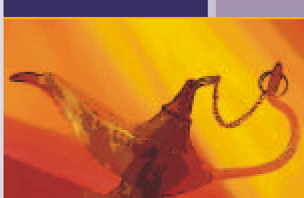
VOICEGENIE

VoiceGenie 7 Speech Resource Management System Reference Guide

August 12th, 2005



VoiceGenie Technologies Inc.
1120 Finch Ave. W. • Toronto, Ontario • M3J 3H7 • Canada
T. +1.416.736.4151 • F. +1.416.736.1551 • support@voicegenie.com
www.voicegenie.com

VoiceGenie

VoiceGenie Contacts

VoiceGenie Technologies Inc.
1120 Finch Avenue West
Toronto, Ontario
Canada
M3J 3H7

T. +1.416.736.4151
F. +1.416.736.1551
support@voicegenie.com

http://www.voicegenie.com/index.html

Proprietary / Copyright Information

Notice

Every effort was made to ensure that the information in this document was complete and accurate at the time of printing. However, this information is subject to change without notice.

Please note that VoiceGenie makes no warranties with respect to future releases. The information provided herein is for informational purposes only. VoiceGenie reserves the right to change product release schedules and/or features allocated to a product release at any time.

Trademarks

All trademarks are the property of their respective owners. Where those designations appear in this document, and VoiceGenie is aware of a trademark claim, the designations have been printed in initial caps or all caps.

Table of Contents

Revision History

Version	Date	Change Summary	Author/Editor
1	March 11th, 2005	Initial release	Alex Lee Johnson Tse Lin Chen Andrew Ho
1.1	March 29th, 2005	Revised version	Andrew Ho
1.2	April 13th, 2005	Final Revision for VoiceGenie 7 Release	Andrew Ho
1.3	August 12th, 2005	Added details for Natural Voices.	Shifeng Wu

1 Introduction

This guide serves as the system reference manual for the VoiceGenie 7 Speech Resource Management (SRM) product. It is intended to provide a complete reference for all aspects related to the configuration, Metrics and alarming of the SRM system – including both the SRM Client and the SRM Server.

The following table gives definitions of some acronyms that are used throughout this document:

Acronyms	Full Definitions
ASR	Automated Speech Recognition (Engines/Technologies)
CLC	Command Line Console -- A command line interface that can be used to query information and issue commands
MRCP	Media Resource Control Protocol -- Adopted by the VoiceGenie Media Platform to control ASR and TTS resources
SRM	Speech Resource Management -- A component integrated into the VoiceGenie Media Platform to provide Speech Recognition and Synthesis functionalities to the application developers
SMC	System Management Console -- A web based tool for administering clusters of VoiceGenie VoiceXML Platforms
OA&M	Operation, Administration and Management
TTS	Text To Speech (Engines/Technologies)

The following sections may contain references to terminology that has become obsolete since the last release, NeXusPoint 6.4.x. Here is a mapping between these terms:

Historical Terms	New Terms
PhoneWeb Software / NeXusPoint 6.4.x Software	VoiceGenie 7 Software
Cluster Management Platform (CMP)	OA&M Framework
Voice Resource Manager (VRM)	Speech Resource Management (SRM)
VoiceGenie Management Console (VMC)	System Management Console (SMC)

2 SRM Client Configuration

The SRM client has a few configuration parameters that define its behaviors. They can be modified via the CMP SMC component through call manager configuration.

The following table lists these configuration parameters:

Parameter Name	Description
vrm.client.dll	This configuration parameter defines the location of the SRM Client library to be used by the Media Platform.
vrm.client.grammar.path	This specifies the location of the built-in grammars residing on the VoiceGenie platform.
vrm.client.tmp.path	This specifies the location of the temporary directory used by the media platform. This must match the `PW_TMP` entry in the `voicexml.cfg` file used by the VoiceXML Interpreter.
vrm.client.vggrammarbase	This specifies the base-URL for translation of grammars residing under the subdirectory vrm.client.tmp.path. For example, in a Linux platform, if vrm.client.tmp.path = /usr/local/phoneweb/tmp/ vrm.client.vggrammarbase = /vggrammarbase/tmp The file /usr/local/phoneweb/tmp/index.txt would be translated by the SRM client into http://205.150.90.166/vggrammarbase/tmp/index.txt. where 205.150.90.166 is the IP address of the MP. These two options allow the temporary grammar generated by the media platform to be fetched by an offboard server. The web server defined in the "HTTP Access to Grammars" section of the SRM Users' Guide is used provide the hotkey grammars, so this configuration item must work together with the configuration defined in the web server.
vrm.client.timeout	This is the timeout value used by the SRM client to wait for a response from the MRCP server, for both the VoiceGenie SRM Server and the native MRCP servers. If a response to an MRCP request has is not received within this timeout period, then the request is deemed to have failed.
vrm.ping.frequency	This parameter defines, in milliseconds, the frequency in which the SRM client pings each of its servers. The MRCP "DESCRIBE" method is used as a ping message for each of the MRCP Servers

	provisioned.
vrm.ping.timeout	This parameter defines, in milliseconds, the timeout period for which we would be waiting for ping response from a MRCP server. If a ping response is not heard back from the server within this timeout, the SRM Client would consider the MRCP server to have become unavailable, and it would then disconnect from the server and periodically re-try connection to the MRCP server again.
vrm.client.max.noinput.timeout	This sets the value, in milliseconds, for the noinput timeout header that is sent to an MRCP engine. This should be set this to a large value, as the VoiceGenie Media Platform handles the no input timer. Default value is 90 seconds.

3 SRM Client Speech Resource Configuration Parameters

The following table contains the full list of the parameters that may be set in the SRM client Speech Resource provisioning:

Note that all parameters should have a prefix "vrm.client." --

Parameter Name	Description
SendVGParams	If this is set to true, then the SRM Client will send some VG vendor-specific parameters in the MRCP messages it sends. This should be set to true for engines that require the VG SRM Server for integration (i.e. the engines without native MRCP support). Default value is false.
DefineGrammarSerial	Sending out the Define Grammar Request serially, that is, the SRM client waits for the DEFINE-GRAMMAR reply from the server before sending the DEFINE-GRAMMAR request for the next grammar. This may not be as efficient, but some MRCP engines does not accept simultaneous grammar define requests. Default value is false. Currently it is required to be set to true for Telisma Telispeech MRCP ASR server and Loquendo MRCP ASR server
TelispeechRecognitionBargein	Set to true to supports Telisma's recognition bargein capability
UniqueGramID	Send a unique Grammar ID for each grammar request, across different channels. Default value is false. Currently it is required to be set to true for NUANCE MRCP ASR server
NuanceTranslateGrammar	For support of legacy application, when this is set to true, the Nuance 7 grammars are translated into Nuance 8 grammars. It is not recommended, however, for new applications to be developed using Nuance 7 grammars. Since the nuance built-in grammars are in Nuance 7 format, this parameter should be set to true for a Nuance MRCP ASR server. Default value is false.
NuanceTranslateBuilt-inTo	Setting this parameter to "static", the Media platform will translate URI of built-in#rule to Nuance desirable URI to support Nuance static

	grammar. Default value is static for a Nuance Engine.
InsertXmlNs	For a Nuance MRCP server we need set value of this parameter to: http://www.w3.org/2001/06/grammar default: not set.
NotEscapeApos	Set this parameter to true for a Nuance MRCP server Default value: false.
LoquendoGrammar	When using the Loquendo MRCP engine, it requires some special handling for its built-in and hotkey grammars. Setting this to true enables such handling. Default value is false.
SkipLoquendoParam	Loquendo requires special handling for its parameter values. Setting this parameter to true enables such special handling. Default value is false.
SendSWMSParams	When this is set, the SRM client will send the OSR 3.0 parameters using the SWMS 3.0 [The MRCP server on top of OSR 3.0 engine] convention. This should set to true, when the VRM client is connected to the native MRCP OSR3.0 server directly. This is not required if we are connected to an OSR 3.0 engine via the VG SRM Server. Default value is false.
NoDuplicatedGramURI	To workaround the problem for some engines that cannot accept duplicated URI in the same recognition session. Default value is false Currently it is required to set to true for SSFT SWMS OSR server
SkipRecognitionTimeout	For some engines, it does not accept the Recognition-Timeout header of MRCP. Setting this parameter to true would cause the SRM client to not send this Header. Default value is false. Currently it is required to be set to true for NUANCE MRCP ASR server
SkipPromptDone	When this is set to true, the SRM client will not send the REGOGNITION-START-TIMER MRCP

	method to the MRCP engines. Some engines do not handle this method well. Default value is false. Currently it is required to be set to true for LOQUENDO MRCP ASR server and SSFT SpeechPerl MRCP ASR server
SkipSetMRCPParams	When this is set to true, SRM client will not send any non-vendor-specific MRCP parameters in the MRCP SET-PARAM method. This helps increase the efficiency in the overall system. Default value is false.
NLSMLEncoding	When this parameter is defined, this is the encoding type that the NLSML will be returned in. This is used for ASR engines that provide its NLSML result in an encoding other than UTF-8, and it does not indicate its encoding type in the XML header. Default value for this parameter is that it's not defined.
SendGrammarContent	Send the DEFINE-GRAMMAR with Grammar content instead of URI for inline grammars For TTY/TDD engine, this parameter must be set to true.
DisableHotWord	Setting this parameter, the platform will treat recognition based barge-in as speech based barge-in. This parameter should be set to true to all the ASR server that does not support recognition based barge-in
HotKeyBasePath	This is the HTTP fetchable location for the hotkey grammars. The IP address is the VG platform IP address. The web server defined in the "HTTP Access to Grammars" section . of the SRM Users' Guide is used provide the hotkey grammars, so this configuration item must work together with the configuration defined in the web server.
HotKeyLocalPath	This is the local path for the hotkey grammars on the VoiceGenie Media Platform. The SRM Client will translate this address, using the HotKeyBasePath, to the appropriate URIs to be sent to the ASR servers.

	For each ASR engine its value should be $VGROOT/grammar/<engine name>/hotkey
PassThruTTSPort	When this option is set the true, the Media Platform will filter out the received TTS RTP packets that are not sent from the source RTP port number when the TTS session was first establish. Default value is false. Currently it is required to be set to true for SSFT RealSpeak MRCP TTS server

4 SRM Server Configuration

4.1 TTS Client Provisioning

The top-level parameter is TTS_CLIENT_LIST. Using this parameter, the SRM server may specify a comma-delimited list of TTS clients to start. An example of a top-level parameter is as follows:

```
TTS_CLIENT_LIST = SPEECHIFY_TOM, SPEECHIFY_JILL
```

For each <TTS_CLIENT_NAME> in the list, the following parameters may be specified:

Parameter Name	Description
TTS_CLIENT.<TTS_CLIENT_NAME>.TYPE	This may be one of the following values: a – this entry is for an alias n – this entry is for a normal TTS client
TTS_CLIENT.<TTS_CLIENT_NAME>.NAME	This is the name of the TTS client. This is used to uniquely identify a type of client.
TTS_CLIENT.<TTS_CLIENT_NAME>.VIRTUAL_DIR	This is the "virtual directory" recognized by the SRM Server. If the suffix of a resource URI of an incoming MRCP message matches this parameter, then a client of this type is used. For example, for the following the resource URI of: rtsp://10.0.0.12/speechify_synthesizer "speechifiy_synthesizer" is the VIRTUAL_DIR. When used in conjunction with the SRM Client, this suffix must match the URI portion of the "Servers" parameter.
TTS_CLIENT.<TTS_CLIENT_NAME>.EXE	This is the full path name to the executable for the client.
TTS_CLIENT.<TTS_CLIENT_NAME>.PROFILE	This is the configuration file to be used by the TTS Client. For more information about how to configure each of the TTS clients, please refer to the sections below.
TTS_CLIENT.<TTS_CLIENT_NAME>.NUM_HOST	This is the number of TTS clients to start.
TTS_CLIENT.<TTS_CLIENT_NAME>.PARAMETER	This is the list of parameters to be appended to the command line when starting the TTS client. Typically, this parameter has value: -i <imt-cmp-config-file> -o imt where <imt-cmp-config-file> is a configuration file used to control the logging behaviour of the TTS client. Each TTS client has its own imt-cmp-config-file.
TTS_CLIENT.<TTS_CLIENT_NAME>.ENV_VAR_FILE	This is an optional file, used to specify environment variables to be set before starting the client. Each line of the environment variable file has the format:

	env_var = value

4.2 ASR Client Provisioning

The top-level parameter is ASR_CLIENT_LIST. Using this parameter, the SRM server may specify a comma-separated list of ASR clients to start. An example of a top-level parameter is as follows:

```
ASR_CLIENT_LIST = SPEECHWORKS
```

For each <ASR_CLIENT_NAME> in the list, the following parameters may be specified:

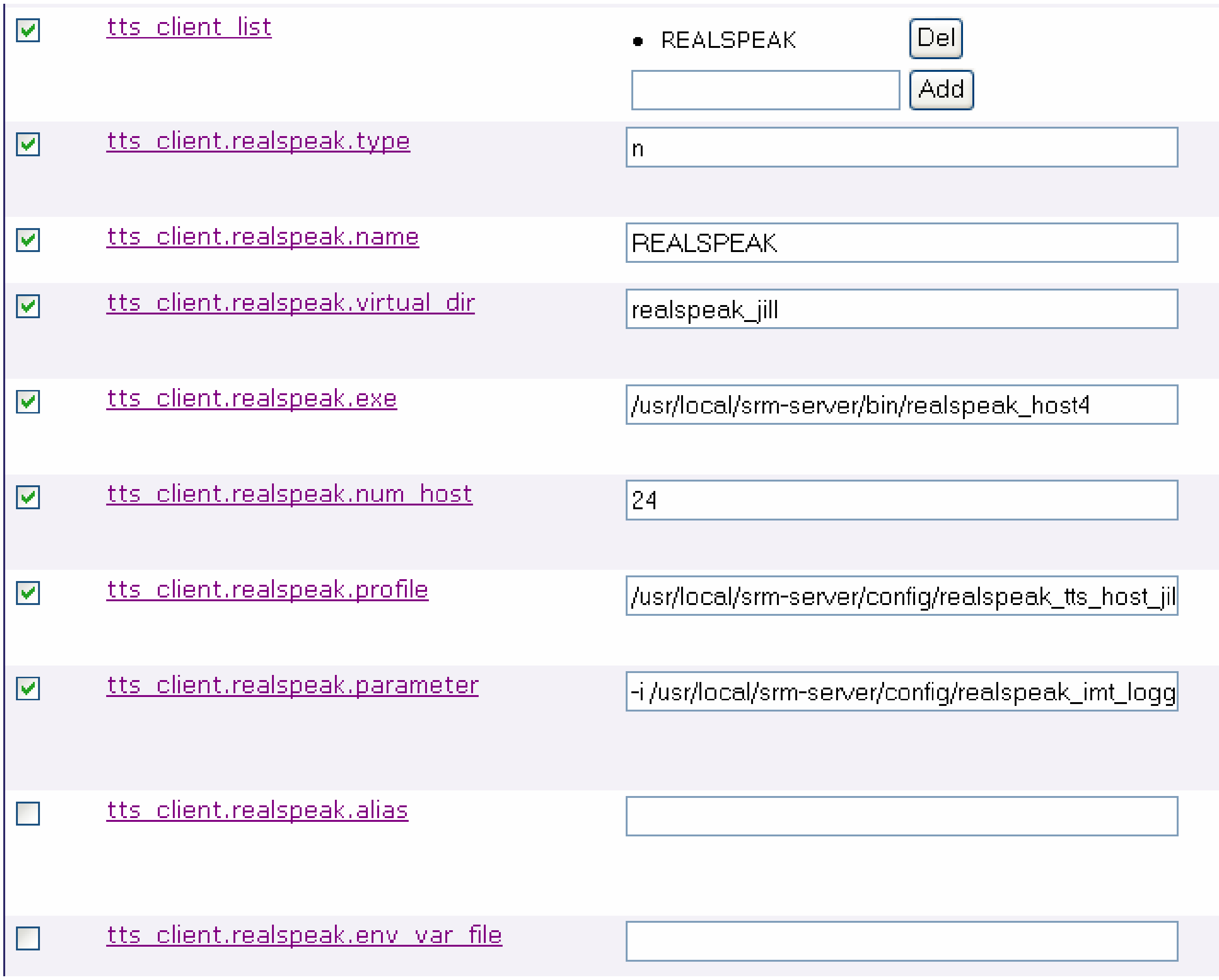
Parameter Name	Description
ASR_CLIENT.<ASR_CLIENT_NAME>.TYPE	This may be one of the following values: a – this entry is for an alias n – this entry is for an ASR client, returning results in the VoiceGenie proprietary format. Phonetics, BBN, and Watson clients are this type of client. x – this entry is for an ASR client which can return results in NLSML format. The client for OSR is this type of client.
ASR_CLIENT.<ASR_CLIENT_NAME>.NAME	This is the name of the ASR client. This is used to uniquely identify a type of client.
ASR_CLIENT.<ASR_CLIENT_NAME>.VIRTUAL_DIR	This is the “virtual directory” recognized by the SRM Server. If the suffix of a resource URI of an incoming MRCP message matches this parameter, then a client of this type is used. For example, for the following the resource URI of: rtsp://10.0.0.12/spwx_recognizer “spwx_recognizer” is the VIRTUAL_DIR. When used in conjunction with the SRM Client, this suffix must match the vrm.client.resource.uri portion in the SRM client Speech Resource provision.
ASR_CLIENT.<ASR_CLIENT_NAME>.EXE	This is the full path name to the executable for the client.
ASR_CLIENT.<ASR_CLIENT_NAME>.PROFILE	This is the configuration file to be used by the ASR Client. For more information about how to configure each of the ASR clients, please refer to sections below.
ASR_CLIENT.<ASR_CLIENT_NAME>.NUM_CLIENT	This is the number of ASR client processes to start.
ASR_CLIENT.<ASR_CLIENT_NAME>.NUM_THREADS	This is the number of threads each ASR client should start. For BBN, Phonetics, and OSR, we should start 1 client with n threads, where n is the number of ASR resources required. For Watson, we should start n clients with 1 thread each.

ASR_CLIENT.<ASR_CLIENT_NAME>.CONTEXT	This is a required parameter for each client, and it specifies a file, whish contains some default ASR parameters used for the initial recognition session when the ASR session is first requested.
ASR_CLIENT.<ASR_CLIENT_NAME>.PARAMETER	This is the list of parameters to be appended to the command line when starting the ASR client. This is optional, and varies for different ASR clients.
ASR_CLIENT.<ASR_CLIENT_NAME>.ENV_VAR_FILE	This is an optional file, used to specify environment variables to be set before starting the client. Each line of the environment variable file has the format: env_var = value

The following screen capture illustrates the GUI part that allows user to add a new TTS engine. The user can type a new TTS engine name in the blank box of the tts_client_list parameter. Clicking the Add button beside the blank box will create the set of parameters to define the new engine. The user must fill in the right parameter values in the respective boxes. Selecting the Update button at the bottom of the GUI is required to have the changes take effect.

To delete a TTS engine, the user can select the Del button beside the TTS engine in the tts_client_list section. Selecting the Update button at the bottom of the GUI is required to have the changes take effect.

It is similar to add and delete an ASR engine.



4.3 ASR / TTS Client Related Configuration

These parameters are used to control where the temporary files are created, and how long they will last in the file system:

Parameter Name	Description
TEMPORARY_PATH	This is the temporary path in the file system used to temporarily store some information by the ASR or TTS clients. For example, the save utterance files are stored in this TEMPORARY_PATH.
TEMPORARY_PATH_URI	This is the HTTP fetchable location for the TEMPORARY_PATH. The IP address of this URI is the IP address of the SRM Server. The web server

	defined in the "Application Server" subsection under the "SRM Server" Section of the SRM Users' Guide is used make the files under TEMPORARY_PATH available. This configuration item must work together with the configuration defined in the web server.
SAVE_TMP_FILES	This parameter is used to indicate whether the SRM server should keep the temporary files after the ASR/TTS sessions have completed.

These parameters are used to control how quickly the TTS and ASR clients are restarted after the SRM server has detected they have unexpectedly disconnected (most likely due to an unintended termination of the ASR/TTS client or ASR/TTS server). The time between restarts increases by RESPAWN_INTERVAL_BAASE each time a client cannot be restarted successfully.

Parameter Name	Description
RESPAWN_INTERVAL_BASE	When an ASR or TTS client has restarted for the n^{th} consecutive time, it will be restarted in RESPAWN_INTERVAL_BASE * (n - 1) seconds, up to a maximum of MAX_RESPAWN_TIMEOUT
RESPAWN_CHECK_INTERVAL	After a TTS/ASR client has restarted, if it hasn't prematurely terminated after RESPAWN_CHECK_INTERVAL seconds, the TTS/ASR client is considered successfully restarted and the counter "n" where "n" is the number of times the client has restarted is reset.
MAX_RESPAWN_TIMEOUT	This is the maximum amount of time, in seconds, between which the clients are restarted. This is intended so the re-spawn interval does not grow without bound

These parameters are used for ASR clients:

Parameter Name	Description
ASR_MANAGER.CLIENT_PORT	This is the port number on which the ASR clients make their socket connections to the SRM Server. It is specified in the format: /dev/tcp/local/9600 where "9600" is the actual port number.
ASR_MANAGER.MAX_PENDING_CONNECTIONS	This specifies the length of the listen socket queue for ASR_MANAGER.CLIENT_PORT
ASR_MANAGER.ONE_CLIENT_PER_CHANNEL	This flag is used when a call session requires two different types of clients. When this flag is set to true, the client of the previous type is released before allocating a new client.
ASR_MANAGER.SPAWN_CLIENT_INTERVAL	This is the duration in milliseconds between starting two clients when the SRM server initially starts up. A delay is put here so that we don't overload the

	machine where the SRM server is.

These parameters are used for TTS clients:

Parameter Name	Description
TTS_MANAGER.CLIENT_PORT	This is the port number on which the TTS clients make their socket connections to the SRM Server. It is specified in the format: /dev/tcp/local/9004 where "9004" is the actual port number.
TTS_MANAGER.MAX_PENDING_CONNECTIONS	This specifies the length of the listen socket queue for TTS_MANAGER.CLIENT_PORT
TTS_MANAGER.SPAWN_CLIENT_INTERVAL	This is the duration in milliseconds between starting two clients when the SRM server initially starts up. A delay is put here so the machine where the SRM server is not overloaded.
TTS_MANAGER.MAX_STRING_LEN	This is the maximum length of a TTS request such that the SRM server would not use a file to forward the TTS request to the client. If the TTS request is less than this size, then the entire request will be sent within the socket message itself. Default value: 2048.
TTS_MANAGER.RESET_INTERVAL	If this value is set to "n", each TTS client will terminate itself after having started for "n" milliseconds. When set to 0, the TTS client will not terminate itself. Default value: 0
TTS_MANAGER.SEARCH_INTERVAL	This is the number of milliseconds between checking each of the TTS clients to determine it is necessary to be restarted because it has been running for longer than TTS_MANAGER.RESET_INTERVAL milliseconds. When this is set to 0, the engine will not restart a client because it has been running for too long. Default value: 0

4.4 Other Configuration

The rest of the following parameters are for configuring the SRM Server in terms of how it accepts requests from the clients:

Parameters	Description
VRMServer.RM.RequestTimeout	This is the time to wait in milliseconds before returning a timeout error back to the client that

	made the request to the SRM server. Default value: 5000
VRMServer.RM.VRMProtocolModules	Semi-colon delimited list of names of .so files of protocol modules. For this version of SRM Server, this should be set to: /usr/local/srm/lib/MRCPProtocolModule.so (Linux) C:\VoiceGenie\srm-server\lib\mrcpprotocolmodule.dll (Windows)
VRMServer.RM.VRMEngineModules	Semi-colon delimited list of names of .so files of engine modules. For example: /usr/local/srm-server/lib/VRMLegacyEngine.so;/usr/local/srm-server/lib/RtspTtsEngine.so;/usr/local/srm-server/lib/VRMTTYEngine.so (Linux) C:\VoiceGenie\srm-server\lib\vrmlegacyengine.dll;C:\VoiceGenie\srm-server\lib\rtsptts.dll;C:\VoiceGenie\srm-server\lib\TTYEngine.dll (Windows)
VRMServer.RM.MRCP.Standard	This can be set to either “standard” or “extended”. This is used to indicate whether the SRM server should accept VG-specific vendor-specific parameters in the MRCP messages.
VRMServer.RM.OpenIdleTimeOut	Time in milliseconds to release the session if there is no subsequent request after SETUP message. Default Value: 900000
VRMServer.DataSrvHdlr.StaleTimeout	When the VoiceGenie CMP component requests a status update from the SRM server, the SRM server returns the status from an internal cache of data kept specifically for sending to the CMP. If the data is older than the timeout specified in this parameter (kept in milliseconds), the SRM Server would not use the cache data, instead it would query the data from various components and return the queried data.
VRMServer.MRCPPM.VRMServerIP	The IP address of the protocol module
VRMServer.MRCPPM.VRMServerPort	Port number that the protocol module can use for incoming requests. When used in conjunction with the SRM Client, this suffix must match the port portion of the “Servers” parameter.
VRMServer.MRCPPM.VRMRestrictConnections	This could be set to true or false. If false, the protocol module will accept connections from any

	computer. If true, the protocol module will only accept connections from the list of valid IP addresses
VRMServer.MRCPPM.VRMValidIps	A semi-colon delimited list of IP addresses that we are permitted to accept connections from. This value is only used when VRMRestrictConnections is set to true.
VRMServer.MRCPPM.FailedSessionTimeout	The length of time (in ms) after a session has been flagged as "failed" to wait before removing the failed session from the automatic error response list.
NotPingClient	This parameter indicates if the MRCP client connected to the SRM server has implemented the DESCRIBE as ping mechanism. If not, this flag should be set to true. By default, the flag is false.
MaxClientIdleTimeOut	Time in seconds to close a socket to a client if there is no client ping message from the client to the server. If not set default is 300 seconds The number will not have effect if VRMServer.RM.NotPingClient is set to true
MaxSessionIdleTimeOut	Time in seconds to close a session if there is no session activity. If not set ,the default value is 3600 seconds
PingCheckFrequency	Time in seconds to check the session and client ping message. If not set the default is 60 seconds

5 ASR / TTS Engines Specific Configuration

5.1 Phonetics Configuration

The configuration files for Phonetics are all located in /usr/local/srm-server/config/. There are 3 files needed for the configuration: profile.phonetic, phonetics.context, and phoneticclnt_imt_logger.cfg. The phonetics.context file is more for internal use. We don't recommend user making any changes on it. Note that these files cannot be modified via the OA&M Framework, but rather must be modified on the VoiceGenie machines directly.

The following are the configuration parameters for profile.phonetic:

Parameter	**Description**
SERVER_NAME	Phonetic Gateway server name (or IP Address)
SERVER_PORT	Phonetic Gateway server port number. This is the parameter "pdsgateway" in the service file on the Phonetics Gateway
SECOND_SERVER_NAME	Backup GW name (IP Address)
SECOND_SERVER_PORT	Backup GW Port
USER_NAME	This is the user name we send to the Phonetics Gateway as the user of the Phonetics Gateway. We typically set this to "PW".
BUILD_NUMBER	This is the Phonetics Operator Build number. It must match the number of the Phonetics server being connected to.
SERVER_TYPE	This is the Phonetics server type we're connecting to. It should always be set to PSRT_GWCLIENT
RECEIVE_TIMEOUT	This is the timeout we have for waiting for a reply from the Phonetics server. If the server doesn't reply within this time, we would consider the request to have failed.
CONNECT_TIMEOUT	This is the socket timeout we use to connect to Phonetics Gateway.
EXIT_VSE_DISCONNECT	Process should restart once it detects VSE disconnected.
MAX_CANDIDATE	Maximum number of candidates to be returned.
EXIT_VSE_DISCONNECT	A flag indicates if Phonetic Client should exit after it detect a VSE socket disconnection. If not set the default value is true

The following are the parameters used to configure the phoneticclnt_imt_logger.cfg:

Parameter	Description
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.
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.
mtinternal.max_sessions	Defines the maximum MTInternal sessions Default is 400
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.
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.
mtinternal.receive_rate_alarm	If greater than 0, 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.
mtinternal.receive_savedata	If specified, received data is saved under the specified directory.
mtinternal.rtp_max_port	The maximum port range for RTP sockets in MTInternal
mtinternal.rtp_min_port	The minimum port range for RTP sockets in MTInternal
mtinternal.transmit_interval	Defines a constant transmission interval in milliseconds. If set to 0, packets will be sent as soon as data arrives.
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.
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.
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.
mtinternal.transmit_rate_alar	If greater than 0, minor alarm is generated if the

m	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.
mtinternal.transmit_savedata	If specified, utterance is saved under the specified directory.
cmp.log_file	The name of the log file for this process. Use full path. default is: /usr/local/srm-server/logs/log.phoneticclnt
cmp.pid_option	This parameter will indicate if you wish to create a new log file every time the process restarts Default value is FALSE
cmp.size_option	This parameter determines whether the log files are rotated by size or time. Set to TRUE to roll over by size, FALSE to roll over by time. Default value is TRUE
cmp.rollover_size	If cmp.size_option is set to TRUE this arameter should be specified. The size limit (in megabytes) for roll over. Default value is 10
cmp.num_rollover_files	If cmp.size_option is set to TRUE this arameter should be specified. The number of files to roll through, i.e they are overwritten Default value is 5
cmp.rollover_mins	If cmp.size_option is set to FALSE this parameter determines how often the files are rolled over. The value is the time interval (in minutes) to roll over Default value is 1440
cmp.rollover_time	If cmp.size_option is set to FALSE and cmp.rollover_mins is not specified, then this parameter determines when the files are rolled over. The time of day (using a 24 hour clock, values from 0:00 to 23:59) to roll over Default value is 4:00
cmp.log_sinks	The sinks that will be used by this component. Possible sinks are: FILE, SYSLOG, EMAIL For this component, we should use FILE
cmp.email	This parameter is only used if the EMAIL sink is specified By default this parameter is commented out. The value should be people's e-mail address
cmp.trace_flag	This flag determines if logs at level log_5 (tracing/debugging) should be logged It overrides the setting in cmp.log_5 Default value is FALSE
cmp.metrics	The log mask for metrics data, default is to log metrics to a metrics sink and send some upstream

	Default value is 0
cmp.log_0	The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_1	The log mask for data logged at log level 1. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_2	The log mask for data logged at log level 2. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_3	The log mask for data logged at log level 3. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_4	The log mask for data logged at log level 4. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_5	The log mask for data logged at log level 5. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.

5.2 RealSpeak Configuration

A different speechify client configuration file is required to be used for each of the RealSpeak voices. Each of the voices would require their own entry in the TTS_CLIENT_LIST, as described 4.1.

Note: A Linux RealSpeak 4.0.4 server, ttsserver, will terminate if a wrong formated User Dictionary is loaded.

To generate a RealSpeak 4.0.4 user dictionary supported by Linux, please follow the following steps:

1. Read Appendix C in the Telecom RealSpeak/Host Programmer's Guide coming with the RealSpeak 4.0.4 installation to understand the steps to create and use user dictionaries.

2.The tool that generates RealSpeak User Dictionary, rsude.exe, is only installed in a Windows version of RealSpeak. This tool provides online help documentation for detailed instructions.

3. After the user dictionary is generated in rsude.exe, please Save As Binary LSB UCS-4 Format (*.bdc) to avoid this problem.

For configuration files of all the RealSpeech voices are located in /usr/local/srm-server/config/. There are 2 files needed for the configuration: realspeak_tts_host_<voice_name>.cfg, and realspeak_imt_logger.cfg . The same realspeak_imt_logger.cfg file can be shared across all the different voices. Note that these files cannot be modified via the OA&M Framework, but rather they must be modified on the VoiceGenie machines directly.

The following are the configuration parameters for realspeak_tts_host_<voice_name>.cfg:

Parameter	**Description**
IP_ADDRESS	The machine name or IP address of the machine with the TTS Engine for the TTS client to connect to.
PORT	The port number of the TTS Engine instance with the required voice installed. Note this port must match with the port number specified by the RealSpeak server configuration file-swittsclient.cfg.
AUDIOFORMAT	Audio format. Valid values are alaw or ulaw
ROOT_DICT	Absolute path of root dictionary
MAIN_DICT	Absolute path of main dictionary
ABBREVIATION_DICT	Absolute path of the abbreviation dictionary
CONTENT_TYPE	This is the content type to be sent to the Speechify engine in the SWIttsSpeak() function call. This value defaults to application/synthesis+ssml. This parameter should not be changed except when using Japanese, in which case it should be set to one of the following: CONTENT_TYPE=charset=UTF-8 CONTENT_TYPE=charset=Shift-JIS CONTENT_TYPE=charset=EUC

The following are the parameters used to configure the realspeak_imt_logger.cfg:

Parameter	Description
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.
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.
mtinternal.max_sessions	Defines the maximum MTInternal sessions Default is 400
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.
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.
mtinternal.receive_rate_alarm	If greater than 0, 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.
mtinternal.receive_savedata	If specified, received data is saved under the specified directory.
mtinternal.rtp_max_port	The maximum port range for RTP sockets in MTInternal
mtinternal.rtp_min_port	The minimum port range for RTP sockets in MTInternal
mtinternal.transmit_interval	Defines a constant transmission interval in milliseconds. If set to 0, packets will be sent as soon as data arrives.
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.
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.
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.
mtinternal.transmit_rate_alarm	If greater than 0, 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.
mtinternal.transmit_savedata	If specified, utterance is saved under the specified directory.
cmp.log_file	The name of the log file for this process. Use full path. default is: /usr/local/srm-server/logs/log.realspeak_host4
cmp.pid_option	This parameter will indicate if you wish to create a new log file every time the process restarts Default value is FALSE
cmp.size_option	This parameter determines whether the log files are rotated by size or time. Set to TRUE to roll over by size, FALSE to roll over by time. Default value is TRUE

cmp.rollover_size	If cmp.size_option is set to TRUE this arameter should be specified. The size limit (in megabytes) for roll over. Default value is 10
cmp.num_rollover_files	If cmp.size_option is set to TRUE this arameter should be specified. The number of files to roll through, i.e they are ove rwritten Default value is 5
cmp.rollover_mins	If cmp.size_option is set to FALSE this parameter determines how often the files are rolled over. The value is the time interval (in minutes) to roll over Default value is 1440
cmp.rollover_time	If cmp.size_option is set to FALSE and cmp.rollover_mins is not specified, then this parameter determines when the files are rolled over. The time of day (using a 24 hour clock, values from 0:00 to 23:59) to roll over Default value is 4:00
cmp.log_sinks	The sinks that will be used by this component. Possible sinks are: FILE, SYSLOG, EMAIL For this component, we should use FILE
cmp.email	This parameter is only used if the EMAIL sink is specified By default this parameter is commented out. The value should be people's e-mail address
cmp.trace_flag	This flag determines if logs at level log_5 (tracing/debugging) should be logged It overrides the setting in cmp.log_5 Default value is FALSE
cmp.metrics	The log mask for metrics data, default is to log metrics to a metrics sink and send some upstream Default value is 0
cmp.log_0	The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_1	The log mask for data logged at log level 1. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_2	The log mask for data logged at log level 2. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_3	The log mask for data logged at log level 3. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_4	The log mask for data logged at log level 4. The log mask for data logged at log level 0. Ple ase consult VoiceGenie before change the default value.
cmp.log_5	The log mask for data logged at log level 5. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.

5.3 Rhetorical Configuration

For each Rhetorical Voice, a different Rhetorical client configuration file need to used for each of the voices. Each of the voices would require their own entry in the TTS_CLIENT_LIST, as described in 4.1.

For configuration files for all the Rhetorical voices are located in /usr/local/srm-server/config/. There are 3 files needed for the configuration: rvoice_tts_host_<voice_name>.cfg, rvoice.env and rvoice_imt_logger.cfg. The same rvoice_imt_logger.cfg file can be shared across all the different voices. Note that these files cannot be modified via the CMP, but rather they need to be modified on the VoiceGenie machines directly.

The following are the configuration parameters for rvoice_tts_host_<voice_name>.cfg. Note that the configuration file should have no spaces before or after the equals '=' signs. Also, the configuration file should be in Unix ASCII format only (ie. Windows edited files with CTRL-M's in them are invalid):

Parameter	**Description**
voice	This is the name of the required voice to be used which must be running on at least one of the servers listed in the “servers” configuration line.
servers	This is a semicolon delimited list of IP addresses that have rVoice Servers running on them. Each server is followed by a colon then the port number that the rVoice Server is listening on. For example: servers=rvoiceserver1:1314;rvoiceserver2:1314;172.0.0.1:1315
timeout	This is the time in milliseconds that the rVoice Client will attempt to contact a server before giving up.
init_timeout	This is the time in milliseconds that the rVoice Client will attempt to establish valid contact with available servers on start-up before giving up.
logdir	This is the directory where the Rhetorical Client log files should be written
loglevel	This controls the amount of logs to be generated. Unless we’re debugging problems with the Rhetorical Client, this should be set to 0000.

The following are the parameters used to configure the rvoice_imt_logger.cfg:

Parameter	Description
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.
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.
mtinternal.max_sessions	Defines the maximum MTInternal sessions Default is 400
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.
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.
mtinternal.receive_rate_alarm	If greater than 0, 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.
mtinternal.receive_savedata	If specified, received data is saved under the specified directory.
mtinternal.rtp_max_port	The maximum port range for RTP sockets in MTInternal
mtinternal.rtp_min_port	The minimum port range for RTP sockets in MTInternal
mtinternal.transmit_interval	Defines a constant transmission interval in milliseconds. If set to 0, packets will be sent as soon as data arrives.
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.
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.
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.
mtinternal.transmit_rate_alarm	If greater than 0, 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.
mtinternal.transmit_savedata	If specified, utterance is saved under the specified directory.
cmp.log_file	The name of the log file for this process. Use full path. default is: /usr/local/srm-server/logs/log.rhetorical
cmp.pid_option	This parameter will indicate if you wish to create a new log file every time the process restarts Default value is FALSE
cmp.size_option	This parameter determines whether the log files are rotated by size or time. Set to TRUE to roll over by size, FALSE to roll over by time. Default value is TRUE
cmp.rollover_size	If cmp.size_option is set to TRUE this arameter should be specified. The size limit (in megabytes) for roll over. Default value is 10
cmp.num_rollover_files	If cmp.size_option is set to TRUE this arameter should be specified. The number of files to roll through, i.e they are overwritten Default value is 5
cmp.rollover_mins	If cmp.size_option is set to FALSE this parameter determines how often the files are rolled over. The value is the time interval (in minutes) to roll over Default value is 1440
cmp.rollover_time	If cmp.size_option is set to FALSE and cmp.rollover_mins is not specified, then this parameter determines when the files are rolled over. The time of day (using a 24 hour clock, values from 0:00 to 23:59) to roll over Default value is 4:00
cmp.log_sinks	The sinks that will be used by this component. Possible sinks are: FILE, SYSLOG, EMAIL For this component, we should use FILE
cmp.email	This parameter is only used if the EMAIL sink is specified By default this parameter is commented out. The value should be people's e-mail address
cmp.trace_flag	This flag determines if logs at level log_5 (tracing/debugging) should be logged It overrides the setting in cmp.log_5 Default value is FALSE
cmp.metrics	The log mask for metrics data, default is to log metrics to a metrics sink and send some upstream Default value is 0

cmp.log_0	The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_1	The log mask for data logged at log level 1. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_2	The log mask for data logged at log level 2. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_3	The log mask for data logged at log level 3. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_4	The log mask for data logged at log level 4. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_5	The log mask for data logged at log level 5. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.

5.4 OSR 3.0 Configuration

OSR 3.0 legacy integration is only supported in Linux.

The configuration files for the OSR 3.0 are all located in /usr/local/srm-server/config/. There are 4 files needed for the configuration: speechworks2.cfg, osr.context, osr.env and spwx_imt_logger.cfg. For more information about the osr.context and the spwx_imt_logger.cfg files, please refer to the section above with ASR Client Provisioning. Note that these files cannot be modified via the OA&M Framework, but rather must be modified on the VoiceGenie machines directly.

The following are the parameters used to configure the speechworks2.cfg:

Parameter	Description
RunAsOSRClient	This parameter controls whether it runs as OSR client or all-in-one mode. If OSR is a client/server setup, this should be TRUE for the robustness reason. When this is set to true, the OSR client will call the SWIrecRecognizerCreate() and the SWIrecRecognizerDestroy() functions at the start and the end of each phone call
WebAddressOfTmpDir	This is used for off-board OSR server that can not access the grammars in "tmp" directory directory. If RunAsOSRClient equals to TRUE, this has to be set. This works in conjunction with the web server configured in the “Application Server” subsection under the “SRM Server” section of the SRM Users’ Guide, to provide an offboard OSR server with grammars found in the SRM Server
IMTConfigFile	This specifies the location of config file of the CMP logger
MaxNumOfLineLogWaveform	MaxNumOfLineLogWaveform controls the maximum number of concurrent calls that can save waveform. On production systems, the platform operator would want to keep this value to a small number so that waveform logging won’t affect system performance too mauch. This needs to work with the voicexml property swirec_suppress_waveform_logging set to 0, which can be set in defaults.vxml or application page. If MaxNumOfLineLogWaveform equals to 0, there is no limit at all (i.e. all lines will log waveform data). Default value: 0.
SupportServerSelection	SupportServerSelection controls whether or not application can select server with client/server setup. To support this, "SWIsvcServerSelectionMode" should be set to "explicit" in the User.xml on the client side. Default value: false
NomatchOnInvalidSlotValue	NomatchOnInvalidSlotValue controls which event the

	platform will throw if there is no valid slot value pair. If it is set to TRUE, a nomatch event will be thrown. Otherwise, an error.grammar.asr event will be thrown. Default value: false
AudioBufferThreshold	This parameter is used to control the minimum number of bytes of audio data to deliver to the endpointer. Default value: 800
ResetEpAcousticStatePerUtterance	ResetEpAcousticStatePerUtterance controls whether or not the platform will reset endpointer acoustic state per utterance. In most cases, it should be false. Default value: false
AllocateEPLicensePerCall	This controls whether or not to allocate endpointer license on a per-call basis. This facilitate better license sharing between servers, but it may lead to more delays in the endpointing of the first recognition session. Default value: false
LegacyResultFormat	This controls whether the OSR3 Client should return the result in the proprietary VG ASR result format. This should always be set to false Default value: false

5.5 AT&T NaturalVoices Configuration

AT&T NaturalVoices integration is only supported on Linux.

The configuration files for AT&T NaturalVoices are all located in /usr/local/srm-server/config/. There are 2 files needed for the configuration: nv_tts_host_all.cfg and nv_imt_logger.cfg. Note that these files cannot be modified via the OA&M Framework, but rather they must be modified on the VoiceGenie machines directly.

The following are the configuration parameters for nv_tts_host_all.cfg:

Parameter	**Description**
IP_ADDRESS	The machine name or IP address of the machine with the TTS Engine for the TTS client to connect to.
PORT	The port number of the TTS Engine instance with the required voice installed.
AUDIOFORMAT	Audio format. Valid values are alaw or ulaw
PING_INTERVAL	The interval to ping the TTS Engine.

The following are the parameters used to configure the nv_imt_logger.cfg:

Parameter	Description
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.
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.
mtinternal.max_sessions	Defines the maximum MTInternal sessions Default is 400
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.
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.
mtinternal.receive_rate_alarm	If greater than 0, 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.
mtinternal.receive_savedata	If specified, received data is saved under the specified directory.
mtinternal.rtp_max_port	The maximum port range for RTP sockets in MTInternal
mtinternal.rtp_min_port	The minimum port range for RTP sockets in MTInternal
mtinternal.transmit_interval	Defines a constant transmission interval in milliseconds. If set to 0, packets will be sent as soon as data arrives.
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.
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.
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.
mtinternal.transmit_rate_alarm	If greater than 0, 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.
mtinternal.transmit_savedata	If specified, utterance is saved under the specified directory.
cmp.log_file	The name of the log file for this process. Use full path. default is: /usr/local/srm-server/logs/log.attnv_tts_host
cmp.pid_option	This parameter will indicate if you wish to create a new log file every time the process restarts Default value is FALSE
cmp.size_option	This parameter determines whether the log files are rotated by size or time. Set to TRUE to roll over by size, FALSE to roll over by time. Default value is TRUE
cmp.rollover_size	If cmp.size_option is set to TRUE this arameter should be specified. The size limit (in megabytes) for roll over. Default value is 10

cmp.num_rollover_files	If cmp.size_option is set to TRUE this arameter should be specified. The number of files to roll through, i.e they are overwritten Default value is 5
cmp.rollover_mins	If cmp.size_option is set to FALSE this parameter determines how often the files are rolled over. The value is the time interval (in minutes) to roll over Default value is 1440
cmp.rollover_time	If cmp.size_option is set to FALSE and cmp.rollover_mins is not specified, then this parameter determines when the files are rolled over. The time of day (using a 24 hour clock, values from 0:00 to 23:59) to roll over Default value is 4:00
cmp.log_sinks	The sinks that will be used by this component. Possible sinks are: FILE, SYSLOG, EMAIL For this component, we should use FILE
cmp.email	This parameter is only used if the EMAIL sink is specified By default this parameter is commented out. The value should be people's e-mail address
cmp.trace_flag	This flag determines if logs at level log_5 (tracing/debugging) should be logged It overrides the setting in cmp.log_5 Default value is FALSE
cmp.metrics	The log mask for metrics data, default is to log metrics to a metrics sink and send some upstream Default value is 0
cmp.log_0	The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_1	The log mask for data logged at log level 1. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_2	The log mask for data logged at log level 2. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_3	The log mask for data logged at log level 3. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_4	The log mask for data logged at log level 4. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.
cmp.log_5	The log mask for data logged at log level 5. The log mask for data logged at log level 0. Please consult VoiceGenie before change the default value.

6 Metrics Entries

Label	Description
asr_audioready Logged by: SRMS Level: all	**Audio Channel Established** This is logged when the ASR client is ready to receive audio. **The format is:** asr_audioready *<Client>*| *<CLIENT_ID>*| *<STATE>*| *<Dest>*| *<DEST>* *<Client>* It is a lable. *<CLIENT_ID>* It is an integer representing an ASR client *<STATE>* The state of the srmserver when this metric is logged. *<Dest>* It is a lable. *<DEST>* This is the IP address and port number of the RTP destination in format of dst_rtp=IP address:port **Example:** asr_audioready Client 0 [LOADING] - Dest: dst_rtp=10.0.0.114:30000
asr_bargein Logged by: SRMS Level: all	**Start Of Speech Is Detected** This is logged when a start-of-speech is detected by the ASR. **The format is:** asr_bargein *<Client>*| *<CLIENT_ID>*| *<STATE>*| *<Info>*| *<INFO>* *<Client>* It is a lable. *<CLIENT_ID>*

	It is an integer representing an ASR client *<STATE>* The state of the srmserver when this metric is logged. *<Status>* It is a lable. *<STATUS>* This is an integer representation whether the request was successful or not. 0 always represents successful. *<Info>* It is a lable. *<INFO>* This logs the recognition result, and failure reason representing extra info sent back to the Media Platform. **Example:** asr_bargein Client 0 [RECOGNIZING] - Info: _bargein_
asr_done Logged by: SRMS Level: all	**Recognition Completed** This is logged when a recognition has been completed. **The format is:** asr_done *<Client>*| *<CLIENT_ID>*| *<STATE>*| *<Result>*| *<RESULT>* *<Client>* It is a lable. *<CLIENT_ID>* It is an integer representing an ASR client *<STATE>* The state of the srmserver when this metric is logged. *<Status>*

	It is a lable. *<STATUS>* This is an integer representation whether the request was successful or not. 0 always represents successful. *<Result>* It is a lable. *<RESULT>* This logs the recognition result sent back to the Media Platform. **Example:** asr_done Client 0 [RECOGNIZING] - Result: <?xml version='1.0'?><result><interpretation grammar="session:_gram1" confidence="93"><input mode="speech">hello</input><instance><X confidence=" 93">valueX</X><SWI_meaning>{X:valueX}</SWI_meaning></instance></interpretation></result>
asr_fail Logged by: SRMS Level: all	**Recognition Failed** This is logged when a recognition cannot be completed, either due to a nomatch, or due to a failure in the recognizer. **The format is:** asr_fail *<Client>*|*<CLIENT_ID>*|*<STATE>*|*<Info>*|*<INFO>* *<Client>* It is a lable. *<CLIENT_ID>* It is an integer representing an ASR client *<STATE>* The state of the srmserver when this metric is logged. *<Status>* It is a lable. *<STATUS>* This is an integer representation whether the request was successful or not. 0 always represents successful.

<table>
<tr><td></td><td>

<Reason>

It is a lable.

<REASON>

This logs the recognition failure reason representing extra info sent back to the Media Platform.

<hr>

Example:
asr_fail Client 0 [RECOGNIZING] - Reason: -914
</td></tr>
<tr><td>asrload_begin
Logged by: SRMS
Level: all</td><td>

Load Grammar Request Sent
This is logged when a grammar load request is sent to the ASR client.

<hr>

The format is:
asrload_begin *<Client>*| *<CLIENT_ID>* | *<STATE>*| *<Context>*| *<CONTEXT_FILE>*

<Client>

It is a lable.

<CLIENT_ID>

It is an integer representing an ASR client

<STATE>

The state of the srmserver when this metric is logged.

<Context>

It is a lable.

<CONTEXT_FILE>

This is the path of temporary asr context file used to store grammar and recognition information.

<hr>

Example:
asrload_begin Client 0 [USED] - Context: /usr/local/srm-server/tmp/00020023-101D5B3D-asr/asrcontext.00020023-101D5B3D-asr.0
</td></tr>
<tr><td>asrload_end
Logged by: SRMS
Level: all</td><td>

Load Grammar Completed
This is logged when loading grammars has finished.

<hr>

The format is:
asrload_end *<Client>*| *<CLIENT_ID>* | *<STATE>*| *<Status>*| *<STATUS>*
</td></tr>
</table>

	<Client> It is a lable. *<CLIENT_ID>* It is an integer representing an ASR client *<STATE>* The state of the srmserver when this metric is logged. *<Status>* It is a lable. *<STATUS>* This is an integer representation whether the request was successful or not. 0 always represents successful. **Example:** asrload_end Client 0 [LOADING] - Status: 0
asrstart_begin Logged by: SRMS Level: all	**Recognition Request Sent** This is logged when the ASR starts to perform recognition. **The format is:** asrstart_begin *<Client>*| *<CLIENT_ID>* | *<STATE>*| *<Status>*| *<STATUS>* *<Client>* It is a lable. *<CLIENT_ID>* It is an integer representing an ASR client *<STATE>* The state of the srmserver when this metric is logged. **Example:** asrstart_begin Client 0 [LOADING]
asrstart_end Logged by: SRMS	**Engine Started Recognition** This is logged when the ASR has started.

Level: all	**The format is:** asrstart_end *<Client>*| *<CLIENT_ID>*| *<STATE>* *<Client>* It is a lable. *<CLIENT_ID>* It is an integer representing an ASR client *<STATE>* The state of the srmserver when this metric is logged. *<Status>* It is a lable. *<STATUS>* This is an integer representation whether the request was successful or not. 0 always represents successful. **Example:** asrstart_end Client 0 [STARTING] - Status: 0
asrstop_begin Logged by: SRMS Level: all	**Stop Recognition Request Sent** This is logged when the ASR is stopped by the Media Platform. **The format is:** asrstop_begin *<Client>*| *<CLIENT_ID>*| *<STATE>*| *<Status>* *<STATUS>*| *<Reply>*| *<REPLY>* *<Client>* It is a lable. *<CLIENT_ID>* It is an integer representing an ASR client *<STATE>* The state of the srmserver when this metric is logged. *<Reply>*

	It is a lable. *<REPLY>* This indicates if a reply messge to the stop requst is reqired. true: reply required; false: no reply reqired. **Example:** asrstop_begin Client 0 [RECOGNIZING] - Reply: true
asrstop_end Logged by: SRMS Level: all	**Engine Started Recognition** This is logged when the ASR has started. **The format is:** asrstop_end *<Client>*| *<CLIENT_ID>* | *<STATE>* *<Client>* It is a lable. *<CLIENT_ID>* It is an integer representing an ASR client *<STATE>* The state of the srmserver when this metric is logged. *<Status>* It is a lable. *<STATUS>* This is an integer representation whether the request was successful or not. 0 always represents successful. **Example:** asrstop_end Client 0 [STOPPING] - Status: 0
asrunload_begin Logged by: SRMS Level: all	**Free ASR Resource Request Sent** This is logged when an ASR resource is being freed up. **The format is:** asrunload_begin *<Client>*| *<CLIENT_ID>* | *<STATE>* *<Client>* It is a lable.

	<CLIENT_ID> It is an integer representing an ASR client *<STATE>* The state of the srmserver when this metric is logged. **Example:** asrunload_begin Client 0 [USED]
asrunload_end Logged by: SRMS Level: all	**Free ASR Resource Completed** This is logged when an ASR resource has been freed up. **The format is:** asrunload_end *<Client>*| *<CLIENT_ID>* | *<STATE>*| *<Status>*| *<STATUS>* *<Client>* It is a lable. *<CLIENT_ID>* It is an integer representing an ASR client *<STATE>* The state of the srmserver when this metric is logged. *<Status>* It is a lable. *<STATUS>* This is an integer representation whether the request was successful or not. 0 always represents successful. **Example:** asrunload_end Client 0 [UNLOAD] - Status: 0
tts_done Logged by: SRMS Level: all	**A TTS Session Completed** This is logged when a TTS request is completed. **The format is:** tts_done *<Host>*| *<HOST_NAME>*| *<STATE>*| *<Status>*| *<STATUS>* *<Host>*

	It is a lable. *<HOST_NAME>* It represents a TTS client *<STATE>* The state of the srmserver when this metric is logged. *<Status>* It is a lable. *<STATUS>* This is an integer representation whether the request was successful or not. 0 always represents successful. **Example:** tts_done Host H0009 [SPEAKING] - Status: 0
tts_play Logged by: SRMS Level: all	**Start A TTS Request** This is logged when a TTS request is sent. **The format is:** tts_play *<Host>*| *<HOST_NAME>*| *<STATE>*| *<Status>*| *<STATUS>* *<Host>* It is a lable. *<HOST_NAME>* It represents a TTS client *<STATE>* The state of the srmserver when this metric is logged. *<Status>* It is a lable. *<STATUS>* This is an integer representation whether the request was successful or

	not. 0 always represents successful. **Example:** tts_play Host H0009 [SPEAK] - Status: 0
ttsstop_end Logged by: SRMS Level: all	**A TTS Session Is Stopped** This is logged when the TTS synthesis has been stopped. **The format is:** ttsstop_end *<Host>*| *<HOST_NAME>*| *<STATE>*| *<Status>*| *<STATUS>* *<Host>* It is a lable. *<HOST_NAME>* It represents a TTS client *<STATE>* The state of the srmserver when this metric is logged. *<Status>* It is a lable. *<STATUS>* It is an integer representation whether the request was successful or not. 0 always represents successful **Example:** ttsstop_end Host H0009 [IDLE] - Status: 0
ttsstop_engin Logged by: SRMS Level: all	**Sending Request To Stop A TTS Session** This is logged when the TTS synthesis is requested to be stopped. **The format is:** ttsstop_engin *<Host>*| *<HOST_NAME>*| *<STATE>*| *<Reason>*| *<REASON>* *<Host>* It is a lable. *<HOST_NAME>* It represents a TTS client

	<STATE> The state of the srmserver when this metric is logged. *<Reason>* It is a lable. It is optional *<REASON>* It is the reason why a stop request was issued. It is optional. If it appears, it is one of "reset", "session closed", "session disconnected" and "barge-in-occurred". **Example:** ttsstop_begin Host H0009 [SPEAKING]

7 SNMP Traps

Name	OID	Type	Description
Started	.1.3.6.1.4.1.7469.3.9.12.1.1.1	Scalar	The time the server was started
ASRDisplayName	.1.3.6.1.4.1.7469.3.9.13.1.1.x	Tabular	The name of the engine
ASRURI	.1.3.6.1.4.1.7469.3.9.13.1.2.x	Tabular	The URI used to access the engine
ASRAvailable	.1.3.6.1.4.1.7469.3.9.13.1.3.x	Tabular	The number of available clients
ASRTotal	.1.3.6.1.4.1.7469.3.9.13.1.4.x	Tabular	The total number of clients
ASRPeak	.1.3.6.1.4.1.7469.3.9.13.1.5.x	Tabular	The lowest number of clients that were available at any given
ASRDied	.1.3.6.1.4.1.7469.3.9.13.1.6.x	Tabular	The number of clients that died unexpectedly
ASRFailed	.1.3.6.1.4.1.7469.3.9.13.1.7.x	Tabular	The number of sessions that ended with a failure code
ASRSuccess	.1.3.6.1.4.1.7469.3.9.13.1.8.x	Tabular	The number of successfully completed sessions
TTSDisplayName	.1.3.6.1.4.1.7469.3.9.13.1.9.x	Tabular	The name of the engine
TTSURI	.1.3.6.1.4.1.7469.3.9.13.1.10.x	Tabular	The URI used to access the engine
TTSAvailable	.1.3.6.1.4.1.7469.3.9.13.1.11.x	Tabular	The number of available clients
TTSTotal	.1.3.6.1.4.1.7469.3.9.13.1.12.x	Tabular	The total number of clients
TTSPeak	.1.3.6.1.4.1.7469.3.9.13.1.13.x	Tabular	The lowest number of clients that were available at any given time
TTSDied	.1.3.6.1.4.1.7469.3.9.13.1.14.x	Tabular	The number of clients that died unexpectedly
TTSFailed	.1.3.6.1.4.1.7469.3.9.13.1.15.x	Tabular	The number of sessions that ended with a failure code
TTSSuccess	.1.3.6.1.4.1.7469.3.9.13.1.16.x	Tabular	The number of successfully completed sessions
PMName	.1.3.6.1.4.1.7469.3.9.13.1.17.x	Tabular	The name of the protocol module
PMOpenMin	.1.3.6.1.4.1.7469.3.9.13.1.18.x	Tabular	The minimum amount of time an open request took
PMCloseMin	.1.3.6.1.4.1.7469.3.9.13.1.19.x	Tabular	The minimum amount of time a close request took
PMStopMin	.1.3.6.1.4.1.7469.3.9.13.1.20.x	Tabular	The minimum amount of time a stop request took
PMSetParamsMin	.1.3.6.1.4.1.7469.3.9.13.1.21.x	Tabular	The minimum amount of time a set params request took
PMGetParamsMin	.1.3.6.1.4.1.7469.3.9.13.1.22.x	Tabular	The minimum amount of time a get params request took
PMLoadGrammarMin	.1.3.6.1.4.1.7469.3.9.13.1.23.x	Tabular	The minimum amount of time a load grammar request took
PMRecognizeMin	.1.3.6.1.4.1.7469.3.9.13.1.24.x	Tabular	The minimum amount of time a recognize request took
PMPromptDoneMin	.1.3.6.1.4.1.7469.3.9.13.1.25.x	Tabular	The minimum amount of time a prompt done request took
PMSpeakMin	.1.3.6.1.4.1.7469.3.9.13.1.26.x	Tabular	The minimum amount of time a speak request took

PMPauseMin	.1.3.6.1.4.1.7469.3.9.13.1.27.x	Tabular	The minimum amount of time a pause request took
PMResumeMin	.1.3.6.1.4.1.7469.3.9.13.1.28.x	Tabular	The minimum amount of time a resume request took
PMControlMin	.1.3.6.1.4.1.7469.3.9.13.1.29.x	Tabular	The minimum amount of time a control request took
PMBargeInMin	.1.3.6.1.4.1.7469.3.9.13.1.30.x	Tabular	The minimum amount of time a barge in occurred request took
PMOpenMax	.1.3.6.1.4.1.7469.3.9.13.1.31.x	Tabular	The maximum amount of time an open request took
PMCloseMax	.1.3.6.1.4.1.7469.3.9.13.1.32.x	Tabular	The maximum amount of time a close request took
PMStopMax	.1.3.6.1.4.1.7469.3.9.13.1.33.x	Tabular	The maximum amount of time a stop request took
PMSetParamsMax	.1.3.6.1.4.1.7469.3.9.13.1.34.x	Tabular	The maximum amount of time a set params request took
PMGetParamsMax	.1.3.6.1.4.1.7469.3.9.13.1.35.x	Tabular	The maximum amount of time a get params request took
PMLoadGrammarMax	.1.3.6.1.4.1.7469.3.9.13.1.36.x	Tabular	The maximum amount of time a load grammar request took
PMRecognizeMax	.1.3.6.1.4.1.7469.3.9.13.1.37.x	Tabular	The maximum amount of time a recognize request took
PMPromptDoneMax	.1.3.6.1.4.1.7469.3.9.13.1.38.x	Tabular	The maximum amount of time a prompt done request took
PMSpeakMax	.1.3.6.1.4.1.7469.3.9.13.1.39.x	Tabular	The maximum amount of time a speak request took
PMPauseMax	.1.3.6.1.4.1.7469.3.9.13.1.40.x	Tabular	The maximum amount of time a pause request took
PMResumeMax	.1.3.6.1.4.1.7469.3.9.13.1.41.x	Tabular	The maximum amount of time a resume request took
PMControlMax	.1.3.6.1.4.1.7469.3.9.13.1.42.x	Tabular	The maximum amount of time a control request took
PMBargeInMax	.1.3.6.1.4.1.7469.3.9.13.1.43.x	Tabular	The maximum amount of time a barge in occurred request took
PMOpenAvg	.1.3.6.1.4.1.7469.3.9.13.1.44.x	Tabular	The average amount of time an open request took
PMCloseAvg	.1.3.6.1.4.1.7469.3.9.13.1.45.x	Tabular	The average amount of time a close request took
PMStopAvg	.1.3.6.1.4.1.7469.3.9.13.1.46.x	Tabular	The average amount of time a stop request took
PMSetParamsAvg	.1.3.6.1.4.1.7469.3.9.13.1.47.x	Tabular	The average amount of time a set params request took
PMGetParamsAvg	.1.3.6.1.4.1.7469.3.9.13.1.48.x	Tabular	The average amount of time a get params request took
PMLoadGrammarAvg	.1.3.6.1.4.1.7469.3.9.13.1.49.x	Tabular	The average amount of time a load grammar request took
PMRecognizeAvg	.1.3.6.1.4.1.7469.3.9.13.1.50.x	Tabular	The average amount of time a recognize request took
PMPromptDoneAvg	.1.3.6.1.4.1.7469.3.9.13.1.51.x	Tabular	The average amount of time a prompt done request took

PMSpeakAvg	.1.3.6.1.4.1.7469.3.9.13.1.52.x	Tabular	The average amount of time a speak request took
PMPauseAvg	.1.3.6.1.4.1.7469.3.9.13.1.53.x	Tabular	The average amount of time a pause request took
PMResumeAvg	.1.3.6.1.4.1.7469.3.9.13.1.54.x	Tabular	The average amount of time a resume request took
PMControlAvg	.1.3.6.1.4.1.7469.3.9.13.1.55.x	Tabular	The average amount of time a control request took
PMBargeInAvg	.1.3.6.1.4.1.7469.3.9.13.1.56.x	Tabular	The average amount of time a barge in occurred request took

8 Alarms

8.1 SRM Client

Alarm#	Level	Definition and Possible Message/Info	Impacts	Causes	Detailed Recommended Actions
061003E9	CRIT	Invalid engine type specified	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Configuration error	Check VRM Client configuration, each engine must have a type which must listed in vrm.client.enginetypelist
061003EA	CRIT	Invalid engine URI specified	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Configuration error	Check VRM Client configuration
061003EB	CRIT	Invalid engine entry in config file	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Configuration error	Check VRM Client configuration, Platform must configurea at least one server or backup server
061003EC	CRIT	Invalid IP or port for the engine	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Configuration error	Check VRM Client configuration and ASR/TTS server configuration
061003ED	CRIT	Engine list is empty	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Configuration error	Check VRM Client configuration
061003EE	CRIT	EROR when parsing engine info	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Configuration error	Check VRM Client configuration
061003EF	CRIT	The eng type list is missing or empty	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Configuration error	Check VRM Client configuration
061003F0	CRIT	EROR creating stack	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Configuration error	Check VRM Client configuration
061003F1	CRIT	EROR initializing the engine type map	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is	Configuration error	Check VRM Client configuration

			restarted		
061003F2	CRIT	EROR initializing the stack	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Software error	Report to VG
061003F3	CRIT	EROR initializing the request manager	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Software error	Report to VG
061003F4	CRIT	EROR initializing connection manager	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Software error	Report to VG
061003F5	CRIT	EROR initializing the stack handler	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Software error	Report to VG
061003F5	CRIT	EROR to read vrmclient provision file	Any ASR/TTS will fail until ASR/TTS is fixed and VGPlatform is restarted	Configuration error	Check VRM Client configuration in cm_provision.dat
061007D1	EROR	Failed to obtain information about the file	Any ASR/TTS will fail for current call	Software error	Report to VG
061007D2	EROR	Grammar file size after stripping header is less than 0	Any ASR will fail for current call	Software error	Report to VG
061007D3	EROR	Unable to find grammar file	Any ASR will fail for current call	Software error	Report to VG
061007D4	EROR	Unable to locate information in the grammar file	Any ASR will fail for current call	Software error	Report to VG
061007D5	EROR	Failed to allocate memory	Any ASR/TTS will fail for current call	Server error	Check system resources
061007D6	EROR	Failed to read from grammar file	Any ASR will fail for current call	System error	Check server disk
061007D7	EROR	Failed to connect to server	Any ASR/TTS will fail until ASR/TTS server is up, no need to restart VG Platform	Configuration error	Check VRM Client configuration, Check if server is up.
061007D8	EROR	Failed to find info about server	Any ASR/TTS will fail until ASR/TTS server is up, no need to restart VG Platform	Configuration error	Check VRM Client configuration
061007D9	EROR	Invalid configuration for the parameter	Any ASR/TTS will fail until configuration is	Configuration error	Check VRM Client configuration

			fixed and VGPlatform is restarted		
061007DA	EROR	Unable to read the grammar base path	Any ASR/TTS will fail until configuration is fixed and VGPlatform is restarted	Configuration error	Check VRM Client configuration
061007DB	EROR	Unable to get the information for all servers	Any ASR/TTS will fail until problem is fixed and VGPlatform is restarted	Software error	Report to VG
061007DC	EROR	Unable to lookup connection information for the specified vrm engine	Any ASR/TTS will fail until configuration is fixed and VGPlatform is restarted	Configuration error	Check VRM Client configuration
061007DD	EROR	EROR when storing info about a session	Any ASR/TTS will fail for current call	Software error	Report to VG
061007DE	EROR	EROR when changing the state of a session	Any ASR/TTS will fail for current call	Software error	Report to VG
061007DF	EROR	Received an invalid timer event in the client library	Any ASR/TTS will fail for current call	Software error	Report to VG
061007E0	EROR	An EROR occurred while trying to remove info about a session	Any ASR/TTS will fail for current call	Software error	Report to VG
061007E1	EROR	The client timed out while waiting for a response	Any ASR/TTS will fail for current call	Server/Network error	Check server connections
061007E2	EROR	Received an unknown message id in a message	Any ASR/TTS will fail for current call	Software error	Report to VG
061007E3	EROR	Received a time out for an unknown event	Any ASR/TTS will fail for current call	Software error	Report to VG
061007E4	EROR	EROR looking up the request type for a session	Any ASR/TTS will fail for current call	Software error	Report to VG
061007E5	EROR	EROR removing a timer event	Any ASR/TTS will fail for current call	Software error	Report to VG
061007E6	EROR	Received an EROR code in the response message from the server	Any ASR/TTS will fail for current call	Server/Network error	Check server connections / ASR/TTS Engine software if it is running
061007E7	EROR	EROR when removing a request from the request manager	Any ASR/TTS will fail for current call	Software error	Report to VG
061007E8	EROR	Unable to find the request in internal structures	Any ASR/TTS will fail for current call	Software error	Report to VG
061007E9	EROR	Received an unexpected socket disconnect	Any ASR/TTS will fail for current call	Server/Network error	Check server connections / ASR/TTS Engine software if it is running
061007EA	EROR	Received an invalid audio codec	Any ASR/TTS will fail for current call	Configuration error	Check VRM Client configuration
061007EB	EROR	Unable to send request	Any ASR/TTS will	Server/Network error	Check server connections / ASR/TTS

			fail for current call		Engine software if it is running
061007EC	EROR	An EROR occurred while processing a message, most likely a memory allocation EROR	Any ASR/TTS will fail for current call	Server error	Check system resources
061007ED	EROR	A method that should not be called was called	Any ASR/TTS will fail for current call	Software error	Report to VG
061007EF	EROR	Lost a connection to an ASR/TTS server	Any ASR/TTS will fail for current call	Server/Network error	Check server connections / ASR/TTS Engine software if it is running
06100BB9	WAR N	Received an EROR message in the recognize session	Any ASR/TTS will fail for current call	Server/Network error	Check server connections / ASR/TTS Engine software if it is running
06100BBA	WAR N	Successfully reconnected to server	None		
06100BBB	WAR N	Received speak complete before speak response	None		
06100BBC	WAR N	the NLSML format is not correct.	Any ASR/TTS will fail for current call	Software error	Report to VG
06100BBD	WAR N	Decoding EROR failed	Any ASR/TTS will fail for current call	Software error	Report to VG
06100BBE	WAR N	The grammar file does not exist	Any ASR/TTS will fail for current call	Server error	Check system resources
06100BBF	WAR N	EROR is encountered when reading grammar file	Any ASR/TTS will fail for current call	Server error	Check system resources
06101389	INFO	VRM Client sends a TTS request	None		
0610138A	INFO	VRM Client receives a SPEAK-COMPLETE	None		
0610138B	INFO	VRM Client uses default TTS/ASR engine	None		
0610138C	INFO	VRM Client service quailty analyst metrics	None		

8.2 SRM Server

Alarm#	Level	Definition and Possible Message/Info	Impacts	Causes	Detailed Recommended Actions
063003E8	CRIT	EROR reading configuration file	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	Configuration error	Check server configurations
063003E9	CRIT	EROR opening a socket	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	Server/Network error	Check server connections
063003EA	CRIT	Unable to initialize OA&M	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	Software error	Report to VG
063003EB	CRIT	Unable to initialize the license manager	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	License is missing or expired	Verify Software Configuration or Liecense
063003EC	CRIT	Unable to load at least 1 protocol module	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	Configuration error	Check server configurations
063003ED	CRIT	Unable to load at least 1 engine module	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	Configuration error	Check server configurations
063007D0	CRIT	Engine had an internal failure	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	Software error	Check ASR/TTS engine if it is up and running
06300BB8	EROR	Invalid config parameter	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	Configuration error	Check server configurations
06300BB9	WARN	Invalid audio codec in sdp	ASR/TTS will fail for current call	Configuration error	Check ASR/TTS engine configuration
06300BBA	WARN	Invalid media line in sdp	ASR/TTS will fail for current call	Configuration error	Check ASR/TTS engine configuration
06300BBB	WARN	Invalid origin line in sdp	ASR/TTS will fail	Configuration error	Check ASR/TTS engine configuration

			for current call		
06300BBC	WARN	RML is stopping or pointer is invalid	ASR/TTS will fail for current call	Software error	Report to VG
06300BBD	WARN	Received invalid header in message	Any ASR/TTS will fail for the current call	Software error	Report to VG
06300BBE	WARN	Machine with IP not in restricted IP list attempted to connect to server	Any ASR/TTS will fail from the Machine with restricted IP. After configuration change VG Platform needs to be restarted	Configuration error	Check VRM Server configuration
06300BBF	WARN	Failed to create a directory	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	System error	Check server disk
06300BC0	WARN	Requested grammer is not found	Any ASR/TTS will fail for the current call	Configuration error	Check VRM Server/ASR Server configuration
06300BC1	WARN	Received request in the wrong state	Any ASR/TTS will fail for the current call	Software error	Report to VG
06300BC2	WARN	a client has disconnected from the engine	Any ASR/TTS will fail for the calls to the ASR/TTS engine disconnected until server is up. It is not necessary to restart VG Platform as it will recovers itself	Server/Network error	Check server connections / ASR/TTS Engine software if it is running
06300BC3	WARN	Failed to send a message on a socket	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	Server/Network error	Check server connections
06300BC4	WARN	Failed to create/write to file	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	System error	Check server disk
06300BC5	WARN	Received malformed result from legacy client	Any ASR/TTS will fail for the current call	Software error	Report to VG
06300BC6	WARN	Received an invalid session ID	Any ASR/TTS will fail for the current call	Software error	Report to VG
06300BC7	WARN	Received an invalid event in the engine	Any ASR/TTS will fail for the current	Software error	Report to VG

			call		
06300BC8	WARN	The environment variable is not set, set it to an appropriate value	Any ASR/TTS will fail until problem is fixed and VG Platform is restarted	Configuration error	Check VRM Server configuration
06300BC9	WARN	Received malformed message from legacy client	Any ASR/TTS will fail for the current call	Software error	Report to VG
06300BCA	WARN	Attempted to use invalid client id	Any ASR/TTS will fail for the current call	Software error	Report to VG
06300BCB	WARN	Received invalid/unknown message from client	Any ASR/TTS will fail for the current call	Software error	Report to VG
06300BCC	WARN	Unable to create/open socket	Any ASR/TTS will fail for the calls to the ASR/TTS engine disconnected until server is up. It is not necessary to restart VG Platform as it will recovers itself	Server/Network error	Check server connections / ASR/TTS Engine software if it is running
06300BCD	WARN	Received incorrect intialization status from legacy client	Any ASR/TTS will fail from the Machine with restricted IP. After configuration change VG Platform needs to be restarted	Server/Network error	Check server connections / ASR/TTS Engine software if it is running
06300BCE	WARN	Tried to free a license but there was no license allocated to the session	None	Software error	Report to VG
06300BCF	WARN	Engine attempted to update the number of resources to a negative number	None	Software error	Report to VG
06300BD0	WARN	Attempted to use a URI that is not registered with the server	Any ASR/TTS will fail for the current call	Configuration error	Check VRM Client configuration
06300BD1	WARN	Failed to register a data point with the data service	Any ASR/TTS will fail for the current call	Configuration error	Check VRM Server / CMP configuration
06300BD2	WARN	Open session idle timeout occured	Any ASR/TTS will fail for the current call	Server/Network error	Check server connections
06300FBB	NOTICE	Reconnected to the client	None	Server/Network error	Check server connections

8.3 RTSP-TTS Engine Error Events

The following error events will be thrown by the application, in case of an error:

Description of the Error	Error returned by the Stream Server	MRCP Response to Media Platform	Error thrown to the VXML application
URI cannot be found on RTSP server	DESCRIBE REPLY 404	SPEAK REPLY with 405	error.tts.noresource
RTSP server has a problem during stream setup or URI cannot be found on the server	SETUP REPLY 404	SPEAK REPLY with 405	error.tts.noresource
URI as specified in the VXML page is malformed.	PLAY request's URI is malformed (i.e., IP address cannot be determined or it is not a valid rtsp URI)	SPEAK REPLY with 408	error.tts.badtext
Errors happen during RTSP streaming, and RTSP server reports the error	PLAY REPLY 500 (Internal Server Error)	SPEAK REPLY with 407	error.tts
RTSP server disconnects from the RTSP client.	Socket Error happening while Playing	SPEAK-COMPLETE with completion cause 004	error.tts
RTSP client hasn't received audio data for a configurable amount of time.	Audio Data Receive timeout	SPEAK-COMPLETE with completion cause 004	error.tts
Cannot connect to the RTSP server	Initial Socket Connection Error when PLAY request is received	SPEAK REPLY with 407	error.tts
Internal error within VoiceGenie software	Any MTInternal Error	SPEAK REPLY with 407 or SPEAK-COMLPETE with completion cause 004 if state is SPEAKING	error.tts

8.4 TTY Error Events

The following error events will be thrown by the application, in case of an error:

Description of the Error	MRCP Response to Media Platform	Error thrown to the VXML application
URI cannot be found on SRMServer	SETUP REPLY with 404	error.asr.noresource
Maximum number of sessions exceed	SETUP REPLY with 407	error.asr.noresource
A grammar error has occurred	DEFINE-GRAMMAR REPLY with 407	error.grammar.asr
Internal errors happen during any step in RECOGNITION	MRCP Reply with 407 status code; RECOGNITION-COMPLETE with 006 completion clause	error.asr