



VoiceGenie 7

ICM Connector

User's Guide

May 27, 2008

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 © 2005–2008 Genesys Telecommunications Laboratories, Inc. All rights reserved.

About Genesys

Genesys Telecommunications Laboratories, Inc., a subsidiary of Alcatel-Lucent, 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 DVD, 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)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Japan	+81-3-6361-8950	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: May 27, 2008



Table of Contents

Chapter 1	Introduction.....	5
Chapter 2	Benefits of CTI	7
Chapter 3	VoiceGenie and CTI.....	9
	3.1 Application Server Integration.....	9
	3.2 Media Platform Integration.....	9
Chapter 4	Architectural Overview – VoiceGenie and ICM	11
Chapter 5	Access to ICM From VoiceXML	17
	5.1 Passing Data to ICM.....	17
	5.2 Asking ICM to Route a Call.....	19
Chapter 6	Operations, Administration, and Maintenance	21
	6.1 Starting and Stopping	21
	6.2 Health Status	21
	6.3 Logging	22
	6.4 Cluster Management.....	22
Chapter 7	Directory Structure	23
Appendix A	Health SNMP Gets	25
Appendix B	Logging Traps.....	27
	B.1 Severity – Critical (LOG_0)	27
	B.2 Severity – Error (LOG_1)	28
	B.3 Severity – Warning (LOG_2)	30



Chapter

1 Introduction

This document provides an overview of VoiceGenie ICM Connector.

The VoiceGenie platform is the industry leading VoiceXML media platform, and includes integrations with leading speech engines, advanced resource management, VoiceXML 2.0 support, and leading-edge call control.

The Cisco ICM infrastructure is a pre and post-call routing platform, allowing end-to-end management of a telephone call and associated data within a geographically distributed network of call centers and IVRs.

This document includes the following sections:

- [Introduction \(this section\);](#)
- [Benefits of CTI](#)
- [VoiceGenie and CTI](#)
- [Summary](#)



Chapter

2 Benefits of CTI

Use of a CTI infrastructure provides a number of tangible benefits within an organization.

- Cradle to grave tracking of a call within a geographically distributed organization;
- Retrieval of call-associated data that is not normally available;
- Intelligent routing of calls within a network;
- Attachment of caller-associated data to the call;
- Ability to ‘screen-pop’ data as part of call routing;
- Abstraction of the (often obscure) interface to telephony components;

CTI does however have some issues to be aware of:

- CTI infrastructure can be expensive;
- CTI integration has historically been complex;



Chapter

3 VoiceGenie and CTI

The VoiceGenie platform provides a number of ways to support CTI integration. The two most common are:

- Application Server Side Integration
- Media Platform Integration

3.1 Application Server Integration

The VoiceGenie platform delivers call-related data to the application server as part of the initial page fetch related to a VoiceXML call. This data includes the information required to interface to an external CTI server – information such as port number, session identifier, and so on. This allows the application server to manage interaction with the CTI infrastructure.

3.2 Media Platform Integration

Media platform integration allows for CTI infrastructure interaction to take place on the media platform itself. The most common features are provided completely transparently to the application. This method has benefits to the application developer.

The remainder of this white paper focuses on the VoiceGenie media platform integration with the Cisco ICM package.

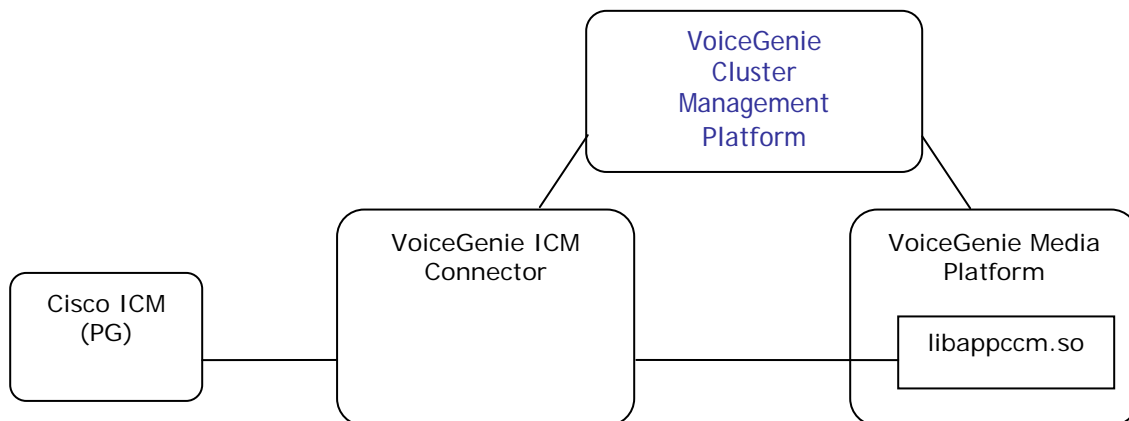


Chapter

4

Architectural Overview – VoiceGenie and ICM

When ICM support is configured, the component architecture is as shown in the following figure.



The VoiceGenie Media Platform loads an optional dynamic library named “libappccm.so”. “CCM” is an acronym for “Call Control Module”. When there is an incoming call to the Media Platform, control of the call is routed to the CCM instead of to a VoiceXML script. The CCM establishes a dialog with the Call Control Platform, using an internal “CCI” (Call Control Interface) protocol. The CCP uses the CCI protocol to control the call. For example, the CCP can select and launch VoiceXML scripts to handle the call, and can then initiate a call transfer when the VoiceXML script has completed.

The VoiceGenie ICM Connector is a protocol converter, between the CCI protocol, and the Cisco “ICM / VRU Interface” protocol which is supported by ICM. A “VRU” (Voice Response Unit) is Cisco’s terminology for a media server platform. A VoiceGenie CCP, and the Media Platforms to which it’s

connected, look like one VRU as far as ICM is concerned. The VoiceGenie CCP can drive one or more media platforms.

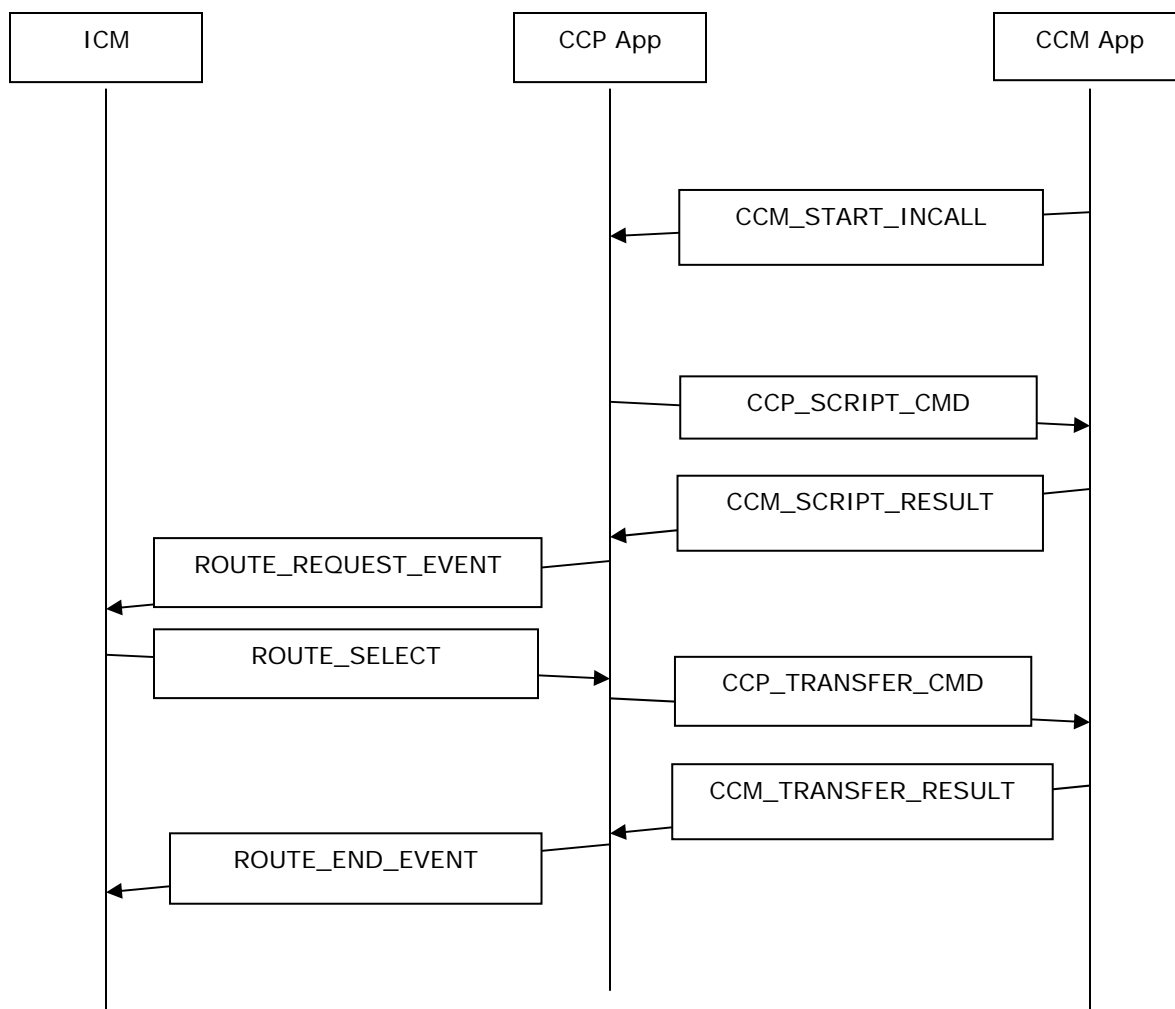
The Cisco ICM protocol supports two different interfaces:

- **“Routing” and “Event Data Feed” interface:** VoiceGenie can use these to tell ICM when a call is connected and disconnected, and to request a route (a transfer destination address) for a given call. Using these interfaces, it helps to think of the VRU as being the client, and the ICM as being a server: the VRU issues route requests, and gets route responses from ICM.
- **“Service Control” interface:** ICM can use this to tell VoiceGenie what VoiceXML scripts it should run, as well as when and where to transfer calls. Using this interface, it helps to think of ICM as being the client, and the VRU as being the server: the ICM issues requests to start a VoiceXML script or to transfer a call, which are obeyed by the VRU.

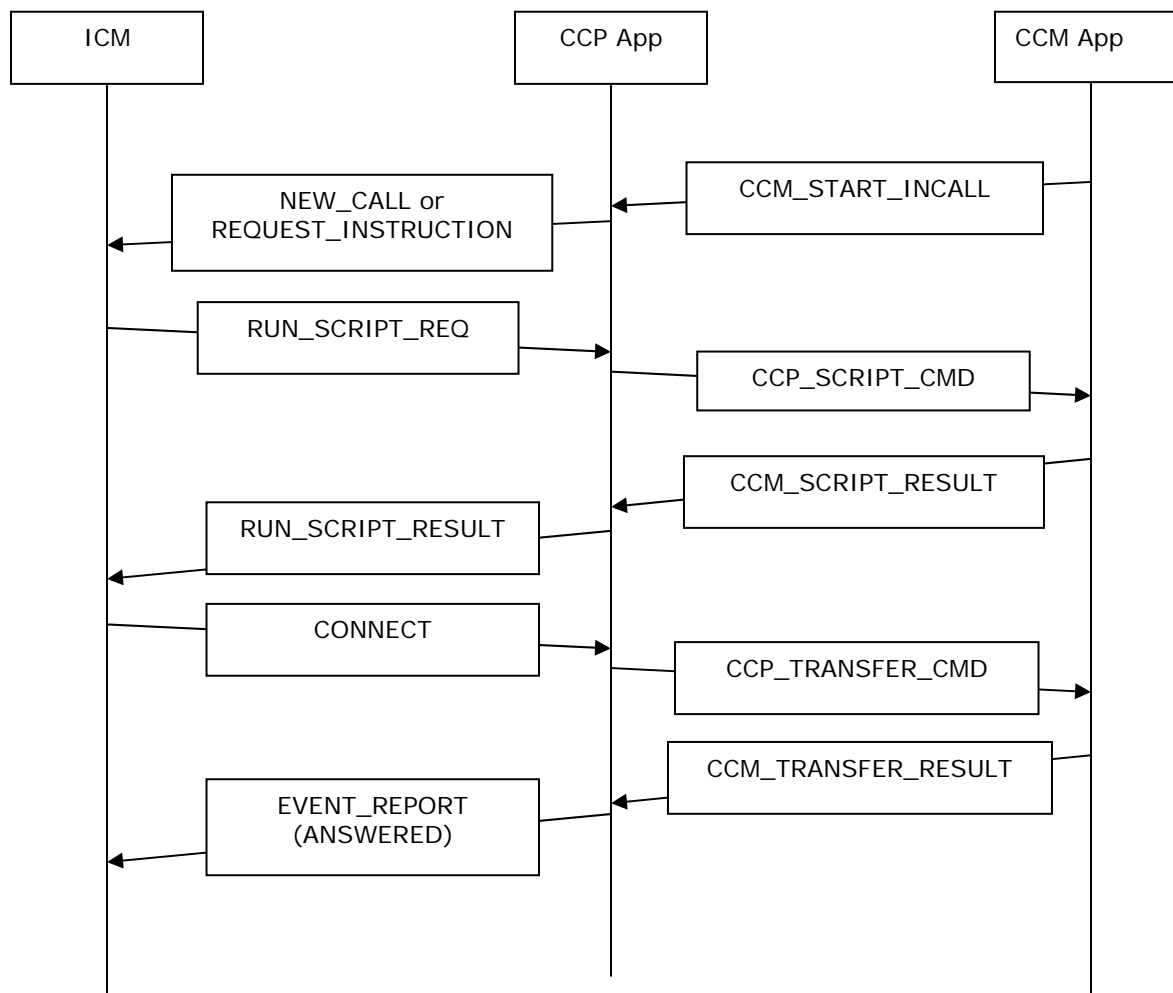
VoiceGenie supports both of these interfaces. However, at run-time ICM supports one or the other of these interfaces for a given VRU (you cannot use both interfaces simultaneously).

The following two diagrams illustrate typical call flow scenarios. On the right are CCI messages exchanged between the VoiceGenie CCP and CCM. On the left are ICM messages exchanged between ICM and CCP.

The first diagram shows call flow when the ICM “Routing” interface is being used.



The second diagram shows call flow when the ICM “Service Control” interface is being used:



Note that in both cases, the dialog is initiated by an incoming call being received by the media platform.

The above figures show the ‘message type’, i.e. the meaning of each message. Most messages carry additional data specific to that message, for example:

- The CCM_START_INCALL messages include fields that specify the call ID, the DNIS, the ANI, and so on;
- The CCP_SCRIPT_CMD message includes fields that specify which VoiceXML script should be run;
- The CCP_TRANSFER_CMD message includes a field that specifies the destination of the call transfer;

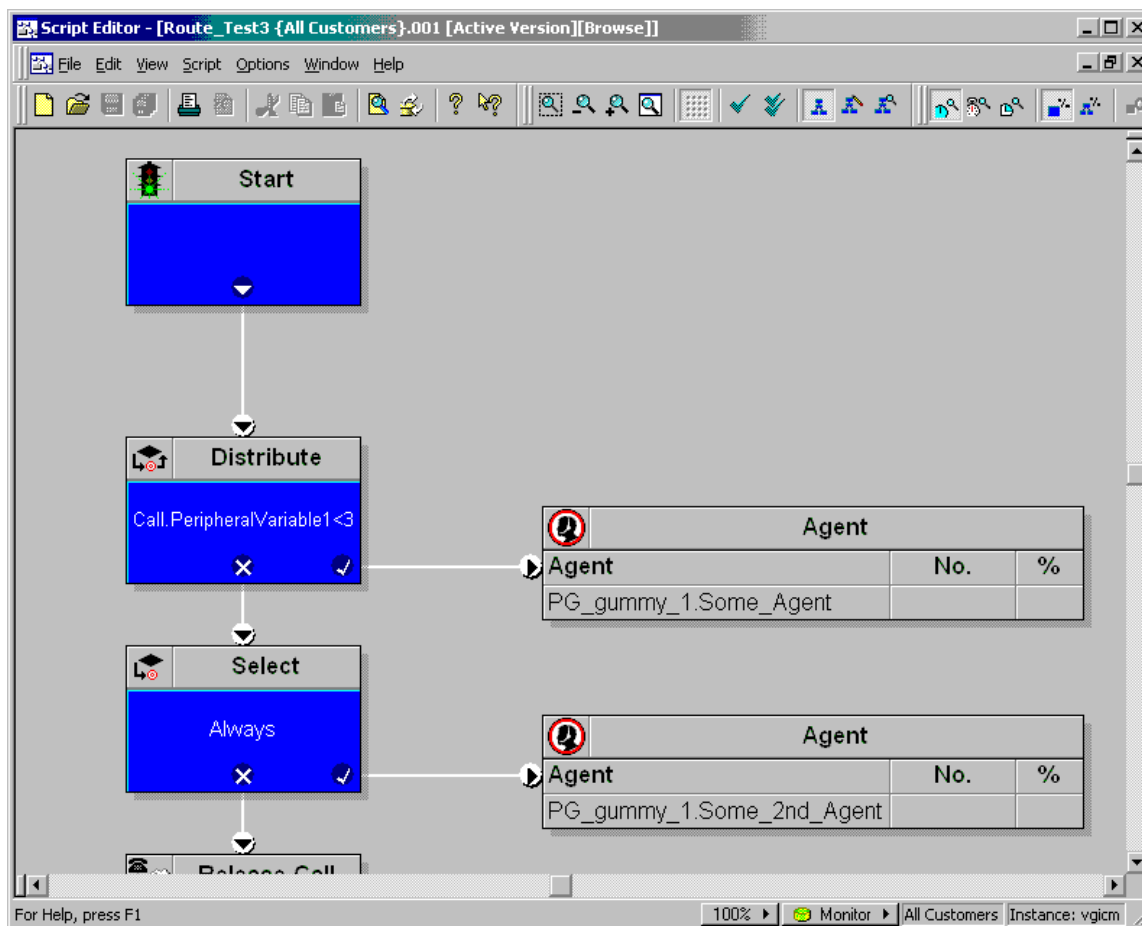
One type of data that is especially interesting is the ‘call variables’:

When a VoiceXML script is running it can capture data from the caller. This data is stored in VoiceXML script variables.

When the VoiceXML script exits, it can name some of these VoiceXML script variables in its <exit> tag. The specified variable names and values are then passed to the CCM (which had initiated the script). The CCM further returns these variable names and values to the CCP, in the CCP_SCRIPT_RESULT message.

The ICM VRU protocol allows up to ten 40-byte values to be carried in various messages, including the ROUTE_REQUEST_EVENT message (used in the “Routing” interface) and the RUN_SCRIPT_RESULT message (used in the “Service Control” interface). When ICM receives these values they are stored in ICM script variables (named “PeripheralVariable1” through “PeripheralVariable10”), which can be used in ICM scripts.

For example, the following shows an ICM script that uses a captured variable to decide which agent should be the target of a call transfer.



Entries in the CCP “ccp.cfg” file (described later) are used to specify which VoiceXML script variables are mapped to each of these 10 ICM variables.



Chapter

5

Access to ICM From VoiceXML

The ICM integration to the platform is virtually transparent to the application. The integration points are discussed in the following examples.

5.1 Passing Data to ICM

If a VoiceXML script contains an `<exit>` tag, and if the DNIS-URL mapping table has specified that the CCM is handling the call, then when the VoiceXML script exits, the call is not automatically disconnected. This means that the call is still connected, and available for a subsequent call transfer (when the ICM "Routing" interface is used), or for a subsequent script or subsequent transfer (when the "Service Control" interface is being used).

The VoiceXML page shown below demonstrates such a page.

```
<?xml version="1.0"?>
<vxml version="1.0">
  <meta name="application" content="Test"/>
  <form id="Welcome">
    <block name="Hello">
      <audio src="builtin:prompts/sting.vox"/>
      <audio src="builtin:prompts/welcome_to_voicegenie.vox"/>
      <exit/>
    </block>
  </form>
</vxml>
```

Caution

If the script does not contain an exit tag then the call will be disconnected when the script ends, and the ICM integration will therefore not work properly. This behavior is in order to maximize backward compatibility.

In the namelist attribute of the <exit> tag, it is possible to specify the names of VoiceXML script variables. These script variables may be mapped to ICM script variables (as shown in the illustration of the "ccp.cfg" file, earlier in this document). ICM can then use these values to make routing decisions.

The VoiceXML page shown below demonstrates a page providing a namelist in the <exit> tag.

```
<?xml version="1.0"?>
<vxml version="1.0">
  <property name="universals" value="none"/>
  <property name="documentfetchhint" value="safe"/>
  <property name="timeout" value="10s"/>
  <property name="bargein" value="false"/>
  <form id="myform">
    <field name="foo" type="digits?length=1" endbeep="true">
      <dtmf>1</dtmf>
      <prompt>
        Please enter a digit
      </prompt>
      <noinput><reprompt/></noinput>
      <nomatch><reprompt/></nomatch>
      <filled>
        <prompt>
          you entered <value expr="foo"/>
        </prompt>
      </filled>
    </field>
    <field name="bar" type="digits?length=1" endbeep="true">
      <prompt>
        Please enter a digit
      </prompt>
      <noinput><reprompt/></noinput>
      <nomatch><reprompt/></nomatch>
      <filled>
        <prompt>
          you entered <value expr="bar"/>
        </prompt>
        <exit namelist="foo bar"/>
      </filled>
    </field>
  </form>
</vxml>
```

The overall behavior when the ICM "Routing" interface is used is not configurable, and is described below:

- When there is an incoming call, the first thing the CCP does is request that the CCM start the VoiceXML script specified in the DNIS-URL mapping table.

- After the VoiceXML script ends, then (assuming that the call hasn't yet disconnected) the CCM will consult the ICM to make a call transfer decision.

The overall behavior when the ICM "Service Control" interface is more varied - however the variation is entirely encoded in the ICM script (to get a specific behavior, the appropriate ICM script much be written).

5.2 Asking ICM to Route a Call

When the ICM "Routing" interface is being used, the VoiceXML script may specify "icm" as the value of the dest attribute of the transfer tag. When a VoiceXML script that is controlled by the CCM issues such a transfer, the CCP issues a ROUTE_REQUEST_EVENT to ICM before the script has finished. When the VoiceXML script ends, the CCP will release the call (in fact, the call may already have pulled from the VoiceGenie platform by the ICM infrastructure, due to the route request). Note that in this case, processing of the <exit> tag will not trigger an ICM transfer request as this has already been done.

The same behavior can be triggered if the "destexpr" attribute evaluates to the string "icm".

When the ICM "Service Control" model is being used, the ICM script can transfer the call to the VoiceGenie platform, request that the platform play a script that emulates an on-hold condition while all agents are busy, and cancel the script (take back the call from the VoiceGenie platform, and then transfer the call to an agent) when an agent becomes available.

```
<?xml version="1.0"?>
<vxml version="1.0">
<meta name="application" content="Test"/>
  <form id="Welcome">
    <block name="Hello">
      <audio src="builtin:prompts/sting.vox"/>
      <audio src="builtin:prompts/welcome_to_voicegenie.vox"/>
    </block>
    <transfer name="newcall" dest="icm" connecttimeout="60s"
      bridge="true" connectwhen="immediate" analysis="true">
      <filled>
        <prompt>
          Your call lasted
          <value expr="newcall$.duration"/>
          seconds.
        </prompt>
        <if cond="newcall == 'busy'">
          All our customer care agents are
          currently busy, please call back later.
        </if>
      </filled>
    </transfer>
    <exit/>
  </form>
</vxml>
```




Chapter

6

Operations, Administration, and Maintenance

6.1 Starting and Stopping

Like all other VoiceGenie 7 components, the System Management Console offers a dedicated page to start or stop the ICM Connector. Click on the Operations tab and click on “Start/Stop Software” on the left hand column. Click on the server(s) that you want to start/stop and then click the Start/Stop button.

Note that “Suspending” an ICM Connector is not supported.

6.2 Health Status

Health status summary can be made available through the command line console (CLC) or the SMC. Individual health status items can be retrieved through SNMP traps.

In CLC, type: `health icm`

In SMC, go to “Monitoring” tab and click on the SIP Proxy item.

Here is a sample output:

```
Health for ICM Call Control (icm) on 10.0.0.192
Started: 2005-04-11/13:26:03.012
CCPICM status: NOT connected to ICM server
Apps: Current 0 Total 0
Run scripts: Current 0 Total 0
Transfers: Current 0 Total 0
Failure: 0
Media platforms(network-id:status): 5:not-connected
```

The following are explanations of each parameter in the health status summary:

- **Started** – The time which ICM Connector was started;
- **CCPICM Status** – reports the connection state between ICM connector and ICM server;
- **Apps** – reports the number of applications running and total number of applications executed;
- **Run scripts** – reports the number of Run script command is executing and the total number of run scripts executed;
- **Transfers** – reports the current number of transfers and total number of call transfers;
- **Failure** – reports the total number of failures;
- **Media Platforms** – reports the status of the Media Platforms that are provisioned to the ICM connector. The network-id is the CMP assigned network ID of the Media Platform.

The health status parameters are available through SNMP get. The complete list of parameters and description is listed in Appendix A. Please also refer to the OA&M Framework User Guide for details on retrieving the information.

6.3 Logging

ICM Connector generates logging information using the VoiceGenie OA&M Framework. All logs of level Critical (LOG_0), Error (LOG_1), and Warning (LOG_2) sent upstream and to the log file. The default log file is in /usr/local/ccp-proxy/log/CMP.log.ccpicm.

Log levels for Notice (LOG_3) and Information (LOG_4) are stored in the log file.

Trace logs (LOG_5) is disabled by default. Trace is not recommended for deployment environment as trace will flood the trace files quickly and decreases system performance.

To enable trace, go to ICM Configuration and select false for cmp.trace_flag. Click Update to submit the configuration change.

6.4 Cluster Management

Details for how to create and administer Clusters can be found in section 3.3.1 of the following document:

VoiceGenie 7 OA&M Framework – SMC Guide



Chapter

7 Directory Structure

ICM Connector home directory will reside in /usr/local/ccp-icm. The following table lists the subdirectories/files and their description:

File (relative to ICM Connector home):	Description
bin/ccpicm	ICM Connector executable
config/ccpicm.cfg	Local ICM Connector configuration file
config/ccpicm_provision.dat	Local ICM Connector provisioning file
config/ccp-icm.xml	ICM Connector definition file
logs/CMP.log.ccpicm*	ICM Connector log files



Appendix

A Health SNMP Gets

Using SNMP Get, a number of health parameters about the VoiceGenie software are retrievable. This section outlines what health information can be retrieved for ICM Connector.

The name prefix is
 “.iso.org.dod.internet.private.enterprises.vg.voiceXMLGateway.vgData.sippScalarTable.sippScalarTableEntry.”

Name	OID	Type	Description
CCPICM-HEALTHDATA-STARTED	.1.3.6.1.4.1.7469.3.9.22.1.50	Scalar	CCPICM Start Time
CCPICM-HEALTHDATA-APPS	.1.3.6.1.4.1.7469.3.9.22.1.51	Scalar	Current Applications
CCPICM-HEALTHDATA-APPSTOTAL	.1.3.6.1.4.1.7469.3.9.22.1.52	Scalar	Total Applications
CCPICM-HEALTHDATA-RUNSCRIPTS	.1.3.6.1.4.1.7469.3.9.22.1.53	Scalar	Current Run Scripts
CCPICM-HEALTHDATA-RUNSCRIPTSTOTAL	.1.3.6.1.4.1.7469.3.9.22.1.54	Scalar	Total Run Scripts
CCPICM-HEALTHDATA-TRANSFERS	.1.3.6.1.4.1.7469.3.9.22.1.55	Scalar	Current Transfers
CCPICM-HEALTHDATA-TRANSFERSTOTAL	.1.3.6.1.4.1.7469.3.9.22.1.56	Scalar	Total Transfers
CCPICM-HEALTHDATA-FAILURES	.1.3.6.1.4.1.7469.3.9.22.1.57	Scalar	Failures
CCPICM-HEALTHDATA-ICMSTATUS	.1.3.6.1.4.1.7469.3.9.22.1.58	Scalar	CCICM Status
CCPICM-HEALTHDATA-MP	.1.3.6.1.4.1.7469.3.9.23.1.55	Tabular	Media Platform Network ID
CCPICM-HEALTHDATA-MPSTATUS	.1.3.6.1.4.1.7469.3.9.23.1.60	Tabular	Media Platform Status



Appendix

B Logging Traps

The OID prefix is “.1.3.6.1.4.1.7469.251.1.304”. To get the OID of a trap, just append the prefix with the suffix column. For example, VGLOG-CCPICM-ICM-INITIALIZATION-FAILED has an OID of .1.3.6.1.4.1.7469.251.1.304.150995200.

B.1 Severity – Critical (LOG_0)

Name	OID suffix	Response Code	Impacts	Causes	Recommended Actions	Message
VGLOG-CCPICM-ICM-INITIALIZATION-FAILED	150995200	CKCFG CKHW	Fail to start ccp-icm	Incorrect configuration	Check CCP-ICM configuration and provisioning	Initialization failure

B.2 Severity – Error (LOG_1)

Name	OID suffix	Response Code	Impacts	Causes	Recommended Actions	Message
VGLOG-CCPICM-CCI-UNEXPECTED-STATE	153092614	CKAPP CKCFG CKHW	Connection with Media Platform may not be working	Incorrect protocol state between ccp-icm and Media Platform	Restart ccp-icm and check configuration	Unexpected CCI State
VGLOG-CCPICM-ERROR-TRANSFER-DATA	153092615	NOTE CKAPP CKCFG CKHW	Fail to send messages to ICM server	Network issues	Check network connections	Transfer data
VGLOG-CCPICM-ICMScript-NOTFOUND	153092616	CKAPP CKCFG	Cannot start script; drops the call	Fail to find script given script ID	Check provisioning and make sure there is script mapping for the given script ID	Script not found
VGLOG-CCPICM-ICM-UNEXPECTED	153092617	CKAPP REVG	Unexpected event but no impact on calls	Incorrect timer events	Restart ccp-icm	Unexpected timeout
VGLOG-CCPICM-INVALID-CONFIG	153092609	CKCFG	Invalid configuration parameters or provisioning entries are not used	Invalid values entered	Check ccp-icm configuration and provisioning	Invalid configuration or provision
VGLOG-CCPICM-INVALID-MSG	153092610	CKAPP CKCFG	Incorrect application behaviour; may lose call	Received unexpected message	Check ICM application and make sure ICM configuration is correct	Message invalid or contains unexpected parameter
VGLOG-CCPICM-MP-	153092613	CKAPP CKCFG	Inconsistent state between	Incorrect configuration	Check operation	Unexpected Media

Name	OID suffix	Response Code	Impacts	Causes	Recommended Actions	Message
UNEXPECTED-STATE		CKHW	ccp-icm and media platform		state of Media Platform and configuration	Platform State
VGLOG-CCPICM-UNEXPECTED-RESULT	153092611	CKCFG	Inconsistent trunk view between ccp-icm and media platform	Incorrect ccp-icm trunk provisioning	Check ccp-icm provisioning	Unexpected response for sent messages
VGLOG-CCPICM-UNEXPECTED-STATE	153092612	CKAPP CKCFG	Inconsistent state between ccp-icm and media platform	Incorrect application or configuration	Check voicexml application and media platform configuration	Unexpected ICM, Dialog or Call State

B.3 Severity – Warning (LOG_2)

Name	OID suffix	Response Code	Impacts	Causes	Recommended Actions	Message
VGLOG-CCPICK-DIALOG-DELETED	153092868	NOTE	Discard message against the dialog	Call has already been disconnected	N/A	Receiving message on deleted dialog; message discarded
VGLOG-CCPICK-WARNING-CCI	153092865	NOTE	Disconnect from media platform; will reconnect	Media Platform may not be responding	Check Media Platform operation state	CCI connection failure
VGLOG-CCPICK-WARNING-DIALOG-FAILURE	153092866	NOTE	Drops call	ICM server sent the request to drop the dialog	N/A	ICM Dialog failure
VGLOG-CCPICK-WARNING-ICMAPP	153092867	CKAPP CKCFG	Application error; may lose call	Application is sending incorrect events to ccp-icm	Check application	Application failure

Revision History

Version	Date	Change Summary	Author/Editor
1.0	November 11th, 2003	Initial release	Chris Wells Rob Marchand
1.1	March 3rd, 2005	Updated for VoiceGenie 7	Andrew Ho
1.2	April 11th, 2005	Added OA&M information	Henry Lum
1.3	April 13th, 2005	Final revision for VoiceGenie 7	Andrew Ho
	May 27, 2008	Converted to Genesys Word Template	Wasser Studios