

VOICEGENIE

VoiceGenie 7

OA&M Framework – CLC Guide

April 13th, 2005



VoiceGenie Contacts

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

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

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

Proprietary / Copyright Information

This material contains proprietary and/or copyright information of VoiceGenie Technologies Inc. and may not be copied, used, or disclosed without the permission of VoiceGenie Technologies Inc.

© COPYRIGHT 2005 VoiceGenie Technologies Inc.

Notice

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

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

Trademarks

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

Table of Contents

1	Introduction	7
2	Configuration	8
3	Accessing the CLC	8
4	Commands Overview	8
5	General Commands	11
5.1	Query	11
5.2	Health	11
5.3	Snapshot	11
5.4	Alarm	12
5.5	Run Script	12
5.6	Help	12
5.7	Exit	13
6	General Component Operation Commands	14
6.1	Tracelevel	14
6.2	Start / Stop	14
6.3	Resume / Suspend	15
7	General Configuration Commands	16
7.1	Cfginfo	16
7.2	Cfggetid	16
7.3	Cfgxmladd	17
8	Configuration Editing Commands	17
8.1	Cfgcreate	17
8.2	Cfgupdate	18
8.3	Cfgdelete	18
8.4	Cfgtarget	19
9	Configuration Parameter Editing Commands	20
9.1	Paramquery	20
9.2	Paramupdate	20
9.3	Paramenable	20
9.4	Paramdisable	21
10	Provision Editing Commands	22
10.1	Prvquery	22
10.2	Prvadd	23
10.3	Prvupdate	23
10.4	Prvdelete	24
10.5	Prvtarget	24
10.6	Prvuntarget	25
11	OA&M Network Editing Commands	26
11.1	Compadd	26

11.2	Compupdate	26
11.3	Compdelete	27
12	Other Commands	28
12.1	Rollover	28
12.2	Sendevent	28
13	Call Manager Specific Commands	30
13.1	Setstate	30
14	Call Manager Agent Mode Specific Commands	32
15	SS7 Call Control Specific Commands	33
16	Appendix A – VoiceGenie SNMP Traps	34
16.1	CLC Traps	35

Revision History

Version	Date	Change Summary	Author/Editor
1.0	August 13 th 2003	Initial release	Rakesh Tailor
2.0	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
2.1	September 23 rd 2003	Updated sections on SMC configuration as well as details on Hunt Groups and Dialing Rules.	Wen Wang
2.2	December 17 th 2003	Updated document to reflect changes for CMP2.1.	Rakesh Tailor
2.3	March 2 nd 2004	Updated document to reflect changes for CMP2.2	Rakesh Tailor
2.4	June 19 th 2004	Added details for new features in CMP2.3, including Logging, Alarming and SNMP changes	Rakesh Tailor
2.5	February 28 th 2005	Added details for new features in VoiceGenie 7.0.0	Rakesh Tailor
2.6	April 13 th , 2005	Final Revision for VoiceGenie 7 Release	Andrew Ho

1 Introduction

The Command Line Console (CLC) provides a command line interface to many of the functionalities provided by the OA&M Framework. To access this information, users can issue commands through a telnet session to the CLC or via shell commands.

2 Configuration

The Command Line Console configuration file is located at /usr/local/cmp-proxy/config/cmpclc.cfg. The parameter cmp.clc_port determines the port that can be used to access the CLC. By default this value is set to 8999. Also, the parameter cmp.externally_accessible_ips can be used to allow access to the CLC. By default, access to the CLC is limited to localhost. Adding a hostname or IP to the cmp.externally_accessible_ips parameter will allow any connection requests from that hostname or IP to be accepted. Multiple hostnames/IPs can be specified by providing a pipe (|) delimited list of hostnames or IPs. For example:

```
cmp.externally_accessible_ips = fibula.voicegenie.com|cmpdev.voicegenie.com
```

3 Accessing the CLC

To access the CLC users can telnet to the port defined by cmp.clc_port, which is by default port 8999. Also, under Linux or Solaris users can simply type clc at the command prompt of any VoiceGenie platform. On Windows it is preferable to use a telnet client like Putty to access the CLC via telnet.

On Linux a script called clccmd is included to allow scripts to be written that use the CLC to submit commands to the OA&M Framework.



Note: the telnet client that comes with the Windows 2000 Server operating system does not automatically echo back commands sent to the CLC. It is preferable to use another telnet client to access the CLC such as Putty. However, user can set local echo on by performing the following:

Access the CLC using telnet:
 telnet localhost 8999

Enter the escape characters:
 CTRL +]

Set the telnet client to echo:
 set LOCAL_ECHO

4 Commands Overview

The CLC supports the following commands, where parameters in [] are required and those in <> are optional:

General Commands:

- query <filter string>
- health <service> <host> <instance>
- snapshot [service] <host> <instance>
- alarm [priority] [number] [data]
- runscript [host] [instance] [wait] [scriptlabel] <arguments>
- help <category>, ? <category>
- exit, x

General Component Operation Commands:

- tracelevel [service] <host> <instance> <lev>

- start <service> <host> <instance>
- stop <service> <host> <instance>

General Configuration Commands:

- cfginfo <service> <version>
- cfggetid [service] <host> <instance>
- cfgxmladd [service] "[subtype]" [version] [filename]

Configuration Editing Commands:

- cfgcreate [service] "[subtype]" [version] [filename] [configName]
- cfgupdate [cfgID] [filename]
- cfgdelete [cfgID]
- cfgtarget [cfgID] [service] <host> <instance>

Configuration Parameter Editing Commands:

- paramquery [cfgID] <name>
- paramupdate [cfgID] [name] [value]
- paramenable [cfgID] [name]
- paramdisable [cfgID] [name]

Provision Editing Commands:

- prvquery <type> <service> <host> <instance>
- prvadd [type] [entry]
- prvupdate [entryID] [entry]
- prvdelete [entryID]
- prvtarget [entryID] [service] <host> <instance>
- prvuntarget [entryID] [service] <host> <instance>

OA&M Network Editing Commands:

- compadd [service] <host> <instance>
- compupdate [networkID] [host]
- compdelete [networkID]

Other Generic Commands:

- clearstats [service] <host> <instance> <args>
- clienttrace [service] <host> <instance> <args>
- login [service] <host> <instance> <args>
- logout [service] <host> <instance> <args>
- makeready [service] <host> <instance> <args>
- restart [service] <host> <instance> <args>
- resume [service] <host> <instance> <args>
- rollover [service] <host> <instance> <args>
- sendevent [service] <host> <instance> <args>
- setstate [service] <host> <instance> <args>
- shutdown [service] <host> <instance> <args>
- ss7isupmgt [service] <host> <instance> <args>
- ss7mtpmgt [service] <host> <instance> <args>

- sscps7 [service] <host> <instance> <args>
- suspend [service] <host> <instance> <args>



Note: The CLC expects the result of a command within 4 seconds. If the result of the command is not returned to the CLC within 4 seconds the CLC will print out *No response*.

5 General Commands

5.1 Query

The query command allows users to see what platforms and components are connected to the OA&M network. This command has the following format:

```
query <filter string>
```

where <filter string> is an optional argument that filters the results and shows only those lines that contain the specified string.

ex. *Example:* 'query galahad' shows all components with galahad in its hostname or component name; 'query Call Manager' or 'query callmgr' would show all Call Managers on the network; 'query offline' would show all offline components.

Note: the CLC supports the use of short command names that are non-ambiguous, as a result, the command query can be shortened to q.

5.2 Health

The health command allows users to get the latest health string for a particular component on a server. The command has the following format:

```
health <service> <host> <instance>
```

where all parameters are optional. If no parameters are specified, the health for all components on the local system will be returned. <service> is the name of the service or component whose health information is being requested, i.e. callmgr, clc, the <host> is the hostname of where the service is located, and <instance> is the instance of the service, by default this is 1. The format of the health status string returned is dependent on the specific component.

ex. *Example:* 'health' shows the health for all components on the local machine, 'health callmgr' shows the health for the Call Manager on the local machine, 'health callmgr galahad' shows the health of the Call Manager on galahad.

5.3 Snapshot

The snapshot command gets a snapshot of the real time port status of a service. The command has the following format:

```
snapshot [service] <host> <instance>
```

where [service] is a service, usually Call Manager, VoiceXML Interpreter or SS7, <host> is an optional parameter that specifies the hostname and <instance> specifies the instance. If <host> is not specified a snapshot of the local <service> is returned.

5.4 Alarm

Using the CLC it is possible to inject alarms into the OA&M infrastructure. This is useful in writing monitoring scripts that need to send an alarm that is handled by the framework. The OA&M framework can route these alarms to syslog, SNMP or to a number of other sinks. This command has the following format:

```
alarm [priority] [number] [data]
```

where parameters in [] are required. The parameters are defined as follows:

[priority] – a number between 0 and 5, where 0 is critical (highest priority) and 5 is the lowest priority.
[number] – a user defined number from 0 to 1048575 that can be used to uniquely identify the alarm.
[data] – any user defined data that describes the alarm

This command will return “Alarm Sent” when it gets a reply that the log was sent correctly.

5.5 Run Script

The runscript command can be used to run scripts on a remote server. The scripts that can be run are defined in the CMP Proxy configuration using the configuration parameter `cmp.script_labels`. Only scripts that are defined in the `cmp.script_labels` parameter can be run. This command has the following format:

```
runscript [host] [instance] [wait] [scriptlabel] <arguments>
```

where parameters in [] are required and parameters in <> are optional. The parameters are defined as follows:

[host] – the name of the host that the command is directed towards, a '-' can be used to denote localhost
[instance] – the instance of the service that the command is directed towards, a '-' can be used to denote 1
[wait] – determines if the confirmation message sent back to the CLC should wait for the result of the script; valid values are 'true' (i.e. wait for result) and 'false' (i.e. do not wait for result)
[scriptlabel] – the script label as defined in the CMP Proxy configuration parameter `cmp.script_labels`
<arguments> - arguments for the script if any

ex. *Example:* The following runs the script `viewlicense (/usr/local/cmp-proxy/scripts/viewlicense.pl)` and waits to return the result:

```
CLC> runscript 10.0.0.72 - true viewlicense get
Script /usr/local/cmp-proxy/scripts/viewlicense.pl get succeeded, result is
vggateway in 2037/12/31 500
vggateway out 2037/12/31 500
vggateway asr 2037/12/31 500
vggateway tts 2037/12/31 500
```

5.6 Help

The help or ? command returns general usage information for the CLC. The command is of the following format:

```
help <category>
```

where <category> is one of 'config', 'provision', 'network' or 'generic'. If <category> is not specified the general help information is returned, otherwise details for that specific category are returned.

5.7 Exit

The exit or x command can be used to gracefully close the connection to the CLC.

6 General Component Operation Commands

6.1 Tracelevel

The tracelevel command allows users to get or set the cmp.trace_flag parameter of a service. This flag is responsible for turning on the detailed tracing information that is necessary for debugging. The command has the following format:

```
tracelevel [service] <host> <instance> <lev>
```

where the parameters are defined as follows:

[service] – the name of the service that the command is directed towards

<host> – the name of the host that the command is directed towards, a '-' can be used to denote localhost

<instance> – the instance of the service that the command is directed towards, a '-' can be used to denote 1

<lev> –if this parameter is not specified, the command returns the existing cmp.trace_flag state (enabled/disabled) for a service, other wise it can be set to 'enable' or 'disable' to enable or disable tracing.

ex. *Example:* 'tracelevel callmgr' returns the trace level of the local callmgr, whereas 'tracelevel callmgr - 0xf0099999' sets the local callmgr's trace level to 0xf0099999



Note: any changes in tracing made by the tracelevel command, i.e. enable or disable, are not permanent and will be reset to the configuration value of cmp.trace_flag once the service is restarted.

6.2 Start / Stop

The CMP Proxy is responsible for starting and stopping VoiceGenie software. The start and stop commands provide users with the ability to start or stop the platform or any individual component. The start and stop commands have the following format:

```
stop/start <service> <host> <instance>
```

The parameters are defined as follows:

<service> – the name of the service that the command is directed towards, if this parameters is not specified it defaults to 'all', which specifies the entire platform.

<host> – the name of the host that the command is directed towards, a '-' can be used to denote localhost.

<instance> – the instance of the service that the command is directed towards, a '-' can be used to denote 1



Note: only components that are defined in cmp.components can be started or stopped using the CLC.



Note: VoiceGenie software can take up to 1 minute to stop. As a result, please be sure to wait at least 1 minute before trying to restart the software.

6.3 Resume / Suspend

The resume and suspend commands have the following format:

```
resume/suspend [service] [host] [instance] <force drop>
```

where parameters in [] are required and parameters in <> are optional. The parameters are defined as follows:

- [service] – the service that will be suspended or resumed
- [host] – the host that the service resides on, '-' can be used to denote localhost
- [instance] – the instance of the service, the default value is 1
- <force drop> – this parameter can be 'true' or 'false', determines if all existing processing should be stopped forcefully

Once the command is issued, CLC will display the command's result. The following table lists the possible command results:

Result	Meaning
Command is in processing	Command is valid and accepted by the platform.
Usage:	The command entered was of the wrong format.
Failed to process the command	The system can not process the command.



Note: The 'resume' command is an alias for 'setadminstate duplex'; the 'suspend' command is an alias for 'setadminstate disable'.

7 General Configuration Commands

7.1 Cfginfo

The `cfginfo` command returns information to the user about what configurations exist for a given service and version. The command has the following format:

```
cfginfo <service> <version>
```

where parameters in `<>` are optional. The parameters are defined as follows:

- `<service>` – the service type of the configuration you are interested in
- `<version>` – the version of configuration you are querying for, version is specified in the format `x.y.z`



Note: if neither `<service>` or `<version>` is specified, information about all configurations is returned by the command.

ex.

Example: The following is an example using the `cfginfo` command, it returns a list of all configurations since service and version are not specified:

```
CLC>cfginfo
CfgID   Component Type           Version  Subtype   Name
-----
2       CMP Server (cmpserver)    7.0.0   AS3.0     default
1       CMP Proxy (cmpproxy)     7.0.0   AS3.0     default-1
5       CMP Proxy (cmpproxy)     7.0.0   AS3.0     default-2
6       VG SNMP Agent (vgsnmp)   7.0.0   AS3.0     default
7       Call Manager (callmgr)   7.0.0   AS3.0     default
11      Call Manager (callmgr)   7.0.0   AS3.0     SIP
8       VXML Interpreter (vxmli) 7.0.0   AS3.0     default
12      VXML Interpreter (vxmli) 7.0.0   AS3.0     SIP
3       System Mgmt Cnsl (smc)   7.0.0   AS3.0     default
4       Command Line Cnsl (clc)  7.0.0   AS3.0     default
15      Command Line Cnsl (clc)  7.0.0   AS3.0     new config
9       Web Proxy (iproxy)       7.0.0   AS3.0     default
13      Web Proxy (iproxy)       7.0.0   AS3.0     SIP
10      Squid Cache (squid)      7.0.0   AS3.0     default
14      Squid Cache (squid)      7.0.0   AS3.0     SIP
```

ex.

Example: The following is an example using the `cfginfo` command, it returns a list of all configurations for the Call Manager since service is specified:

```
CLC> cfginfo callmgr
CfgID   Component Type           Version  Subtype   Name
-----
7       Call Manager (callmgr)   7.0.0   AS3.0     default
11      Call Manager (callmgr)   7.0.0   AS3.0     SIP
```

7.2 Cfggetid

The `cfggetid` command can be used to find out what configuration is being used by a service. The command has the following format:


```
cfggetid [service] <host> <instance>
```

where parameters in [] are required and parameters in <> are optional. The parameters are defined as follows:

- [service] – the name of the service for which the configuration ID is being queried
- <host> – the name of the host for where the service whose configuration ID is being queried, a '-' can be used to denote localhost
- <instance> – the instance number of the service whose configuration ID is being queried, default value is 1, a '-' can be used to denote 1

ex. *Example:*

The following is an example of the `cfggetid` command. This example returns the configuration ID, configuration name and version of the configuration that is being used by the local Call Manager service:

```
CLC> cfggetid callmgr  
Configuration ID: 11, Name: SIP, Version: 7.0.0
```

7.3 Cfgxmladd

The `cfgxmladd` command can be used to upload a new configuration XML for a service. This command should not be used under normal circumstances. The command has the following format:

```
cfgxmladd [service] "[subtype]" [version] [filename]
```

where parameters in [] are required. The parameters are defined as follows:

- [service] – the name of the service for which the XML is being added
- [subtype] – the subtype as specified in the XML file
- [version] – the version of the XML being added, the version is specified in the format x.y.z
- [filename] – the full path and filename of the XML file that is being added



Note: when an XML that already exists is replaced the CMP Server should be restarted so that changes in the XML take effect.

8 Configuration Editing Commands

8.1 Cfgcreate

The `cfgcreate` command can be used to create a new configuration. The command has the following format:

```
cfgcreate [service] "[subtype]" [version] [filename] [configName]
```

where parameters in [] are required. The parameters are defined as follows:

- [service] – the type of service for which the configuration is being created
- [subtype] – the subtype of the configuration to be created
- [version] – the version of the configuration to be created, the version is specified in the format x.y.z
- [filename] – the full path and filename of the file to use to update values of the configuration, for the default values the user can specify '-'
- [configName] – the name of the new configuration being created

ex. Example:

The following is an example of the `cfgcreate` command. This example creates a configuration called “New Configuration” with all default values, it returns the configuration ID of the newly created configuration:

```
CLC> cfgcreate cmpproxy "AS3.0" 7.0.0 - New Configuration
Config ID: 16
cmp.log_dll.file: Parameter Saved
cmp.log_dll.upstream: Parameter Saved
cmp.log_dll.metrics: Parameter Saved
...
```

8.2 Cfgupdate

The `cfgupdate` command can be used to update an existing configuration with new parameter values. The command has the following format:

```
cfgupdate [cfgID] [filename]
```

where parameters in [] are required. The parameters are defined as follows:

- [cfgID] – the configuration ID of the configuration that should be updated
- [filename] – the full path and filename of the file to use to update values of the configuration

ex. Example:

The following is an example of the `cfgupdate` command. This example updates the configuration with ID 16 with values in the configuration file `/usr/local/cmp-proxy/config/cmpproxy.cfg`, it returns a list of parameters that were updated:

```
CLC> cfgupdate 16 /usr/local/cmp-proxy/config/cmpproxy.cfg
cmp.backup_cmpe: Parameter Updated
cmp.components: Parameter Updated
cmp.primary_cmpe: Parameter Updated
pml_i: Parameter Updated
cmgr: Parameter Updated
prxy: Parameter Updated
pml_i_limits: Parameter Updated
cmgr_limits: Parameter Updated
prxy_limits: Parameter Updated
pml_i_start_script: Parameter Updated
cmgr_start_script: Parameter Updated
prxy_start_script: Parameter Updated
pml_i_restart_script: Parameter Updated
cmgr_restart_script: Parameter Updated
prxy_restart_script: Parameter Updated
pml_i_stop_script: Parameter Updated
cmgr_stop_script: Parameter Updated
prxy_stop_script: Parameter Updated
```

8.3 Cfgdelete

The `cfgdelete` command can be used to delete an existing configuration. The command has the following format:

```
cfgdelete [cfgID]
```

where parameters in [] are required. The parameters are defined as follows:

- [cfgID] – the configuration ID of the configuration to be deleted

ex. *Example:*

The following is an example of the `cfgdelete` command. This example deletes the configuration with ID 16, it returns a confirmation of the delete:

```
CLC> cfgdelete 16  
Configuration Deleted
```

8.4 Cfgtarget

The `cfgtarget` command allows users to target a configuration to a specific service. The command is of the following format:

```
cfgtarget [cfgID] [service] <host> <instance>
```

where parameters in [] are required and parameters in <> are optional. The parameters are defined as follows:

- [cfgID] – the configuration ID of the configuration to be targeted
- [service] – the name of the service that this configuration is being targeted towards
- <host> – the hostname of machine where the service resides, a '-' can be used to denote localhost.
- <instance> – the instance of the service that the command is directed towards, a '-' can be used to denote 1

ex. *Example:*

The following is an example of the `cfgtarget` command. This example targets configuration with ID 16 to the CMP Proxy on the localhost:

```
CLC> cfgtarget 16 cmpproxy  
Configuration Targeted Successfully
```

9 Configuration Parameter Editing Commands

9.1 Paramquery

The paramquery command can be used to query the entire contents or a single parameter of an existing configuration. The command has the following format:

```
paramquery [cfgID] <name>
```

where parameters in [] are required and parameters in <> are optional. The parameters are defined as follows:

- [cfgID] – the configuration ID of the configuration to be queried
- <name> – the configuration parameter to query, if blank the entire contents are queried

ex. *Example:*

The following is an example of the paramquery command. This example queries the value of the cmp.trace_flag parameter from the configuration with ID 16, it returns the value (FALSE) and whether the parameter is Enabled or Disabled:

```
CLC> paramquery 16 cmp.trace_flag  
cmp.trace_flag: FALSE: Enabled
```

9.2 Paramupdate

The paramupdate command can be used to update the value of a parameter in a configuration. The command has the following format:

```
paramupdate [cfgID] [name] [value]
```

where parameters in [] are required. The parameters are defined as follows:

- [cfgID] – the configuration ID of the configuration to be updated
- [name] – the configuration parameter to update
- [value] – the new value of the parameter

ex. *Example:*

The following is an example of the paramupdate command. This example updates the value of the cmp.trace_flag parameter with the value TRUE:

```
CLC> paramupdate 16 cmp.trace_flag TRUE  
cmp.trace_flag Parameter Updated
```

9.3 Paramenable

The paramenable command can be used to enable a parameter in a configuration. The command has the following format:

```
paramenable [cfgID] [name]
```

where parameters in [] are required. The parameters are defined as follows:

- [cfgID] – the configuration ID of the configuration to be enabled
- [name] – the configuration parameter to enable

ex. *Example:*

The following is an example of the paramenable command. This example enables the cmp.email parameter:

```
CLC> paramenable 16 cmp.email  
cmp.email Parameter Updated
```

9.4 Paramdisable

The paramdisable command can be used to disable a parameter in a configuration. The command has the following format:

```
paramdisable [cfgID] [name]
```

where parameters in [] are required. The parameters are defined as follows:

- [cfgID] – the configuration ID of the configuration to be disabled
- [name] – the configuration parameter to disable

ex. *Example:*

The following is an example of the paramdisable command. This example disables the cmp.trace_flag parameter:

```
CLC> paramdisable 16 cmp.trace_flag  
cmp.trace_flag Parameter Updated
```

10 Provision Editing Commands

10.1 Prvquery

The prvquery command can be used to query information about provisioning. The command has the following format:

```
prvquery <type> <service> <host> <instance>
```

where parameters in <> are optional. The parameters are defined as follows:

- <type> – the type of provisioning, a number, if it is not defined the list of valid types is returned
- <service> – the name of the component that the command is directed towards
- <host> – the name of the host that the command is directed towards, a '-' can be used to denote localhost
- <instance> – the instance on the component that the command is directed to, default value is 1

ex. *Example:*

The following is an example of the prvquery command with no arguments. This command returns a list of valid provision types that can be used for the <type> parameter:

```
CLC> prvquery
Entry Type Description
-----
1 DNIS - URL Mapping
2 Dialing Rules
3 SNMP Trap Filter
4 Hunt Groups
100 ICM Platform Mapping
101 ICM Service Mapping
102 ICM Trunk Mapping
103 SS7 DNIS - URL Mapping
104 SS7 Circuit ID/Media Platform Mapping
```

The following is an example of the prvquery command where the <type> is specified. In this case a list of all provisioning entries of that type is returned:

```
CLC> prvquery 1
Entry ID Entry Type Entry
-----
1 1 <key name="DNIS" value="XXXX"/>
<application module="VXML">
<param name="url"
value="file:///usr/local/phoneweb/samples/helloaudio.vxml"/>
<param name="default" value="defaults.vxml"/>
</application>
```

The result is a table that contains a listing on Entry ID, Entry Type and the actual contents of the entry.

If the prvquery command is used where the <type> is specified as well as a specific component (i.e. service, host, instance), then all entries of that <type> that are targeted to that component are returned in the table along with the Entry ID.

10.2 Prvadd

The prvadd command can be used to add provisioning entries. The command has the following format:

```
prvadd [type] [entry]
```

where parameters in [] are required. The parameters are defined as follows:

- [type] – the type of provisioning, a number
- [entry] – the contents of the provision record, a new line is specified using ‘\n’

ex. *Example:* The following is an example using the prvadd command, in the example an entry with type 1 is added, also, ‘\n’ is used to insert new line characters where required:

```
CLC> prvadd 1 <key name="DNIS" value="XXXX"/>\n<application
module="VXML">\n<param name="url"
value="file:///usr/local/phoneweb/samples/helloaudio.vxml"/>\n<param
name="default" value="defaults.vxml"/>\n</application>
Entry Added with Entry ID 3
```

The following table outlines the return values and their meanings:

Result	Meaning
Entry Added with Entry ID <X>	Success, entry added with ID <X>.
Invalid Entry Key for Provision Type	Failed, entry key inside entry is not valid.
Database Error: Unable to Add Provision Entry	Failed, encountered a database error.
Unable to Add Entry: Duplicate Not Allowed	Failed, entry being added would create a duplicate value which is not allowed.

10.3 Prvupdate

The prvupdate command can be used to update a provisioning entry. The command has the following format:

```
prvupdate [entryID] [entry]
```

where parameters in [] are required. The parameters are defined as follows:

- [entryID] – the ID of the entry that needs to be updated
- [entry] – the new contents of the entry, i.e. the updated value, a new line is specified using ‘\n’

ex. *Example:* The following is an example using the prvupdate command, in the example an entry with ID 3 is updated, also, ‘\n’ is used to insert new line characters where required:

```
CLC> prvupdate 3 key name="DNIS" value="XXXX"/>\n<application
module="VXML">\n<param name="url"
value="file:///usr/local/phoneweb/samples/helloaudio.vxml"/>\n<param
name="default" value="defaults.vxml"/>\n</application>
Entry Updated
```

The following table outlines the return values and their meanings:

Result	Meaning
Entry Updated	Success.
Database Error: Unable to Update	Failed, encountered a database error.

Provision Entry	
Invalid Entry Key for Provision Type	Failed, entry key inside the entry is not valid.
Unable to Update Entry: Entry Key <X> should not be Changed	Failed, the entry key of the entry can not be changed using the prvupdate command, a new one must be added using prvadd.
Unable to Update Entry: Entry ID is Invalid	Failed, The entry ID provided is invalid.

10.4 Prvdelete

The prvdelete command can be used to delete provision entries. The command has the following format:

prvdelete [entryID]

The parameter is defined as follows:

- [entryID] – the ID of the entry that needs to be deleted

ex. *Example:* The following is an example using the prvdelete command:

```
CLC> prvdelete 3
Entry Deleted
```

The following table outlines the return values and their meanings:

Result	Meaning
Entry Deleted	Success.
Database Error: Unable to Update Provision Entry	Failed, encountered a database error.
Unable to Update Entry: Entry ID is Invalid	Failed, invalid entry ID.

10.5 Prvtarget

The prvtarget command can be used to assign a provision entry to a component. The command has the following format:

prvtarget [entryID] [service] <host> <instance>

where parameters in [] are required and parameters in <> are optional. The parameters are defined as follows:

- [entryID] – the ID of the entry that will be assigned
- [service] – the component that the entry will be assigned to
- <host> – the hostname where the component is, use '-' or 'localhost' to specify the localhost
- <instance> – the instance on the component that the command is directed to, default value is 1

ex. *Example:* The following is an example using the prvtarget command where entry ID 1 is targeted to the local Call Manager:

```
CLC> prvtarget 1 callmgr
Provision with Entry ID 1 targeted to Network ID 6
```

The following table outlines the return values and their meanings:

Result	Meaning
Provision with Entry ID <X> targeted to Network ID <Y>	Success, entry <X> targeted to component <Y>.
Database Error: Unable to Target Entry ID <X> to Network ID <Y>	Failed, encountered a database error.
Internal Error: Unable to Target Entry ID <X> to Network ID <Y>	Failed, an internal error occurred, report to VoiceGenie.
Invalid Entry ID: <X>	Failed, the entry ID provided is invalid.
Entry ID <X> already targeted to Network ID <Y>	The entry ID is already targeted to that component.
Duplicate Entry Key: Network ID <Y> already has an entry with the same Entry Key as <X>	Failed, entry can not be targeted since the component already has an entry with that entry key targeted to it (i.e. same DNIS).

10.6 Prvuntarget

The prvuntarget command can be used to no longer assign a provision entry to a component. The command has the following format:

```
prvuntarget [entryID] [service] <host> <instance>
```

where parameters in [] are required and parameters in <> are optional. The parameters are defined as follows:

- [entryID] – the ID of the entry that will be assigned
- [service] – the component that the entry will no longer be assigned to
- <host> – the hostname of the component, use '-' or 'localhost' to specify localhost
- <instance> – the instance on the component that the command is directed to, default value is 1

ex. *Example:* The following is an example using the prvuntarget command where entry ID 1 is no longer targeted to the local Call Manager:

```
CLC> prvuntarget 1 callmgr
Entry ID 1 no longer targeted to Network ID 6
```

The following table outlines the return values and their meanings:

Result	Meaning
Entry ID <X> no longer targeted to Network ID <Y>	Success.
Database Error: Unable to Delete Entry ID <X> with target of Network ID <Y>	Failed, encountered a database error.
Invalid Entry ID: <X>	Failed, the entry ID provided is invalid.

11 OA&M Network Editing Commands

11.1 Compadd

The compadd command can be used to add components to the OA&M network. The command has the following format:

```
compadd [service] <host> <instance>
```

where parameters in [] are required and parameters in <> are optional. The parameters are defined as follows:

- [service] – the service name of the component being added
- <host> – the hostname of where the component is being added, can be left blank to specify localhost, or use '-' or 'localhost'
- <instance> - the instance of the component being added, blank or '-' implies instance 1

ex. *Example:* The following is an example using the compadd command, a ICM Call Control component is added:

```
CLC> compadd icm localhost
Component Added with NetworkID 23
```

The following table outlines the return values and their meanings:

Result	Meaning
Component Added with NetworkID <Y>	Success, added with network ID <Y>.
Database Error: Unable to Add Component	Failed, encountered a database error.
Unable to Add Component: Invalid Component Type	Failed, component type is invalid.
Unable to Add Component: Corrupted Results	Failed, internal error, report to VocieGenie.
Unable to Add Component	Failed.

11.2 Compupdate

The compupdate command can be used to update the hostname of a component. The command has the following format:

```
compupdate [networkID] [host]
```

where parameters in [] are required and parameters in <> are optional. The parameters are defined as follows:

- [networkID] – the networkID of the component whose hostname is being changed, it must be a CMP Proxy component.
- [host] – the new hostname of the component

ex. *Example:* The following is an example using the compupdate command, updating the hostname to 'foo.voicegenie.com':

```
CLC> compupdate 9 foo.voicegenie.com
```

Component 9 Updated with Host foo.voicegenie.com and Name foo.voicegenie.com

The following table outlines the return values and their meanings:

Result	Meaning
Component <Y> Updated with Host <Z> and Name <Z>	Success, update <Y> with the hostname <Z> and also changed name to <Z>.
Unable to Update Component: Invalid NetworkID	Failed.
Database Error: Unable to Update Component	Failed, encountered database error.

11.3 Complete

The complete command can be used to delete a component. The command has the following format:

complete [networkID]

The parameter is required and is defined as follows:

- [networkID] – the networkID of the component which is being deleted, a cmp proxy component can only be deleted if it has no child components.

ex. *Example:* The following is an example using the complete command, deleting networkID 23:

```
CLC> complete 23
Component with NetworkID 23 Deleted
```

The following table outlines the return values and their meanings:

Result	Meaning
Component with NetworkID <Y> Deleted	Success.
Unable to Delete Component: Invalid NetworkID	Failed.
Unable to Delete Component: Children must be Deleted First	Failed, can not delete a component that has children, i.e the CMP Proxy can not be deleted without deleting all other components on that server first.
Unable to Delete Component: Component must be offline	Failed, component must be offline in order to delete it.
Database Error: Unable to Delete Component	Failed, encountered database error.

12 Other Commands

12.1 Rollover

The rollover command can be used to rollover the pw_metricsfile if required. The command has the following format:

```
rollover cmpproxy <host> <instance>
```

where parameters in <> are optional. The parameters are defined as follows:

- <host> – the hostname of the server where the rollover should occur, can be left blank to specify localhost, or use '-' or 'localhost'
- <instance> - the instance, blank or '-' implies instance 1

ex. *Example:* The following is an example using the complete command, deleting networkID 23:

```
CLC> rollover cmpproxy  
Metrics file rolled over
```

12.2 Sendevent

The sendevent command can be used to send messages to an active VoiceXML session. The command has the following format:

```
sendevent vxmli [host] [instance] [recipient_address] [sender_address] [message]
```

where all parameters are required. The parameters are defined as follows:

- [host] - the host where the VXMLI is running. For localhost, a '-' can be used
- [instance] - the instance of the interpreter, '-' implies instance 1
- [recipient_address] - the unique ID of the VXMLI session the message will be sent to. The ID of an VXMLI session can be looked up from the session variable:

```
session.com.voicegenie.instance.myself
```

A valid *recipient_address* has the following format:

```
uuu.xxx.yyy.zzz$ttttttttt$0xXXXXXXXX
```

Field	Interpretation
uuu.xxx.yyy.zzz	the IP address of the host where the VXMLI instance resides
tttttttttt	UTC, measured in seconds since midnight, January 1, 1970
0xXXXXXXXX	The VXMLI instance message handler (a unique handle for the instance)

The maximum length allowed for *recipient_address* is 127 characters.

- [sender_address] - can also be specified as a valid ID of a VXMLI session, or any character string whose length is under 127 characters. If the *sender_address* is a valid session ID, the recipient session would be able to send back a message to the session with the ID specified by the *sender_address*, which is dependent on the VoiceXML application.

- [message] - *any* combination of characters, with a maximum length of 2999 characters.

ex. *Example:* The following is an example using the sendevent command:

```
CLC> sendevent vxmli - - 10.0.0.117$1079459239$0x892d5c8 1234 Hello World!
```

The following table outlines the return values and their meanings:

Result	Meaning
Sending Message from <sender_address> to <recipient_address>	Success.
Usage: sendevent [service] [hostname] [inatnce][recipient_address] [sender_address] [message]	Failed, invalid format command.
Can not deliver message	Failed.
Failed to send message	Failed.

13 Call Manager Specific Commands

The CLC supports the following Call Manager specific commands:

- setstate
- restart
- shutdown

13.1 Setstate

The setstate command has the following format:

```
setstate callmgr [Hostname] [Instance] [Mode] <Board> <Channel> <Force Drop>
```



Note: parameters in <> are optional, [] mandatory.

where [Hostname] is the hostname of the machine where the command is targeted.

[Mode] sets the administrative state to inbound/outbound/duplex/disable/config.

- inbound – Set the given platform/board/channel to only allow inbound calls.
- outbound – Set the given platform/board/channel to only allow outbound calls.
- duplex – Set the given platform/board/channel to allow both inbound and outbound calls.
- disable – Set the given platform/board/channel to not allow inbound nor outbound call.
- config – Set the given platform/board/channel to re-read configuration from ~pw/glines.cfg (for dialogic only).



Note: for SIP, only “outbound” and “duplex” states are supported. Also, for SIP, “inbound” state will work the same as “duplex”, while “disable” operates the same as “outbound”.

<Board> is the optional board ID.



Note: For SIP, the board ID should be 0. For Dialogic boards, the board ID should be the board index (starting from 1) plus 100. As a result, valid board IDs are from 101 through 108. These IDs correspond to the board configuration in ~pw/glines.cfg. A board ID of –1 represents all boards and is the default value.

<Channel> is the optional channel ID.



Note: For Dialogic, valid channel IDs are 1 through 30. For SIP, channel ID is ignored. By default, the channel ID is –1, which represents all channels on the given board.

<Force Drop> determines if the call should be dropped immediately. Valid values are true and false. If it is true and if required (say, switching from inbound to outbound mode), it will drop any existing calls immediately. If it is false, the operation will not be executed until the current calls have gracefully terminated. The default value is false.

Once the command is issued, CLC will display the command’s result. The following table lists the possible command results:

Result	Meaning
Command is in processing	Command is valid and accepted by the platform.
Invalid board ID	The Board ID entered is invalid.
Given Board does not exist	The given board does not exist on the platform.
Invalid channel ID	The Channel ID entered by the user is invalid.
Usage:	The command entered was of the wrong format.
Failed to process the command	The system can not process the command.

If the board number is only provided, the operation will be carried out on all channels on the specified board. If neither board nor channel is provided, the operation will be carried out on all boards and channels on the platform.

ex. *Example:* provided that hostname of the desired VG platform is cmpdev:
 'setstate callmgr cmpdev - duplex 101 22' would trigger VG platform to set board 1 (dialogic) channel 22 to duplex mode, and 'setstate callmgr cmpdev duplex' would trigger VG platform to set all channels in duplex mode.

Note that the management of the states for platform, board and channels are all independent. The ultimate operational state of a channel is decided by the combination of the states of all three levels, that the most restricted state will apply. Here are a few examples:

Platform Admin State	Board Admin State	Channel Admin State	Channel Final Operational State
DISABLE	DUPLEX	DUPLEX	DISABLE
DUPLEX	DUPLEX	INBOUND	INBOUND
OUTBOUND	INBOUND	DUPLEX	DISABLE

Considering a VG platform with 2 boards, with all states set to DUPLEX on startup:
 user sets board 1 to INBOUND (i.e. setstate callmgr [Hostname] inbound 101)
 user sets platform to DISABLE (i.e. setstate callmgr [Hostname] disable)
 user sets platform to DUPLEX (i.e. setstate callmgr [Hostname] duplex)

In this scenario after the VoiceGenie server restarts, board 1 will be set to take on inbound calls.

14 Call Manager Agent Mode Specific Commands

The CLC supports the following Call Manager specific commands that are for Agent Mode:

- makeready
- login
- logout

These three commands have the following format:

makeready/login/logout callmgr [Hostname] [Instance] <Board> <Channel> <Force Drop>



Note: Parameters in <> are optional, [] mandatory. And, these three commands does NOT apply to SIP.

In general the meaning of the different parameter values is the same as setstate. Please refer to section 13.1 for the meaning of these parameters. The only difference is that, for makeready/login/logout, board ID can not be 0, since these commands can not be applied to SIP.

Once the command is issued, CLC will display the command's result. The following table lists the possible command results:

Result	Meaning
Command is in processing	Command is valid and accepted by the platform.
Invalid board ID	The Board ID entered is invalid.
Given Board does not exist	The given board does not exist on the platform.
Invalid channel ID	The Channel ID entered by the user is invalid.
Usage:	The command entered was of the wrong format.
Failed to process the command	The system can not process the command.

▪



Note: Issuing the command 'setstate callmgr [Hostname] disable' under agent mode will ensure that the channels are logged out before they are put out of service; however, the commands 'setstate callmgr [Hostname] duplex/inbound/outbound' will NOT login the channels. One must explicitly issue a login command after the channels are active to login the channels.

15 SS7 Call Control Specific Commands

A number of CLC commands exist to enable users to get information from SS7 components, as well as perform certain operations on the SS7 component. The following commands are supported:

- sccpss7
- ss7mtpmgt
- ss7isupmgt

For details please consult the ICM White Paper and SS7 Configuration Documents.

16 Appendix A – VoiceGenie SNMP Traps

This Appendix contains the list of traps that can be produced by a VoiceGenie component. The second last number in the OID corresponds to the type of component. The component types are as follows:

Type	Name
200	CMP Proxy
300	VG SNMP
302	Call Manager
303	VoiceXML Interpreter
308	CLC
309	Web Proxy (IProxy)
311	VRM Server

The last number corresponds to the LogID, this ID uniquely identifies the log and corresponds to the id field in the CallLog table.

Each Description field contains information about the relative severity of the alarm, the severities are:

Severity	Description
CRIT	An alarm event that denotes a critical or fatal condition and results in the failure of the software.
EROR	An alarm event that denotes an error condition that should never happen and that results in the loss of functionality.
WARN	An alarm event that denotes an exceptional situation that may occur legitimately but it is necessary to be aware of.

Also, each alarm has a Response Code specified, the response codes are defined as follows:

Severity	Description
CKAPP	Check application or Web server
CKCFG	Check and correct configuration
CKFS	Check file/directory existence/permission, or disk space
CKHW	Check hardware
CKTY	Check telephony hardware/connection
CKASR	Check ASR server
CKTTS	Check TTS server
CKNW	Check network connection
CKOP	Check operational state of the server
NOTE	Notice/observation
REVG	Report to VoiceGenie [with logs]
SWRS	Software restart: [collect logs] restart VoiceGenie server
HWRS	Hardware restart: [collect logs] reboot VoiceGenie platform

In addition the table list any old alarm code (i.e. pre NexusPoint 6.4) that may relate to the alarm.

16.1 CLC Traps

Name	OID	Description	Response Code	Old Alarm Number
VGLOG-INVALID-MSG-TYPE	.1.3.6.1.4.1.7469.251.1.308.1048597	WARN Invalid Message Type Sent or Received	REVG	-
VGLOG-CANNOT-CREATE-CLIENT	.1.3.6.1.4.1.7469.251.1.308.1048599	WARN Error Creating Client Socket	REVG	-
VGLOG-DSRV-INVALID-QUERY	.1.3.6.1.4.1.7469.251.1.308.6291477	WARN Invalid data query string received	REVG	-
VGLOG-DSRV-UNSUPPORTED-VAR	.1.3.6.1.4.1.7469.251.1.308.6291478	WARN Query for unsupported variable received	REVG	-
VGLOG-SOCKET-SEND-FAILED	.1.3.6.1.4.1.7469.251.1.308.134219731	EROR Socket send failed	CKNW/REVG	10000
VGLOG-VGASSERT	.1.3.6.1.4.1.7469.251.1.308.135267305	CRIT VGASSERT	REVG	-