



VoiceGenie 7.2

OA&M Framework

User's Guide

The information contained herein is proprietary and confidential and cannot be disclosed or duplicated without the prior written consent of Genesys Telecommunications Laboratories, Inc.

Copyright © 2000–2006 Genesys Telecommunications Laboratories, Inc. All rights reserved.

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: 09-2007



Table of Contents

Chapter 1	Introduction	9
Chapter 2	Components	11
	2.1 Database Server	11
	2.2 CMP Server (i.e. Management Server).....	11
	2.3 CMP Proxy & Command Line Console (CLC).....	11
	2.3.1 CMP Proxy.....	11
	2.3.2 Command Line Console (CLC)	12
	2.4 System Management Console (SMC).....	12
	2.5 VoiceGenie SNMP Agent	12
Chapter 3	Architecture	13
	3.1 All-in-One Architecture	13
	3.2 Distributed Architecture	14
Chapter 4	Installation and Licensing	17
Chapter 5	OA&M Framework Configuration	19
	5.1 Overview	19
	5.2 CMP Server	20
	5.2.1 Database Connection Settings.....	20
	5.2.2 CMP Server Redundancy Connection Settings	20
	5.3 CMP Proxy	21
	5.3.1 CMP Server Connection Settings.....	21
	5.3.2 Multiple Network Interface Card (NIC) Settings	22
	5.4 Configuration Synchronization Override	22
Chapter 6	CMP Server	25
	6.1 Overview	25
	6.2 CMP Server Startup and Options	25
	6.3 Starting, Stopping or Restarting the CMP Server.....	26
	6.4 CMP Server Health Information	26
	6.5 Reducing Load on the CMP Server / Database Server.....	27
	6.5.1 Reducing Stored Logging Data	27
	6.5.2 Reducing Stored Historical Data	27

	6.5.3 Reducing Stored Health Data.....	28
Chapter 7	Database Server	29
	7.1 Overview	29
	7.2 Accessing MySQL.....	30
	7.3 Database Summarization and Cleanup	30
	7.3.1 Database Summarization.....	30
	7.3.2 Database Cleanup	31
	7.3.3 Binary Log Cleanup	32
	7.4 Database Backup and Restoration	33
	7.4.1 Database Backup.....	33
	7.4.2 Database Restoration	33
	7.5 Database Monitoring and Alarming.....	34
	7.5.1 Database Queue Monitoring	34
	7.5.2 Database Replication Monitoring	35
Chapter 8	CMP Proxy	39
	8.1 Overview	39
	8.2 CMP Proxy Startup and Options.....	39
	8.3 Starting, Stopping or Restarting the CMP Proxy	40
	8.4 System CPU, Memory and Disk Usage Monitoring.....	40
	8.4.1 Exceeding CPU Usage Limits and Generated Alarms.....	41
	8.4.2 Exceeding Memory Usage Limits and Generated Alarms	42
	8.4.3 Exceeding Disk Usage Limits and Generated Alarms	42
	8.5 Process Startup and Recovery	43
	8.6 Process CPU and Memory Usage Monitoring.....	45
Chapter 9	Alarming and Logging	47
	9.1 Overview	47
	9.2 Severity Levels	47
	9.3 Predefined and User Defined Sink Locations.....	48
	9.3.1 Log File (FILE)	48
	9.3.2 Metrics File (METRICS)	50
	9.3.3 Upstream (UPSTREAM)	50
	9.3.4 Syslog (SYSLOG).....	51
	9.3.5 VoiceGenie SNMP (SNMP)	52
	9.3.6 Email (EMAIL).....	52
	9.3.7 Tools – Log Agent (LOGAGENT).....	52
	9.3.8 User Defined Sinks	53

	9.4 Log Filtering Masks	53
	9.5 Enabling or Disabling Tracing/Debugging.....	55
	9.6 Default Logging Configuration	56
	9.7 UTC Logging	56
	9.8 Log File Cleanup	57
	9.9 Log Queues.....	58
Appendix A	Log IDs: Modules and Specifiers	59
	A.1 CRIT, EROR, WARN, NOTE, INFO or DEBUG Log IDs.....	59
	A.1.1 CMP Modules.....	59
	A.1.2 Call Manager Modules.....	60
	A.1.3 VoiceXML Interpreter Modules	62
	A.1.4 Web Proxy (IProxy) Modules	63
	A.1.5 VRM Server Modules	64
	A.1.6 Common Modules	64
	A.1.7 CCP Modules	65
	A.1.8 Tools Modules.....	65
	A.1.9 Media Processing Component Modules	65
	A.1.10 NGI.....	66
	A.2 Metrics Log IDs	66
Appendix B	Database Schema.....	73
	B.1 NDM Database	73
	B.1.1 CallLog Table	75
	B.1.2 HistHealthStatus Table.....	75
	B.1.3 HistInitStatus Table	76
	B.1.4 HistProcessStatus Table	76
	B.1.5 Network Table	77
	B.1.6 CallData	78
	B.2 CallHistory Database	79
	B.2.1 HistAppDistribution.....	79
	B.2.2 HistCallLength.....	80
	B.2.3 HistCallVolume.....	81
Appendix C	VoiceGenie SNMP Traps.....	83
	C.1 CMP Proxy Traps.....	83
	C.2 External Agent Traps	92
Appendix D	Database Replication Setup	97
	D.1 Replication Architecture.....	97

	D.2 Configuring the Databases.....	98
	D.3 Resetting DB replication.....	102
	D.4 Fail-Over Handling.....	104
Appendix E	Developing a Custom Logging Sink	107
	E.1 VoiceGenie Log Sink Interface.....	107
	E.1.1 Destructor/Constructor.....	110
	E.1.2 GetVersion	111
	E.1.3 Initialize	111
	E.1.4 LogToSink	111
	E.2 Threading Issues.....	112
	E.3 Building and Linking the Library	113
	E.3.1 Linux.....	113
	E.3.2 Windows.....	113
Appendix F	Windows Process Names	115
Appendix G	Security Considerations	117
	G.1 How to use SSH as a Replacement for Telnet and FTP	117
	G.2 Hardening the SSH Server	119
	G.3 Disabling Unused Services.....	119
	G.4 General Password Construction Guidelines.....	120
	G.5 No World Writable Files	120
	G.6 Disabling CTRL+ALT+DELETE Shutdown (Linux).....	120
	G.7 System Passwords	121
	G.8 Database Passwords.....	121
	G.9 System Management Console (SMC) Passwords.....	122
Appendix H	CMP External Agents	123



Chapter

1

Introduction

This document is intended to act as a guide to the VoiceGenie OA&M (Operation, Administration & Management) Framework; also, it introduces the user to the various user interfaces of the OA&M Framework. The OA&M Framework is used by all products within the VoiceGenie product offering; this includes the VoiceXML Platform, MRCP Proxy and SIP Proxy to name a few.

This document includes a high level description of the components of the OA&M Framework, as well as details about its capabilities, configuration and setup.

The purpose of the OA&M Framework is to provide comprehensive operations, administration, management and monitoring capability to any VoiceGenie deployment. This is achieved by providing a communication path between all the components of the VoiceGenie solution. Through this communication infrastructure a number of services are built to provide various OA&M functionalities.

The OA&M Framework consists of five components:

- Database Server
- CMP Server (i.e. Management Server)
- CMP Proxy & Command Line Console (CLC)
- System Management Console (SMC)
- VoiceGenie SNMP Agent

The CMP Proxy & Command Line Console (CLC) must be installed on every VoiceGenie server that needs to be managed or monitored by the OA&M Framework. The CMP Proxy is responsible for monitoring system resources, while the CLC is a command line user interface to the OA&M Framework. The CMP Server provides the centralized logging and configuration capabilities and is responsible for clustering the various servers within a deployment, and the Database Server is where centralized information is stored. The System Management Console (SMC) is a web based tool used to administer the deployment. The VoiceGenie SNMP is VoiceGenie's SNMP

Agent that allows access to various real time statistics as well as SNMP traps produced by the OA&M Framework. The following section describes each component in further detail.



Chapter

2 Components

2.1 Database Server

The Database Server is where all centralized data is stored; this includes logging and configuration information. Currently, only MySQL Server and Oracle 10g is supported for this purpose.

2.2 CMP Server (i.e. Management Server)

The CMP Server, which is also referred to as the CMP Engine, is responsible for all centralized logging and configuration capability. The CMP Server should be deployed in a redundant pair configuration for fail-over scenarios. All CMP Proxies in the VoiceGenie network of servers connect to a CMP Server.

2.3 CMP Proxy & Command Line Console (CLC)

The CMP Proxy & CLC consists of two items:

- CMP Proxy
- Command Line Console

2.3.1 CMP Proxy

The CMP Proxy must run on every server that is managed or monitored by the OA&M Framework. It acts as a single point of communication for all VoiceGenie software running on that server. The CMP Proxy is responsible for server level logging; this includes the metrics logs, alarms and system level logging. Also, the CMP Proxy is responsible for starting and stopping all

VoiceGenie software components. In addition, the CMP Proxy monitors the disk, CPU and memory utilization of the system, as well as the CPU and memory utilization of all VoiceGenie processes and can restart them if required.

2.3.2 Command Line Console (CLC)

The Command Line Console (CLC) is a command line interface to the OA&M Framework. Through this interface, users can query information about the components that are part of the VoiceGenie network of servers. Also, the CLC allows users to inject commands into the OA&M Framework to carry out various tasks.

For detailed information about CLC, please refer to the following document:

VoiceGenie 7.2 OA&M Framework – CLC User’s Guide

2.4 System Management Console (SMC)

The SMC consists of a web interface that can be used to access various monitoring, operations, installation, configuration, and administration capabilities. Through the web interface users can access both real time and historical information about the VoiceGenie software, as well as perform various operations and carry out configuration and provision changes.

For detailed information about SMC, please refer to the following document:

VoiceGenie 7.2 OA&M Framework – SMC User’s Guide

2.5 VoiceGenie SNMP Agent

The VoiceGenie SNMP component is the SNMP agent for all VoiceGenie software. Via the VoiceGenie SNMP Agent users can receive SNMP traps whenever an alarm condition occurs, also, SNMP gets and sets are supported.

For detailed information about SNMP, please refer to the following document:

VoiceGenie 7.2 OA&M Framework – SNMP User’s Guide



Chapter

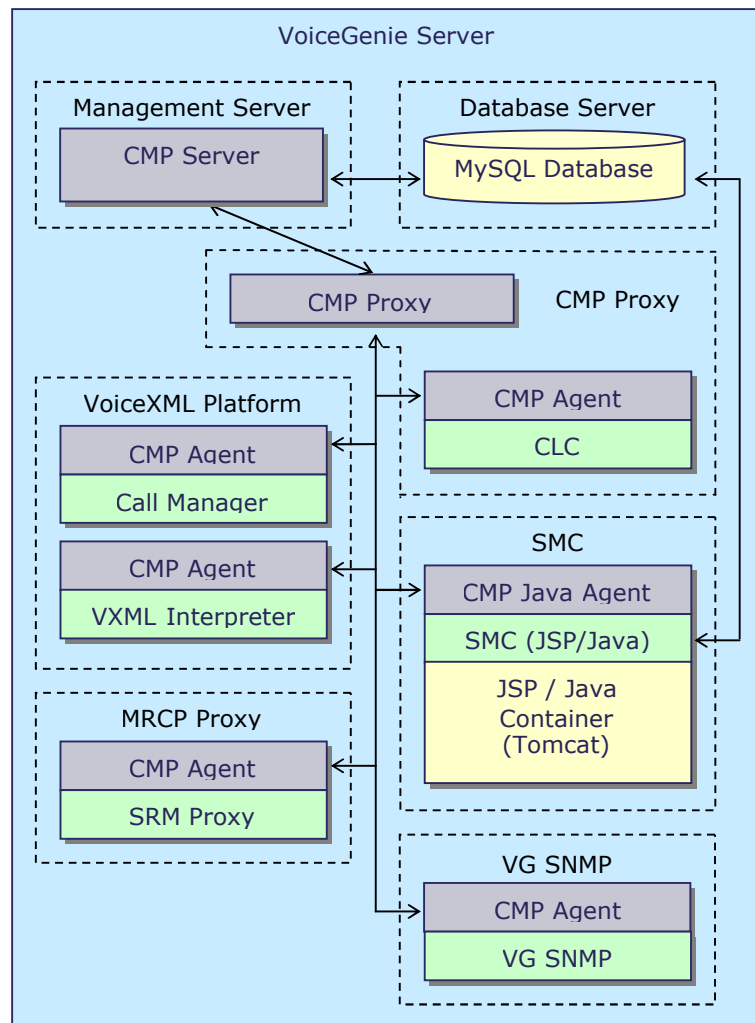
3

Architecture

The OA&M Framework is designed to function in both a *distributed* or *all-in-one* architecture. The distributed architecture should be used in all production deployments. The all-in-one architecture can be used for a lab environment or application development purposes. Please refer to the *VoiceGenie 7.2 Installation Guide* for a detailed discussion of deployment architectures.

3.1 All-in-One Architecture

With the all-in-one architecture all components, i.e. the CMP Proxy & CLC, CMP Server, Database Server, SMC, VoiceGenie SNMP and the rest of the VoiceGenie components are installed on a single server. This architecture is depicted below:

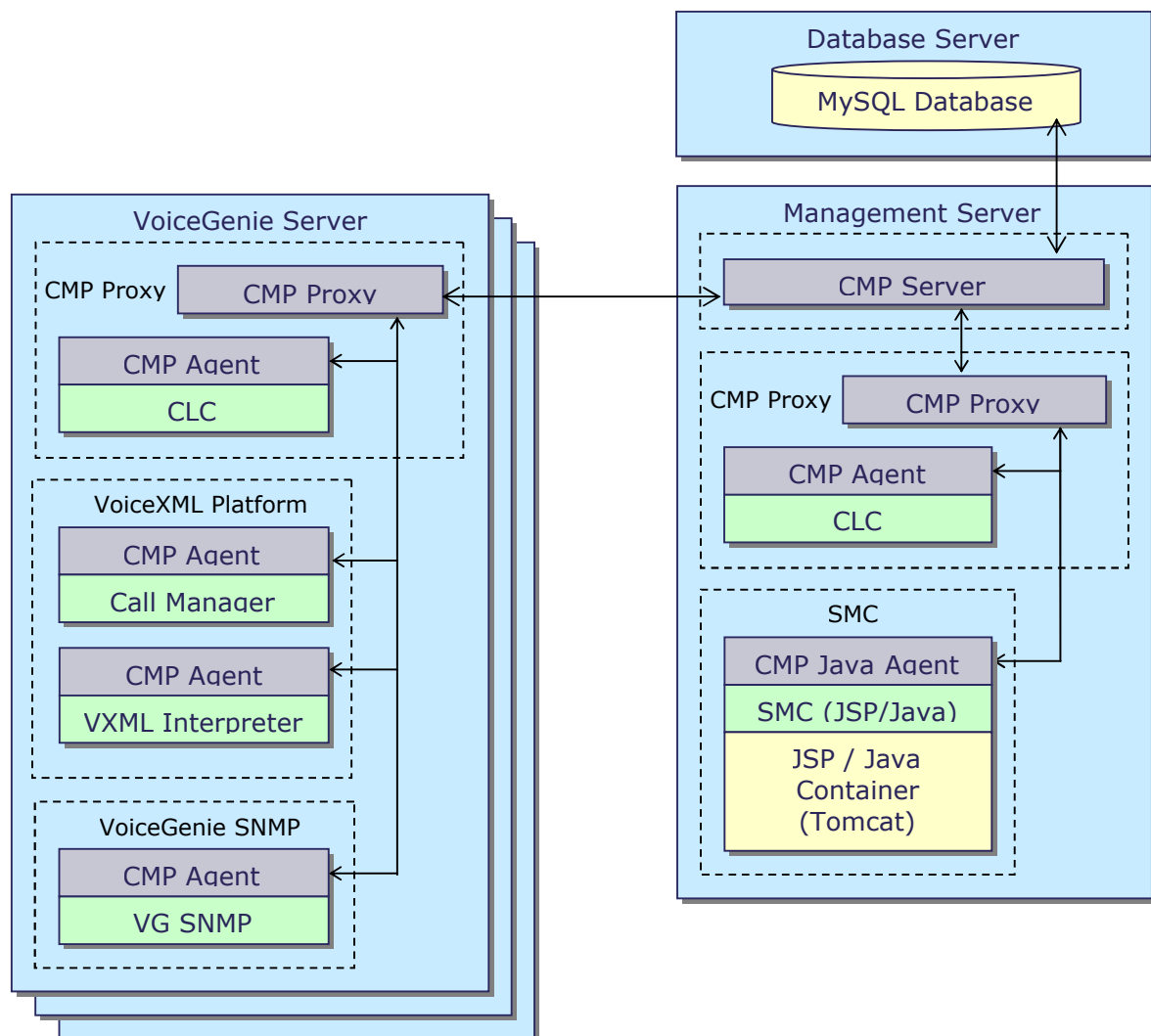


In this architecture all components of the OA&M Framework are located on one machine. This architecture is useful for setting up the VoiceGenie software in a lab environment or when carrying out application development.

Note: The All-in-One setup is not recommended for production systems since database operations can negatively affect call processing.

3.2 Distributed Architecture

In a distributed architecture, the various components of the OA&M Framework are distributed across different servers in the network. The Database Server can be on its own machine; also, the CMP Server can be deployed in a redundant pair (primary/backup) configuration with each CMP Server running on its own machine. An example of this architecture is depicted below:



Note: For increased fault tolerance a MySQL database with replication can be setup on both the Primary and Backup CMP Servers so that full redundancy can be implemented. Note that configuration changes or installations should not be carried out when the system has failed over to the Backup CMP Server.

Using this infrastructure a number of OA&M functionalities will be available, including the following:

- Health Status Monitoring
- Real Time Status Monitoring
- Operations Administration
- Centralized Logging and Alarming
- Centralized Configuration

- Cluster-wide Administration
- Distributed Installations
- Historical Reports
- SNMP

Please refer to the *VoiceGenie 7.2 Installation Guide* for a detailed discussion of deployment architectures.



Chapter

4

Installation and Licensing

Information regarding installation and licensing of the OA&M Framework can be found in the following document: *VoiceGenie 7.2 Installation Guide*.



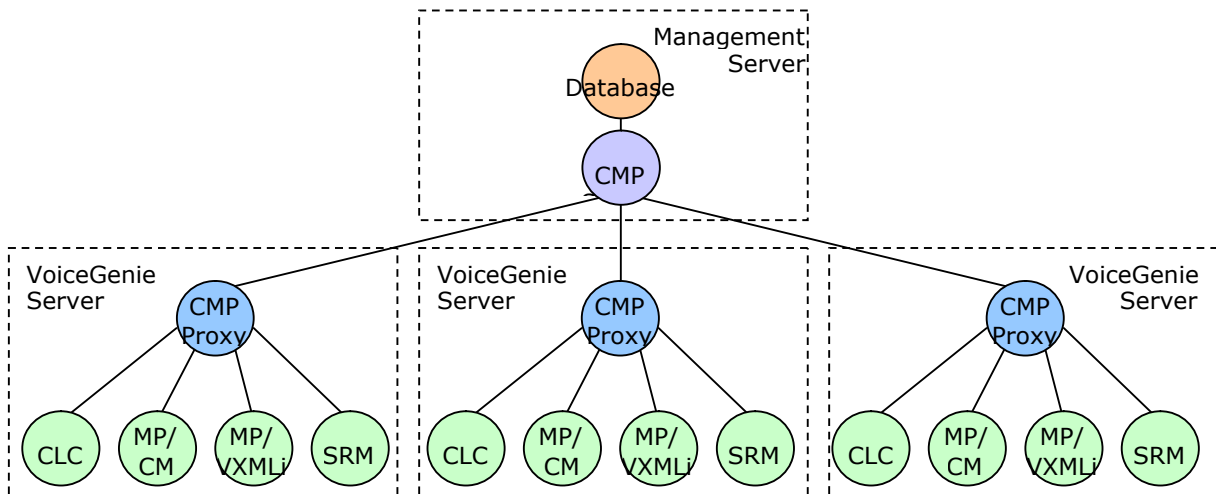
Chapter

5

OA&M Framework Configuration

5.1 Overview

The OA&M Framework is fundamentally a communication infrastructure. A number of services for logging, management, configuration, etc. are built on top of this communication infrastructure. In order to use this functionality the communication infrastructure must be setup correctly. The following diagram depicts the layout of the OA&M Framework infrastructure; it resembles an inverted tree structure:



All centralized data is stored within the database (i.e. logs, configuration, etc.). A group of VoiceGenie Servers is clustered by connecting them to a common CMP Server (Management Server). Also, all the components on a server (i.e. CLC, VoiceXML Platform (MP/CM & MP/VXMLi), SRM – Speech Resource Manager, etc.) connect to the CMP Proxy on the server. This section describes what configuration is required to ensure a proper setup.

5.2 CMP Server

The CMP Server configuration is edited using the SMC. Note that changes made by the SMC may not take effect immediately and may require a software restart. Also, the contents of the configuration are written to the file system. This configuration file is located at `/usr/local/cmp-server/config/cmpengine.cfg` under Linux and `C:\VoiceGenie\cmp\cmp-server\config\cmpengine.cfg` on Windows.

The key parameters in configuring the CMP Server are the CMP Server redundancy connection settings and the database connection settings.

5.2.1 Database Connection Settings

The following three database connection parameters must be specified correctly:

- `cmp.driver`
- `cmp.user`
- `cmp.password`

The `cmp.driver` parameter specifies the ODBC Data Source Name (DSN) to connect to. Under Linux the DSN is specified in the `odbc.ini` file which is located at `/usr/local/cmp-server/config/odbc.ini`. Under Windows the DSN is specified in the Control Panel. The default value is `myodbc`. The `cmp.user` and `cmp.password` parameters are the user name and password that should be used by the CMP Server to connect to the database. The default value is `pw` and `pw`.

5.2.2 CMP Server Redundancy Connection Settings

The following CMP Server redundancy connection parameters should be specified correctly:

- `cmp.primary_cmpe`
- `cmp.primary_port`
- `cmp.backup_cmpe`
- `cmp.backup_port`

The hostname of the Primary CMP Server must be defined using the parameter `cmp.primary_cmpe`, i.e. `primary.voicegenie.com`. In addition, the port number of the Primary CMP Server must be set, the default value is 8600.

Similarly, the hostname of the Backup CMP Server must be set using the parameter `cmp.backup_cmpe`, i.e. `backup.voicegenie.com`. The port number of the Backup CMP Server must also be set, the default is 8650.

Note that the ports defined by `cmp.primary_port` and `cmp.backup_port` (8600 and 8650) should be opened on a firewall so that CMP Proxies from other VoiceGenie servers can connect to the CMP Server.

Note: If no Backup CMP Server exists, these two parameters: `cmp.backup_cmpe` and `cmp.backup_port` should be disabled so that the software does not try to connect to it.

The role of the CMP Server is determined by a configuration parameter in the role configuration file, which is located at `/usr/local/cmp-server/config/cmpengine_role.cfg` on Linux and `C:\VoiceGenie\cmp\cmp-server\config\cmpengine_role.cfg` on Windows. Within this file, the `cmp.primary` parameter determines if the CMP Server is to function as a Primary CMP Server or a Backup CMP Server. If the parameter is set to 1 (i.e. `cmp.primary = 1`) the CMP Server will function as a Primary CMP Server. If the parameter is set to 0 (i.e. `cmp.primary = 0`) the CMP Server will function as a Backup CMP Server.

Note: In the All-in-One setup only one CMP Server exists, as a result, it should always be set as the primary.

5.3 CMP Proxy

The CMP Proxy configuration is edited using the SMC. However, to change fundamental connection settings the parameter values should be changed both in the SMC as well as on disk. The on disk configuration file is located at `/usr/local/cmp-proxy/config/cmpproxy.cfg` under Linux and at `C:\VoiceGenie\cmp\cmp-proxy\config\cmpproxy.cfg` under Windows.

The most important parameters for configuring the CMP Proxy are the CMP Server connection settings. These are required for the CMP Proxy to connect with the CMP Server.

5.3.1 CMP Server Connection Settings

The following CMP Server connection parameters should be specified correctly:

- `cmp.primary_cmpe`
- `cmp.primary_port`
- `cmp.backup_cmpe`
- `cmp.backup_port`

The hostname of the Primary CMP Server must be defined using the parameter `cmp.primary_cmpe`, i.e. `primary.voicegenie.com`. In addition, the port number of the Primary CMP Server must be set. The default value is 8600.

Similarly, the hostname of the Backup CMP Server must be set using the parameter `cmp.backup_cmpe`, i.e. `backup.voicegenie.com`. The port number of the Backup CMP Server must be set, the default is 8650. The Backup CMP Server will only be contacted if the Primary CMP Server is unreachable or unavailable.

5.3.2 Multiple Network Interface Card (NIC) Settings

The CMP Proxy connects to the CMP Server using the default Ethernet interface, this is usually the first IP address on the first network interface card on the system. If you would like the CMP Proxy to use a difference IP address to connect to the CMP Server (i.e. for multiple or dual NIC systems) you can configure it in the Local IP Configuration file which is located at `/usr/local/cmp-proxy/config/local_ip.cfg` on Linux and `C:\VoiceGenie\cmp\cmp-proxy\config\local_ip.cfg` on Windows. Within this file, the `cmp.local_ip` parameter should be set to the desired local IP to use for the CMP Server connection.

5.4 Configuration Synchronization Override

Currently, when a component connects to the OA&M Framework, it synchronizes its local configuration on disk with what is stored in the database. However, there are instances where it may be useful to override database synchronization and only use the values stored in the local configuration file. The `cmp.sync` parameter allows a component to control this option. After a user sets a component's `cmp.sync` parameter to `FALSE` and restarts that component, all synchronization and parameter update attempts made by the CMP Server to the component will be rejected. Consequently, neither the local configuration on disk nor the parameter values in memory will ever be overwritten or updated, and the component will only use the parameter values in the configuration file. A user will have to update the configuration file to change the `cmp.sync` parameter value back to `TRUE` in order to revert back to normal synchronization mode. This option could be useful for testing a configuration change without modifying the configuration stored in the database or for people who do not want to use SMC to make configuration changes and prefer to use a text editor. However, it is recommended that the `cmp.sync` option be used for temporary changes only.

Note: The `cmp.sync` value should be changed directly in the configuration file and not through the SMC.



Chapter

6

CMP Server

6.1 Overview

This section outlines some of the configuration details about the CMP Server along with information on how to start and stop the CMP Server. See for information about advanced replication support.

Each VoiceGenie deployment must have at least one CMP Server installed. Ideally, the CMP Server should be deployed in a redundant pair configuration for fail-over scenarios in a production environment. The CMP Server requires a database server to centrally store information. Details on the Database server can be found in the next section.

6.2 CMP Server Startup and Options

The CMP Server starts automatically at system boot up. Under Linux the startup information is stored in the `inittab` file, which is located at `/etc/inittab`. Under Windows the CMP Server is started up as a service, the details of which can be found in the `Services` section under the `Administrative Tools` section of the Control Panel.

The CMP Server takes two command line parameters.

- The `-r` option determines the root location of the CMP Server. This parameter should be set to `/usr/local/cmp-server/` under Linux and `C:\VoiceGenie\cmp\cmp-server` under Windows.
- The `-c` command line parameter specifies the name and location of the configuration file. Under Linux, the location can be specified with an absolute path by starting with a slash (i.e. `/usr/local/cmp-server/config/cmpengine.cfg`) or as a relative path to the root (`-r`) by not starting with a slash (i.e. `config/cmpengine.cfg`). Under

Windows, the location should be specified as an absolute path and set to
 C:\VoiceGenie\cmp\cmp-server\config\cmpengine.cfg.

6.3 Starting, Stopping or Restarting the CMP Server

On Linux, you must be the root user to start, stop or restart the CMP Server. To become the root user log in to the CMP Server system and type in `su`, then enter the root password when prompted.

Then, to start the CMP Server, issue the following command:

```
/etc/init.d/cmp-server start
```

To stop the CMP Server, issue the following command:

```
/etc/init.d/cmp-server stop
```

To restart the CMP Server, issue the following command:

```
/etc/init.d/cmp-server restart
```

On Windows, the CMP Server can be started, stopped or restarted from the Services window, which can be accessed from the Administrative Tools section under the Control Panel. To start the CMP Server, click on the CMP Server Service entry and click the Start Service button. To stop the CMP Server, click on the CMP Server Service entry and click the Stop Service button. To restart the CMP Server, click on the CMP Server Service entry and click the Restart Service button.

6.4 CMP Server Health Information

Health information about the CMP Server can be retrieved using the CLC `health cmpserver` command. The following information is provided:

Health for Primary CMP Server (cmpserver)

Started: <date/time CMP Server(Engine) started>

DB Connection: <Database connection status (Online or Offline)>, Attempts <number of connection attempts to the Database>

DB Requests: Success <successful database requests>, Queued <currently queued database requests>, Error <total database request errors>, Failed <total database requests that failed to execute due to connection failure>, Total <total database requests>

DB Request Time (ms): Min <minimum DB request processing time (ms)>, Max <maximum DB request processing time (ms)>, Moving Avg <moving average for the DB request processing time>

6.5 Reducing Load on the CMP Server / Database Server

The CMP Server is responsible for centralizing all data for an entire cluster of VoiceGenie Servers. It stores metrics data for sessions, performance data and component health information. Also, other external processes connect to the database and calculate summarized statistics for call processing and call quality. As the size of a cluster grows the load that is handled by the centralized CMP Server increases. As a result, a number of configuration parameters can be tweaked in order to decrease the total load on the CMP Server and the Database Server.

6.5.1 Reducing Stored Logging Data

One option in reducing load is to decrease the amount of data logged to the CMP Server and eventually the Database Server. This can be accomplished by adjusting the log filters in the CMP Proxy configuration. Details on how the log filters work and how they can be changed can be found in “9.4 Log Filtering Masks” section of this document.

6.5.2 Reducing Stored Historical Data

The CMP Server logs historical data about the initialization, health and process status of all VoiceGenie processes. The logging of this data to the database can be adjusted through configuration. The following table describes the associated configuration parameters:

CMP Server Parameter	Description
cmp.log.process_data.histhealth	Determines if historical health information is stored in the HistHealthStatus table. The effect of storing this information is 1 database request per component every 20 seconds.
cmp.log.process_data.histinit	Determines if historical process initialization information is stored in the HistInitStatus table. The effect of storing this information is 1 database request per component startup and shutdown.
cmp.log.process_data.histproc	Determines if historical process statistics (CPU and memory) information is stored in the HistProcessStatus table. The effect of storing this information is 1 database request per component every 20 seconds.

6.5.3 Reducing Stored Health Data

The CMP Server receives health information from each process on a periodic basis. By default this occurs every 20 seconds for each process. The logging of this data to the database can be adjusted through configuration. The following table describes the associated configuration parameters:

CMP Server Parameter	Description
cmp.log.process_data.health	Determines if current process health information is stored in the Network table. The effect of storing this information is 1 database request per component every 20 seconds.



Chapter

7

Database Server

7.1 Overview

This section outlines some of the details about the Database Server.

The CMP Server stores all centralized information in a database server. Currently, the CMP Server only supports connection to a MySQL database server or an Oracle 10g database server. The OA&M Framework requires that the following 2 databases reside within the database server:

- NDM – contains the configuration and OA&M Framework details
- CallHistory – contains historical log information

The Database Server can be located on the same machine as the CMP Server or on a separate off board server.

Note: If you choose to deploy the Database Server off board from the CMP Server please ensure that the time on the various machines is synchronized otherwise errors may be logged by the CMP Server. The *VoiceGenie 7.2 Installation Guide* documents how a machine can be set up to use an NTP time server.

Connectivity to the database is most efficient when the database server is on the same physical server as the CMP Server. This must be taken into account when choosing an architecture, since logging large amounts of data to the database could result in the CMP Server falling behind and rendering the system unusable.

In addition, the MySQL database server can be setup in a redundant pair configuration with replication, this is useful for added fault tolerance.

7.2 Accessing MySQL

After installation, MySQL is setup with default usernames and passwords. The username used by CMP Server to connect to the database server is `pw` and it has the default password `pw`. Also, the default root password for the MySQL server is `rootsql`. For further details about MySQL please refer to the documentation provided at www.mysql.com.

7.3 Database Summarization and Cleanup

The Database Server comes bundled with a java application called `dbadmin.jar` that is run periodically to summarize and cleanup the database. This application is located in the directory `/usr/local/cmp-db/bin/` under Linux and `C:\VoiceGenie\cmp\cmp-db\bin` under Windows.

Note: The summarization and cleanup scripts can be CPU and memory intensive, as a result, the database server should never be on the same machine as VoiceGenie call processing software such as the VoiceXML Platform. Therefore, the All-in-One configuration is not recommended for production systems.

7.3.1 Database Summarization

The Java application is run periodically to summarize call data records as well as summarize statistics for the various reports generated by the SMC. The call data summarization occurs every 10 minutes. The statistics for reports such as the Call Volume, Call Length and Application Distribution are calculated hourly. The Call Length report groups calls based on their length. These groups are defined by a set of values stored in the `MetaData` table of the database. These values are defined as described in the table below:

MetaData Table Name	Default Value	Description
group0	10	Defines first group as all call lengths > 0 and <= 10 seconds
group1	30	Defines second group as all call lengths > 10 and <= 30 seconds
group2	60	Defines third group as all call lengths > 30 and <= 60 seconds
group3	180	Defines fourth group as all call lengths > 60 and <= 180 seconds

MetaData Table Name	Default Value	Description
group4	300	Defines fifth group as all call lengths > 180 and <= 300 seconds and defines sixth group as all call lengths > 300 seconds

7.3.2 Database Cleanup

The database cleanup occurs once a day. The application queries the MetaData table for the values that determine how many hours of data should be kept. Users can change these values in the MetaData table to adjust the amount of data that is kept. The default numbers of hours of data that are kept is:

Table	Hours	MetaData Parameter Name	Table Description
CallLog	24	CallLogThreshold	The raw metrics and billing information
CallData	168	CallDataThreshold	The summary Call Data Records
HistCallVolume	672	HistCallVolumeThreshold	Call Volume summarization data
HistCallLength	672	HistCallLengthThreshold	Call Length summarization data
HistAppDistribution	672	HistAppDistributionThreshold	Application Distribution summarization data
HistHealthStatus	24	HistHealthStatusThreshold	Historical Health Status information
HistInitStatus	672	HistInitStatusThreshold	Historical Initialization Status information
HistProcessStatus	24	HistProcessStatusThreshold	Historical Process (CPU & memory) information

Note: The database cleanup script may take some time to clean up the CallData/CallLog tables if they are very large. During the clean up process the CMP Server may need to queue logs intended for the CallLog table and call records intended for the CallData table. If necessary, reduce the appropriate thresholds to help minimize the impact on the CMP Server.

7.3.3 Binary Log Cleanup

When the database server is setup for replication it will create Binary Log files which contain information on the changes that have been made to the database. These file are read by the MySQL replication slave to keep the databases synchronized. However, these Binary Log files are not automatically deleted by MySQL and need to be deleted once they have been used by MySQL to synchronize the database. As a result, the `dbadmin.jar` application can be setup to perform this cleanup.

Under Linux the following line can be added to `/etc/crontab` to perform the Binary Log cleanup every hour at 50 minutes past the hour, note that this should be added to cron on both the primary and backup database servers:

```
50 * * * * pw /usr/java/jdk/bin/java -jar /usr/local/cmp-db/bin/dbadmin.jar checkbinlog
```

Under Windows, a scheduled task should be added to run the following command every hour at 50 minutes past the hour:

```
java -jar C:\VoiceGenie\cmp\cmp-db\bin\dbadmin.jar checkbinlog
```

Also, the `dbadmin.jar` application must connect to the database using the root user. The root account access to the database from the system must be granted. This can be done by logging into the database using the root user and executing the following MySQL command:

```
mysql> grant ALL on *.* to root@'IP_address' identified by 'rootsql';
```

Where the `IP_address` should be replaced by the IP address of the system.

The `dbadmin.jar` also must store the root password in an encrypted manner in a flat file. To add the root password to this flat file you will need to run the following command, note that `<root password>` is your specific root user account password:

Under Linux:

```
java -jar /usr/local/cmp-db/bin/dbadmin.jar addpass <root password>
```

Under Windows:

```
java -jar C:\VoiceGenie\cmp\cmp-db\bin\dbadmin.jar addpass <root password>
```

The Binary Log file cleanup application generates the following alarms when an error condition is detected:

Name	Log ID	Level	Description	Causes	Response Code/Detailed Recommended Action
VGLOG-DB-BINLOG-EXEC-ERROR-ALARM	1009	ERROR	There was an error in obtaining the bin log status.	The script could not access MySQL to get the status of the binary log files or could not purge the files, MySQL may not be running, or replication may not be setup correctly.	Restart MySQL or resetup replication.
VGLOG-DB-BINLOG-DELETE-ALARM	1010	NOTE	Binary logs have been deleted up to log %s.	Used to inform the customer that a purge has occurred.	– (Normal event)

7.4 Database Backup and Restoration

The Database Server comes bundled with a java application called `dbadmin.jar` that can be used to backup and restore the database. This application is located in the directory `/usr/local/cmp-db/bin/` under Linux and `C:\VoiceGenie\cmp\cmp-db\bin` under Windows.

7.4.1 Database Backup

The application uses the MySQL utility `mysqldump` to backup the CMP database structure and some of the configuration data. The backup data is stored in the CMP DB scripts directory under `/usr/local/cmp-db/scripts/` under Linux and `C:\VoiceGenie\cmp\cmp-db\scripts\` under Windows. The script is run daily by cron. Each time the script is run two files are created, the filenames contain the timestamp, e.g. `cmp_db_backup_data_20040620041201.sql` and `cmp_db_backup_structure_20040620041201.sql`.

The structure of all databases and tables will be included in the backup files. The data contained within these tables will also be stored except for any tables that contain log information.

7.4.2 Database Restoration

The application can be used to restore the database structure and configuration information. It takes the path of a backup file as an argument. It also takes the

root password of the MySQL database server as another argument. This script deletes all the existing CMP databases and tables; then creates the database structure and inserts configuration data into the tables based on the information stored in the backup file. After restoration, the CMP database will have the same structure and data as when the backup was made, however, the log tables will be empty.

Before running the restore script, please ensure that the CMP Server is not running.

7.5 Database Monitoring and Alarming

The Database Server comes bundled with a java application called `dbadmin.jar` that can be used to monitor the health of the database and can send out alarms when critical conditions are reached. This application is located in the directory `/usr/local/cmp-db/bin/` under Linux and `C:\VoiceGenie\cmp\cmp-db\bin` under Windows.

7.5.1 Database Queue Monitoring

Between the CMP Server and the Database Server all requests are queued. The `dbadmin.jar` application monitors this queue to ensure that database requests are being handled in a timely manner and that the queue is not growing uncontrollably. Under Linux, the database queue monitoring is run every minute; under Windows the scheduled task needs to be added manually by adding a task that runs the following application:

```
java -jar C:\VoiceGenie\cmp\cmp-db\bin\dbadmin.jar
checkqueue -l 1000 -i 10
```

The application takes 2 command line options; they are outlined in the table below:

Command Line Option	Purpose	Default Value
-l	This parameter defines the maximum queue length allowed before an alarm is sent.	1000
-i	This parameter defines the maximum number of times in a row that the queue is found to be increasing in length before an alarm is sent.	10

The queue monitoring application generates the following alarms when an error condition is detected:

Name	Log ID	Level	Description	Causes	Response Code/Detailed Recommended Action
VGLOG-DB-QUEUE-EXEC-ERROR-ALARM	1004	WARN	There was an error obtaining the queue length from the CLC.	The database monitoring scripts could not access the CLC or could not parse the queue length information from the response given by the CLC.	REVG – Check that the CLC is operational and that the health information from the CMP Server (CLC> health cmpserver) is gettable. Send output to VoiceGenie.
VGLOG-DB-QUEUE-LENGTH-ALARM	1005	EROR	The queue has exceeded the maximum length allowed, %d.	This may be caused by the database (MySQL) being used by another application, causing the MySQL server to slow down and increasing the queue. Also, it could be the case that the CMP Server is being inundated with logs and can not handle the data throughput.	REVG – Log in to MySQL (mysql -u pw -ppw NDM) and get the out put of show processlist; to see what processes are using the database. Also, turn down any excessive logging going to the database (i.e. metrics, tracing, etc.)
VGLOG-DB-QUEUE-INCREASING-ALARM	1006	CRIT	The queue has increased in size for too many consecutive sample periods, %d.	This may be caused by the database (MySQL) being used by another application, causing the MySQL server to slow down and increasing the queue. Also, it could be the case that the CMP Server is being inundated with logs and can not handle the data throughput.	REVG – Log in to MySQL (mysql -u pw -ppw NDM) and get the out put of show processlist; to see what processes are using the database. Also, turn down any excessive logging going to the database (i.e. metrics, tracing, etc.)

7.5.2 Database Replication Monitoring

For added redundancy and failover the database server can be run in a replicated pair configuration. In such a scenario it is useful to be able to

monitor the replication status of the database servers. As a result, the `dbadmin.jar` application can be setup to monitor database replication status.

Under Linux the following line can be added to `/etc/crontab` to check replication status every minute, note that this should be added to cron on both the primary and backup database servers:

```
* * * * * pw /usr/java/jdk/bin/java -jar /usr/local/cmp-db/bin/dbadmin.jar checkreplication
```

Under Windows, a scheduled task should be added to run the following command every minute:

```
java -jar C:\VoiceGenie\cmp\cmp-db\bin\dbadmin.jar checkreplication
```

The `dbadmin.jar` application must connect to the database using the root user. As a result, the `dbadmin.jar` stores the root password in an encrypted manner in a flat file. To add the root password to this flat file you will need to run the following command, note that `<root password>` is your specific root user account password:

Under Linux:

```
java -jar /usr/local/cmp-db/bin/dbadmin.jar addpass <root password>
```

Under Windows:

```
java -jar C:\VoiceGenie\cmp\cmp-db\bin\dbadmin.jar addpass <root password>
```

The database replication monitoring application generates the following alarms when an error condition is detected:

Name	Log ID	Level	Description	Causes	Response Code/Detailed Recommended Action
VGLOG-DB-REPLICATION-EXEC-ERROR-ALARM	1007	WARN	There was an error in obtaining the replication status.	The database monitoring scripts could not access mysql to get the status of the replication.	Check that the root password is added correctly to the configuration using the <code>dbadmin.jar addpass</code> command. Check that the mysql client works and connects to the database.
VGLOG-DB-REPLICATION-BROKEN-ALARM	1008	CRIT	The replication has been broken and needs to be repaired.	The database slave is no longer running correctly on the database server and as a result database replication is broken.	Database replication must be set up again.



Chapter

8

CMP Proxy

8.1 Overview

This section outlines some of the configuration details about the CMP Proxy as well as information about its process and system monitoring capabilities.

8.2 CMP Proxy Startup and Options

The CMP Proxy is configured to be started at system boot up, unless explicitly disabled. Under Linux the startup information is stored in the inittab file, which is located at `/etc/inittab`. Under Windows the CMP Proxy is started up as a service, the details of which can be found in the `Services` section under the `Administrative Tools` section of the Control Panel.

The CMP Proxy takes two command line parameters. The `-r` option determines the root location of the CMP Proxy. This parameter should be set to `/usr/local/cmp-proxy/` under Linux and `C:\VoiceGenie\cmp\proxy` under Windows. Also, the `-c` command line parameter specifies the name and location of the configuration file. Under Linux the location can be specified with an absolute path by starting with a slash (i.e. `/usr/local/cmp-proxy/config/cmpproxy.cfg`) or as a relative path to the root (`-r`) by not starting with a slash (i.e. `config/cmpproxy.cfg`). Under Windows the location is an absolute path and should be set to `C:\VoiceGenie\cmp\cmp-proxy\config\cmpproxy.cfg`.

8.3 Starting, Stopping or Restarting the CMP Proxy

On Linux, you must be the root user to start, stop or restart the CMP Proxy. To become the root user log in to the system and type in `su`, then enter the root password when prompted.

Then, to start the CMP Proxy, issue the following command:

```
/etc/init.d/cmp-proxy start
```

To stop the CMP Proxy, issue the following command:

```
/etc/init.d/cmp-proxy stop
```

To restart the CMP Proxy, issue the following command:

```
/etc/init.d/cmp-proxy restart
```

On Windows, the CMP Proxy can be started, stopped or restarted from the Services window, which can be accessed from the Administrative Tools section under the Control Panel. To start the CMP Proxy, click on the CMP Proxy Service entry and click the Start Service button. To stop the CMP Proxy, click on the CMP Proxy Service entry and click the Stop Service button. To restart the CMP Proxy, click on the CMP Proxy Service entry and click the Restart Service button.

Note: In the event that the CMP Proxy gets into a state where the service's status is continually in the Stopping state. The machine must be rebooted in order for the service to resume normal operation. This only happens in rare circumstances.

In the unlikely event that this occurs often, you can configure the service to log on as a specific user type (normally the Administrator user account). To do this, right click on the service and select Properties, then click on the Log On tab. Select the This Account radial and enter the account name and password under which the service should be started. Note that if the password for that account is changed, the password value for this service will also have to be manually updated. Subsequently the machine will need to be rebooted for the new settings to take effect. Now if the service enters into a bad state the service can be killed using the Task Manager and restarted using the Services interface.

8.4 System CPU, Memory and Disk Usage Monitoring

The CMP Proxy is responsible for monitoring and reporting system CPU, memory and disk usage information. This information is reported as part of a

periodic heartbeat message to the CMP Server. The heartbeat reporting interval is dictated by the CMP Proxy's `cmp.heartbeat` configuration parameter. By default this value is set to 20 seconds. In general this value should not be modified.

The system usage information is viewable via the SMC from the Status Monitor page under the Monitoring tab. Also, it can be queried via an SNMP get and is part of the VoiceGenie MIB, the OID is

.1.3.6.1.4.1.7469.3.9.3). In addition, the system usage information can be queried via the CLC using the `health` command for the CMP Proxy (i.e. `health cmpproxy`). The information is part of the first line, the following is an example:

```
[2|0.94%|164MB|/|23%|/dev/shm|0%|/usr/local|11%|/var|2%]
```

The first item is the networkID of the CMP Proxy on the system, followed by the system CPU utilization (0.94%), total system memory utilization (164MB) and the disk usage on all local partitions (23% on /, 0% on /dev/shm, 11% on /usr/local, 2% on /var).

8.4.1 Exceeding CPU Usage Limits and Generated Alarms

The CMP Proxy raises an alarm if the total CPU Usage on a machine exceeds a configured limit. The limits are configured in the CMP Proxy configuration; this configuration is changeable via the SMC. These limits are outlined in the table below:

Configuration Parameter Name	Severity	Default Value (%)	SNMP Trap ID	Clearing SNMP Trap ID
<code>cmp.cpu.warn.limit</code>	Warning	70	.1.3.6.1.4.7469.251.1 .200.8389389	.1.3.6.1.4.7469.251.1 .200.8389642
<code>cmp.cpu.eror.limit</code>	Error	75	.1.3.6.1.4.7469.251.1 .200.8389127	.1.3.6.1.4.7469.251.1 .200.8389642
<code>cmp.cpu.crit.limit</code>	Critical	80	.1.3.6.1.4.7469.251.1 .200.8388868	.1.3.6.1.4.7469.251.1 .200.8389642

When a threshold is exceeded the appropriate alarm is raised. When the CPU usage decreases below a threshold a corresponding clearing alarm is raised. Details about the alarm are sent within the log message of the alarm.

During database cleanup, cron jobs may consume significant CPU resources, and some of the above alarms may be triggered. In such situations, CPU usage alarms can be safely ignored.

8.4.2 Exceeding Memory Usage Limits and Generated Alarms

The CMP Proxy raises an alarm if the total Memory Usage on a machine exceeds a configured limit. The limits are configured in the CMP Proxy configuration; this configuration is changeable via the SMC. These limits are outlined in the table below:

Configuration Parameter Name	Severity	Default Value (MB)	SNMP Trap ID	Clearing SNMP Trap ID
cmp.memory.warn.limit	Warning	1900	.1.3.6.1.4.7469.251.1 .200.8389390	.1.3.6.1.4.7469.251.1 .200.8389643
cmp.memory.eror.limit	Error	1950	.1.3.6.1.4.7469.251.1 .200.8389128	.1.3.6.1.4.7469.251.1 .200.8389643
cmp.memory.crit.limit	Critical	2000	.1.3.6.1.4.7469.251.1 .200.8388869	.1.3.6.1.4.7469.251.1 .200.8389643

When a threshold is exceeded the appropriate alarm is raised. When the memory usage decreases below a threshold a corresponding clearing alarm is raised. Details about the alarm are sent within the log message of the alarm.

Note: The above defaults are based on a system with 2GB of memory. Anyone with a different system setup should modify these parameters accordingly.

8.4.3 Exceeding Disk Usage Limits and Generated Alarms

The CMP Proxy raises an alarm if the Disk Usage on a partition exceeds a configured limit. The limits are configured in the CMP Proxy configuration; this configuration is changeable via the SMC. These limits are outlined in the table below:

Configuration Parameter Name	Severity	Default Value (%)	SNMP Trap ID	Clearing SNMP Trap ID
cmp.disk.warn.limit	Warning	85	.1.3.6.1.4.7469.251.1 .200.8389388	.1.3.6.1.4.7469.251.1 .200.8389641
cmp.disk.eror.limit	Error	90	.1.3.6.1.4.7469.251.1 .200.8389126	.1.3.6.1.4.7469.251.1 .200.8389641

Configuration Parameter Name	Severity	Default Value (%)	SNMP Trap ID	Clearing SNMP Trap ID
cmp.disk.crit.limit	Critical	95	.1.3.6.1.4.7469.251.1 .200.8388867	.1.3.6.1.4.7469.251.1 .200.8389641

When a threshold is exceeded the appropriate alarm is raised. When the disk usage decreases below a threshold a corresponding clearing alarm is raised. Details about the alarm are sent within the log message of the alarm.

8.5 Process Startup and Recovery

The CMP Proxy is also responsible for starting up most processes that run on the VoiceGenie Server. This includes the Media Platform, Speech Resource Manager and SIP Proxy processes to name a few. The CMP Proxy also monitors the CPU and memory usage of all processes it starts and restarts them if they exceed certain operational limits or are no longer operational.

The list of processes that are started by the CMP Proxy is specified in the CMP Proxy's `cmp.components` configuration parameter. Only processes that are started by the CMP Proxy are monitored for CPU and memory usage. The `cmp.components` parameter consists of a bar delimited list of labels that refer to other configuration parameters that specify the command line of the process, operational limits of the process and any scripts to run before the process is started. The following is an example:

```
cmp.components = PMLI|CMGR

pml_i = /usr/local/phoneweb/bin/pwvxmli
pml_i_limits = 85|1000|10000
pml_i_start_script =
/usr/local/phoneweb/bin/pml_i_start_script
pml_i_restart_script =
/usr/local/phoneweb/bin/pml_i_restart_script
pml_i_stop_script =
/usr/local/phoneweb/bin/pml_i_stop_script

cmgr = /usr/local/phoneweb/bin/pwcallmgr
cmgr_limits = 95|1000|60000
cmgr_start_script =
/usr/local/phoneweb/bin/cmgr_start_script
cmgr_restart_script =
/usr/local/phoneweb/bin/cmgr_restart_script
cmgr_stop_script =
/usr/local/phoneweb/bin/cmgr_stop_script
```

The above parameters specify that the process labels PMLI and CMGR should be started and monitored. The command line to start PMLI is specified in the

parameter `pmli`. The command line to start CMGR is specified in the parameter `cmgr`.

The operational limits of the process are specified in the `<label>_limits` parameter. This parameter consists of three bar delimited values. The first two values specify the maximum allowed CPU utilization and the maximum allowed memory utilization (in MB). If either of these values is exceeded for 2 consecutive heartbeat intervals the process is terminated and restarted. In the example above PMLI can use up to 85% of the CPU and up to 1000 MB of memory, also, CMGR can use up to 95% of the CPU and up to 1000 MB of memory. The third parameter in the `<label>_limits` parameter specifies the shutdown delay time (in milliseconds), which is the amount of time the process has to shutdown gracefully when stopped. If the process does not stop within the allotted time it will be forced to terminate. In the example above PMLI has up to 10000 ms to stop gracefully while CMGR has up to 60000 ms to stop gracefully.

Note: It is not safe to stop the CMP Proxy while components started by the CMP Proxy are still running. A component can still run for some time after a stop command is issued to it, the time depends on the shutdown delay time explained above.

The `<label>_start_script` parameter can be used to define a script that should be run before the process is started.

The `<label>_restart_script` parameter can be used to define a script that should be run if the process stops unexpectedly or is stopped since it exceeds an operational limit, this script is run just before the process is restarted.

The `<label>_stop_script` parameter can be used to define a script that should be run after the process is stopped.

Note: In the event that a process unexpectedly terminates or exceeds an operational limit the CMP Proxy will restart the process. Also, the CMP Proxy will restart any process that is CMP enabled if it fails to send a heartbeat for a period exceeding the CMP Proxy's `cmp.timeout` parameter value, or if it unexpectedly closes its network connection to the CMP Proxy.

The CMP Proxy logs an event whenever a process is stopped, started or recovered. The following table outlines the traps that are generated:

Event Description	Severity	SNMP Trap ID
Component Died Unexpectedly	Warning	.1.3.6.1.4.7469.251.1.200.8389392
Component Running after it Unexpectedly Died	Notice	.1.3.6.1.4.7469.251.1.200.8389646

8.6 Process CPU and Memory Usage Monitoring

Only processes that are started by the CMP Proxy (i.e. defined in `cmp.components`) are monitored for CPU and memory usage. The process usage information is viewable via the SMC by looking at the `Cluster Status` page under the `Monitoring` tab, or clicking on the process name on the `Status Monitor` page under the `Monitoring` tab. Also, it can be queried via an SNMP get and is part of the VoiceGenie MIB, (OID is `.1.3.6.1.4.1.7469.3.9.3`). In addition, the process usage information can be queried via the CLC using the `health` command for the CMP Proxy (i.e. `health cmpproxy`). The following is an example:

```
[9|0%|2MB]
[8|0%|59MB]
[3|0%|1MB]
[6|5.34%|25MB]
[4|1.8%|50MB]
[5|0.09%|6MB]
[7|25.42%|323MB]
```

Each line represents a different process. The first item in each entry is the `networkID` of the process, followed by the CPU utilization of the process and the memory utilization of the process.

When the CMP Proxy detects that an operational limit has been exceeded, or that a process is no longer operational, an event is logged.

Also, the historical values of CPU and memory utilization can be graphed using the `Process Status Report` that is available on the `Monitoring` tab of the SMC. Note that by default only the last 24 hours of data are retained.



Chapter

9

Alarming and Logging

9.1 Overview

One of the chief functionalities of the OA&M Framework is its rich alarming and logging capabilities. The OA&M Framework allows alarms and logs to be filtered at a very granular level to a number of predefined and user defined locations. This section explains how alarming and logging works and how it can be configured for optimal use.

9.2 Severity Levels

All logs that are generated by the VoiceGenie system have a severity level. The following table describes the severity levels that exist:

Severity	Log Level	Type	Description
Metric	METRICS	METRICS	The metrics information that is usually written to the <code>pw_metricsfile</code> , it contains detailed information about calls and call flow.
Critical	log_0	CRIT	An alarm event that denotes a critical or fatal condition and results in the failure of the software.
Error	log_1	EROR	An alarm event that denotes an error condition that should never happen and that results in the loss of functionality.

Severity	Log Level	Type	Description
Warning	log_2	WARN	An alarm event that denotes an exceptional situation that may occur legitimately but it is necessary to be aware of.
Notice	log_3	NOTE	A log event that denotes a significant system level condition or event.
Informational	log_4	INFO	A log event that denotes a normal event that may be of interest.
Trace/Debug	log_5	DEBUG	A log event that denotes information that is only required for component level debugging.

9.3 Predefined and User Defined Sink Locations

Logs can be directed to a number of different locations, known as sinks, through the use of configuration settings. These sinks can be either a predefined sink or a user defined sink. The predefined sinks include the log file (FILE), metrics file (METRICS), upstream (UPSTREAM), Syslog (SYSLOG), VG SNMP (SNMP) or email (EMAIL). User defined sinks consist of user developed DLLs that are loaded by the CMP Proxy. The `cmp.log_sinks` parameter in the configuration file determines what sinks are actively being used by a process. The list of active sinks consists of a bar delimited list. The following is an example:

```
cmp.log_sinks = FILE|UPSTREAM|METRICS|SNMP|SYSLOG
```

The following section describes each sink.

9.3.1 Log File (FILE)

If the FILE sink is enabled, alarms or logs can be logged to a local file. The name of the file is specified in the `cmp.log_file` configuration parameter. This value should be an absolute path. A number of options are available for how the file is rotated; they are outlined below:

Parameter	Values	Prerequisite	Description
cmp.pid_option	TRUE/FALSE	none	If this parameter is specified a new log file will be created every time the process restarts. This is useful for debugging purposes since the log file for a process is not overwritten when it is restarted.
cmp.size_option	TRUE/FALSE	none	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 minutes or time.
cmp.rollover_size	# of megabytes	cmp.size_option = TRUE	If cmp.size_option is set to TRUE this parameter specifies at what size the file should be rolled over. The size is specified in megabytes.
cmp.num_rollover_files	# of files	cmp.size_option = TRUE	If cmp.size_option is set to TRUE this parameter specifies the number of files to roll through before overwriting.
cmp.rollover_mins	# of minutes	cmp.size_option = FALSE	If cmp.size_option is set to FALSE this parameter determines how often the files are rolled over. The number specified is the time interval in minutes.
cmp.rollover_time	time	cmp.size_option = FALSE and cmp.rollover_mins not specified	If cmp.size_option is set to FALSE and cmp.rollover_mins is not specified, then this parameter determines the time of day when the files are rolled over. The value is the time using the 24 hour clock, i.e. values from 0:00 to 23:59.
cmp.log_write_buffer	# of bytes	none	Writes to file are done using block writes to increase performance. This parameter determines the size of the buffer, in bytes, to be used for block writing to the disk. A value of 0 implies no block writing.
cmp.log_write_buffer_stale_timeout	# of milliseconds	cmp.log_write_buffer not 0	The longest time that a log can remain in the block write buffer before being written to disk

Parameter	Values	Prerequisite	Description
cmp.log_write_buffer_idle_timeout	# of milliseconds	cmp.log_write_buffer not 0	The amount of time during which no logs are received after which the block write buffer is written to disk
cmp.log_reorder_window	# of milliseconds	Only useful in the CMP Proxy.	The time window, in milliseconds, during which logs received by the CMP Proxy can be reordered by time if received out of order. A value of 0 turns off this feature so that no reordering occurs.

Note: If the `cmp.size_option` is set to TRUE the number and size of the log files will be limited, thus reducing the likelihood of the disk becoming full. On the other hand, if the `cmp.size_option` is set to FALSE the log file is rotated by time, as a result the size and number of log files can grow. In this case users should be careful to monitor how much disk space is used by these files. By default, log files are rotated by size and can consist of up to 5 files of 10 MB each.

9.3.2 Metrics File (METRICS)

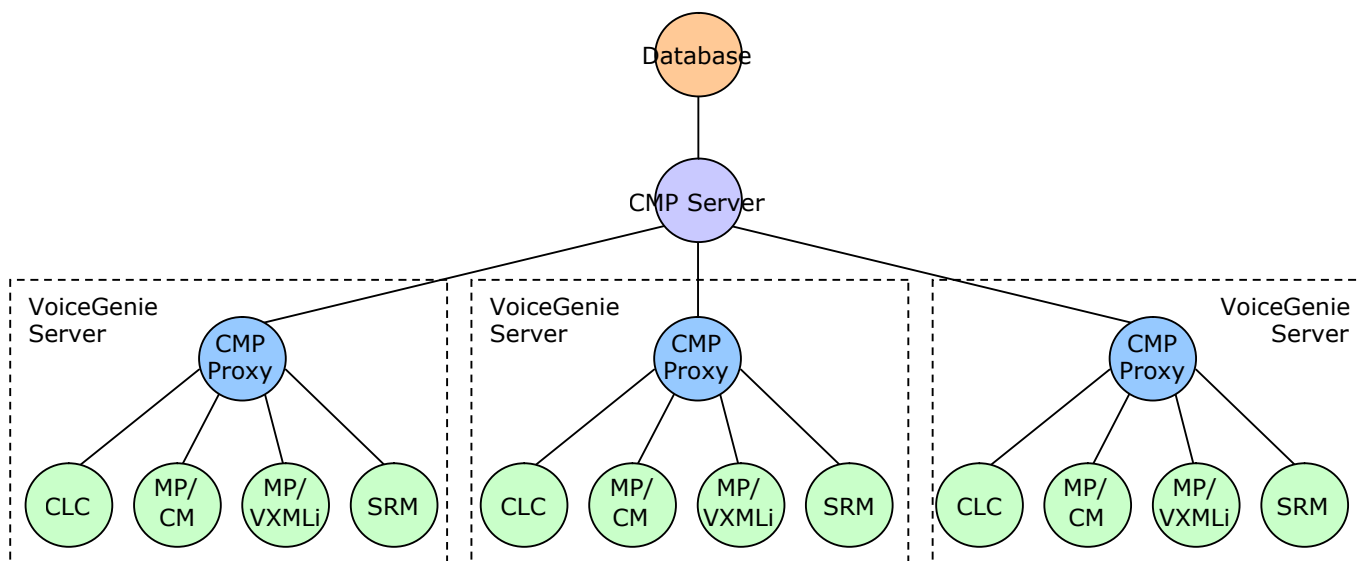
If the METRICS sink is enabled, metrics data can be logged to a file. The name of the file is specified in the `cmp.metrics_file` configuration parameter and is set to `/usr/local/phoneweb/logs/pw_metricsfile` by default on Linux and `C:\VoiceGenie\mp\logs\pw_metricsfile` on Windows. The METRICS file is rotated using the same options as those for the FILE sink. They are described above. This sink should only be enabled in the CMP Proxy configuration.

Note: Much like the FILE sink, users should be cognizant of the amount of disk space used by metrics files that are created. By default, the metrics file is rotated after 24 hours (1440 minutes).

There is no longer a `pw_billing` file since this information is no also logged as metrics data. As a result, all billing information should be gathered from the metrics file or database.

9.3.3 Upstream (UPSTREAM)

As a reminder, the OA&M Framework is a communication infrastructure that resembles an inverse tree structure:



When the UPSTREAM sink is enabled alarms and logs can be sent upstream in the OA&M Framework. For example, logs from the Media Platform components (Call Manager and VoiceXML Interpreter) can be sent upstream to the CMP Proxy. These logs, as well as logs from the CMP Proxy can be sent upstream to the CMP Server. Any logs received by the CMP Server can be sent upstream to the database. By default all alarms information is sent to the database where it can be easily viewed by the Alarm Browser which is found under the Monitoring tab of the SMC.

Note: The CMP Proxy is responsible for writing server level log files such as the `pw_metricsfile`, as a result, it must receive information from the Call Manager and VoiceXML Interpreter. These metrics logs can be received out of time order. As a result, some log data may be written to file out of chronological sequence.

9.3.4 Syslog (SYSLOG)

If the SYSLOG sink is enabled alarms and logs can be sent to the system log. Under Linux the logs are sent to Syslog, which is a daemon process that listens for data on port 514. All data received is written to a log file, by default this file is found at `/usr/local/phoneweb/logs/pw_logfile`. Under Windows the logs are sent to the Application Log in Event Viewer, which can be accessed under the Administrative Tools section of the Control Panel.

Note: On Linux Syslog writes logs to the `/usr/local/phoneweb/logs/pw_logfile` file, if this file is deleted the Syslog needs to be restarted to recreate this file. To start, stop or restart Syslog you must be the root user. To become the root user log in to the system and type in `su`, then enter the root password when prompted.

Then, to start the Syslog, issue the following command:

```
/etc/init.d/syslogd start
```

Then, to stop the Syslog, issue the following command:

```
/etc/init.d/syslogd stop
```

Then, to restart the Syslog, issue the following command:

```
/etc/init.d/syslogd restart
```

9.3.5 VoiceGenie SNMP (SNMP)

The SNMP sink should only be enabled in the CMP Proxy configuration. If this sink is enabled alarms can be sent to the VG SNMP component. These alarms can then be sent to the Network Management Systems that are configured to receive SNMP Traps in the VoiceGenie SNMP configuration. See the section on VoiceGenie SNMP for more details.

9.3.6 Email (EMAIL)

If the EMAIL sink is enabled alarms and logs can be sent to an email address. The email address where alarms are sent is specified by the `cmp.email` configuration parameter. The value of this parameter is a valid email address. Currently, only one email address is supported.

Note: The EMAIL sink does not work under Windows.

9.3.7 Tools – Log Agent (LOGAGENT)

The LOGAGENT sink should only be enabled in the CMP Proxy configuration. If this sink is enabled, metrics information can be sent to the Tools Log Agent component. These metrics are used to generate various statistics and call processing information. See the Tools user documentation for more details.

9.3.8 User Defined Sinks

The OA&M Framework allows users to create their own user defined sinks. This can be extremely useful for customers who have specific logging requirements. For example, many customers require logs to be written in a certain specific format, or have existing billing systems that need to be integrated with.

A user defined sink consists of a DLL (dynamically linked library) developed by either VoiceGenie or the customer that is loaded by either the CMP Proxy, or CMP Server. Based on configuration, logs can be directed towards that user defined sink. To specify a custom sink add a label to the `cmp.log_sinks` parameter that is not one of the predefined sinks, the following is an example:
`cmp.log_sinks = FILE|UPSTREAM|METRICS|SNMP|SYSLOG|CUSTOM1`

The label `CUSTOM1` is the name of the custom user defined sink. Also, the location of the DLL that this sink refers to needs to be specified, the following is an example:

```
cmp.log_dll.CUSTOM1 = /usr/local/cmp-proxy/bin/libCustom1sink.so
```

The parameter `cmp.log_dll.<sink label>`, where `<sink label>` is the label specified in the `cmp.log_sinks` parameter, specifies the location of the DLL or shared object. To direct logs to the user defined sink use the log filter mask as defined in the next section.

Please refer to information on how to create your own user defined sink.

9.4 Log Filtering Masks

VoiceGenie software logs a great deal of information at various severity levels as outlined in 9.2 Severity Levels; as a result, a mechanism is required to be able to filter only the desired logs to the desired sinks. This is accomplished through a number of log filters that are defined in configuration. A log filter exists for each severity level; the following configuration parameters define the log filters:

- `cmp.metrics`
- `cmp.log_0`
- `cmp.log_1`
- `cmp.log_2`
- `cmp.log_3`
- `cmp.log_4`
- `cmp.log_5`

The following is an example of a typical log filter as viewed from the SMC:

[cmp_log_0](#) hide

Log mask for data logged at log level 0 (Dynamic)

file	upstream	metrics	snmp	syslog	
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Logs from the System via CMP
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Base CMP Communication
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Configuration Service
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Logging Service
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Provision Service
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Subscription Service
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Data Service
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CMP Agent
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CMP Proxy
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Command Line Console
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Command Line Console Telnet Server Communication
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	VG-SNMP Agent
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CMP Proxy Process Utilization
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Component XML Definition Parser
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Media transport base
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Media transport stored
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Media transport Dialogic
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Media transport RTP
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Media transport Brooktrout
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Line manager base
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Line manager Dialogic
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Line manager dialogic
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Line manager SIP
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Line manager METROONE
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Line manager H323
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Line manager Brooktrout
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Line manager SIP2

Each column in the configuration represents a different log sink (destination), the label at the top specifies the log sinks (i.e. file, upstream, etc.). Each row is a separate module within the VoiceGenie software that is capable of producing logs. When the checkbox in a given sink column and a given module row is checked, all logs from that module will be directed to that sink. In addition, if only specific logs within that module are required, the user can specify only those logs in the textbox provided. The format of this configuration is a comma separated list of number and ranges, the following is an example:

4001-4004, 5001, 5002

For metrics the configuration is slightly different. In this case the column still represents the sink, however the row represents the specific metric and no additional textbox is required. The following is an example of the metrics filter mask:

[cmp.metrics](#) log mask for metrics data (Dynamic)

hide

file	upstream	metrics	snmp	syslog	
<input type="checkbox"/> all <input type="checkbox"/> none	<input type="checkbox"/> all <input type="checkbox"/> none	<input type="checkbox"/> all <input type="checkbox"/> none	<input type="checkbox"/> all <input type="checkbox"/> none	<input type="checkbox"/> all <input type="checkbox"/> none	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Inbound Call Begin (incall_begin)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Inbound Call End (incall_end)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Inbound Call Reject (incall_reject)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Bridge Call Begin (bridge_begin)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Bridge Call End (bridge_end)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Bridge Call Reject (bridge_reject)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Call Begin (call_begin)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Call End (call_end)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Call Reject (call_reject)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Outbound Call Begin (outcall_begin)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Outbound Call End (outcall_end)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Outbound Call Reject (outcall_reject)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Application Name (app_name)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Transfer Begin (transfer_begin)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Transfer End (transfer_end)
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Transfer Result (transfer_result)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	WF Request (wf_request)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	WF Response (wf_response)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Prompt Begin (prompt_begin)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Prompt End (prompt_end)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Record Begin (record_begin)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Record End (record_end)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Subdialog Begin (subdialog_begin)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Subdialog End (subdialog_end)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Input Begin (input_begin)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Input End (input_end)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ASR Trace (asr_trace)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	DTMF (dtmf)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Form Enter (form_enter)
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Form Select (form_select)

Users can click on the all or none button to select all or deselect all checkboxes for a given sink. Also, these filters are usually hidden unless explicitly shown using the show button. To hide the filters click on hide.

9.5 Enabling or Disabling Tracing/Debugging

Logs that are from severity CRIT (`cmp.log_0`) to INFO (`cmp.log_4`) are filtered as defined by the log filters defined above. Tracing (`cmp.log_5`) information has a special configuration parameter in addition to the filter mask. This configuration parameter is called `cmp.trace_flag` and can be used to turn tracing completely on or off. The following table outlines the values for this parameter and what it means:

Parameter	Value	Description
cmp.trace_flag	TRUE	Tracing/Debugging On Depending on Masks; log as defined by the configuration parameter <code>cmp.log_5</code> .
cmp.trace_flag	FALSE	Tracing/Debugging Always Off; this should be the setting in production machines to ensure faster processing.

In the SMC this configuration is represented as follows:

- [cmp.trace_flag](#)
- Tracing/Debugging Always Off
 - Tracing/Debugging On Depending on Masks
- Determines if any logs at level `log_5` (tracing/debugging) should be logged (Dynamic)

This value can be changed permanently in configuration via SMC, or can be changed at runtime in a transient manner using the `tracelevel` command, which is described in *VoiceGenie 7.2 OA&M Framework – CLC User's Guide*.

9.6 Default Logging Configuration

The default logging configuration is to have all alarms (Critical, Error and Warning) and notice (NOTE) events sent upstream to the database as well as to SYSLOG and SNMP; all Metrics sent to the Metrics file and significant Metric events sent to the database; and all other logs (Informational, Trace/Debug) written to a local file only. In addition, the `cmp.trace_flag` is set to FALSE, so that tracing is turned off, but can easily be turned on by simply using the `tracelevel` command. For further details please look at the default configuration files located in the Appendices.

9.7 UTC Logging

Each OA&M log message stores its construction time in UTC (Universal Time Coordinated) format. UTC is similar to Greenwich Mean Time and is based on the number of seconds that have elapsed since 00:00:00 UTC January 1, 1970. The local time of a system is offset from UTC by an amount based on the operating system's time zone. The log's construction time can be left as UTC or converted to the system's local time before a log message is sent to a sink. This is controlled by the following configuration parameter:

Parameter	Value	Description
cmp.utc.<sink>	TRUE	This <sink> will save log construction times in UTC format.

Parameter	Value	Description
cmp.utc.<sink>	FALSE (default)	This <sink> will save log construction times using the system's local time, by adding or subtracting a specific amount of time from the UTC time. This is based on the system's time zone and daylight savings time settings.

The <sink> name can be any of the supported sinks (FILE, UPSTREAM, METRICS, SNMP, SYSLOG, LOGAGENT). However, SYSLOG does not show log construction times, so the `cmp.utc.syslog` parameter has no effect. Also, for UPSTREAM sinks, `cmp.utc.upstream` only has an effect for the CMP Server. It will control whether UTC format or the CMP Server's local time is used when writing to the `CallLog` table in the NDM database. For all other components, the `cmp.utc.upstream` parameter has no effect because, in these cases, the OA&M framework requires that log construction times remain in UTC format. This helps avoid confusion when systems of a single cluster are running in different time zones.

In order to help maintain accurate UTC time, it is recommended that all systems synchronize on a regular basis using NTP (Network Time Protocol).

9.8 Log File Cleanup

Some of the log sinks above result in the creation of log files on the system, notably the FILE sink and the METRICS sink. Also, there are a number of other system log files that can grow large and fill up the disk if they are not cleaned up periodically. As a result, there are a number of settings in the system crontab that have been added to ensure that these files are deleted in a timely fashion. These settings are as follows:

In `/etc/crontab`:

```
13 4 * * * root find /usr/local/tomcat/logs/ -name
'localhost*.txt' -mtime +13 -exec rm -f {} \;
13 4 * * * root find /usr/local/tomcat/logs/ -name
'cmpagent*.log' -mtime +13 -exec rm -f {} \;
13 4 * * * root find /usr/local/tomcat/logs/ -name
'catalina*.txt' -mtime +13 -exec rm -f {} \;
13 4 * * * root find /usr/local/tomcat/logs/ -name
'catalina*.out' -mtime +13 -exec rm -f {} \;
10 4 * * * root find /usr/local/cmp-proxy/logs -name
'CMP.log.cmpproxy*' -mtime +13 -exec rm -f {} \;
10 4 * * * root find /usr/local/phoneweb/logs -name
'pw_logfile*' -mtime +59 -exec rm -f {} \;
10 4 * * * root find /usr/local/phoneweb/logs -name
'pw_metricsfile*' -mtime +59 -exec rm -f {} \;
```

The last 2 lines ensure that all `pw_metricsfile`s (created by the METRICS sink) and all `pw_logfile` (created by Syslog) that are older than 60 days are

deleted. The first 4 lines ensure that Tomcat logs and CMP Proxy logs that are older than 2 weeks are deleted. Any other files that may take up disk space should also be added to this list.

Note: In Windows these logs are not automatically scheduled to be removed, extra scheduled tasks need to be added.

9.9 Log Queues

All VoiceGenie processes have a separate logging thread that is responsible for logging data to the sinks outlined above. The logging thread has a queue associated with it; other threads within the VoiceGenie process queue up log requests into this queue. This queue has a limit associated with it and is defined by the configuration parameter `cmp.log_queue_limit`. Once the queue limit is reached the calling threads are throttled so that the queue does not grow uncontrolled. This information is provided for informational purposes.



Appendix

A

Log IDs: Modules and Specifiers

Every item that is logged contains a Log ID, this ID contains information about the specific type of information logged. What the Log ID represents depends on whether the log is a metrics log or a CRIT, EROR, WARN, NOTE, INFO or DEBUG log.

A.1 CRIT, EROR, WARN, NOTE, INFO or DEBUG Log IDs

The Log IDs for these log messages contain 2 pieces of important information:

1. Module
2. Specifier

The Module is the particular module within the VoiceGenie software components that generated the log, while the Specifier is the specific information that is being logged from that module. The Log ID is always represented as an 8 digit hexadecimal number. The first three numbers are the Module, the last five are the Specifier. The following sections outline all the Modules that exist in the VoiceGenie software.

A.1.1 CMP Modules

Name	Hex Module Number	Description
CMPLOGMOD_SYSTEM	000	Reserved for logs from the system via CMP
CMPLOGMOD_CMPCOMM	001	Base CMP Communication

Name	Hex Module Number	Description
CMPLOGMOD_CFGSERVICE	002	Configuration Service
CMPLOGMOD_LOGSERVICE	003	Logging Service
CMPLOGMOD_PRVSERVICE	004	Provision Service
CMPLOGMOD_SUBSCRSERVICE	005	Subscription Service
CMPLOGMOD_DATASERVICE	006	Data Service
CMPLOGMOD_CMPAGENT	007	CMP Agent
CMPLOGMOD_CMPPROXY	008	CMP Proxy
CMPLOGMOD_CMPENGINE	009	CMP Engine (Server)
CMPLOGMOD_DBTHREAD	00A	Database Access and Working Thread
CMPLOGMOD_CMPCLC	00B	Command Line Console
CMPLOGMOD_CLCCOMM	00C	Commmand Line Console Telnet Server Communication
CMPLOGMOD_VGSNMP	00D	VG-SNMP Agent
CMPLOGMOD_VGPROCESSUTIL	00E	Process Utilization
CMPLOGMOD_COMPDEFPARSER	00F	Component XML Definition Parser
CMPLOGMOD_SBCLOGSINK	070	SBC Logging Sink Logs
CMPLOGMOD_CDRCOLLECTOR	071	Call Detail Record Collector (used by CMP Engine)
CMPLOGMOD_CHANGEREQMGR	072	Configuration and Provision Change Request Manager
CMPLOGMOD_EXAGENT	073	External Agent Processes

A.1.2 Call Manager Modules

Name	Hex Module Number	Description
CMLOGMOD_MTBASE	010	Media transport base
CMLOGMOD_MTSTORED	011	Media transport stored

Name	Hex Module Number	Description
CMLOGMOD_MTDLGC	012	Media transport Dialogic
CMLOGMOD_MTRTP	013	Media transport RTP
CMLOGMOD_MTBRKT	014	Media transport Brooktrout
CMLOGMOD_LMBASE	015	Line manager base
CMLOGMOD_LMDLGC	016	Line manager Dialogic
CMLOGMOD_DLGCEV	017	Line manager dialogic
CMLOGMOD_LMSIP	018	Line manager SIP
CMLOGMOD_LMMETROONE	019	Line manager METROONE
CMLOGMOD_LMH323	01A	Line manager H323
CMLOGMOD_LMBRKT	01B	Line manager Brooktrout
CMLOGMOD_SESSMGR	01C	Session manager SESSMGR
CMLOGMOD_CALLSESSION	01D	Session manager CALLSESSION
CMLOGMOD_SMPWMAPI	01E	Session manager pwmapi
CMLOGMOD_SMMAIN	01F	Session manager main
CMLOGMOD_CMGRCMPAGENT	020	Session manager callmgr cmp agent
CMLOGMOD_CMUTIL	021	Session manager callmgr util
CMLOGMOD_APPMODULE	022	CMAPI application
CMLOGMOD_VXMLMODULE	023	CMAPI vxml
CMLOGMOD_APPCCM	024	CMAPI APPCCM
CMLOGMOD_CONTCHECK	025	CMAPI CONTCHECK
CMLOGMOD_REMDIAL	026	CMAPI REMDIAL
CMLOGMOD_SBCPOLICY	027	CMAPI SBCPOLICY
CMLOGMOD_LMSIP2	028	Line manager SIP2
CMLOGMOD_CONFERENCE	029	CMAPI CONFERENCE
CMLOGMOD_PORTCOUNT	02A	CMAPI PORTCOUNT
CMLOGMOD_SQA	02B	Service Quality Analyst

Name	Hex Module Number	Description
CMLOGMOD_MTSNOWSHORE	02C	Media transport Snowshore
CMLOGMOD_LMSNOWSHORE	02D	Line manager Snowshore
CMLOGMOD_SNOWSHORE	02E	Snowshore Main
CMLOGMOD_MTMPC	02F	MPC Wrapper

A.1.3 VoiceXML Interpreter Modules

Name	Hex Module Number	Description
VXMLILOGMOD_MAIN	030	main application events like startup, shutdown, initialize components, load configuration etc.
VXMLILOGMOD_CMPAGENT	031	CMP interface (CMP Agent initialization, cleanup etc.)
VXMLILOGMOD_FMINTERFACE	032	fetching module interface
VXMLILOGMOD_VGCOMM	033	VGComm interface
VXMLILOGMOD_SHCOMM	034	Shared memory communication
VXMLILOGMOD_SHMEM	035	Shared memory general use
VXMLILOGMOD_UTIL	036	Util functions
VXMLILOGMOD_CALLCTRL	037	call control (transfer, call, join)
VXMLILOGMOD_LOGTAG	038	log (<logtag)
VXMLILOGMOD_RSLTPROC	039	result processor
VXMLILOGMOD_DTMFRECO	03A	DTMF recognizer
VXMLILOGMOD_SUBDLG	03B	subdialogs
VXMLILOGMOD_FLOWCTRL	03C	flow control
VXMLILOGMOD_PARSER	03D	parser
VXMLILOGMOD_COMPILER	03E	compiler
VXMLILOGMOD_GRAMMARS	03F	grammars
VXMLILOGMOD_PROMPTS	040	prompts

Name	Hex Module Number	Description
VXMLILOGMOD_FIA	041	FIA
VXMLILOGMOD_EVENTS	042	events
VXMLILOGMOD_EMAILER	043	emailer interface
VXMLILOGMOD_SCRIPTENG	044	script engine
VXMLILOGMOD_EXCONTENT	045	executable content
VXMLILOGMOD_FORKING	046	forking vxmli instance
VXMLILOGMOD_INTSESSMSG	047	inter-session messaging
VXMLILOGMOD_RECORDING	048	recording tag
VXMLILOGMOD_FORMITEMS	049	<block> <object>
VXMLILOGMOD_FILLED	04A	<filled>
VXMLILOGMOD_INPUTITEM	04B	<menu> <field> <initial>
VXMLILOGMOD_FORM	04C	<form>
VXMLILOGMOD_STATEMACHINE	04D	state machine general
VXMLILOGMOD_DICT	04E	The dictionaries
VXMLILOGMOD_SQA	04F	SQA Related Logs

A.1.4 Web Proxy (IProxy) Modules

Name	Hex Module Number	Description
FMLOGMOD_MAIN	050	main
FMLOGMOD_HTTP	051	HTTP: request, response
FMLOGMOD_CMPAGENT	052	CMP agent interface
FMLOGMOD_SHCOMM	053	Shared memory communication
FMLOGMOD_SHMEM	054	Shared memory general use
FMLOGMOD_CLIENT	055	Client interface
FMLOGMOD_REQUEST	056	Fetch request handling

Name	Hex Module Number	Description
FMLOGMOD_RESPONSE	057	Fetch response handling
FMLOGMOD_CURL	058	cURL lib interface
FMLOGMOD_CACHING	059	Caching

A.1.5 VRM Server Modules

Name	Hex Module Number	Description
VRMLOGMOD_MRCPV1STACK	060	VRM MRCPv1 Stack
VRMLOGMOD_CLIENT	061	VRM Client
VRMLOGMOD_SVR_PROTOCOL	062	VRM Server Protocol Module
VRMLOGMOD_SVR_RML	063	VRM Server RML
VRMLOGMOD_SVR_LEGACY	064	VRM Server LegacyEngine
VRMLOGMOD_SVR_NATIVE_ASR	065	VRM Server NativeClient ASR
VRMLOGMOD_SVR_NATIVE_TTS	066	VRM Server NativeClient TTS
VRMLOGMOD_PROXY	067	VRM Proxy
VRMLOGMOD_SQA	068	SQA Related Logs

A.1.6 Common Modules

Name	Hex Module Number	Description
VGULOGMOD_MAIN	080	Main VG Utility Components
VGULOGMOD_VGLOG	081	VoiceGenie Logging Subsystem
VGULOGMOD_MTINTERNAL	082	Mt internal
VGULOGMOD_VGCOMM	083	Callmgr and VoiceXML Interpreter VG Comm
VGULOGMOD_UNKNOWN	084	This is used when the MODULE_ID is not defined

A.1.7 CCP Modules

Name	Hex Module Number	Description
CCPLOGMOD_CORE	090	main
CCPLOGMOD_SCSS7BRKT	091	ss7 modules
CCPLOGMOD_SCICM	092	ICM modules
CCPLOGMOD_SCCMAPI	093	CMAPI modules
CCPLOGMOD_CCPPROXY	094	ccpproxy modules
CCPLOGMOD_RM	095	ccprm modules
CCPLOGMOD_SCCMPAGENT	096	ccpcmpagent modules
CCPLOGMOD_CCXML	097	ccpccxmli modules
CCPLOGMOD_CCXMLI	098	ccxmli modules

A.1.8 Tools Modules

Name	Hex Module Number	Description
CALOGMOD_CORELIB	0A0	Core NXP Analyst library
CALOGMOD_LOGAGENTLIB	0A1	Log Agent library
CALOGMOD_LOGAGENTAPP	0A2	Log Agent CMP Agent
CALOGMOD_LOGMONLIB	0A3	Log Monitor library
CALOGMOD_LOGMONAPP	0A4	Log Monitor CMP Agent

A.1.9 Media Processing Component Modules

Name	Hex Module Number	Description
MPCLOGMOD_MEDIAMGR	0B0	MPC Media Manager
MPCLOGMOD_CONTROL	0B1	MPC Control Layer
MPCLOGMOD_RTP_INTERFACE	0B3	MPC RTP Interface

Name	Hex Module Number	Description
MPCLOGMOD_DSP	0B4	MPC DSP
MPCLOGMOD_RESERVEDSTART	0B5	MPC reserved start
MPCLOGMOD_RESERVEDEND	0BF	MPC reserved end

A.1.10 NGI

Name	Hex Module Number	Description
LOGMOD_NGI	0C0	NGI Log

A.2 Metrics Log IDs

In the case of metrics there is no module, instead, the Log ID specifies the type of metric that is being logged. The following table outlines the Log IDs and the metric that it relates to:

Metric	Log ID
VGLOG_INCALL_BEGIN	0
VGLOG_INCALL_END	1
VGLOG_INCALL_REJECT	2
VGLOG_BRIDGE_BEGIN	3
VGLOG_BRIDGE_END	4
VGLOG_BRIDGE_REJECT	5
VGLOG_CALL_BEGIN	6
VGLOG_CALL_END	7
VGLOG_CALL_REJECT	8
VGLOG_OUTCALL_BEGIN	9
VGLOG_OUTCALL_END	10
VGLOG_OUTCALL_REJECT	11
VGLOG_APP_NAME	12

Metric	Log ID
VGLOG_TRANSFER_BEGIN	13
VGLOG_TRANSFER_END	14
VGLOG_TRANSFER_RESULT	15
VGLOG_WF_REQUEST	16
VGLOG_WF_RESPONSE	17
VGLOG_PROMPT_BEGIN	18
VGLOG_PROMPT_END	19
VGLOG_RECORD_BEGIN	20
VGLOG_RECORD_END	21
VGLOG_SUBDIALOG_BEGIN	22
VGLOG_SUBDIALOG_END	23
VGLOG_INPUT_BEGIN	24
VGLOG_INPUT_END	25
VGLOG_ASR_TRACE	26
VGLOG_DTMF	27
VGLOG_FORM_ENTER	28
VGLOG_FORM_SELECT	29
VGLOG_FORM_EXIT	30
VGLOG_MENU_ENTER	31
VGLOG_FILLED_ENTER	32
VGLOG_GOTO	33
VGLOG_EVENT	34
VGLOG_LOG	35
VGLOG_APP_END	36
VGLOG_PARSE_ERROR	37
VGLOG_PARSE_WARNING	38

Metric	Log ID
VGLOG_EXEC_ERROR	39
VGLOG_EXEC_WARNING	40
VGLOG_COMPILE_DONE	41
VGLOG_FETCH_REQUEST	42
VGLOG_FETCH_RESPONSE	43
VGLOG_EVAL_COND	44
VGLOG_EVAL_SCRIPT	45
VGLOG_EVAL_VAR	46
VGLOG_EVAL_EXPR	47
VGLOG_INT_BEGIN	48
VGLOG_INT_END	49
VGLOG_INT_TRANS	50
VGLOG_INT_STATE	51
VGLOG_INCALL_INITIATED	52
VGLOG_BRIDGE_INITIATED	53
VGLOG_CALL_INITIATED	54
VGLOG_OUTCALL_INITIATED	55
VGLOG_CALL_REFERENCE	56
VGLOG_ASRLOAD_BEGIN	57
VGLOG_ASRLOAD_END	58
VGLOG_ASRUNLOAD_BEGIN	59
VGLOG_ASRUNLOAD_END	60
VGLOG_ASRSTART_BEGIN	61
VGLOG_ASRSTART_END	62
VGLOG_ASRSTOP_BEGIN	63
VGLOG_ASRSTOP_END	64

Metric	Log ID
VGLOG_ASR_AUDIOREADY	65
VGLOG_ASR_BARGEIN	66
VGLOG_ASR_DONE	67
VGLOG_TTS_PLAY	68
VGLOG_TTSSTOP_BEGIN	69
VGLOG_TTSSTOP_END	70
VGLOG_TTS_DONE	71
VGLOG_TRANSFER_INITIATED	72
VGLOG_TRANSFER_CONNECTED	73
VGLOG_OUTCALL_REQUESTED	74
VGLOG_ASR_FAIL	75
VGLOG_FILLING	76
VGLOG_CHOICE_SELECT	77
VGLOG_SUBMIT	78
VGLOG_ROOT_NAME	79
VGLOG_EVENT_ENTER	80
VGLOG_EVENT_EXIT	81
VGLOG_PXY_SETPARAMS	82
VGLOG_PXY_SETPARAMS_RESP	83
VGLOG_PXY_GETPARAMS	84
VGLOG_PXY_GETPARAMS_RESP	85
VGLOG_PXY_GETSVRINFO	86
VGLOG_PXY_GETSVRINFO_RESP	87
VGLOG_PXY_STOP	88
VGLOG_PXY_STOP_RESP	89
VGLOG_PXY_LOAD	90

Metric	Log ID
VGLOG_PXY_LOAD_RESP	91
VGLOG_PXY_RECOG	92
VGLOG_PXY_RECOG_RESP	93
VGLOG_PXY_PROMPTDONE	94
VGLOG_PXY_PROMPTDONE_RESP	95
VGLOG_PXY_SPEAK	96
VGLOG_PXY_SPEAK_RESP	97
VGLOG_PXY_PAUSE	98
VGLOG_PXY_PAUSE_RESP	99
VGLOG_PXY_RESUME	100
VGLOG_PXY_RESUME_RESP	101
VGLOG_PXY_CONTROL	102
VGLOG_PXY_CONTROL_RESP	103
VGLOG_PXY_GETRESULT	104
VGLOG_PXY_GETRESULT_RESP	105
VGLOG_PXY_BARGEIN	106
VGLOG_PXY_BARGEIN_RESP	107
VGLOG_PXY_STARTOFSPEECH	108
VGLOG_PXY_STARTOFSPEECH_RESP	109
VGLOG_PXY_MARKER	110
VGLOG_PXY_MARKER_RESP	111
VGLOG_PXY_SPEAKCOMPLETE	112
VGLOG_PXY_SPEAKCOMPLETE_RESP	113
VGLOG_PXY_RECOGCOMPLETE	114
VGLOG_PXY_RECOGCOMPLETE_RESP	115
VGLOG_PXY_ERROR	116

Metric	Log ID
VGLOG_PXY_ERROR_RESP	117
VGLOG_PXY_SVRDISCONNECT	118
VGLOG_PXY_CLNTDISCONNECT	119
VGLOG_TTS_MARK	120
VGLOG_PXY_OPEN	121
VGLOG_PXY_OPEN_RESP	122
VGLOG_PXY_CLOSE	123
VGLOG_PXY_CLOSE_RESP	124
VGLOG_PXY_REDIR	125
VGLOG_PXY_REDIR_RESP	126
VGLOG_FILLED_EXIT	127
VGLOG_DTMF_END	128
VGLOG_LINK_TRIGGERED	129
VGLOG_APPL_BEGIN	130
VGLOG_VXMLI3	131
VGLOG_MRCP_TRACE	132
VGLOG_PROMPT_START	133
VGLOG_PROMPT_PLAY	134
VGLOG_MENU_EXIT	135



Appendix

B

Database Schema

The CMP uses MySQL database server to store information, currently, no other database servers are supported. There are 2 databases that store information, the NDM database and the CallHistory database. The default login and password for both databases is pw and pw respectively.

Provided that the default setup has not been changed, users can access the database by typing the following at the console prompt:
`mysql -u pw -ppw NDM` or `mysql -u pw -ppw CallHistory`

B.1 NDM Database

The following tables exist in this database:

Table Name	Description
ActivityLog	Logs all activity that occurs on the SMC including logins, logouts, and operations performed.
ApplicationData	Stores the names of the initial applications accessed in each call that comes into the Media Platform. The application name is defined by the meta tag with the name attribute set to application.
BConfig	Stores configuration information about all the component types supported by the system.
CMPUser	Contains user information.
CacheManifest	Stores the ache manifest used to preload or purge the Squid Web Cache.
CallData	Stores the call detail record (CDR) for each call that comes into the Media Platform.

Table Name	Description
CallLog	Stores all call related and software logs sent upstream by the CMP Server.
ComponentType	Contains the component types that are valid.
Config	Stores all configuration names and their configuration type
ConfigProfile	Stores details about each profile that is created.
ConfigProfileRelation	Stores the IDs of configurations associated with each profile.
DeploymentStatus	Stores information about the status of all deployment attempts.
Groups	Contains information about the user groups that are available.
HistHealthStatus	Stores the historical health status of each component.
HistInitStatus	Stores the historical initialization status of each component.
HistProcessStatus	Stores the historical process (CPU & memory) status of each component.
MetaData	Stores meta data used by the system for certain scripts and configuration profiles.
Network	Contains the list of components in the CMP network as well as the user defined clusters.
Parameter	Stores all parameter names and values for every configuration in the cluster.
ParameterHistory	Stores previous values of all parameters for every configuration in the cluster.
ProductComponentRelation	Stores the mapping between each product and its component types.
ProductFile	Stores information about products that have been added to the cluster.
Provision	Contains the centralized provision information.
ProvisionRelation	Stores the mapping of what provisioning is deployed to what component.

Table Name	Description
ProvisionType	Contains the provision types that are valid.
Relation	Stores the mapping of what configuration is deployed to what component.

The CallLog table, HistHealthStatus table, HistInitStatus table and HistProcessStatus tables contain log and health data; the Network table contains information about the CMP network of components. The schema for these tables is outlined in the sections below.

B.1.1 CallLog Table

This table contains all metrics, billing and log data sent by the platforms. The table structure is as follows:

Field Name	Type	Description	Values
callID	varchar(100)	An ID that is unique for all calls in a single deployment. Consists of a 64 bit number.	i.e. 003603F8-2000A1BA
eventTime	varchar(30)	The time at which the call event occurred, to millisecond accuracy.	i.e. 2003-10-16/19:45:19.137
logType	varchar(20)	The type of log.	METRIC, CRIT, EROR, WARN, NOTE, INFO, DBUG
logID	varchar(20)	The particular log ID or sub-log type. Refer to 9.9 Log Queues.	
info	text	The data contained within the log.	
sourceID	int(11)	The networkID of the software component that generated log.	i.e. 2
callLogID	bigint(20)	An auto increment number to uniquely identify each row.	

B.1.2 HistHealthStatus Table

This table contains all health status information received by the components. The table structure is as follows:

Field Name	Type	Description	Values
networkID	int(32)	The network ID of the component which sent the health status information.	
eventTime	datetime	The time at which the health status information was received, to second accuracy.	i.e. 2003-10-16 19:45:19
health	text	The health string sent by the component.	

B.1.3 HistInitStatus Table

This table records when components either initialize or disconnect from the CMP network. The table structure is as follows:

Field Name	Type	Description	Values
networkID	int(32)	The network ID of the component which either initialized or disconnected with the CMP network.	
eventTime	datetime	The time at which the initialization event occurred, to second accuracy.	i.e. 2003-10-16 19:45:19
event	text	The event that occurred.	i.e. Initialize or Shutdown

B.1.4 HistProcessStatus Table

This table records the process status (CPU and memory utilization) for all connected components. Also, it records the disk usage for platforms. The table structure is as follows:

Field Name	Type	Description	Values
networkID	int(32)	The network ID of the component whose process status is being recorded.	
eventTime	datetime	The time at which the health status information was received, to second accuracy.	i.e. 2003-10-16 19:45:19

Field Name	Type	Description	Values
CPUUsage	varchar(32)	The percentage of the CPU used by the component with 2 decimal accuracy.	i.e. 8.68%
memoryUsage	varchar(32)	The amount of memory used by the component in megabytes.	i.e. 118MB
diskUsage	varchar(32)	The percentage of the /usr/local/ disk space used by the platform.	i.e. 4%

B.1.5 Network Table

This table stores information about the various components that are part of the CMP OA&M network of managed components. The table structure is as follows:

Field Name	Type	Description	Values
networkID	int(32)	The network ID of the component.	
name	varchar(32)	The name of the component, server name in the case of <code>type = 200</code> , otherwise component name.	
componentType	int(32)	The type of component.	0 = CMP Server 200 = CMP Proxy ≥ 300 = other VG software
host	varchar(200)	The hostname or IP address that the component is running on.	10.0.0.10 or foo.voicegenie.com
instance	int(32)	The instance of the component. Usually 1, only larger if using multiple instances of the VoiceXML Interpreter.	i.e. 1
configurationID	int(32)	The configurationID of the configuration used by this component.	
CPUUsage	varchar(32)	The percentage of the CPU used by the component with 2 decimal accuracy.	i.e. 8.68%
memoryUsage	varchar(32)	The amount of memory used by the component in megabytes.	i.e. 118MB

Field Name	Type	Description	Values
healthStatus	text	The last health status of the component.	
diskUsage	varchar(32)	The percentage of the /usr/local/ disk space used by the platform.	i.e. 4%
connStatus	varchar(32)	The connection status of the component.	online or offline
syncStatus	varchar(32)	The synchronization status of the component's configuration.	TRUE or FALSE

B.1.6 CallData

This table stores a call detail record for every call that comes in to the Media Platform. The table structure is as follows:

Field Name	Type	Description	Values
callID	varchar(100)	The unique callID of the call, each entry in this table has a unique callID since each entry represents one call or session.	i.e. 003603F8-2000A1BA
networkID	int(32)	The networkID of the CMP Proxy on the physical machine that the call came in to.	i.e. 2, refer to the Network table to get more information about the server.
span	int(8)	The span that the call occurred on.	0 = SIP or H323 101+ = Dialogic or Brooktrout
port	int(8)	The port that the call occurred on.	0 = SIP or H323 1+ = Dialogic or Brooktrout
eventTime	datetime	The start time of the call.	i.e. 2003-10-16 19:45:19
length	int(32)	The length of the call in seconds.	
connectDuration	int(32)	The length of the connection time in milliseconds	
callType	char(1)	The type of call.	I = inbound O = outbound using remdial C = outbound using call tag B = outbound using transfer

Field Name	Type	Description	Values
callStatus	char(1)	The status of the call.	S – successful & connected F – failed & connected R – rejected
recordStatus	char(2)	The status of the record.	C – complete record NS – no start for call NE – no end for call
dnis	varchar(255)	DNIS of call.	
initialURL	varchar(255)	The initial URL.	
ani	varchar(255)	ANI of call.	
endCode	varchar(100)	The endCode.	

The CallData, CallLog and ApplicationData tables contain call specific details and alarms. To clear these tables execute the following MySQL commands:

```
mysql -u pw NDM -ppw;
mysql> truncate CallData;
mysql> truncate CallLog;
mysql> truncate ApplicationData;
```

B.2 CallHistory Database

The following tables exist in this database:

Table Name	Description
HistAppDistribution	Contains the information required to generate the Application Distribution report.
HistCallLength	Contains the information required to generate the Call Length Distribution report.
HistCallVolume	Contains the information required to generate the Call Volume report.

The schema for these tables is outlined in the sections below.

B.2.1 HistAppDistribution

This table stores summarized application distribution information. This table is populated by the dbadmin script. The table structure is as follows:

Field Name	Type	Description	Values
appDistID	int(32)	A unique auto incremented ID for each entry.	
eventTime	datetime	The time period for the data, the data is summarized hourly.	i.e. 2003-10-16 19:00:00, 2003-10-16 20:00:00, etc.
networkID	int(32)	The networkID of the CMP Proxy on the physical machine that the data is summarized for.	i.e. 2, refer to the Network table to get more information about the server.
appName	varchar(255)	The name of the application.	i.e. Test
count	int(32)	The count for how many times that application (in appName) was accessed on the machine as defined by networkID in the time defined by timestamp.	

B.2.2 HistCallLength

This table stores summarized call length distribution information. This table is populated by the dbadmin script. The table structure is as follows:

Field Name	Type	Description	Values
callLengthID	int(32)	A unique auto incremented ID for each entry.	
eventTime	datetime	The time period for the data, the data is summarized hourly.	i.e. 2003-10-16 19:00:00, 2003-10-16 20:00:00, etc.
networkID	int(32)	The networkID of the CMP Proxy on the physical machine that the data is summarized for.	i.e. 2, refer to the Network table to get more information about the server.
group0Count	int(32)	A count of call lengths that fit in this range for the given timestamp on the given server as defined by the networkID.	
group1Count	int(32)	A count of call lengths that fit in this range for the given timestamp on the given server as defined by the networkID.	

Field Name	Type	Description	Values
group2Count	int(32)	A count of call lengths that fit in this range for the given timestamp on the given server as defined by the networkID.	
group3Count	int(32)	A count of call lengths that fit in this range for the given timestamp on the given server as defined by the networkID.	
group4Count	int(32)	A count of call lengths that fit in this range for the given timestamp on the given server as defined by the networkID.	
group5Count	int(32)	A count of call lengths that fit in this range for the given timestamp on the given server as defined by the networkID.	

B.2.3 HistCallVolume

This table stores summarized call volume information. This table is populated by the dbadmin script. The table structure is as follows:

Field Name	Type	Description	Values
callVolumeID	int(32)	A unique auto incremented ID for each entry.	
eventTime	datetime	The time period for the data, the data is summarized hourly.	i.e. 2003-10-16 19:00:00, 2003-10-16 20:00:00, etc.
networkID	int(32)	The networkID of the CMP Proxy on the physical machine that the data is summarized for.	i.e. 2, refer to the Network table to get more information about the server.
IBsuccess	int(32)	The number of successful inbound calls in the timeframe on a particular machine.	
OBsuccess	int(32)	The number of successful outbound calls using remdial in the timeframe on a particular machine.	

Field Name	Type	Description	Values
CBsuccess	int(32)	The number of successful outbound calls using call tag in the timeframe on a particular machine.	
BBsuccess	int(32)	The number of successful outbound calls using transfer in the timeframe on a particular machine.	
IBFail	int(32)	The number of failed inbound calls in the timeframe on a particular machine.	
OBFail	int(32)	The number of failed outbound calls using remdial in the timeframe on a particular machine.	
CBFail	int(32)	The number of failed outbound calls using call tag in the timeframe on a particular machine.	
BBFail	int(32)	The number of failed outbound calls using transfer in the timeframe on a particular machine.	
IBReject	int(32)	The number of rejected inbound calls in the timeframe on a particular machine.	
OBReject	int(32)	The number of rejected outbound calls using remdial in the timeframe on a particular machine.	
CBReject	int(32)	The number of rejected outbound calls using call tag in the timeframe on a particular machine.	
BBReject	int(32)	The number of rejected outbound calls using transfer in the timeframe on a particular machine.	

The CallHistory Database tables contain call information, and if this information is no longer needed the tables can be cleaned by executing the following MySQL commands:

```
mysql -u pw CallHistory -ppw;
mysql> truncate HistAppDistribution;
mysql> truncate HistCallLength;
mysql> truncate HistCallVolume;
```



Appendix

C

VoiceGenie SNMP Traps

C.1 CMP Proxy Traps

The prefix for all CMP Proxy OIDs is .1.3.6.1.4.1.7469.251.1.200.

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-INVALID-MSG-TYPE-CMPP	1048597	WARN	Invalid Message Type Sent or Received	An OA&M message is being ignored	VoiceGenie software version mismatch	REVG
VGLOG-CANNOT-CREATE-SERVER-CMPP	1048598	WARN	Error Creating Server Socket	The CMP Proxy will not establish connections with any of the CMP Agents.	Ethernet problem or the <code>cmp.proxy_port</code> parameter is not properly set.	REVG
VGLOG-CANNOT-CREATE-CLIENT-CMPP	1048599	WARN	Error Creating Client Socket	The CMP Proxy will not establish a connection with the CMP Server	Ethernet problem or an error in one of the CMP Proxy's connection parameters.	REVG

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-CFG-WRITE-FAIL-CMPP	2097178	WARN	Configuration file could not be written	Configuration file will not be updated.	Disk may be full.	CKCFG
VGLOG-DSRV-INVALID-QUERY-CMPP	6291477	WARN	Invalid data query string received	An SNMP get or set request may fail.	A request for an invalid parameter, perhaps from the SNMP Agent.	REVG
VGLOG-DSRV-UNSUPPORTED-VAR-CMPP	6291478	WARN	Query for unsupported variable received	An SNMP get or set request may fail.	A request for an invalid parameter, perhaps from the SNMP Agent.	REVG
VGLOG-COMPONENT-NOT-STARTED	8388865	CRIT	Component could not be Started	A component that is supposed to run on the platform will not start, making the system unusable.	The configuration may be incorrect, i.e. perhaps this component is not supposed to be started to begin with, or the binary of the component can not be located.	CKCFG/REVG Check CMP Proxy configuration, check the <code>cmp.component</code> s parameter and all related configuration parameters.
VGLOG-DISK-USAGE-CRITICAL	8388867	CRIT	Disk usage exceeded critical threshold	The disk usage on a partition has exceeded a critical threshold, usually 95%.	The disk is filling up.	CKOP Clear out disk space.

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-CPU-USAGE-CRITICAL	8388868	CRIT	CPU usage exceeded critical threshold	The CPU usage has exceeded a critical threshold, usually 80%.	The CPU is being used up by the system and may impact performance and user experience.	CKOP Restart VoiceGenie process that is using up the CPU.
VGLOG-MEM-USAGE-CRITICAL	8388869	CRIT	Mem usage exceeded critical threshold	The memory usage has exceeded a critical threshold, usually 2GB.	The memory on the system is being used up. May be a memory leak.	CKOP Restart VoiceGenie process that is using up memory.
VGLOG-ETH-STATUS-CRITICAL	8388870	CRIT	Ethernet interface down	Network connectivity is down and will effect system operation.	Network interface may be configured incorrectly, or plus may be pulled out.	CKNW Check Ethernet connection and network configuration.
VGLOG-COMPONENT-NOT-STOPPED	8389121	CRIT	A Component could not be Stopped	A VoiceGenie component that should be stoppable by the CMP Proxy could not be stopped. The system will be unstable.	The component may have been started by a process other than the CMP Proxy.	CKOP/REVG Reboot the system to ensure that the system is running in a stable fashion.
VGLOG-SCRIPT-NOT-STARTED-CMPP	8389122	EROR	Script could not be Run	A script that was initiated by a user and supposed to be run by the CMP Proxy could not be run.	The script may not exist.	CKCFG/REVG Check CMP Proxy configuration, check <code>cmp.script_labels</code> parameter.

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-COMPONENT-TIMEDOUT	8389123	EROR	A Component has Timed Out	A VoiceGenie component has not send a periodic heartbeat message to the CMP Proxy.	The process may be in a deadlock situation.	CKOP/REVG Collect the logs of that process and reports to Genesys. Ensure that system is usable, if not reboot the system.
VGLOG-RESOURCES-EXCEEDED	8389124	EROR	A Component has Exceeded its Allowed Resource Usage	A VoiceGenie component exceeded its limits for CPU or Memory usage (usually 95% of CPU and 1 GB of memory). Process will be restarted.	There may be a memory leak or a spin lock condition within the software.	REVG Collect the logs of that process and reports to Genesys. Ensure that system is usable, if not reboot the system.
VGLOG-PROCESS-DIED	8389125	EROR	A Component has Died	A VoiceGenie component has died unexpectedly. The system will not function properly.	The component may have exited.	CKOP/REVG Look for the existence of core files and collect the logs of that process. Report to Genesys. Ensure that system is usable, if not reboot the system.
VGLOG-DISK-USAGE-ERROR	8389126	EROR	Disk usage exceeded error threshold	The disk usage on a partition has exceeded a high threshold, usually 90%.	The disk is filling up.	CKOP Clear out disk space if possible, monitor disk space carefully.

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-CPU-USAGE-ERROR	8389127	EROR	CPU usage exceeded error threshold	The CPU usage has exceeded a high threshold, usually 75%.	The CPU is being used up by the system and may impact performance and user experience.	CKOP Monitor CPU carefully for CPU increases.
VGLOG-MEM-USAGE-ERROR	8389128	EROR	MEM usage exceeded error threshold	The memory usage has exceeded a high threshold, usually 1.95GB.	The memory on the system is being used up. May be a memory leak.	CKOP Monitor memory carefully for memory leaks.
VGLOG-STATE-FILE-NOT-FOUND	8389377	WARN	Execution State file can not be opened	The file that stores the execution state of the platform can not be opened so the operational state after a reboot may be incorrect.	The file does not exist or can not be opened by the CMP Proxy.	CKCFG Check the permission on the /usr/local/cmp-proxy/config/state.cfg and its contents.
VGLOG-NETID-FILE-NOT-FOUND	8389378	WARN	Network ID file can not be opened, using default ID	The file that stores the network ID of the CMP Proxy can not be opened, as a result the call ID generated by the platform may not contain the correct system wide ID.	The file does not exist or can not be opened by the CMP Proxy.	CKCFG Check the permission on the /usr/local/cmp-proxy/config/networkid.cfg and its contents.

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-CONFIG-NOT-SYNCHRONIZED	8389380	WARN	Configuration Not Synchronized with Database	The CMP Proxy will use the local cached version of the configuration which may be out of date.	The CMP Proxy started before it was able to connect to the CMP Server.	CKNW/REVG Restart the CMP Proxy: su /etc/init.d/cmp-proxy restart
VGLOG-KILLING-COMPONENT	8389382	WARN	Killing a Component	A VoiceGenie component was killed because it could not be stopped gracefully.	System instability.	REVG Reboot the system to ensure that the operating system is running in a stable fashion.
VGLOG-AGENT-DISCONNECTED	8389383	WARN	CMP Agent Disconnected from the proxy (socket closed)	A VoiceGenie component disconnected from the CMP Proxy because it stopped.	The component may have been stopped by a user or may have stopped unexpectedly (i.e. exited)	REVG If it was stopped unexpectedly look for the existence of core files and collect the logs of that process. Report to Genesys. Ensure that system is usable, if not reboot the system.
VGLOG-INVALID-SCRIPT-LABEL-CONFIG	8389385	WARN	Invalid script label configuration	A script label configuration is incorrect and may prevent scripts from being run by the CMP Proxy.	The <code>cmp.script_labels</code> parameter is misconfigured.	CKCFG Check CMP Proxy configuration, check <code>cmp.script_labels</code> parameter for invalid values.

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-SCRIPT-FAILED	8389386	WARN	Script failed	A script failed to run. The operation attempted has failed.	The script may not exist, or parameters passed in may be invalid.	CKCFG Check CMP Proxy configuration, check <code>cmp.script_labels</code> parameter for invalid values.
VGLOG-KILL-PROCESS	8389387	WARN	Process killed	A process started by the CMP Proxy was killed because it could not be stopped gracefully.	System instability.	REVG Reboot the system to ensure that the operating system is running in a stable fashion.
VGLOG-DISK-USAGE-WARNING	8389388	WARN	Disk usage exceeded warning threshold	The disk usage on a partition has exceeded a warning threshold, usually 85%.	The disk is filling up.	CKOP Clear out disk space if possible, monitor disk space carefully.
VGLOG-CPU-USAGE-WARNING	8389389	WARN	CPU usage exceeded warning threshold	The CPU usage has exceeded a warning threshold, usually 70%.	The CPU is being used up by the system and may impact performance and user experience.	CKOP Monitor CPU carefully for CPU increases.
VGLOG-MEM-USAGE-WARNING	8389390	WARN	MEM usage exceeded warning threshold	The memory usage has exceeded a warning threshold, usually 1.9GB.	The memory on the system is being used up. May be a memory leak.	CKOP Monitor memory carefully for memory leaks.

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-ENGINE-DIED-WARNING	8389391	WARN	Engine died warning event	Full CMP functionality will not be available.	The CMP Server (CMP Engine) has died	CKOP Collect the logs of the CMP Server and report to Genesys. Ensure that system is usable, if not reboot the system with the CMP Server.
VGLOG-COMP-DIED-WARNING	8389392	WARN	Component died warning event	A VoiceGenie component has died unexpectedly. The system will not function properly.	The component may have exited.	CKOP Look for the existence of core files and collect the logs of that process. Report to Genesys. Ensure that system is usable, if not reboot the system.
VGLOG-COMPONENT-STOPPED	8389637	NOTE	A Component has been Stopped	None, the stop may be caused by the user stopping the system. If it is a more serious stopping event a higher level trap will be raised.	vgstop CLC> stop process kill	-
VGLOG-COMPONENT-STARTED	8389638	NOTE	A Component has been Started	None, someone is attempting to start the VoiceGenie software.	vgstart CLC> start	-

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-DISK-USAGE-CLEAR	8389641	NOTE	Message to clear the disk usage threshold events	System is functioning normally.	The disk usage is no longer above a given threshold.	-
VGLOG-CPU-USAGE-CLEAR	8389642	NOTE	Message to clear the cpu usage threshold events	System is functioning normally.	The CPU usage is no longer above a given threshold.	-
VGLOG-MEM-USAGE-CLEAR	8389643	NOTE	Message to clear the mem usage threshold events	System is functioning normally.	The memory usage is no longer above a given threshold.	-
VGLOG-ETH-STATUS-CLEAR	8389644	NOTE	Message to clear the ethernet status event	System is functioning normally.	The Ethernet adaptor is now working correctly.	-
VGLOG-ENGINE-STATUS-CLEAR	8389645	NOTE	Message to clear the engine died warning event	System is functioning normally.	The CMP Server is operational once again.	-
VGLOG-COMP-STATUS-CLEAR	8389646	NOTE	Message to clear the component died warning event	System is functioning normally.	The component that had died is functional once again.	-
VGLOG-CANT-OPEN-CONFIGFILE	134219730	ERROR	Can't open Configuration file.			CKCFG/REVG

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-SOCKET-SEND-FAILED-CMPP	134219731 (Old Alarm Number: 10000)	EROR	Socket send failed.	A message failed to get sent over the network.	Ethernet problem.	CKNW/REVG
VGLOG-VGASSSERT-CMPP	135267305	CRIT	VGASSSERT			REVG
VGLOG-CANT-OPEN-DLL-CMPP	138413034	CRIT	Can't open DLL library	A custom logging sink failed to open properly.	The custom logging sink's library path may not be configured properly or the library may be invalid.	REVG
VGLOG-CANT-RESOLVE-DLL-SYM-CMPP	138413035	CRIT	Can't resolve symbol in DLL	A custom logging sink failed to open properly.	The custom logging sink library may be invalid or an incorrect version.	REVG

C.2 External Agent Traps

The prefix for all External Agent Traps can as outlined in the following table:

Component	SNMP Prefix	XXXX Value
ASR Client	.1.3.6.1.4.1.7469.251.1.400	EASC
TTS Client	.1.3.6.1.4.1.7469.251.1.401	ETTC
ASR Server	.1.3.6.1.4.1.7469.251.1.403	EASR
TTS Server	.1.3.6.1.4.1.7469.251.1.404	ETTS

Note that the XXXX value is used to make the names of the following traps unique.

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-INVALID-MSG-TYPE-XXXX	1048597	WARN	Invalid Message Type Sent or Received	An OA&M message is being ignored	VoiceGenie software version mismatch	REVG
VGLOG-CANNOT-CREATE-CLIENT-XXXX	1048599	WARN	Error Creating Client Socket	The CMP Proxy will not establish a connection with the CMP Server	Ethernet problem or an error in one of the CMP Proxy's connection parameters.	REVG
VGLOG-DSRV-INVALID-QUERY-XXXX	6291477	WARN	Invalid data query string received	An SNMP get or set request may fail.	A request for an invalid parameter, perhaps from the SNMP Agent.	REVG
VGLOG-DSRV-UNSUPPORTED-VAR-XXXX	6291478	WARN	Query for unsupported variable received	An SNMP get or set request may fail.	A request for an invalid parameter, perhaps from the SNMP Agent.	REVG
VGLOG-VGASSSERT-XXXX	135267305	CRIT	VGASSSERT			REVG
VGLOG-SOCKET-SEND-FAILED-XXXX	134219731	EROR	Socket send failed	A message failed to get sent over the network.	Ethernet problem.	CKNW/REVG
VGLOG-EXAGENT-STOP-SCRIPT-XXXX	120586753	EROR	Stop Script Failed to Run	Can not stop process.	Invalid configuration, stop script may not exist.	REVG

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-EXAGENT-START-SCRIPT-XXXX	120586754	EROR	Start Script Failed to Run	Can not start process.	Invalid configuration, start script may not exist.	REVG
VGLOG-EXAGENT-START-FAIL-XXXX	120586755	EROR	Start Script Failed to Start Monitored Process	Can not start process.	Invalid configuration, start script could not start process.	REVG
VGLOG-EXAGENT-INVALID-SCRIPT-XXXX	120587009	WARN	Invalid Script File Specified	Can not start or stop process.	Script file does not exist.	CKCFG/REVG
VGLOG-EXAGENT-CFG-PROCESS-NAME-XXXX	120587010	WARN	Invalid Value Specified for xagent.process name	Can not monitor process.	Configured name is invalid.	CKCFG/REVG
VGLOG-EXAGENT-MONITOR-PROCESS-XXXX	120587011	WARN	Failed to Notify CMP about new Process being Monitored	CMP may not be up to date on monitoring information.	CMP network down.	CKCFG/REVG
VGLOG-EXAGENT-THREADS-MAX-XXXX	120587012	WARN	A Monitored Process has too many Threads	Process may not be running correctly.	Dependent on process.	REVG

Name	OID Suffix	Level	Description	Impacts	Causes	Response Code/Detailed Recommended Actions
VGLOG-EXAGENT-THREADS-MIN-XXXX	120587013	WARN	A Monitored Process has too few Threads	Process may not be running correctly.	Dependent on process.	REVG
VGLOG-EXAGENT-PROCESS-DIED-XXXX	120587014	WARN	A Monitored Process has died	Functionality provided by process is not available.	Dependent on process.	REVG
VGLOG-EXAGENT-STOP-RETURN-XXXX	120587015	WARN	Problem with Stop Script Return	Process can not be stopped.	Dependent on process.	REVG
VGLOG-EXAGENT-START-RETURN-XXXX	120587016	WARN	Problem with Start Script Return	Process can not be started.	Dependent on process.	REVG



Appendix

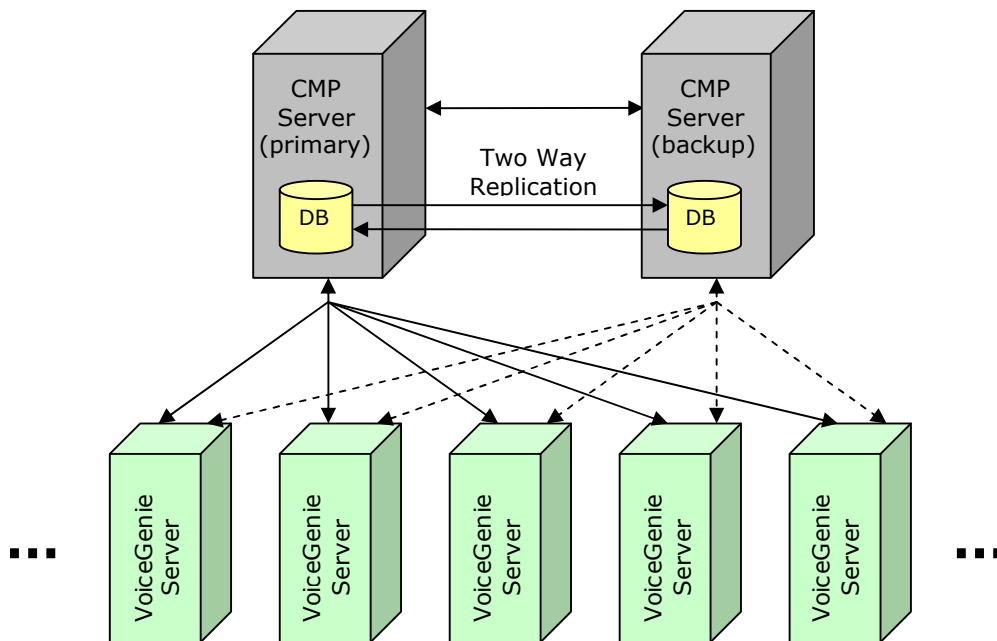
D

Database Replication Setup

This section outlines a method to provide enhanced OA&M Framework redundancy using a backup database along with MySQL's replication features. MySQL replication provides a mechanism to maintain a redundant database by supporting real-time synchronization using a master-slave hierarchy.

D.1 Replication Architecture

The VoiceGenie OA&M Framework architecture revolves around a single CMP Server (Management Server) that provides the clustering capabilities for a deployment of VoiceGenie Servers. In a redundant architecture a Backup CMP Server (Management Server), along with a backup database is deployed using Two-Way Replication to keep the information in the database synchronized. The diagram below depicts this architecture.



Note: MySQL replication is only available in version 3.23.47 or greater of MySQL. Customers running VoiceGenie versions prior to 7.0.0 will need to upgrade their MySQL installation in order to enable replication.

Also, note that the OA&M Framework components on the Backup Management Server should be using the local database for all information; this means that the CMP Server configuration and the SMC configuration should be using the local database. To check this, ensure the following:

1. In CMP Server configuration ensure (via the SMC) `cmp.db` is set to `127.0.0.1`
2. On Linux ensure that `/usr/local/cmp-server/config/odbc.ini` and `/etc/odbc.ini` have the database server set to `127.0.0.1`
3. On Windows ensure that the Registry Key `HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\MySQL\Server` is set to `127.0.0.1`
4. In Management Console configuration ensure (via the SMC) `cmp.db` is set to `127.0.0.1`

D.2 Configuring the Databases

This section explains how to setup a Two-Way Replication for the databases on the Primary and Backup Management Servers. In this scenario, replication occurs in both directions, and as a result, each database is a master and a slave to each other. Replication is setup in this fashion so that any updates that are

made to the database when fail-over occurs are replicated back to the Primary Management Server when the system is restored to normal operation. As noted in the section above, please ensure that the SMC and CMP Server on each system is using the local database.

Assuming that your deployment has one server allocated as the Primary Management Server, and another allocated as the Backup Management Server the following directions can be followed to setup Two-Way Replication:

1. Stop all OA&M Framework components on the Primary and Backup Management Server machines (including the `cmproxy` and `cmpengine` processes)
2. Log in to the Primary Management Server as the root user
3. In the file `/etc/my.cnf` (Linux) or `C:\Program Files\MySQL\MySQL Server 4.1\my.ini` (Windows), update the `[mysqld]` section with the `log-bin`, `server-id=100`, `slave-skip-errors=1062,1053,0` settings, and optionally `set-variable = slave_net_timeout = 120`. The following is a Linux example:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log-bin
server-id=100
slave-skip-errors=1062,1053,0
slave_net_timeout=120
```

4. Restart MySQL, i.e. `/etc/init.d/mysql restart` (Linux) or restart the `mysql` service using Services in Administrative Tools (Windows)
5. Log in to MySQL as the root user (i.e. at the command prompt type: `mysql -u root -prootsql mysql`)
6. In MySQL, execute the following: `GRANT ALL ON *.* to pw@'%' IDENTIFIED BY 'pw';`
7. In MySQL, execute the following: `FLUSH TABLES WITH READ LOCK;`
8. In MySQL, execute the following: `SHOW MASTER STATUS;` and note down the information
9. In MySQL, execute the following: `UNLOCK TABLES;`
10. On Linux, go to the directory `/var/lib/mysql` and Tar the NDM and CallHistory directories:


```
tar -cvf NDMsnapshot NDM
tar -cvf CallHistorysnapshot CallHistory
```

On Windows, backup or copy the C:\Program Files\MySQL\MySQL Server 4.1\data\NDM and \CallHistory directories to a shared location.

11. Login to the Backup Management Server as the root user
12. In the file /etc/my.cnf (Linux) or C:\Program Files\MySQL\MySQL Server 4.1\my.ini (Windows), update the [mysqld] section with the log-bin, server-id=101, slave-skip-errors=1062 settings, and optionally set-variable = slave_net_timeout = 120. The following is a Linux example:


```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log-bin
server-id=101
slave-skip-errors=1062,1053,0
slave_net_timeout=120
```
13. Restart MySQL, i.e. /etc/init.d/mysql restart (Linux) or restart the mysql service using Services in Administrative Tools (Windows)
14. Log in to MySQL as the root user (i.e. at the command prompt type: mysql -u root -prootsql mysql)
15. In MySQL, execute the following: GRANT ALL ON *.* to pw@"%" IDENTIFIED BY 'pw';
16. In MySQL, execute the following: RESET SLAVE;
17. Stop MySQL, i.e. /etc/init.d/mysql stop (Linux) or stop the mysql service using Services in Administrative Tools (Windows)
18. On Linux, go to the directory /var/lib/mysql and remove the existing NDM and CallHistory directories:


```
rm -rf NDM
rm -rf CallHistory
```

On Windows, go to the directory C:\Program Files\MySQL\MySQL Server 4.1\data and delete the existing NDM and CallHistory directories.
19. On Linux, copy over the Tar files from the Primary Management Server (NDMsnapshot and CallHistorysnapshot) and extract the contents of the files:


```
tar -xvf NDMsnapshot
tar -xvf CallHistorysnapshot
```

On Windows, copy over the NDM and CallHistory database directories, which were backed up in step 10, to replace the directories deleted in the previous step.

20. Start MySQL, i.e. `/etc/init.d/mysql start` (Linux) or start the mysql service using Services in Administrative Tools (Windows)
21. In MySQL, execute the following:

```
CHANGE MASTER TO MASTER_HOST='<master_ip>',
MASTER_USER='pw',MASTER_PASSWORD='pw',
MASTER_LOG_FILE='<bin-log-file>', MASTER_LOG_POS=<bin-
log-pos>;
```

where `<bin-log-file>` and `<bin-log-pos>` are the values noted down from step 8 and `<master_ip>` is the Primary Management Server's IP address.
22. In MySQL, execute the following: `SHOW MASTER STATUS;` and note down the information
23. In MySQL, execute the following: `SLAVE START;`
24. Log back in to the Primary Management Server as the root user
25. In MySQL, execute the following:

```
CHANGE MASTER TO MASTER_HOST='<master_ip>',
MASTER_USER='pw',MASTER_PASSWORD='pw',
MASTER_LOG_FILE='<bin-log-file>', MASTER_LOG_POS=<bin-
log-pos>;
```

where `<bin-log-file>` and `<bin-log-pos>` are the values noted down from step 22 and `<master_ip>` is the Backup Management Server's IP address.
26. In MySQL, execute the following: `SLAVE START;`
27. Use `show processlist`, `show master status`, and `show slave status` to see if the replication is working properly. The MySQL error log file in `/var/log` may also be useful for troubleshooting
28. Start the Primary and Backup Management Server software and any other related OA&M Framework components running on the two machines

Note: Users should consider setting the `master-connect-retry` value in `my.cnf/my.ini` to less than 60 seconds (default value) to reduce the time it takes for the databases to re-synchronize after network or other system problems. A reasonable value is 20 seconds. It should be set on both the primary and backup server.

Users should set the `slave_net_timeout` value in `my.cnf/my.ini` to a lower value (the default is one hour) since this setting determines how often the backup tries to reconnect to the primary when the connection between the two is idle. This helps to deal with situations where the connections between the primary and backup database are not properly closed. This parameter should be set to the same value in both databases. A reasonable value is 120 seconds.

D.3 Resetting DB replication

If the DB replication needs to be reset, after the Database Replication has been reconfigured, then please use the following procedure:

1. Logon to primary server
2. Run `mysql -uroot -prootsql`, then run `slave stop`; then run `reset slave`;
3. Run `mysql -upw -ppw NDM`, then run


```
truncate CallLog;
truncate HistHealthStatus;
truncate HistInitStatus;
truncate HistProcessStatus;
use CallHistory;
truncate HistAppDistribution;
truncate HistCallLength;
truncate HistCallVolume;
```
4. Run `mysql -uroot -prootsql`, then run `show master status` and write down the position number and the file name, i.e.,

File	Position	Binlog_Do_DB	Binlog_Ignore_DB
dps12-bin.000001	114359		
5. `cd /var/lib/mysql`, and run “`tar -cvf NDMsnapshot NDM`” and “`tar -cvf CallHistorysnapshot CallHistory`”
6. ftp over the two snapshot files to the back server to `/var/lib/mysql`
7. Logon to the backup server

8. Issue `mysql -uroot -prootsql`, and run “`slave stop`”, “`reset slave`”
9. Stop `mysql` by running `/etc/init.d/mysqld stop`
10. Run the followings:


```
cd /var/lib/mysql
rm -rf NDM
rm -rf CallHistory
cd /var/lib/mysql
tar -xvf NDMsnapshot
tar -xvf CallHistorysnapshot
```
11. Start `mysql` by running `/etc/init.d/mysqld start`
12. Run `mysql -uroot -prootsql`, and issue


```
“CHANGE MASTER TO MASTER_HOST='hostname_primary_server',
MASTER_USER='pw', MASTER_PASSWORD='pw',
MASTER_LOG_FILE='bin_file_name_primary',MASTER_LOG_POS=position_number_primary;”
```

e.g. “`CHANGE MASTER TO MASTER_HOST='10.0.0.103', MASTER_USER='pw', MASTER_PASSWORD='pw', MASTER_LOG_FILE='narita-bin.001',MASTER_LOG_POS=73;`”
13. Issue “`SHOW Master Status;`” and write down the backup db server’s position and file name
14. Issue “`Slave start;`”
15. Verify the replication is working by:
 - a. `mysql -uroot -prootsql`:
 - b. Show `processlist`, to see if there’s any error
 - c. Compare “`Show slave status`” on backup server and “`show master status`” on the primary server; see if the position numbers are growing in the same pace.
16. Run the CLC query, `CLC> health cmpserver`, to verify if the DB queries work.
17. Logon to the primary sever
18. Repeat the “`CLC> health cmpserver`” to see if the DB Query successful number is growing
19. Run “`mysql -uroot -prootsql`”, and issue


```
“CHANGE MASTER TO MASTER_HOST='hostname_backup_server',
MASTER_USER='pw', MASTER_PASSWORD='pw',
MASTER_LOG_FILE='bin_file_name_backup',MASTER_LOG_POS=position_number_backup;”
```

e.g. “`change master to MASTER_HOST='10.0.0.104', MASTER_USER='pw', MASTER_PASSWORD='pw', MASTER_LOG_FILE='moon-bin.001',MASTER_LOG_POS=33;`”

20. Issue “Slave start;”
21. Verify the replication is working by running:
 - a. `mysql -uroot -prootsql:`
 - b. Show `processlist`, to see if there’s any error
 - c. Compare “Show slave status” on backup server and “show master status” on the primary server, see if the position numbers are growing in the same pace.
22. Run the CLC query, `CLC> health cmpserver`, to verify if the CMP Primary Server is taking control and if the DB queries work.

D.4 Fail-Over Handling

The fail-over provided by two-way replication of the databases provides higher availability and lessens the chances of lost data; however, there is the possibility that some data is lost. This section outlines the expected behavior when replication is used.

1. If the primary database (i.e. MySQL) goes down, then the Primary Management Server (CMP Server) software will not automatically use the backup database. In this case data will be lost and the Primary Management Server will not function normally.
2. When fail-over has occurred and the Backup Management Server has taken over, at this point, if the backup database (i.e. MySQL) goes down, then the Backup Management Server (CMP Server) software will not have access to the database. In this case data will be lost and the Backup Management Server will not function normally.
3. When fail-over has occurred, changes to the database should be minimized, the following rules must be followed:
 - a. No new machines should be added to the network and no software can be deployed when the Backup Management Server is in control.
 - b. Provisioning and configuration changes can be made, but should be kept at a minimum.
 - c. When recovery occurs and the Primary Management Server comes back online, users should wait until the primary database has re-synchronized before making changes to the database (i.e. deploying software, etc.). Users can check this via the `SHOW MASTER STATUS` and `SHOW SLAVE STATUS` commands in MySQL.
4. During fail-over and recovery, logs related to call starts and ends will not be lost, however, there is a possibility that a handful of Call Data Records may be lost. It is expected that during fail-over/recovery some calls may be logged as incomplete, the data from these can be restored from the general log data.

5. The SMC on the Backup Management Server uses the database on the Primary Management Server as its data source. As a result, the SMC on the Backup Management Server will not work correctly during fail-over.
6. Replication requires extra disk space since it must keep track of every SQL statement that modifies the database content; as a result, these logs need to be cleared periodically. This can be done using the following MySQL command: `PURGE MASTER LOGS TO 'logname'`, where `logname` is the name of a Binary Log file that can be deleted (i.e. `mysql-bin.010`). For details on purging the Binary Logs please refer to MySQL's online documentation, section 13.6.1. Note that only Binary Logs that are not in use can be deleted.
7. On the Management Servers a number of database clean up scripts are run periodically by cron. These scripts should only be run on the Master Database Server and the Slave Database Server on alternated date to avoid conflicts

The line on the Primary server can be:

```
29 3 1-31/2 * * pw /usr/java/jdk/bin/java -jar /usr/local/cmp-  
db/bin/dbadmin.jar cleanup
```

The line on the Backup server can be:

```
29 3 2-30/2 * * pw /usr/java/jdk/bin/java -jar /usr/local/cmp-  
db/bin/dbadmin.jar cleanup
```

The following table lists all the tables in the NDM and CallHistory databases and outlines potential problems that may occur during a Management Server fail-over and recovery.

Table Name	Potential Failover Impact
ActivityLog	Logs may be lost during failover. Tables may become inconsistent during recovery.
ApplicationData	–
BConfig	–
CMUser	–
CacheManifest	–
CallData	Records may be incomplete due to failover and recovery. Tables may become inconsistent during recovery.
CallLog	Tables may become inconsistent during recovery.

Table Name	Potential Failover Impact
ComponentType	–
Config	–
ConfigProfile	–
ConfigProfileRelation	–
Configuration	–
DeploymentStatus	–
Groups	–
HistHealthStatus	–
HistInitStatus	–
HistProcessStatus	–
MetaData	–
Network	–
Parameter	–
ParameterHistory	–
ProductComponentRelation	–
ProductFile	–
Provision	–
ProvisionRelation	–
ProvisionType	–
Relation	–
HistAppDistribution	–
HistCallLength	–
HistCallVolume	–

Visit www.mysql.com for further information on replication support for version 4.1.20.



Appendix

E

Developing a Custom Logging Sink

This section explains how to develop a custom user defined C++ sink to integrate with the OA&M Framework's logging service. Refer to the section for a general explanation of the OA&M Framework's logging capabilities.

The OA&M Framework allows users to develop their own custom logging sink to filter and process VoiceGenie logs in a customized manner. A custom sink can receive all types of log, including metrics logs, and can be integrated at either the cluster level, by plugging into the CMP Server, or at the machine level, by plugging into the CMP Proxy. On Windows, a custom sink must be a DLL (Dynamically Linked Library); on Linux, it must be a shared object.

E.1 VoiceGenie Log Sink Interface

The custom sink must support the proper log sink interface in order for the OA&M Framework to load it correctly. The main interface consists of only one function called `GetSink()`. Below is a code snippet showing one way to define the function to make it accessible to the OA&M Framework under both Linux and Windows (using Microsoft Visual C++). This should be included as part of your C++ source file:

```
extern "C"  
{  
#ifdef WIN32  
    __declspec(dllexport) VGLogSinkInterface* GetSink();  
#else // Linux  
    VGLogSinkInterface* GetSink();  
#endif  
}
```

Notice that the `GetSink` function is supposed to return a pointer to a `VGLogSinkInterface` object. VoiceGenie defines the interface for this

abstract class in a C++ header file (VLogSinkInterface.h). The contents of this file are shown below:

```
/*VLogSinkInterface.h: Definition of class
VLogSinkInterface */

#ifndef VGLOGSINKINTERFACE_H
#define VGLOGSINKINTERFACE_H

class VLogSinkInterface
{
public:
    VLogSinkInterface(){};
    virtual ~VLogSinkInterface() {};
    virtual const char * GetVersion() const = 0;
    virtual bool Initialize(int uNetworkID) = 0;
    virtual bool LogToSink(unsigned int nLogType, unsigned
int nLogID,
        const char* strCallID, const char* strTimestamp,
        const char* strData, unsigned int
nOriginalSenderID) = 0;

protected:
    bool m_bInitialized;
};

#endif // !defined(VGLOGSINKINTERFACE_H)
```

When coding the custom sink, a new class needs to be derived from VLogSinkInterface and the GetSink function needs to return an instance of this new class. Implementation specifics of the VLogSinkInterface functions are up to the developer. The following is an example of a simple implementation:

```
CustomSink.h
#include <stdio.h>
#include "VLogSinkInterface.h"

#ifndef CUSTOMSINK_H
#define CUSTOMSINK_H

class CustomSink: public VLogSinkInterface
{
public:
    CustomSink(){};
    ~CustomSink() {};
    const char * GetVersion() const;
    bool Initialize(int uNetworkID);
    bool LogToSink(unsigned int nLogType, unsigned int
nLogID,
        const char* strCallID, const char* strTimestamp,
```

```

        const char* strData, unsigned int
nOriginalSenderID);
protected:
    bool m_bInitialized;
};

#endif // !defined(CUSTOMSINK_H)
CustomSink.cpp:
#include "CustomSink.h"

const char * CustomSink::GetVersion() const
{
    return "Custome Sink";
}

// global sink object
static CustomSink TheSink;

extern "C"
{
#ifdef WIN32
    __declspec(dllexport) VGLogSinkInterface* GetSink();
#else // Linux
    VGLogSinkInterface* GetSink();
#endif
}

VGLogSinkInterface* GetSink()
{
    return &TheSink;
}

bool CustomSink::Initialize(int uNetworkID)
{
    // set value to true or will result in this error -
08100000 Invalid DLL Sink Access

```

```

        m_bInitialized=true;
        return m_bInitialized;
};

bool CustomSink::LogToSink(unsigned int nLogType,
unsigned int nLogID, const char* strCallID, const char*
strTimestamp, const char* strData, unsigned int
nOriginalSenderID)
{
    if (!m_bInitialized)
        return false;

    FILE * LogFile = fopen("/tmp/test_file.txt", "a");
    fprintf(LogFile, "Log Type:%d\nLog ID:%d\nCall
ID:%s\nTime Stamp:%s\nData:%s\nOrig Sender ID:%d\n",
nLogType, nLogID, strCallID, strTimestamp, strData,
nOriginalSenderID);
    fclose(LogFile);
    return true;
};

```

The sections below describes how the OA&M Framework uses these functions.

E.1.1 Destructor/Constructor

The constructor of the `VGLogSinkInterface` is never used directly. The developer needs to make sure it gets called at some point before `GetSink()` returns.

There is no exposed mechanism for destroying the `VGLogSinkInterface` object, so if the object is allocated on the heap it stays in memory until the process that loaded the library dies. The one way around this is to define the object globally so that it gets constructed while the library loads. This way the destructor gets called implicitly when the OA&M Logging Service shuts down. It does not really matter too much, since in either case, the object is loaded for most of the life of the OA&M process. Regardless, it should be safe to assume that the destructor is not called while other `VGLogSinkInterface` functions are running in separate threads.

E.1.2 GetVersion

This function is not really used at this time.

E.1.3 Initialize

This function is called once during the initialization of the OA&M process. After this function is called the `VGLogSinkInterface` object should be ready to accept logs. No logs are sent to the sink before this call. Note that the `uNetworkID` that is passed in is the network ID of the process that is loading the custom sink.

E.1.4 LogToSink

The `LogToSink` function is called every time a log is sent to the custom sink via the OA&M Framework. The following table provides details about each parameter in the `LogToSink` function.

Parameter	Description	Possible Values/Examples
unsigned int nLogType	The log type of the log.	LOG_0 = 300, LOG_1 = 301, LOG_2 = 302, LOG_3 = 303, LOG_4 = 304, LOG_5 = 305, METRICS = 100
unsigned int nLogID	The log ID of the log. This includes the module identifier (the 12 most significant bits) and the log specifier (the 20 least significant bits). For metrics logs the module ID is always 0.	See for details on module and specifier bits for logs and metrics.
const char* strCallID	An ID that is unique for all calls in a single deployment. Consists of a 64 bit number. In this case it is represented as a null terminated, 16 digit hexadecimal string with a - character in between the first and last 8 digits.	Example string: 00020023-100003E9

Parameter	Description	Possible Values/Examples
const char* strTimestamp	A timestamp indicating the time the log was created. The string is a 24 hour with milliseconds format: YYYY-MM-DD/HH:MM:SS.mmm	Example string: 2005-03-10/13:42:33.335
const char* strData	This is a string containing the entire log message.	Metrics example string: inCall_initiated 13:1 Regular log example string: Starting CMP Proxy
unsigned int nOriginalSenderID	This is the network ID of the component that originally created the log.	A positive integer.

Each time a log is sent to the sink, the above values are passed through the `LogToSink` function. At this point, the custom sink can filter and process the logs as desired.

E.2 Threading Issues

In VoiceGenie 7.0.0+, the main OA&M Framework logging is done in a separate thread within each component's process. All logging sinks, including custom user defined sinks, carry out their processing within this logging thread. This can help minimize the impact of logging on the performance of each VoiceGenie component, especially when logging sinks block while writing to disk. Nevertheless under heavy logging, the length of time a sink takes to process a log can directly impact the lag associated with other logging sinks, and any heavy CPU usage obviously impacts the performance of the entire machine.

E.3 Building and Linking the Library

E.3.1 Linux

Building and linking the custom shared object on Linux can be done using g++ (GCC) 3.2.2 on a machine running the VoiceGenie Linux 3.0 operating system. The example below shows what commands and options to use:

```
g++ -c -g -Wall -D_REENTRANT -D_NO_LARGEFILE64_SOURCE -fPIC CustomSink.cpp -o CustomSink.o
```

```
g++ -g -Wall -D_REENTRANT -D_NO_LARGEFILE64_SOURCE -fPIC -shared -Wl,-soname,libMySink.so -o libMySink.so CustomSink.o -lpthread
```

Make sure to use the `-fPIC` compiler option to make the code ‘position independent’ and the `-shared` linker flag to tell the linker that this is a shared object. After the shared object file is created it can be used by the OA&M Framework; see 9.3.8 User Defined Sinks for details on how to do this.

E.3.2 Windows

Building and linking on Windows may depend on the compiler available. For Microsoft Visual C++, create a Win32 DLL project. Make sure that in the property pages of the project, the Configuration Properties > General > Configuration Type is set to Dynamic Library (.dll), and the Configuration Properties > C/C++ > Code Generation > Runtime Library is set to Multi-threaded DLL (or Multi-threaded Debug DLL). Please consult your compiler’s documentation for assistance with other compilers.



Appendix

F

Windows Process Names

The following is a list of VoiceGenie and VoiceGenie-related executables that may run on your Windows system once VoiceGenie is installed.

VoiceGenie Software:

Executable	Product	Application Component
pwcallmgr.exe	VoiceXML Media Platform	Call Manager
pwproxy.exe	VoiceXML Media Platform	Web Proxy
pwvxmli.exe	VoiceXML Media Platform	VoiceXML Interpreter
srmsserver.exe	Speech Resource Manager (SRM) Server	Speech Resource Manager (SRM) Server
srmpoxy.exe	Speech Resource Manager (SRM) Proxy	Speech Resource Manager (SRM) Proxy
cmpproxy.exe	OA&M Framework	CMP* Proxy
cmpengine.exe	OA&M Framework	CMP* Server
cmpclc.exe	OA&M Framework	CMP* Command Line Console
cmpexagent.exe	OA&M Framework	CMP* External Agent

* CMP = Cluster Management Platform (another name for the OA&M Framework)

Third Party Software:

- Apache Tomcat: tomcat.exe
- Java 2 Runtime Environment: java.exe
- MySQL: mysqld.exe
- Squid Web Proxy Cache: squid.exe

Also, ASR/TTS Software processes may run on the system, these process names will be configuration specific.



Appendix

G

Security Considerations

This section provides suggestions on improving the security of your VoiceGenie systems. These suggestions are only general guidelines and will not guarantee security for any specific system. Use this section as an aid to improve the security of your particular environment.

G.1 How to use SSH as a Replacement for Telnet and FTP

SSH or Secure SHell is a secure replacement for telnet. When using telnet, the characters you type on your keyboard, to send to the remote server, are sent in clear text. This means that anyone with access to the network can use a packet sniffer to find out exactly what you are typing, including your password. An attacker could then use your password to gain unauthorized access to the remote server. Afterward, your administrator would examine the server logs to determine the nature of the intrusion, and these logs would reveal that your account was the culprit. SSH encrypts the text being transferred between you and the remote server.

SCP (Secure CoPy) lets you use SSH to copy files to and from your remote server without having to actually login. SFTP (SSH FTP) looks and acts just like FTP except it uses SSH to communicate. SSH also compresses data before sending it to the remote host providing faster file transfers and a decrease in bandwidth usage.

The examples below apply mostly to Linux operating systems since telnet use is extremely common with Linux. However, SSH clients are also available for Windows 2000 for those who want an alternative to telnet on that operating system as well.

SSH authenticates using two possible methods: normal password and key pair. The normal password method simply prompts the user for their password.

Linux Example:

```
ssh samiam@greeneggs  
samiam@greeneggs's password:
```

Now enter your password and you have access.

The key pair method of authentication involves generating a public and private key. Before you can begin you must first generate a key pair:

```
ssh-keygen -t rsa
```

You will be asked to enter a *passphrase*. A passphrase is a password that you must enter before you use your key. This prevents anyone from stealing your key. Without the passphrase the key is useless. SSH then creates a directory in your home directory called `.ssh`.

Inside this directory you will see two files:

1. `id_rsa` – This is your private key. Never give this out.
2. `id_rsa.pub` – This is your public key. This is the key you place on remote servers.

Next, you give a copy of your public key (`id_rsa.pub`) to the administrator of the server you wish to access. He will then create a directory called `.ssh` in your home directory on that server. Your public key is placed in the directory as a file called `authorized_keys2`. When this is done you should be able to login to the remote server using:

```
ssh <your username>@<remote hostname or IP address>
```

SSH will ask you for your passphrase. After you've entered it you will be given a shell prompt on the remote server. For example say the remote server was called `greeneggs` and your username was `samiam`. To SSH to that server you would type: `ssh samiam@greeneggs`

If your user name was the same on both your workstation and the remote server then you could omit the username: `ssh greeneggs`

SCP is similar. To copy a file on Linux:

```
scp <full path to files> <your username>@<remote hostname  
or IP address>:<full path to file location>
```

You could also reverse it to copy from the remote server to your work station. Suppose you wanted to copy `/tmp/file.tar` on the remote server to `/tmp/file.tar` locally:

```
scp samiam@greeneggs:/tmp/file.tar /tmp/file.tar
```

SFTP works much like SSH. To SFTP from a Linux client to a remote server:

```
sftp samiam@greeneggs
```

You will then have a prompt which can use in a manner similar to FTP.

G.2 Hardening the SSH Server

If the SSH server is located on a busy or public network you should add restrictions to the server to lessen the chance of unwanted intruders. The file `/etc/ssh/sshd_config` controls how the server (`sshd`) acts. The following options are useful:

- `PermitRootLogin no`

This prevents someone from logging on as `root` via SSH.

- `AllowUsers`

Using this option, you can specifically allow certain users the right to login via SSH while disallowing all others. For example, to allow only the user `samiam` to login, the line would read:

```
AllowUsers samiam samiam@hostname
```

Hostname is the hostname of the SSH server. Users are separated by white space. You can add as many users as you like.

- `Banner /etc/ssh/banner`

This option has SSH display a banner during the login process. The banner file is a plain text file. Example:

```
#####
#      AUTHORIZED ACCESS ONLY      #
# Your activities are being LOGGED and REPORTED #
#####
```

If you make these changes be sure to restart the SSH server with the command (as root) `service sshd restart`.

Note: If you disable a user in the future by locking their account (`passwd -l`) or by changing their password they will still have SSH access as long as they use the key pair authentication method. Be sure to delete their `.ssh` directory or make it unreadable by them.

G.3 Disabling Unused Services

It is good practice to turn off any unneeded services running on the system. This mitigates the security risk to your system. When using the SSH server, both telnet and FTP services can be disabled.

G.4 General Password Construction Guidelines

A strong password usually has the following characteristics:

- Contains both upper and lower case characters
- Has digits, punctuation characters and letters
- Contains at least eight alphanumeric characters
- Is not a word in any language, slang, dialect or jargon
- Is not a commonly used word such as: Names of family, pets, friends, co-workers, fantasy characters, computer terms and names, commands, sites, companies, hardware, software etc.
- Is not a word similar to `sanjose`, `sanfran` or any similar derivation
- Is not a birth-date or other personal information such as an address or phone number.
- Is not a pattern like `aaabbb`, `qwerty`, `zyxwvuts`, `123321`, etc., or any of these spelled backwards, or any of them preceded or followed by a digit (e.g., `secret1`, `1secret`).

Never write your password down or store it on-line. Use acronyms and word plays to help create passwords that can be easily remembered. For instance, “My Favorite Ball-Player is Number 34!” could help you remember the password `mFB-Pi#34!`.

Note: Do not use the previous example as a password!

Please try to follow these guidelines when creating any new password for the system.

G.5 No World Writable Files

On Linux, it is important to ensure that added files are not world writable. World writable files are files that can be overwritten by anyone. As such, you will not be able to trust the contents of these files. The world writable attribute can be removed by issuing the command `chmod o-w <filename>`.

G.6 Disabling CTRL+ALT+DELETE Shutdown (Linux)

By default, the `init` tab allows any console user to shut down a Linux platform using `CTRL+ALT+DEL`. While the default behavior is probably better in most

cases, it allows anyone with access to the system console to cause a system shutdown. To disable this feature:

1. Login as `root`
2. Edit `inittab` (`vi /etc/inittab`)
3. Look for this line:

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```
4. Comment it out by changing the line to this:

```
#ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```
5. Save the file and exit to shell
6. Reboot or execute `init q`

G.7 System Passwords

The VoiceGenie Linux 3.0 operating system installation creates 2 user accounts by default. The account login and password are listed in the table below:

Account	Default Password	Description
root	root1	Account for system administration
pw	pw	Account used to run most VoiceGenie software.

Note that the above passwords should be changed once the system is installed to limit unauthorized access. To change the password, login to the system as that user (i.e. `pw` or `root`) and enter the command `passwd` at the command prompt to change the password.

G.8 Database Passwords

The OA&M Framework uses a MySQL database to centrally store information. By default this database is setup with 2 accounts: the `root` account, which is used for administration purposes, and the `pw` account, which is used by the software to update and query the database. The default settings for these are listed below:

User	Password
root	rootsql
pw	pw

The password of the root account should be changed for security purposes. This can be done by logging into the database using the root user and executing the following MySQL command:

```
mysql> SET password FOR 'root' = password('newpass');  
mysql> flush privileges;
```

Where newpass is the new password that should be used.

The password of the pw account requires more steps to change. First, before changing the pw password, the CMP Server and the SMC's configuration should be updated with the new password. Change the parameter `cmp.password` in both configurations to the new desired password.

Next, change the pw password in MySQL; log in to the database as the root user and execute the following commands:

```
mysql> SET password FOR 'pw' = password('newpass');  
mysql> flush privileges;
```

Where newpass is the new password that should be used.

The next step is to restart the CMP Server and the SMC (Tomcat server) so that they use the new password to connect to the database. To restart the CMP Server and the SMC run the following commands as the super user on the command line:

```
/etc/init.d/cmp-server restart  
/etc/init.d/vgtomcat restart
```

In addition, if you have database replication setup you will need to set up replication once again with the new database password. During replication setup there are two commands where the database password is required, they are as follows:

```
GRANT ALL ON *.* to pw@"%" IDENTIFIED BY 'pw';
```

```
CHANGE MASTER TO MASTER_HOST='<master_ip>',  
MASTER_USER='pw',MASTER_PASSWORD='pw',  
MASTER_LOG_FILE='<bin-log-file>', MASTER_LOG_POS=<bin-  
log-pos>;
```

For further details on MySQL please consult the MySQL documentation (<http://dev.mysql.com/doc/mysql/en/connection-access.html>).

G.9 System Management Console (SMC) Passwords

To change the password to access the SMC, simply log in to the SMC and click on the Administration tab. From the left hand menu click on the Manage Users link and use the interface to change the password.



Appendix

H

CMP External Agents

External Agents are CMP related processes that communicate with the CMP Proxy and provide configuration and monitoring capabilities for third party components. Examples of third party processes monitored by external agent components are Squid, TTS Clients, ASR Clients, TTS Servers and ASR Servers. The external agents have the process name `cmpexagent(.exe)` and allow the OA&M Framework to interact with third party components. Sometimes the external agent processes are also used to start/stop the third party processes via the CLC.

Note that squid can not be started/stopped via the external agent. To start/stop squid use one of the following methods:

1. Using Linux shell, run `/etc/rc3.d/S90squid start/stop`
2. Using CLC, run `runscript - - false startcache stop` or `runscript - - false startcache start`
3. Using SMC, go to `Operations > start/stop cache`

Note: CMP External Agent processes will not start properly if one of its log files or configuration files cannot be accessed due to insufficient permissions.

Appendix H: CMP External Agents

Revision History

Version	Date	Change Summary	Author/Editor
1	August 13 th 2003	Initial release	Rakesh Tailor
	September 19 th 2003	Updated configuration files in appendices. Update sections 2.1.1, 5.1, 7. Added 6.4 for Provisioning service.	Monti Ghai
	September 23 rd 2003	Updated sections on SMC configuration as well as details on Hunt Groups and Dialing Rules.	Wen Wang
2	December 17 th 2003	Updated document to reflect changes for CMP2.1.	Rakesh Tailor
3	March 2 nd 2004	Updated document to reflect changes for CMP2.2	Rakesh Tailor
4	June 19 th 2004	Added details for new features in CMP2.3, including Logging, Alarming and SNMP changes	Rakesh Tailor
5	February 28 th 2005	Added details for new features in VoiceGenie 7.0.0	Rakesh Tailor
6	March 7 th , 2005	Moved various components out of this guide to create their respective reference guides, including CLC, SMC, SNMP and CMP	Andrew Ho
7	March 21 th , 2005	Updating all the Appendices to reflect changes in VoiceGenie 7.0.0. Added Appendices C, D, E, F, G	Monti Ghai
8	March 23 rd , 2005	Merged document with CMP document and added more 7.0.0 details.	Rakesh Tailor
9	June 1 st , 2005	Added details concerning replication.	Monti Ghai, Rakesh Tailor
10	February 20 th , 2006	Updates for 7.1.	Rakesh Tailor
11	September 5 th , 2006	Updates for 7.1	Monti Ghai
12	February 2 nd , 2007	Updates for 7.1, Appendix D, for database cleanup scripts to be run on alternative days on Primary and Slave DB Server	Johnson Tse
13	March 8 th , 2007	ER115575424, updated Custom Sink example error as well the typo	Johnson Tse
14	September 21 st , 2007	Updates for 7.2	Wen Wang

