: `rtsp://media.server.com/recognizer`
vrmproxy.allocation_algorithm	The session allocation algorithm. This can be set to `ROUND_ROBIN` or `LEAST_USED`. Default: `ROUND_ROBIN`
vrmproxy.resource_type	Media Resource type. This can be set to `ASR` or `TTS`.
vrmproxy.max_sessions	The maximum number of sessions can be allocated to this resource. Default: `200`

Parameter Name	Description
vrmproxy.routing_mode	The media routing mode setting. This can be set to `REGULAR` or `CAPABILITY`. When set to `CAPABILITY`, the MRCP client must be able to support the MRCP extension specified in section 2, "The MRCP Protocol." Currently language is the only capability. Default: `REGULAR`
vrmproxy.languages	The languages supported by the given media resource. This parameter only applies when routing mode is set to `CAPABILITY` Default: `en-us`
vrmproxy.server_address	The remote media server IP address
vrmproxy.server_port	The remote media server Port number
vrmproxy.protocol	The protocol mode should be set to `MRCP1.0`
vrmproxy.ping_interval	The interval (in ms) that the VRMProxy will ping this media resource. If it is not set or it is set to `0`, there will be no ping message at all.

3.2 MRCP/SRM Proxy Configuration

Parameter	Description
srmproxy configuration	
vrmproxy.mrcpv1.protocol_server_port	Server port of `MRCPv1` protocol module Default: `11000`
vrmproxy.register_port	Port for MRCP servers to register Default: `10000` **Note:** MRCP server register feature is not supported in VoiceGenie 7

Parameter	Description
vrmproxy.stream_relay	Relay media stream or not. When this parameter is set to `true`, the MRCP proxy will redirect the audio stream between a MRCP client and server. When it is set to `false`, the audio stream will commute directly between a MRCP client and server. Possible values: `true`, `false` Default: `false`
vrmproxy.clients_allowed	IP addresses or ranges of clients allowed to connect to this proxy, e.g. `10.0.0.1-10.0.0.255, 10.0.1.2` If not set, there is no limitation on the client that can connect to the MRCP Proxy.
vrmproxy.timeout.open_session	Timeout of `Open-Session` in milliseconds Default: `10000`
vrmproxy.timeout.close_session	Timeout of `Close-Session` in milliseconds Default: `10000`
vrmproxy.timeout.set_params	Timeout of `SET-PARAMS` in milliseconds Default: `10000`
vrmproxy.timeout.get_params	Timeout of `GET-PARAMS` in milliseconds Default: `10000`
vrmproxy.timeout.stop	Timeout of `STOP` in milliseconds Default: `10000`
vrmproxy.timeout.define_grammar	Timeout of `DEFINE-GRAMMAR` in milliseconds Default: `10000`
vrmproxy.timeout.get_result	Timeout of `GET-RESULT` in milliseconds Default: `10000`
vrmproxy.timeout.recog_start_timers	Timeout of `RECOGNITION-START-TIMERS` in milliseconds Default: `10000`
vrmproxy.timeout.speak	Timeout of `SPEAK` in milliseconds Default: `10000`
vrmproxy.timeout.recognize	Timeout of `RECOGNIZE` in milliseconds Default: `10000`

Parameter	Description
vrmproxy.timeout.control	Timeout of `CONTROL` in milliseconds Default: `10000`
vrmproxy.timeout.pause	Timeout of `PAUSE` in milliseconds Default: `10000`
vrmproxy.timeout.resume	Timeout of `RESUME` in milliseconds Default: `10000`
vrmproxy.timeout.barge_in_occurred	Timeout of `BARGE-IN-OCCURRED` in milliseconds Default: `10000`
vrmproxy.timeout.waiting_for_server	Timeout in milliseconds for a client to wait for server Default: `10000`
vrmproxy.timeout.back_in_service	Timeout in milliseconds for a server to be put back in service after it encounters errors Default: `10000`
vrmproxy.timeout.get_server_info	Timeout in milliseconds to get a response for `Get-Server-Info` request (Ping) Default: `10000`
vrmproxy.timeout.reconnect_interval	Interval in milliseconds to reconnect to a server Default: `10000`
vrmproxy.timeout.clean_loop	Interval to clean idle sessions that exceed `vrmproxy.timeout.max_idle` Default: `60000`
vrmproxy.timeout.max_idle	Max session idle time in milliseconds. A session will be terminated after this time. Default: `180000`

Parameter	Description
vrmproxy.metrics_direction	Indicated the direction of the MRCP messages to be logged into metrics file. Possible values: `IN, OUT, BOTH, NONE` • `IN:` only log the MRCP messages that are received from MRCP clients and servers • `OUT:` only log the MRCP messages that are sent to MRCP clients and servers • `BOTH:` log MRCP messages that are received and sent from/to MRCP clients and servers • `NONE:` Not log any MRCP messages to `metricsfile` Default: `IN`
vrmproxy.latency_tracking_level	It is a bit mask to specify the level of the latency data that should be tracked and logged into `metricsfile.` The levels are defined below. • `OPEN_SESSION: 0x00000001` • `CLOSE_SESSION: 0x00000002` • `SET_PARAMS: 0x00000004` • `GET_PARAMS: 0x00000008` • `STOP: 0x00000010` • `SPEAK: 0x00000020` • `SPEAK_COMPLETE: 0x00000040` • `DEFINE_GRAMMAR: 0x00000080` • `RECOGNIZE: 0x00000100` • `RECOGNITION_COMPLETE: 0x00000200` • `GET_RESULT: 0x00000400` • `GET_SERVER_INFO: 0x00000800` • `RECOGNITION_START_TIMERS: 0x00001000` • `START_OF_SPEECH: 0x00002000` • `BARGE_IN_OCCURRED: 0x00004000` • `PAUSE: 0x00008000` • `RESUME: 0x00010000` • `CONTROL: 0x00020000` • `SPEECH_MARKER: 0x00040000` • `REDIRECT_MEDIA_STREAM: 0x00080000` Default: `0x00000000`

Parameter	Description
vrmproxy.latency_calculate_method	Latency calculation method Possible values: `number_of_messages`, `period_of_time` • `Number_of_message`: The latency data (minimum, maximum and average) is calculated based on the last number of messages specified by `vrmproxy.latency_calculate_base` • `period_of_time`: The latency data (minimum, maximum and average) is calculated based on the latest period of time specified by `vrmproxy.latency_calculate_base` Default: `number_of_messages`
vrmproxy.latency_calculate_base	It specifies the base to calculate latency data. If the calculation method is `number_of_message` in `vrmproxy.latency_calculate_method`, it specifies the number of messages that the max/min/average latency data should be calculated. If the calculation method is `period_of_time` in `vrmproxy.latency_calculate_method`, it specifies the time (in sec) that the max/min/average latency data should be calculated. Default is zero, which means no latency calculation. Default: `0`
vrmproxy.status_stale_timeout	Specifies the timeout in milliseconds to refresh SRM Proxy status. Default: `5000`

3.3 MRCP Stack Configuration

Parameter	Description
MRCP Stack Configuration	
stack.trace.debug	Enable and disable `mrcp stack trace`. Possible values: `false`, `true` Default: `true`
stack.connection.timeout	Connection timeout in milliseconds Default: `20000`

Parameter	Description
stack.socket.onesend	Send a message once as a whole or not Possible values: `true, false` Default: `true`
stack.transport.type	MRCP transport type Default: `RTSP`

3.4 Internal Media Transport

Parameter	Description
Internal Media Transport	
mtinternal.rtp_min_port	The minimum port range for RTP sockets in `MTInternal` Default: `20000`
mtinternal.rtp_max_port	The maximum port range for RTP sockets in `MTInternal` Default: `30000`
mtinternal.max_sessions	Defines the maximum `MTInternal` sessions Default: `400`
mtinternal.transmit_interval	Defines a constant transmission interval in milliseconds. If set to `0,` packets will be sent as soon as data arrives. Default: `0`
mtinternal.transmit_rate	When `mtinternal.transmit_interval` is non-zero, this parameter specifies the maximum number of packets to be sent for each transmission interval. Set to `0` to turn off this restriction. Default: `0`
mtinternal.transmit_min_size	Defines the minimum data size in bytes that can be sent. Note that this number is applied to all codecs with fixed frame size. It will be rounded down to the nearest multiple of the codec frame size. This parameter will be disabled when variable frame size codec is used. Set to `-1` to disable the limit. Default: `160`

Parameter	Description
mtinternal.transmit_max_size	Defines the maximum data size in bytes that can be sent. Note that this number is applied to all codecs with fixed frame size. It will be rounded down to the nearest multiple of the codec frame size. This parameter will be disabled when variable frame size codec is used. Set to `-1` to disable the limit. Default: `160`
mtinternal.receive_min_size	Defines the minimum packet sample size that will be notified to the receiver. Note that this number is applied to all codecs with fixed frame size. It will be rounded down to the nearest multiple of the codec frame size. This parameter will be disabled when variable frame size codec is used. Set to `-1` to disable the limit. Default: `-1`
mtinternal.receive_max_size	Defines the maximum packet sample size that will be notified to the receiver. Note that this number is applied to all codecs with fixed frame size. It will be rounded down to the nearest multiple of the codec frame size. This parameter will be disabled when variable frame size codec is used. Set to `-1` to disable the limit. Default: `-1`
mtinternal.jitter_log	Defines the logging period in terms of number of received packets. If less than `1`, Jitter logging is turned off. Jitter logging will be disabled if variable frame size codec is used for received packets. Default: `0`
mtinternal.transmit_rate_alarm	If greater than `0`, a minor alarm is generated if the transmission rate of outgoing packets is slower the real time by the specified delay in milliseconds. This alarm will be disabled if variable frame size codec is used for transmitted packets. Default: `500`
mtinternal.receive_rate_alarm	If greater than `0`, a minor alarm is generated if the transmission rate of incoming packets is slower the real time by the specified delay in milliseconds. This alarm will be disabled if variable frame size codec is used for received packets. Default: `500`
mtinternal.transmit_savedata	If specified, utterance is saved under the directory specified by this parameter.

Parameter	Description
mtinternal.receive_savedata	If specified, received data is saved under the directory specified by this parameter.
mtinternal.max_concurrent_savedata	If specified as an integer *n*, and `mtinternal.transmit_savedata` or `mtinternal.receive_savedata` is enabled, then only a maximum of *n* concurrent files will be open for writing data. Default value is `-1`, which would place no limit. Default: `-1`

3.5 Logging Configuration Settings

Parameter	Description
Logging Configuration Settings	
cmp.log_file	Path of `srmproxy` log file Default (Linux/Solaris): `/usr/local/srm-proxy/logs/CMP.log.srmproxy` Default (Windows): `C:\VoiceGenie\srm-proxy\logs\CMP.log.srmproxy`
cmp.size_option	Rollover all log files by size or by time Possible values: `FALSE, TRUE` Default: `FALSE`
cmp.rollover_size	The size limit, in MB, for rollover when rolling over by size Default: `100`
cmp.num_rollover_files	The number of files to roll through before they are overwritten when rolling over by size Default: `5`
cmp.rollover_mins	The interval of time, in minutes, between rollover when rolling over by time Default: `1440`
cmp.rollover_time	The time at which the log files are rolled over when rolling over by time Default: `4:00`

Parameter	Description
Email parameters	
cmp.email	If the `EMAIL` sink is specified, the email address will be used Default: `name@domain.com`

Parameter	Description
Logging Service parameters	
cmp.log_sinks	Sinks that will be used by this component, possible sinks are: `FILE, UPSTREAM, SYSLOG, SNMP, EMAIL` Default: `FILE|UPSTREAM`
cmp.proxy	This specifies call manager proxy. Default: `localhost`
cmp.proxy_port	The server port of this CMP Proxy, other VoiceGenie software components connect to this port Default: `8700`
cmp.trace_flag	Determines if logs at level `log_5` (tracing/debugging) should be logged Possible values: `FALSE, TRUE` Default: `FALSE`
cmp.pid_option	Appends PID of the process to the name of the trace file so that they are not overwritten when the process restarts Possible values: `FALSE, TRUE` Default: `FALSE`
cmp.metrics	log mask for metrics data Default: `0|1`

cmp.log_0	Log mask for data logged at log level `0`
	Default:
	`1111111111111111111111111111111111111111111111`
	`1111111111111111111111111111111111111111111111`
	`1111111111111111111111111111111111111111111111`
	`11|1111111111111111111111111111111111111111111`
	`1111111111111111111111111111111111111111111111`
	`1111111111111111111111111111111111111111111111`
	`11111`

Parameter	**Description**
cmp.log_1	Log mask for data logged at log level 1 Default: 1111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 11|1111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 11111|0000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000 00000000|1111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 11111111111|1111111111111111111111111111111111 1111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 11111111111111
cmp.log_2	Log mask for data logged at log level 2 Default: 1111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 11|1111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 11111
cmp.log_3	Log mask for data logged at log level 3 Default: 1111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 1111111111111111111111111111111111111111111111 11|0000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000 00000

Parameter	Description
cmp.log_4	Log mask for data logged at log level 4 Default: `11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111|00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000`
cmp.log_5	Log mask for data logged at log level 5 Default: `11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111|00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000`

3.6 Guaranteed Logs Parameters

Parameter	Description
Guaranteed Logs parameters	
cmp.guaranteed_logs_to_file	Specify if logs that are guaranteed to be sent upstream should be logged to a temp file Possible values: `FALSE`, `TRUE` Default: `true`
cmp.unsent_log_file	Specify the name of the temp log file to log to if `cmp.guaranteed_logs_to_file` Default (Linux/Solaris): `/usr/local/srm-proxy/logs/guaranteed.log.srmproxy` Default (Windows): `C:\VoiceGenie\srm-proxy\logs\guaranteed.log.srmproxy`

Chapter

4 Metrics/Logging

Label	Description
barge_in_occurred OUT Logged by: SRMP Level: 0x00004000 && (vrmproxy.metrics_direction =OUT || BOTH)	**BARGE-IN-OCCURRED Request Sent to A Resource** This is logged when a `BARGE-IN-OCCURRED` request is sent to the server. The format is: `barge_in_occurred OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>` • `<server>`: It is a label. • `<SERVER_ID>`: It is an integer representing a MRCP resource id • `<message>`: It is a label. • `<MESSAGE_ID>`: It is an integer representing a RTSP message id Example:
barge_in_occurred IN Logged by: SRMP Level: 0x00004000 && (vrmproxy.metrics_direction =IN || BOTH)	**BARGE-IN-OCCURRED Request Received** This is logged when a `BARGE-IN-OCCURRED` request for a MRCP session is received. The format is: `barge_in_occurred IN <message>|<MESSAGE_ID>` • `<message>`: It is a label. • `<MESSAGE_ID>`: It is an integer representing a RTSP message id Example:

Label	Description
barge_in_occurred_resp IN Logged by: SRMP Level: 0x00004000 && (vrmproxy.metrics_direction =IN || BOTH)	**BARGE-IN-OCCURRED Response Received From A MRCP Server** This is logged when a `BARGE-IN-OCCURRED` response is received from a MRCP server. The format is: `barge_in_occurred_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>` • `<server>`: It is a label. • `<SERVER_ID>`: It is an integer representing a MRCP resource id • `<message>`: It is a label. • `<MESSAGE_ID>`: It is an integer representing a RTSP message id • `<status>`: It is a label. • `<STATUS>`: RTSP status code of the message Example:
barge_in_occurred_resp OUT Logged by: SRMP Level: 0x00004000 && (vrmproxy.metrics_direction =OUT || BOTH)	**BARGE-IN-OCCURRED Response Sent to A MRCP Client** This is logged when a `BARGE-IN-OCCURRED` response is sent to the requesting MRCP client. The format is: `barge_in_occurred_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>` • `<message>`: It is a label. • `<MESSAGE_ID>`: It is an integer representing a RTSP message id • `<status>`: It is a label. • `<STATUS>`: RTSP status code of the message • `<State>`: It is a label. • `<STATE>`: RTSP state of the message Example:
close IN Logged by: SRMP Level: 0x00000002 && (vrmproxy.metrics_direction =IN || BOTH)	**TEARDOWN Request Received** This is logged when a `TEARDOWN` request for a MRCP session is received. Example: `close IN`

Label	Description
close OUT Logged by: SRMP Level: 0x00000002 && (vrmproxy.metrics_direction =OUT || BOTH)	**TEARDOWN Request Sent to A Resource** This is logged when a TEARDOWN request for a MRCP session is sent to the resource. The format is: close OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id Example: close OUT - server 382, message 4294967295
close_resp IN Logged by: SRMP Level: 0x00000002 && (vrmproxy.metrics_direction =IN || BOTH)	**TEARDOWN Response Received From A MRCP Server** This is logged when a TEARDOWN response is received from a MRCP server. The format is: close_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message Example: close_resp IN - server 382, status 200

Label	Description

close_resp OUT
Logged by: SRMP
Level: 0x00000002 && (vrmproxy.metrics_direction =OUT || BOTH)
	TEARDOWN Response Sent to A MRCP Client
	This is logged when a `TEARDOWN` response is sent to the requesting MRCP client.
	The format is:
	`close_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>`
	• `<message>:` It is a label.
	• `<MESSAGE_ID>:` It is an integer representing a RTSP message id
	• `<Status>:` It is a label.
	• `<STATUS>:` RTSP status code of the message
	Example:
	`close_resp OUT - message 4294967295, status 200`

control IN Logged by: SRMP Level: 0x00020000 && (vrmproxy.metrics_direction =IN || BOTH)	**CONTROL Request Received** This is logged when a `RESUME` request for a MRCP session is received. The format is: `control IN <message>|<MESSAGE_ID>` • `<message>:` It is a label. • `<MESSAGE_ID>:` It is an integer representing a RTSP message id Example:
control OUT Logged by: SRMP Level: 0x00020000 && (vrmproxy.metrics_direction =OUT || BOTH)	**CONTROL Request Sent to A Resource** This is logged when a `CONTROL` request is sent to the server. The format is: `control OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>` • `<server>:` It is a label. • `<SERVER_ID>:` It is an integer representing a MRCP resource id • `<message>:` It is a label. • `<MESSAGE_ID>:` It is an integer representing a RTSP message id Example:

Label	Description
control_resp IN Logged by: SRMP Level: 0x00020000 && (vrmproxy.metrics_direction =IN || BOTH)	**CONTROL Response Received From A MRCP Server** This is logged when a `CONTROL` response is received from a MRCP server. The format is: `control_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>` • `<server>`: It is a label. • `<SERVER_ID>`: It is an integer representing a MRCP resource id • `<message>`: It is a label. • `<MESSAGE_ID>`: It is an integer representing a RTSP message id • `<Status>`: It is a label. • `<STATUS>`: RTSP status code of the message Example:
control_resp OUT Logged by: SRMP Level: 0x00020000 && (vrmproxy.metrics_direction =OUT || BOTH)	**CONTROL Response Sent to A MRCP Client** This is logged when a `CONTROL` response is sent to the requesting MRCP client. The format is: `control_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>` • `<message>`: It is a label. • `<MESSAGE_ID>`: It is an integer representing a RTSP message id • `<Status>`: It is a label. • `<STATUS>`: RTSP status code of the message • `<State>`: It is a label. • `<STATE>`: RTSP state of the message Example:

Label	Description

define_gram IN
Logged by: SRMP
Level: 0x00000080 && (vrmproxy.metrics_direction =IN || BOTH)
	DEFINE-GRAMMAR Request Received
	This is logged when a DEFINE-GRAMMAR request for a MRCP session is received.
	The format is:
	`define_gram IN <message>|<MESSAGE_ID>`
	• `<message>:` It is a label.
	• `<MESSAGE_ID>:` It is an integer representing a RTSP message id
	Example:
	`define_gram IN - message 14061`

define_gram OUT
Logged by: SRMP
Level: 0x00000080 && (vrmproxy.metrics_direction =OUT || BOTH)
	DEFINE-GRAMMAR Requst Sent to A Resource
	This is logged when a DEFINE-GRAMMAR request is sent to the server.
	The format is:
	`define_gram OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>`
	• `<server>:` It is a label.
	• `<SERVER_ID>:` It is an integer representing a MRCP resource id
	• `<message>:` It is a label.
	• `<MESSAGE_ID>:` It is an integer representing a RTSP message id
	Example:
	`define_gram OUT - server 382, message 14061`

Label	Description
define_gram_resp IN Logged by: SRMP Level: 0x00000080 && (vrmproxy.metrics_direction =IN || BOTH)	**DEFINE-GRAMMAR Response Received From A MRCP Server** This is logged when a DEFINE-GRAMMAR response is received from a MRCP server. The format is: define_gram_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message Example: define_gram_resp IN - server 382, message 14061, status 200
define_gram_resp OUT Logged by: SRMP Level: 0x00000080 && (vrmproxy.metrics_direction =OUT || BOTH)	**DEFINE-GRAMMAR Response Sent to A MRCP Client** This is logged when a DEFINE-GRAMMAR response is sent to the requesting MRCP client. The format is: define_gram_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE> • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message • <State>: It is a label. • <STATE>: RTSP state of the message Example: define_gram_resp OUT - message 14088, status 200, state 0

Label	Description

get_params IN
Logged by: SRMP
Level: 0x00000008 && (vrmproxy.metrics_direction =IN || BOTH)
	GET-PARAMS Request Received
	This is logged when a GET-PARAMS request for a MRCP session is received.
	The format is:
	`get_params IN <message>|<MESSAGE_ID>`
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	Example:
	`get_params IN - message 14055`

get_params OUT
Logged by: SRMP
Level: 0x00000008 && (vrmproxy.metrics_direction =OUT || BOTH)
	GET-PARAMS Request Sent to A Resource
	This is logged when a GET-PARAMS request is sent to the server.
	The format is:
	`get_params OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>`
	• `<server>`: It is a label.
	• `<SERVER_ID>`: It is an integer representing a MRCP resource id
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	Example:
	`get_params OUT - server 382, message 14055`

Label	Description

get_params_resp IN
Logged by: SRMP
Level: 0x00000008 && (vrmproxy.metrics_direction =IN || BOTH)
	GET-PARAMS Response Received From A MRCP Server
	This is logged when a GET-PARAMS response is received from a MRCP server.
	The format is:
	`get_params_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>`
	• <server>: It is a label.
	• <SERVER_ID>: It is an integer representing a MRCP resource id
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	• <Status>: It is a label.
	• <STATUS>: RTSP status code of the message
	Example:
	`get_params_resp IN - server 382, message 14055, status 200`

get_params_resp OUT
Logged by: SRMP
Level: 0x00000008 && (vrmproxy.metrics_direction =OUT || BOTH)
	GET-PARAMS Response Sent to A MRCP Client
	This is logged when a GET-PARAMS response is sent to the requesting MRCP client.
	The format is:
	`get_params_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>`
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	• <Status>: It is a label.
	• <STATUS>: RTSP status code of the message
	• <State>: It is a label.
	• <STATE>: RTSP state of the message
	Example:
	`get_params_resp OUT - message 14057, status 200, state 0`

Label	Description
get_result IN Logged by: SRMP Level: 0x00000400 && (vrmproxy.metrics_direction =IN || BOTH)	**GET-RESULT Request Received** This is logged when a GET-RESULT request for a MRCP session is received. The format is: get_result IN <message>|<MESSAGE_ID> • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id Example:
get_result OUT Logged by: SRMP Level: 0x00000400 && (vrmproxy.metrics_direction =OUT || BOTH)	**GET-RESULT Request Sent to A Resource** This is logged when a GET-RESULT request is sent to the server. The format is: get_result OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id Example:
get_result OUT Logged by: SRMP Level: 0x00001000 && (vrmproxy.metrics_direction =OUT || BOTH)	**RECOGNITION-START-TIMERS Request Sent to A Resource** This is logged when a RECOGNITION-START-TIMERS request is sent to the server. The format is: get_result OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id Example: recog_start_timer OUT - server 382, message 14059

Label	Description
get_result_resp IN Logged by: SRMP Level: 0x00000400 && (vrmproxy.metrics_direction =IN || BOTH)	**GET-RESULT Response Received From A MRCP Server** This is logged when a `GET-RESULT` response is received from a MRCP server. The format is: `get_result_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>` • `<server>`: It is a label. • `<SERVER_ID>`: It is an integer representing a MRCP resource id • `<message>`: It is a label. • `<MESSAGE_ID>`: It is an integer representing a RTSP message id • `<Status>`: It is a label. • `<STATUS>`: RTSP status code of the message Example:

get_result_resp IN
Logged by: SRMP
Level: 0x00000400 && (vrmproxy.metrics_direction =IN || BOTH)
	RECOGNITION-START-TIMERS Response Received From A MRCP Server
	This is logged when a `RECOGNITION-START-TIMERS` response is received from a MRCP server.
	The format is:
	`get_result_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>`
	• `<server>`: It is a label.
	• `<SERVER_ID>`: It is an integer representing a MRCP resource id
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	• `<Status>`: It is a label.
	• `<STATUS>`: RTSP status code of the message
	Example:
	`recog_start_timer_resp IN - server 382, message 14059, status 200`

Label	Description
get_result_resp OUT Logged by: SRMP Level: 0x00000400 && (vrmproxy.metrics_direction =OUT || BOTH)	**GET-RESULT Response Sent to A MRCP Client** This is logged when a `GET-RESULT` response is sent to the requesting MRCP client. The format is: `get_result_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>` • `<message>`: It is a label. • `<MESSAGE_ID>`: It is an integer representing a RTSP message id • `<Status>`: It is a label. • `<STATUS>`: RTSP status code of the message • `<State>`: It is a label. • `<STATE>`: RTSP state of the message Example:

get_result_resp OUT
Logged by: SRMP
Level: 0x00001000 && (vrmproxy.metrics_direction =OUT || BOTH)
	RECOGNITION-START-TIMERS Response Sent to A MRCP Client
	This is logged when a `RECOGNITION-START-TIMERS` response is sent to the requesting MRCP client.
	The format is:
	`get_result_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>`
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	• `<Status>`: It is a label.
	• `<STATUS>`: RTSP status code of the message
	• `<State>`: It is a label.
	• `<STATE>`: RTSP state of the message
	Example:
	`recog_start_timer_resp OUT - message 14059, status 200, state 0`

Label	Description
get_server_info In Logged by: SRMP Level: 0x00000800 && (vrmproxy.metrics_direction =IN || BOTH)	**DESCRIBE Received** This is logged when a DESCRIBE message is received to query a resource information. The format is: get_server_info In <message>|<MESSAGE_ID>|<Status>|<STATUS>|<URI>|<URI_CONTEXT> • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message • <URI>: It is a label. • <URI_CONTEXT>: This is the rtsp request URI coming from a MRCP client Example: message 5989, status 200, URI rtsp://10.0.0.92:4900/media/speechrecognizer
get_server_info OUT Logged by: SRMP Level: 0x00000800 && (vrmproxy.metrics_direction =OUT || BOTH)	**DESCRIBE Request Sent to A Resource** This is logged when a DESCRIBE request is sent to a MRCP server to query a resource. The format is: get_server_info OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<URI>|<URI_CONTEXT> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <URI>: It is a label. • <URI_CONTEXT>: This is the rtsp request URI coming from a MRCP client Example: get_server_info OUT - server 382, message 5988, URI rtsp://10.0.0.70/recognizer

Label	Description

get_server_info_resp OUT
Logged by: SRMP
Level: 0x00000800 && (vrmproxy.metrics_direction =OUT || BOTH)
	DESCRIBE Response Sent to A MRCP Client
	This is logged when a `DESCRIBE` response is sent to the requesting MRCP client.
	The format is:
	`get_server_info_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>`
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	• `<Status>`: It is a label.
	• `<STATUS>`: RTSP status code of the message
	Example:
	`get_server_info_resp OUT - message 14501, status 200`

marker IN Logged by: SRMP Level: 0x00004000 && (vrmproxy.metrics_direction =IN || BOTH)	**SPEECH-MARKER Event Received** This is logged when a `SPEECH-MARKER` Event from MRCP server is received. The format is: `marker IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<request>|<REQUEST_ID>` • `<server>`: It is a label. • `<SERVER_ID>`: It is an integer representing a MRCP resource id • `<message>`: It is a label. • `<MESSAGE_ID>`: It is an integer representing a RTSP message id • `<request>`: It is a label. • `<REQUEST_ID>`: It is an integer representing a MRCP request id Example:

Label	Description
marker OUT Logged by: SRMP Level: 0x00004000 && (vrmproxy.metrics_direction =OUT || BOTH)	**SPEECH-MARKER Event Sent to A Client** This is logged when a SPEECH-MARKER event is sent to the client. The format is: `marker OUT <message>|<MESSAGE_ID>|<request>|<REQUEST_ID>"/>|<State>|<STATE>` • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <request>: It is a label. • <REQUEST_ID>: It is an integer representing a MRCP request id • <State>: It is a label. • <STATE>: RTSP state of the message Example:
marker_resp IN Logged by: SRMP Level: 0x00004000 && (vrmproxy.metrics_direction =IN || BOTH)	**SPEECH-MARKER Response Received From A MRCP Client** This is logged when a SPEECH-MARKER response is received from a MRCP client. The format is: `marker_resp IN <message>|<MESSAGE_ID>|<Status>|<STATUS>` • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message Example:

Label	Description
marker_resp OUT Logged by: SRMP Level: 0x00004000 && (vrmproxy.metrics_direction =OUT || BOTH)	**SPEECH-MARKER Response Sent To A MRCP Server** This is logged when a SPEECH-MARKER response is sent to the MRCP server. The format is: `marker_resp OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>| <State>|<STATE>` • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message Example:

open IN
Logged by: SRMP
Level: 0x00000001 && (vrmproxy.metrics_direction =IN || BOTH)
	SETUP Request Received
	This is logged when a SETUP request for a MRCP session is received.
	The format is:
	`open IN <client>|<CLIENT_ID>|<message>|<MESSAGE_ID>|<URI>|<URI_CONTEXT>`
	• <client>: It is a label.
	• <CLIENT_ID>: It is an integer representing a MRCP client id
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	• <URI>: It is a label.
	• <URI_CONTEXT>: This is the rtsp request URI coming from a MRCP client
	Example:
	`open IN - client 2, message 14048, URI rtsp://srmproxy.voicegenie.com/osr`

Label	Description

open OUT
Logged by: SRMP
Level: 0x00000001 && (vrmproxy.metrics_direction =OUT || BOTH)
	SETUP Request Sent to A Resource
	This is logged when a SETUP request is sent to a MRCP server.
	The format is:
	`open OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<URI>|<URI_CONTEXT>`
	• <server>: It is a label.
	• <SERVER_ID>: It is an integer representing a MRCP resource id
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	• <URI>: It is a label.
	• <URI_CONTEXT>: This is the rtsp request URI coming from a MRCP client
	Example:
	`open OUT - server 382, message 14048, URI rtsp://10.0.0.70/recognizer`

open_resp IN
Logged by: SRMP
Level: 0x00000001 && (vrmproxy.metrics_direction =IN || BOTH)
	SETUP Response Received From A MRCP Server
	This is logged when a SETUP response is received from a MRCP server.
	The format is:
	`open_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>| <State>|<STATE>`
	• <server>: It is a label.
	• <SERVER_ID>: It is an integer representing a MRCP resource id
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	• <Status>: It is a label.
	• <STATUS>: RTSP status code of the message
	• <State>: It is a label.
	• <STATE>: RTSP state of the message
	Example:
	`open_resp IN - server 382, message 14048, status 200`

Label	Description

open_resp OUT
Logged by: SRMP
Level: 0x00000001 && (vrmproxy.metrics_direction =OUT || BOTH)
	SETUP Response Sent to A MRCP Client
	This is logged when a SETUP response is sent to the requesting MRCP client.
	The format is:
	`open_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>`
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	• <Status>: It is a label.
	• <STATUS>: RTSP status code of the message
	• <State>: It is a label.
	• <STATE>: RTSP state of the message
	Example:
	`open_resp OUT - message 14048, status 200`

pause IN Logged by: SRMP Level: 0x00008000 && (vrmproxy.metrics_direction =IN || BOTH)	**PAUSE Request Received** This is logged when a PAUSE request for a MRCP session is received. The format is: `pause IN <message>|<MESSAGE_ID>` • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id Example:
pause OUT Logged by: SRMP Level: 0x00008000 && (vrmproxy.metrics_direction =OUT || BOTH)	**PAUSE Request Sent to A Resource** This is logged when a PAUSE request is sent to the server. The format is: `pause OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>` • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id Example:

Label	Description
pause_resp IN Logged by: SRMP Level: 0x00008000 && (vrmproxy.metrics_direction =IN || BOTH)	**PAUSE Response Received From A MRCP Server** This is logged when a PAUSE response is received from a MRCP server. The format is: `pause_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>` • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message Example:
pause_resp OUT Logged by: SRMP Level: 0x00008000 && (vrmproxy.metrics_direction =OUT || BOTH)	**PAUSE Response Sent to A MRCP Client** This is logged when a PAUSE response is sent to the requesting MRCP client. The format is: `pause_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>` • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message • <State>: It is a label. • <STATE>: RTSP state of the message Example:

Label	Description

recog_complete IN
Logged by: SRMP
Level: 0x00000200 && (vrmproxy.metrics_direction =IN || BOTH)
	RECOGNITION-COMPLETE Event Received
	This is logged when a `RECOGNITION-COMPLETE` Event from MRCP server is received.
	The format is:
	`recog_complete IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<request>|<REQUEST_ID>`
	• `<server>:` It is a label.
	• `<SERVER_ID>:` It is an integer representing a MRCP resource id
	• `<message>:` It is a label.
	• `<MESSAGE_ID>:` It is an integer representing a RTSP message id
	• `<request>:` It is a label.
	• `<REQUEST_ID>:` It is an integer representing a MRCP request id
	Example:
	`recog_complete IN - server 320, message 10, request 11205`

recog_complete OUT
Logged by: SRMP
Level: 0x00000200 && (vrmproxy.metrics_direction =OUT || BOTH)
	RECOGNITION-COMPLETE Event Sent to A Client
	This is logged when a `RECOGNITION-COMPLETE` event is sent to the client.
	The format is:
	`recog_complete OUT <message>|<MESSAGE_ID>|<request>|<REQUEST_ID>"/>|<State>|<STATE>`
	• `<message>:` It is a label.
	• `<MESSAGE_ID>:` It is an integer representing a RTSP message id
	• `<request>:` It is a label.
	• `<REQUEST_ID>:` It is an integer representing a MRCP request id
	• `<State>:` It is a label.
	• `<STATE>:` RTSP state of the message
	Example:
	`recog_complete OUT - message 14041, request 2159, state 0`

Label	Description

recog_complete_resp IN
Logged by: SRMP
Level: 0x00000200 && (vrmproxy.metrics_direction =IN || BOTH)
	RECOGNITION-COMPLETE Response Received From A MRCP Client
	This is logged when a `RECOGNITION-COMPLETE` response is received from a MRCP client.
	The format is:
	`recog_complete_resp IN <message>|<MESSAGE_ID>|<Status>|<STATUS>`
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	• `<Status>`: It is a label.
	• `<STATUS>`: RTSP status code of the message
	Example:
	`recog_complete_resp IN - message 2159, status 200`

recog_complete_resp OUT
Logged by: SRMP
Level: 0x00000200 && (vrmproxy.metrics_direction =OUT || BOTH)
	RECOGNITION-COMPLETE Response Sent To A MRCP Server
	This is logged when a `RECOGNITION-COMPLETE` response is sent to the MRCP server.
	The format is:
	`recog_complete_resp OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>`
	• `<server>`: It is a label.
	• `<SERVER_ID>`: It is an integer representing a MRCP resource id
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	• `<Status>`: It is a label.
	• `<STATUS>`: RTSP status code of the message
	Example:
	`recog_complete OUT - server 382, message 29, status 200`

Label	Description

recog_start_timer IN
Logged by: SRMP
Level: 0x00001000 && (vrmproxy.metrics_direction =IN || BOTH)
	RECOGNITION-START-TIMERS Request Received
	This is logged when a `RECOGNITION-START-TIMERS` request for a MRCP session is received.
	The format is:
	`recog_start_timer IN <message>|<MESSAGE_ID>`
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	Example:
	`recog_start_timer IN - message 14059`

recognize IN
Logged by: SRMP
Level: 0x00000100 && (vrmproxy.metrics_direction =IN || BOTH)
	RECOGNIZE Request Received
	This is logged when a `RECOGNIZE` request for a MRCP session is received.
	The format is:
	`recognize IN <message>|<MESSAGE_ID>|<request>|<REQUEST_ID>`
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	• `<request>`: It is a label.
	• `<REQUEST_ID>`: It is an integer representing a MRCP request id
	Example:
	`recognize IN - message 14058, request 2159`

Label	Description
recognize OUT Logged by: SRMP Level: 0x00000100 && (vrmproxy.metrics_direction =OUT || BOTH)	**RECOGNIZE Request Sent to A Resource** This is logged when a RECOGNIZE request is sent to the server. The format is: recognize OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<request>|<REQUEST_ID> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <request>: It is a label. • <REQUEST_ID>: It is an integer representing a MRCP request id Example: recognize OUT - server 382, message 14058, request 2159
recognize_resp IN Logged by: SRMP Level: 0x00000100 && (vrmproxy.metrics_direction =IN || BOTH)	**RECOGNIZE Response Received From A MRCP Server** This is logged when a RECOGNIZE response is received from a MRCP server. The format is: recognize_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message Example: recognize_resp IN - server 382, message 14063, status 200

Label	Description

recognize_resp OUT
Logged by: SRMP
Level: 0x00000100 && (vrmproxy.metrics_direction =OUT || BOTH)
	RECOGNIZE Response Sent to A MRCP Client
	This is logged when a `RECOGNIZE` response is sent to the requesting MRCP client.
	The format is:
	`recognize_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>`
	• `<message>:` It is a label.
	• `<MESSAGE_ID>:` It is an integer representing a RTSP message id
	• `<Status>:` It is a label.
	• `<STATUS>:` RTSP status code of the message
	• `<State>:` It is a label.
	• `<STATE>:` RTSP state of the message
	Example:
	`recognize_resp OUT - message 14063, status 200, state 1`

redirect OUT
Logged by: SRMP
Level: 0x00008000 && (vrmproxy.metrics_direction =IN || BOTH)
	Redirect (ANNOUNCE) Request Sent
	This is logged when a `ANNOUNCE (redirect)` request for a MRCP session is sent to a MRCP client.
	The format is:
	`redirect OUT <message>|<MESSAGE_ID>`
	• `<message>:` It is a label.
	• `<MESSAGE_ID>:` It is an integer representing a RTSP message id
	Example:
	`redirect OUT - message 14047`

Label	Description

redirect_resp IN
Logged by: SRMP
Level: 0x00008000 && (vrmproxy.metrics_direction =IN || BOTH)
	Redirect (ANNOUNCE) Response Received
	This is logged when a `ANNOUNCE (redirect)` request for a MRCP session is received from a MRCP client.
	The format is:
	`redirect_resp IN <message>|<MESSAGE_ID>|<Status>|<STATUS>`
	• `<message>:` It is a label.
	• `<MESSAGE_ID>:` It is an integer representing a RTSP message id
	• `<Status>:` It is a label.
	• `<STATUS>:` RTSP status code of the message
	Example:
	`redirect_resp IN - message 14047, status 200`

resume IN Logged by: SRMP Level: 0x00010000 && (vrmproxy.metrics_direction =IN || BOTH)	**RESUME Request Received** This is logged when a `RESUME` request for a MRCP session is received. The format is: `resume IN <message>|<MESSAGE_ID>` • `<message>:` It is a label. • `<MESSAGE_ID>:` It is an integer representing a RTSP message id Example:
resume OUT Logged by: SRMP Level: 0x00010000 && (vrmproxy.metrics_direction =OUT || BOTH)	**RESUME Request Sent to A Resource** This is logged when a `RESUME` request is sent to the server. The format is: `resume OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>` • `<server>:` It is a label. • `<SERVER_ID>:` It is an integer representing a MRCP resource id • `<message>:` It is a label. • `<MESSAGE_ID>:` It is an integer representing a RTSP message id Example:

Label	Description
resume_resp IN Logged by: SRMP Level: 0x00010000 && (vrmproxy.metrics_direction =IN || BOTH)	**RESUME Response Received From A MRCP Server** This is logged when a `RESUME` response is received from a MRCP server. The format is: `resume_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>` • `<server>:` It is a label. • `<SERVER_ID>:` It is an integer representing a MRCP resource id • `<message>:` It is a label. • `<MESSAGE_ID>:` It is an integer representing a RTSP message id • `<Status>:` It is a label. • `<STATUS>:` RTSP status code of the message Example:
resume_resp OUT Logged by: SRMP Level: 0x00010000 && (vrmproxy.metrics_direction =OUT || BOTH)	**RESUME Response Sent to A MRCP Client** This is logged when a `RESUME` response is sent to the requesting MRCP client. The format is: `resume_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>` • `<message>:` It is a label. • `<MESSAGE_ID>:` It is an integer representing a RTSP message id • `<Status>:` It is a label. • `<STATUS>:` RTSP status code of the message • `<State>:` It is a label. • `<STATE>:` RTSP state of the message Example:

Label	Description

set_params IN
Logged by: SRMP
Level: 0x00000004 && (vrmproxy.metrics_direction =IN || BOTH)
	SET-PARAMS Request Received
	This is logged when a SET-PARAMS request for a MRCP session is received.
	The format is:
	`set_params IN <message>|<MESSAGE_ID>`
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	Example:
	`set_params IN - message 14055`

set_params OUT
Logged by: SRMP
Level: 0x00000004 && (vrmproxy.metrics_direction =OUT || BOTH)
	SET-PARAMS Request Sent to A Resource
	This is logged when a SET-PARAMS request is sent to the server.
	The format is:
	`set_params OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>`
	• `<server>`: It is a label.
	• `<SERVER_ID>`: It is an integer representing a MRCP resource id
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	Example:
	`set_params OUT - server 382, message 14055`

Label	Description

set_params_resp IN
Logged by: SRMP
Level: 0x00000004 && (vrmproxy.metrics_direction =IN || BOTH)
	SET-PARAMS Response Received From A MRCP Server
	This is logged when a SET-PARAMS response is received from a MRCP server.
	The format is:
	`set_params_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>`
	• <server>: It is a label.
	• <SERVER_ID>: It is an integer representing a MRCP resource id
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	• <Status>: It is a label.
	• <STATUS>: RTSP status code of the message
	Example:
	`set_params_resp IN - server 382, message 14055, status 200`

set_params_resp OUT
Logged by: SRMP
Level: 0x00000004 && (vrmproxy.metrics_direction =OUT || BOTH)
	SET-PARAMS Response Sent to A MRCP Client
	This is logged when a SET-PARAMS response is sent to the requesting MRCP client.
	The format is:
	`set_params_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>`
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	• <Status>: It is a label.
	• <STATUS>: RTSP status code of the message
	• <State>: It is a label.
	• <STATE>: RTSP state of the message
	Example:
	`set_params_resp OUT - message 14057, status 200, state 0`

Label	Description

speak IN
Logged by: SRMP
Level: 0x00000020 && (vrmproxy.metrics_direction =IN || BOTH)
	SPEAK Request Received
	This is logged when a SPEAK request for a MRCP session is received.
	The format is:
	`speak IN <message>|<MESSAGE_ID>|<request>|<REQUEST_ID>`
	• `<message>:` It is a label.
	• `<MESSAGE_ID>:` It is an integer representing a RTSP message id
	• `<request>:` It is a label.
	• `<REQUEST_ID>:` It is an integer representing a MRCP request id
	Example:
	`speak IN - message 14053, request 2153`

speak OUT
Logged by: SRMP
Level: 0x00000020 && (vrmproxy.metrics_direction =OUT || BOTH)
	SPEAK Request Sent to A Resource
	This is logged when a SPEAK request is sent to the server.
	The format is:
	`speak OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<request>|<REQUEST_ID>`
	• `<server>:` It is a label.
	• `<SERVER_ID>:` It is an integer representing a MRCP resource id
	• `<message>:` It is a label.
	• `<MESSAGE_ID>:` It is an integer representing a RTSP message id
	• `<request>:` It is a label.
	• `<REQUEST_ID>:` It is an integer representing a MRCP request id
	Example:
	`speak OUT - server 382, message 14053, request 2153`

Label	Description
speak_complete IN Logged by: SRMP Level: 0x00000040 && (vrmproxy.metrics_direction =IN || BOTH)	**SPEAK-COMPLETE Event Received** This is logged when a SPEAK-COMPLETE Event from MRCP server is received. The format is: speak_complete IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<request>|<REQUEST_ID> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <request>: It is a label. • <REQUEST_ID>: It is an integer representing a MRCP request id Example: speak_complete IN - server 382, message 29, request 14058
speak_complete OUT Logged by: SRMP Level: 0x00000040 && (vrmproxy.metrics_direction =OUT || BOTH)	**SPEAK-COMPLETE Event Sent to A Client** This is logged when a SPEAK-COMPLETE event is sent to the client. The format is: speak_complete OUT <message>|<MESSAGE_ID>|<request>|<REQUEST_ID>"/>|<State>|<STATE> • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <request>: It is a label. • <REQUEST_ID>: It is an integer representing a MRCP request id • <State>: It is a label. • <STATE>: RTSP state of the message Example: speak_complete OUT - message 14041, request 2159, state 0

Label	Description
speak_complete_resp IN Logged by: SRMP Level: 0x00000040 && (vrmproxy.metrics_direction =IN || BOTH)	**SPEAK-COMPLETE Response Received From A MRCP Client** This is logged when a SPEAK-COMPLETE response is received from a MRCP client. The format is: speak_complete_resp IN <message>|<MESSAGE_ID>|<Status>|<STATUS> • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message Example: speak_complete_resp IN - message 2159, status 200
speak_complete_resp OUT Logged by: SRMP Level: 0x00000040 && (vrmproxy.metrics_direction =OUT || BOTH)	**SPEAK-COMPLETE Event Response Sent To A MRCP Server** This is logged when a SPEAK-COMPLETE event response is sent to the MRCP server. The format is: speak_complete_resp OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message Example: speak_resp OUT - server 382, message 29, status 200

Label	Description
speak_resp OUT Logged by: SRMP Level: 0x00000020 && (vrmproxy.metrics_direction =OUT || BOTH)	**SPEAK Response Sent to A MRCP Client** This is logged when a SPEAK response is sent to the requesting MRCP client. The format is: speak_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE> • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message • <State>: It is a label. • <STATE>: RTSP state of the message Example: speak_resp OUT - message 14053, status 200, state 1
start_of_speech IN Logged by: SRMP Level: 0x00002000 && (vrmproxy.metrics_direction =IN || BOTH)	**START-OF-SPEECH Event Received** This is logged when a START-OF-SPEECH Event from MRCP server is received. The format is: start_of_speech IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<request>|<REQUEST_ID> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <request>: It is a label. • <REQUEST_ID>: It is an integer representing a MRCP request id Example: start_of_speech IN - server 382, message 28, request 14058

Label	Description

start_of_speech OUT
Logged by: SRMP
Level: 0x00002000 && (vrmproxy.metrics_direction =OUT || BOTH)
	START-OF-SPEECH Event Sent to A Client
	This is logged when a START-OF-SPEECH event is sent to the client.
	The format is:
	`start_of_speech OUT <message>|<MESSAGE_ID>|<request>|<REQUEST_ID>"/>|<State>|<STATE>`
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	• `<request>`: It is a label.
	• `<REQUEST_ID>`: It is an integer representing a MRCP request id
	• `<State>`: It is a label.
	• `<STATE>`: RTSP state of the message
	Example:
	`start_of_speech OUT - message 14042, request 2159, state 1`

start_of_speech_resp IN
Logged by: SRMP
Level: 0x00002000 && (vrmproxy.metrics_direction =IN || BOTH)
	START-OF-SPEECH Response Received From A MRCP Client
	This is logged when a START-OF-SPEECH response is received from a MRCP client.
	The format is:
	`start_of_speech_resp IN <message>|<MESSAGE_ID>|<Status>|<STATUS>`
	• `<message>`: It is a label.
	• `<MESSAGE_ID>`: It is an integer representing a RTSP message id
	• `<Status>`: It is a label.
	• `<STATUS>`: RTSP status code of the message
	Example:
	`start_of_speech_resp IN - message 2159, status 200`

Label	Description
start_of_speech_resp OUT Logged by: SRMP Level: 0x00002000 && (vrmproxy.metrics_direction =OUT || BOTH)	**START-OF-SPEECH Response Sent To A MRCP server** This is logged when a START-OF-SPEECH response is sent to the MRCP server. The format is: start_of_speech_resp OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>| <State>|<STATE> • <server>: It is a label. • <SERVER_ID>: It is an integer representing a MRCP resource id • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id • <Status>: It is a label. • <STATUS>: RTSP status code of the message Example: start_of_speech_resp OUT - server 382, message 28, status 200
stop IN Logged by: SRMP Level: 0x00000010 && (vrmproxy.metrics_direction =IN || BOTH)	**STOP Request Received** This is logged when a STOP request for a MRCP session is received. The format is: stop IN <message>|<MESSAGE_ID> • <message>: It is a label. • <MESSAGE_ID>: It is an integer representing a RTSP message id Example: stop IN - message 14065

Label	Description

stop OUT
Logged by: SRMP
Level: 0x00000010 && (vrmproxy.metrics_direction =OUT || BOTH)
	STOP Request Sent to A Resource
	This is logged when a STOP request is sent to the server.
	The format is:
	`stop OUT <server>|<SERVER_ID>|<message>|<MESSAGE_ID>`
	• <server>: It is a label.
	• <SERVER_ID>: It is an integer representing a MRCP resource id
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	Example:
	`stop OUT - server 382, message 14065`

stop_resp IN
Logged by: SRMP
Level: 0x00000010 && (vrmproxy.metrics_direction =IN || BOTH)
	STOP Response Received From A MRCP Server
	This is logged when a STOP response is received from a MRCP server.
	The format is:
	`stop_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>`
	• <server>: It is a label.
	• <SERVER_ID>: It is an integer representing a MRCP resource id
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	• <Status>: It is a label.
	• <STATUS>: RTSP status code of the message
	Example:
	`stop_resp IN - server 382, message 14065, status 200`

Label	Description

stop_resp IN
Logged by: SRMP
Level: 0x00000020 && (vrmproxy.metrics_direction =IN || BOTH)
	SPEAK Response Received From A MRCP Server
	This is logged when a SPEAK response is received from a MRCP server.
	The format is:
	`stop_resp IN <server>|<SERVER_ID>|<message>|<MESSAGE_ID>|<Status>|<STATUS>`
	• <server>: It is a label.
	• <SERVER_ID>: It is an integer representing a MRCP resource id
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	• <Status>: It is a label.
	• <STATUS>: RTSP status code of the message
	Example:
	`speak_resp IN - server 382, message 14053, status 200`

stop_resp OUT
Logged by: SRMP
Level: 0x00000010 && (vrmproxy.metrics_direction =OUT || BOTH)
	STOP Response Sent to A MRCP Client
	This is logged when a STOP response is sent to the requesting MRCP client.
	The format is:
	`stop_resp OUT <message>|<MESSAGE_ID>|<Status>|<STATUS>|<State>|<STATE>`
	• <message>: It is a label.
	• <MESSAGE_ID>: It is an integer representing a RTSP message id
	• <Status>: It is a label.
	• <STATUS>: RTSP status code of the message
	• <State>: It is a label.
	• <STATE>: RTSP state of the message
	Example:
	`stop_resp OUT - message 14065, status 200, state 0`

Chapter

5 Alarms

Note that all SRM Clients, SRM Servers and MRCP/SRM Proxies are implemented with the MRCP Protocol/stack, and thus all of them may throw the alarms listed in the following table.

Alarm#	Level	Definition and Possible Message/Info	Impacts	Causes	Detailed Recommended Actions
060007D1	EROR	New failed for <datatype>	Any ASR/TTS will fail for the current call	Software EROR	Report to VoiceGenie
060007D2	EROR	Invalid configuration setup	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	Configuration EROR	Check SRM Client configuration
060007D3	EROR	operation while uninitialized	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	ASR/TTS during VRMClient shutdown	Restart callmgr
060007D4	EROR	Unable to construct malformed message	Any ASR/TTS will fail for the current call	Software EROR	Report to VoiceGenie
060007D5	EROR	Unable to parse malformed message	Any ASR/TTS will fail for the current call	Software EROR	Report to VoiceGenie
060007D6	EROR	Received invalid request	Any ASR/TTS will fail for the current call	Software EROR	Report to VoiceGenie

Alarm#	Level	Definition and Possible Message/Info	Impacts	Causes	Detailed Recommended Actions
060007D7	EROR	Socket EROR	Any ASR/TTS will fail until anASR/TTS server is started, it is not necessary to start VoiceGenie Platform as it will recover	Server/Network EROR	Check server connections
06000BB9	WARN	Unexpected Socket Event	Any ASR/TTS will fail until anASR/TTS server is started, it is not necessary to start VoiceGenie Platform as it will recover	Server/Network EROR	Check server connections

The following alarm list is specific for the MRCP Proxy:

Alarm#	Level	Definition and Possible Message/Info	Impacts	Causes	Detailed Recommended Actions
067003E8	CRIT	Failed to initialize license manager	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	License is missing or expired	Verify Software Configuration or License
067003E9	CRIT	Failed to initialize proxy core	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	Software Error	Report to VoiceGenie
067003EA	CRIT	Failed to initialize stream module	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	Software Error	Report to VoiceGenie
067003EB	CRIT	Failed to initialize client protocol module	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	Software Error	Report to VoiceGenie
067003EC	CRIT	Failed to initialize server protocol module	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	Software Error	Report to VoiceGenie

Alarm#	Level	Definition and Possible Message/Info	Impacts	Causes	Detailed Recommended Actions
067003ED	CRIT	Failed to initialize resource manager	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	Software Error	Report to VoiceGenie
067007D0	CRIT	License expired	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	License is missing or expired	Verify Software Configuration or License
067007D1	EROR	Configuration error	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	Configuration Error	Check SRM Proxy Configuration
067007D2	EROR	Failed to create session list	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	Software Error	Report to VoiceGenie
06700BB8	EROR	Failed to obtain license	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	License is missing or expired	Verify Software Configuration or License
06700BB9	WARN	Server module system error	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	Software Error	Report to VoiceGenie
06700BBA	WARN	Client module system error	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	Software Error	Report to VoiceGenie
06700BBB	WARN	Unknown event	Any ASR/TTS will fail until problem is fixed and VoiceGenie Platform is restarted	Software Error	Report to VoiceGenie
06700BBC	WARN	Failed to open audio stream	Any ASR/TTS will fail for the current call	Software Error	Report to VoiceGenie
06700BBD	WARN	Failed to update server stream destination	Any ASR/TTS will fail for the current call	Software Error	Report to VoiceGenie

Alarm#	Level	Definition and Possible Message/Info	Impacts	Causes	Detailed Recommended Actions
06700BBE	WARN	Failed to build SDP message	Any ASR/TTS will fail for the current call	Software Error	Report to VoiceGenie
06700BBF	WARN	Unsupported SDP message	Any ASR/TTS will fail for the current call	Software Error	Report to VoiceGenie
06700BC0	WARN	Ping error	Any ASR/TTS will fail	Server/Network Error	Check server connections
06700BC0	WARN	Failed to register data point	Any ASR/TTS will fail for the current call	Software Error	Report to VoiceGenie
06700FA0	WARN	VRM Proxy is running	None		
06700FA1	NOTE	VRM Proxy shutdown	None		
06700FA2	NOTE	License notice	None		
06701388	NOTE	CMP is unable to synchronize config	None	Configuration Error	Check CMP Proxy configuration
06701389	INFO	Client is not allowed to connect to this proxy	Any ASR/TTS will fail for this client	Configuration Error	Check SRM Proxy configuration
0670138A	INFO	Client connected	None		
0670138B	INFO	Client disconnected	None	Server/Network Error	Check server connections
0670138C	INFO	Termination notification	None		
0670138D	INFO	Server connected	None		
0670138E	INFO	Failed to connect server	Any ASR/TTS will fail until ASR/TTS server is up, it is not necessary to restart the VoiceGenie Platform since it will recover itself	Server/Network Error	Check server connections

Alarm#	Level	Definition and Possible Message/Info	Impacts	Causes	Detailed Recommended Actions
0670138F	INFO	Server disconnected	Any ASR/TTS will fail until ASR/TTS server is up, it is not necessary to restart the VoiceGenie Platform since it will recover itself	Server/Network Error	Check server connections
06701390	INFO	Dynamic parameter update	None		
06701391	INFO	Generic CLC command	None		
06701392	INFO	Provision handler registration fail.	None		

Revision History

Version	Date	Change Summary
1.0	March 23rd, 2005	Initial release
1.1	April 13th, 2005	Revised version
1.2	March 1st, 2006	Updates for VoiceGenie 7.1
1.3	September 5th	Updates for VoiceGenie 7.1
1.4	September 3rd, 2007	Updates for VoiceGenie 7.2