VOICEGENIE

VoiceGenie 7 CCXML Platform

Installation and Configuration Guide

June 2005

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

VoiceGenie
TECHNOLOGIES INC

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

VoiceGenie
TECHNOLOGIES INC

Proprietary / Copyright Information

Notice

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

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

Trademarks

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

Table of Contents

Revision History

Version	Date	Change Summary	Edited By
1	2005/03/28	Initial release	Henry Lum
2	2005/05/20	Updates	Henry Lum
3	2005/06/15	Updates	Henry Lum

1.0 Introduction

This manual details the process for installing and configuring the VoiceGenie 7 CCXML Platform.

VoiceGenie CCXML Platform provides a CCXML interpreter that integrates with existing VoiceGenie infrastructure such as the Media Platform and SIP Proxy. The underlying network protocol for CCXML Platform is SIP; this means that CCXML Platform can interoperate with other conferencing server or dialog server.

Although VoiceGenie has traditionally provided extended call control capabilities through proprietary extensions to VoiceXML, the development of Call Control XML (CCXML) provides a standard, XML-based language for scripting call control logic. Like VoiceXML, CCXML is independent of the environment in which it operates, and can run in environments ranging from VOIP-based softswitch products to integrated residential gateways that manage a single telephone call.

1.1 Compatibility and Support

CCXML Platform follows the W3C Working Draft dated January 11, 2005.

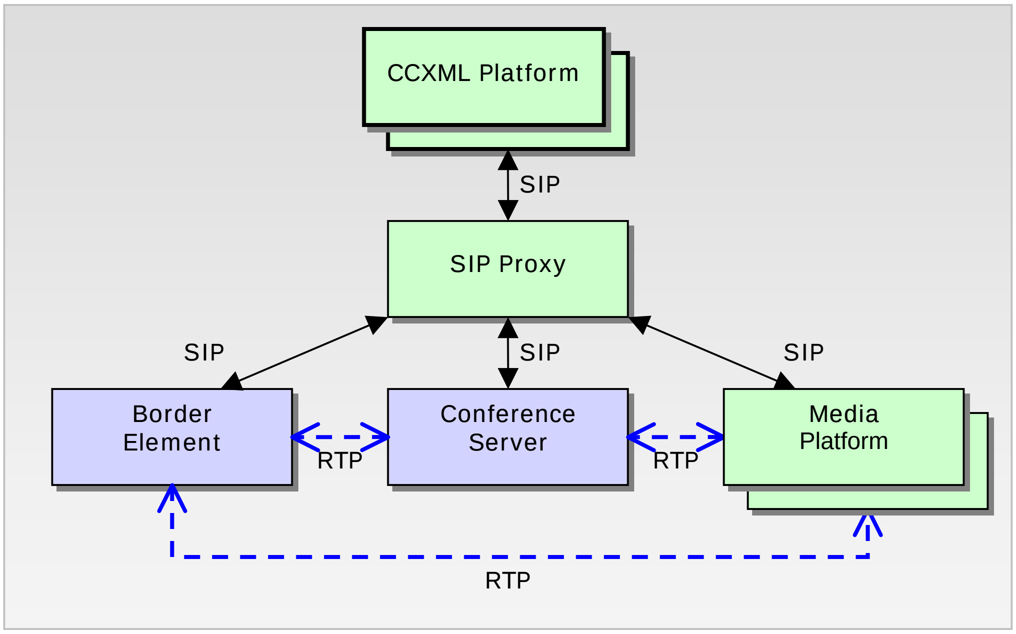
VoiceGenie CCXML Platform is designed to work with VoiceGenie 7 Media Platform and SIP Proxy. Older versions of the Media Platforms are not supported.

2.0 System Architecture

The CCXML Platform can be installed in a clustered environment or an all-in-one environment.

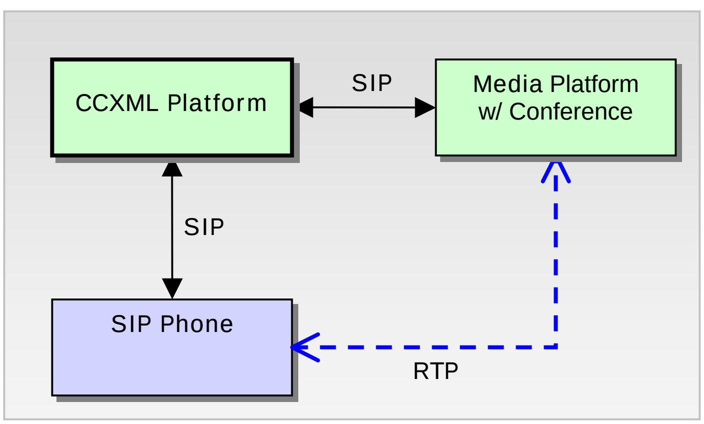
2.1 Clustered Environment

In a clustered environment, SIP Proxy manages multiple instances of CCXML Platforms and Media Platforms. Border elements such as media gateway send SIP Requests to the SIP Proxy and the requests are forwarded the appropriate SIP services, such as Media Platform or conference server. The following diagram shows elements managed by the SIP Proxy:



2.2 All-in-one Environment

The minimal configuration for CCXML does not require SIP Proxy if Media Platform assumes the role of both dialog server and conference server. CCXML Platform will always use the same Media Platform to prepare dialogs and create conferences. SIP Proxy can be installed in this environment but it is optional. The following diagram shows the configuration:



3.0 Prerequisites

For distributed installation, CCXML Platform requires the following components to be installed and configured:

1) VoiceGenie OA&M framework
2) VoiceGenie Media Platform (SIP enabled)

Optional Components:

3) VoiceGenie SIP Proxy (includes Redundancy Manager and Linux bonding driver)

For all-in-one installation, CCXML Platform requires the following components to be installed and configured:

1) VoiceGenie OA&M framework
2) VoiceGenie Media Platform with SIP and Conference enabled

Optional Components:

3) VoiceGenie SIP Proxy (includes Redundancy Manager and Linux bonding driver)

3.1 Deployment Order

After installing OA&M framework, please deploy the rest of the components in the following order:

i) SIP Proxy
ii) Media Platform (plus ASR/TTS)
iii) CCXML Platform

CCXML Platform must be the last component to be deployed.

Please refer to their respective installation and configuration guides to install the above components. This document also provides additional configuration instructions for Media Platform and SIP Proxy to integrate CCXML Platform into the cluster.

For convenience, the following subsection provides abbreviated installation instructions for SIP Proxy. For complete installation instructions, please refer to the SIP Proxy User's Guide.

3.2 SIP Proxy Installation

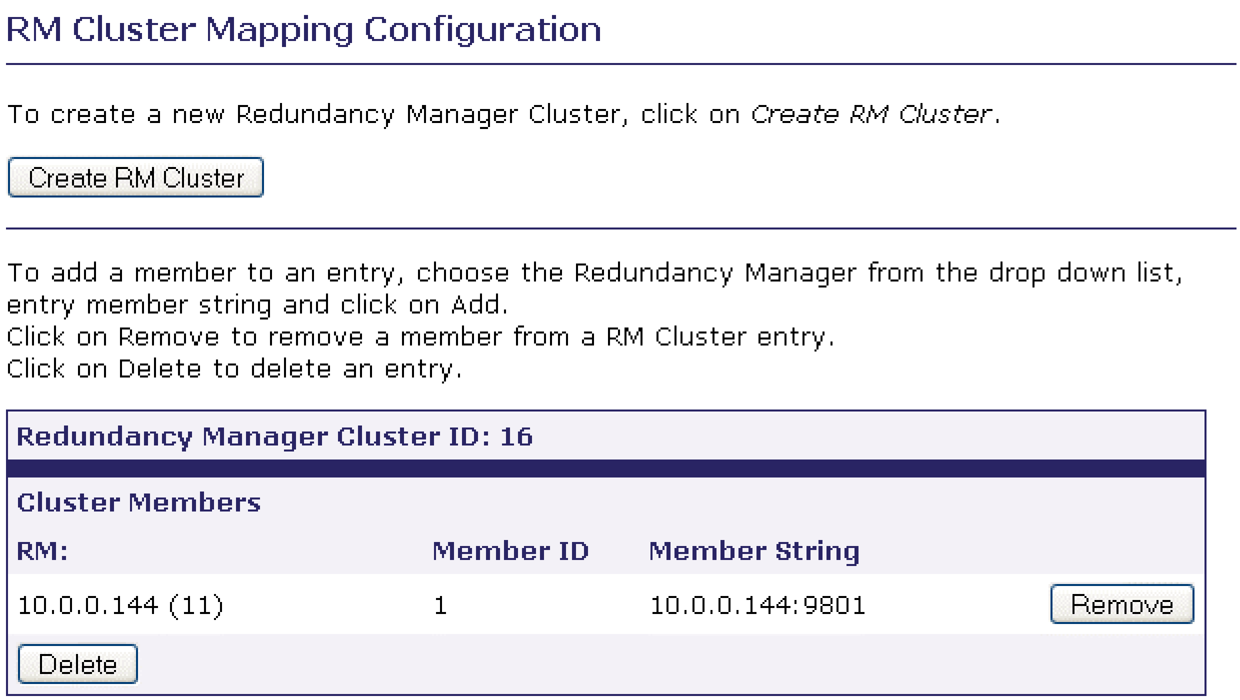
Required files:

Component	Filename
Linux Bonding Driver (Linux only)	bonding-7.0.0-3.tar.gz
CCP Redundancy Manager	ccp-rm-7.0.1-x.tar.gz
SIP Proxy	ccp-proxy-7.0.1-x.tar.gz

Installation Steps:

2) Open System Management Console homepage and find Product Manager. Upload the required files.
3) Create configuration profiles in "Config Profile Manager" for each component listed above.
4) Use Deployment manager to deploy each component in the order listed in the above table. Select the machine(s) that SIP Proxy will be deployed on and ensure all 3 components are deployed.
5) Check Deployment History to make sure the product is deployed successfully.
6) Apply license keys to /usr/local/phoneweb/config/vglicense.txt; create the /usr/local/phoneweb/config directory manually if the directory does not already exist.

7) Set up Redundancy Manager Cluster Mapping. Go to Configuration tab and click on "RM Cluster Mapping".



Click on "Create RM Cluster" and a new box with a cluster will be shown. For each SIP Proxy machine, you will find the IP address available in a dropdown box. Enter the IP address with **port 9801** and click "Add". The resulting screenshot with one SIP Proxy is shown above.

8) Instructions to configure SIP Proxy depend on the system architecture. Please go onto the next section to install and configure CCXML Platform where instructions to configure SIP Proxy are provided.

4.0 CCXML Platform Installation

Required files:

Component	Filename
CCXML Platform	ccp-ccxml-7.0.0-x.tar.gz

Installation Steps:

1) Product Manager

Open a web browser and go to the System Management Console homepage;
Login and click on "Configuration" tab;
On the left hand column click on "Product Manager";
Click Browse and select one of the filenames listed above and click OK;
Click Upload to upload the component to Product Manager.

2) Config Profile Manager

In System Management Console, go to “Configuration” tab;
On the left hand column click on “Config Profile Manager”;
At the Product drop-down list, select “CCXML Platform – AS3.0-CCXML – 7.0.0”;
Enter a name for this profile;
Click Create, and you will notice the profile name will show up at the table below;

3) Deployment Manager

In System Management Console, go to “Configuration” tab;
On the left hand column click on “Deployment Manager”;
Click on the radio button that corresponds to “CCXML Platform”;
Select the configuration profile from the drop-down list that corresponds to this row;
Select the server(s) where CCXML Platform will be deployed;
Click Deploy button to start deploying;
Check Deployment History to make sure the product is deployed successfully;

4) Apply License keys

Please collect CCXML license keys from VoiceGenie. Please place the license file named vglicense.txt into /usr/local/phoneweb/config/vglicense.txt. If the directory does not already exist (ie. Media Platform is not installed on the machine), please create the directory manually and place the license file.

To verify the validity of the license key, please enter:

```
pw@ccxml > /usr/local/ccp-ccxml/bin/ccpccxml -l
VoiceGenie CCXML Platform
Build $Name:  $
Compiled by g++ (GCC) 3.2.2 Mon Mar 28 14:19:24 EST 2005
License Validation:
License file /usr/local/phoneweb/config/vglicense.txt is valid
License Content:
vggateway       in,out   Thu Dec 31 00:00:00 2037        0
vggateway       asr      Thu Dec 31 00:00:00 2037        0
vggateway       tts      Thu Dec 31 00:00:00 2037        0
ccxml   connection       Thu Dec 31 00:00:00 2037        5000
```

The summary shows that there are 5000 licenses available on the CCXML Platform. This license key allows up to 5000 concurrent connections, dialogs, and conference participants.

Please contact licenses@voicegenie.com if you do not have an appropriate license key.

5) Configure CCXML Platform

For all-in-one configuration, please read section 5.1
For all-in-one configuration with SIP Proxy, please read section 5.2
For distributed configuration, please read section 5.3

6) Configure Media Platform; please read section 6 to configure Media Platform

7) Configure SIP Proxy; please read section 7 to configure SIP Proxy

8) Reboot the server

5.0 Configuring CCXML Platform

5.1 All-in-one installation (CCXML Platform and Media Platform on same machine)

After deploying both CCXML Platform and Media Platform, please modify the following configuration parameters to set up all-in-one installation with CCXML Platform and Media Platform running on the same machine.

CCXML Platform
1) *ccpccxml.sip.proxy* – change to "**127.0.0.1:5062**"

Call Manager (Media Platform)
1) *sip.transport.0* – enable and change to "**transport0 udp:any:5062**"
2) Please also follow the rest of configuration instructions for Media Platform in *Configuring Media Platform*.

5.2 All-in-one installation (CCXML Platform, SIP Proxy and Media Platform on same machine)

After deploying CCXML Platform, SIP Proxy, and Media Platform, please modify the following configuration parameters from SMC to set up all-in-one installation.

CCXML Platform
1) sip.transport.0 – change to "**transport0 udp:any:5068**"
2) *ccpccxml.sip.proxy* – change to IP address of this machine

Call Manager (Media Platform)
1) *sip.transport.0* – enable and change to "**transport0 udp:any:5066**"
2) Please also follow the rest of configuration instructions for Media Platform in *Configuring Media Platform*.

SIP Proxy
1) *proxy.sip.proxy.respaddr* – add the IP addresses of this machine
2) Please also follow the rest of the SIP Proxy configuration instructions in *Configuring SIP Proxy*.

5.3 Distributed Installation

In a distributed environment, SIP Proxy can be used to handle multiple instances of Media Platform and other devices as the conference server. Please modify the following configuration parameters for each component listed below.

CCXML Platform
1) *ccpccxml.sip.proxy* – change to the IP address of the SIP Proxy

Call Manager (Media Platform)

1) Please also follow the rest of configuration instructions for Media Platform in Configuring Media Platform.

SIP Proxy
1) *proxy.sip.proxy.respaddr* – add the IP addresses of the SIP Proxy machine
2) Please also follow the rest of the SIP Proxy configuration instructions in Configuring SIP Proxy.

6.0 Configuring Media Platform

Call Manager requires the following configuration changes in order to integrate with the CCXML Platform. To change the configuration, click on Call Manager under SMC->Configuration tab.

sessmgr.modules = VXML Remdial Conference
sessmgr.appmodules = VXML:VXML Remdial:RemoteDial Conference:Conference
sessmgr.conference.conference = Conference
sip.transfermethods = REFER
sip.defaultblindxfer = REFER
sip.referxferhold = 0 (please enable this parameter by clicking the checkbox besides this parameter)

Click Update button to apply the changes. Media Platform must be restarted in order for changes to take effect.

7.0 Configuring SIP Proxy

SIP Proxy must be configured with proper mapping in order for SIP Proxy to correctly forward SIP requests to and from Media Platform and CCXML Platform. There are 4 tables defined by the SIP Proxy that will need to be modified; they are described in detail in the SIP Proxy System Reference Manual:

1) **SIP Service Table** defines all available SIP services within the cluster. Load balancing scheme can be defined for each SIP Service.

2) **SIP Services Mapping** Table defines a set of rules that maps incoming requests to a SIP Service. This table can also translate SIP Request URI based on regular expression rules.

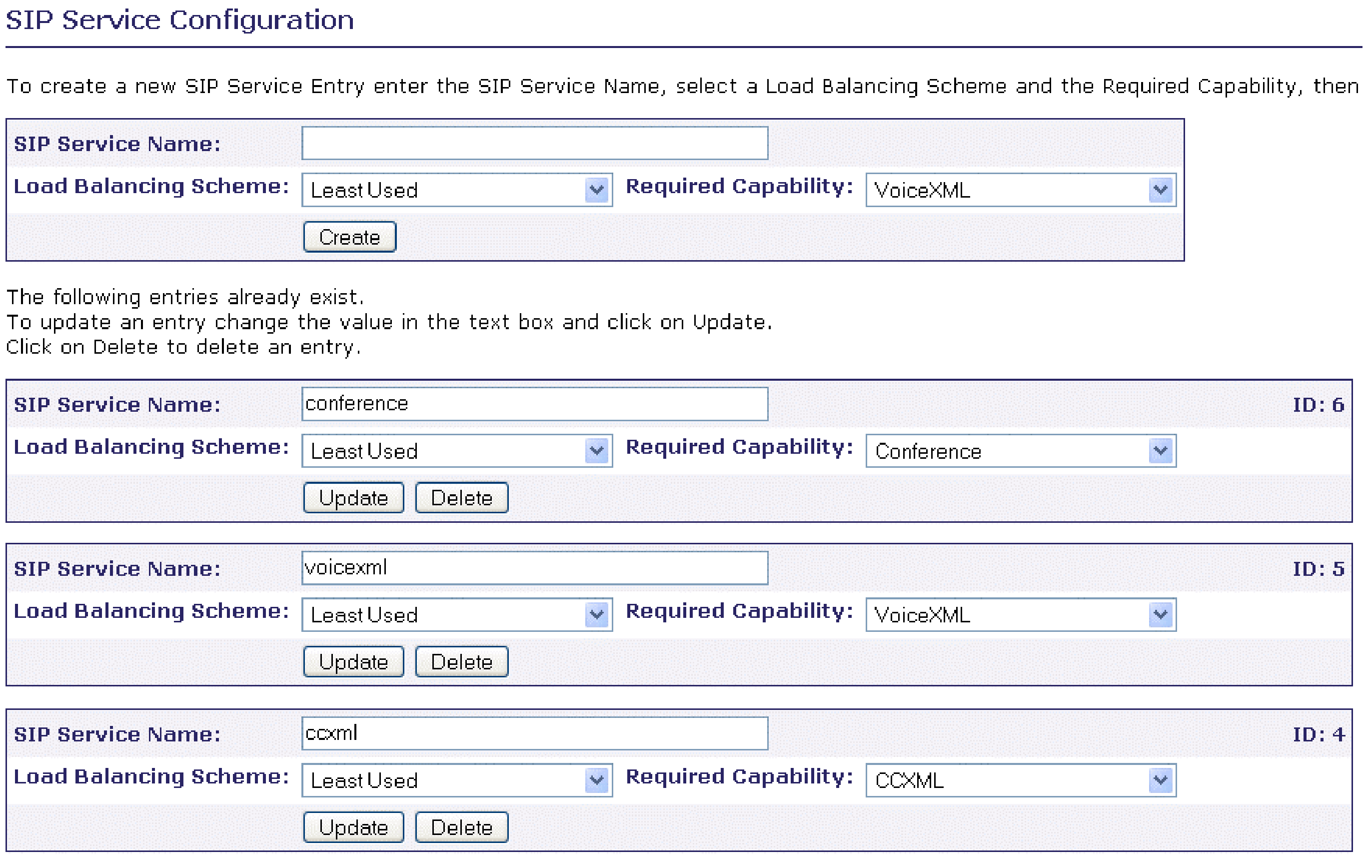
3) **SIP Resource Types Table** defines a template for SIP resources that share common attributes.

4) **SIP Resources Table** defines the list of SIP resources. Each SIP Resource provides one or more SIP Service and each SIP Resource belongs to only one SIP Resource Type.

Here are the step-by-step instructions for defining the table entries for CCXML configuration:

1) Define SIP Services

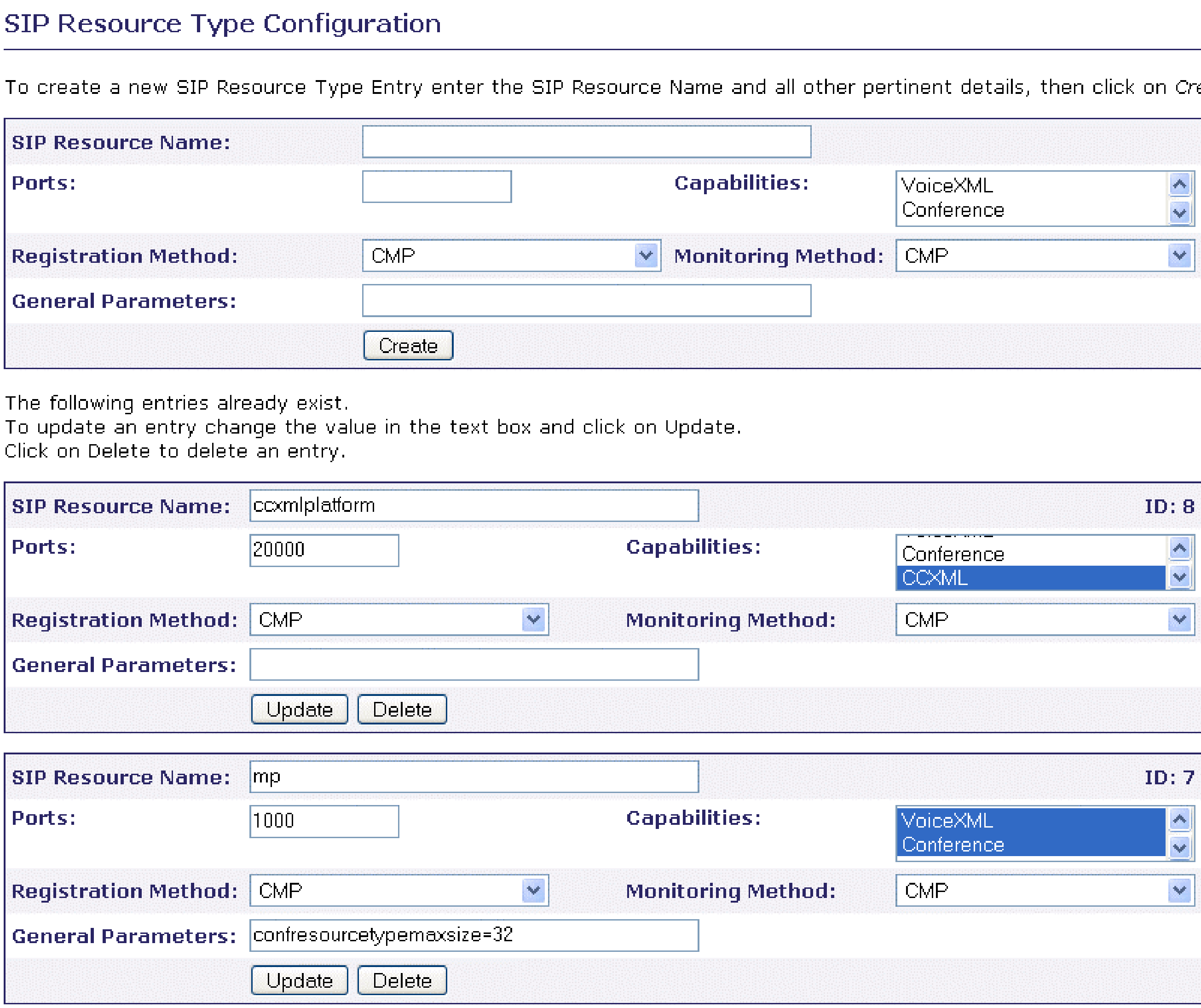
Define 3 SIP Services called ccxml, mp, and conference.



2) Define SIP Resource Types

There are only two types of SIP Resources: CCXML Platform and Media Platform. Set the number of ports available on each SIP Resource Type. The number of ports should be equivalent to the number of licenses for the CCXML or Media Platform.

For Media Platform, if conference is enabled, please add "confresourcetypemaxsize=32" into the General Parameters field as shown in the diagram below (replace 32 with the number of ports you are licensed for).

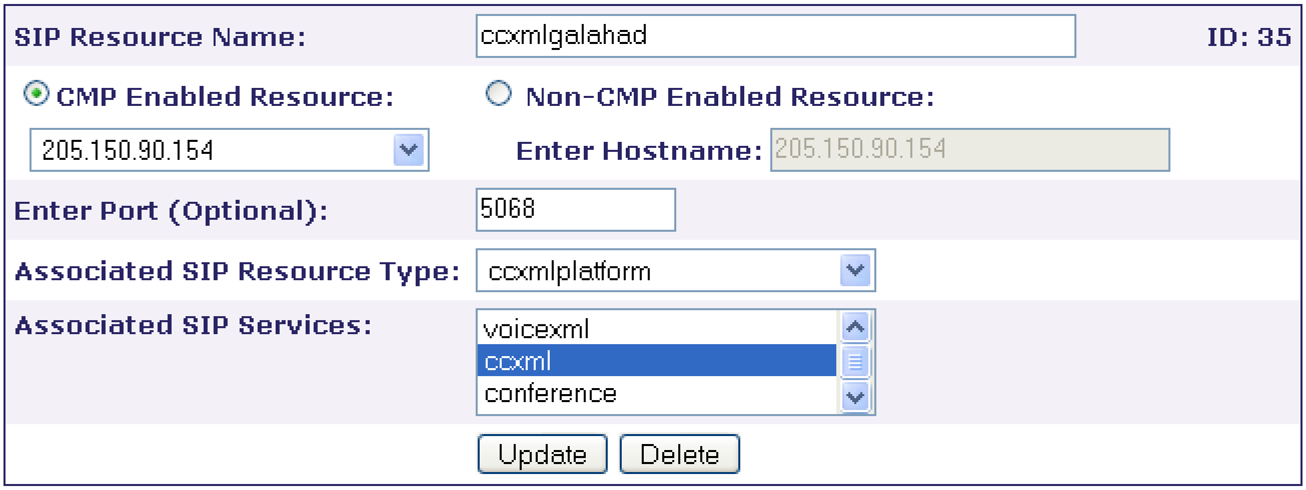


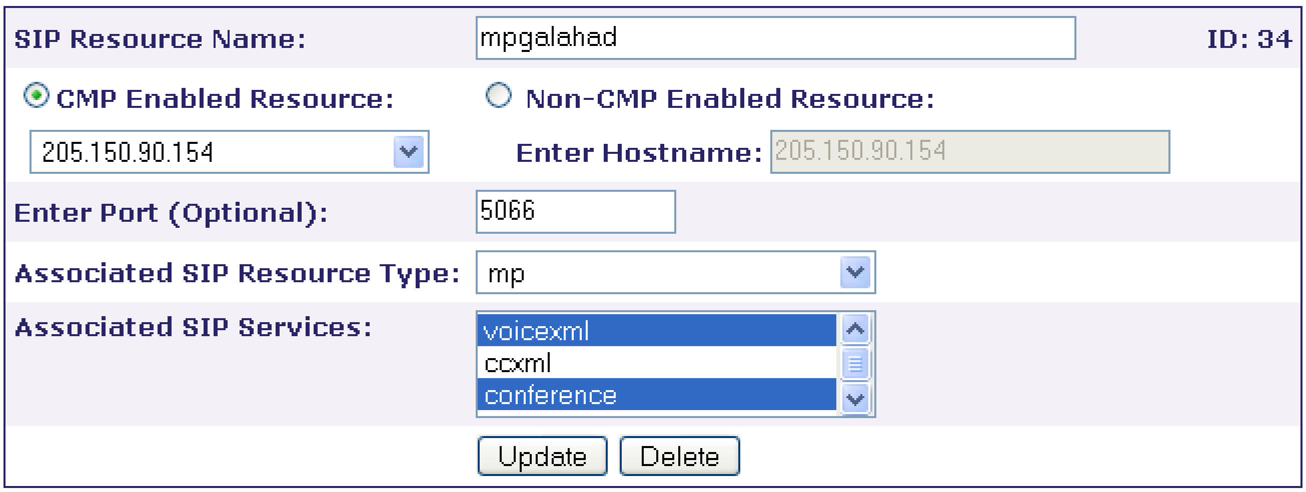
3) Define SIP Resources

For each SIP Resource (CCXML Platform or Media Platform) installed in the cluster, create a SIP Resource entry that identifies the SIP port number and the SIP Services that it provides.

For CCXML Platforms, select “ccxmlplatform” as the SIP Resource Type and “ccxml” as the SIP Service. For Media Platforms, select “mp” as the SIP Resource Type and “voicexml,conference” as the SIP Services (hold on the Shift key to select multiple SIP Services).

The following diagram assumes CCXML Platform, Media Platform, and SIP Proxy are running on the same machine (all-in-one).





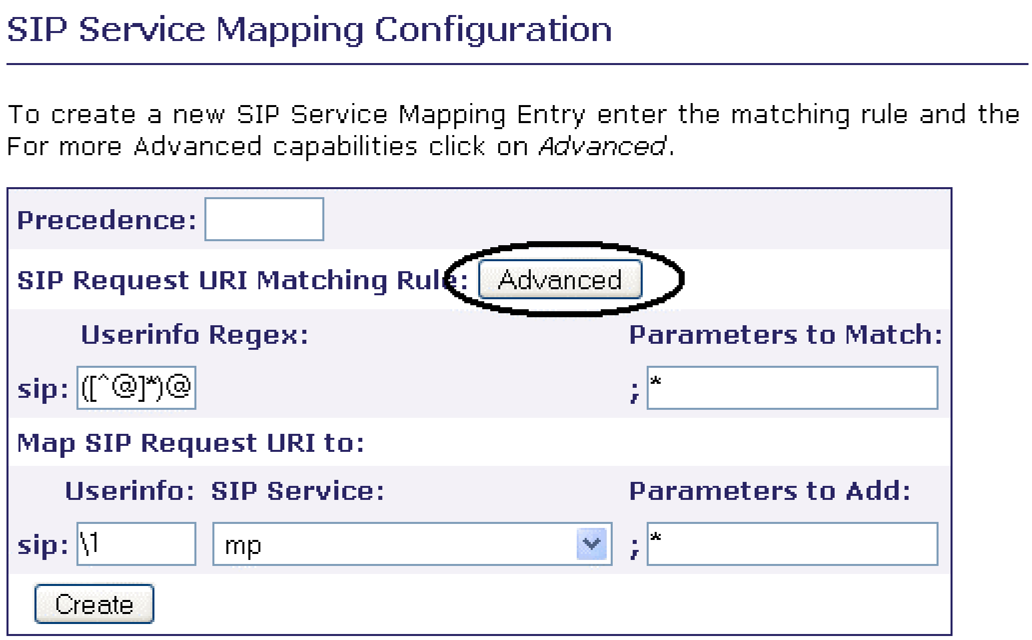
Note about All-in-one installation

For All-in-one installation, there will be two entries with identical names under CMP Enabled Resource. Please select the first one for Media Platform and select the second identical entry for CCXML Platform.

For a distributed configuration, the port numbers should be 5060.

4) Define SIP Service Mapping

SIP Proxy can forward requests from CCXML Platform to a VoiceXML dialog or a conference by mapping SIP request URI. Please click on the circled button to open the Advanced Configuration:



Advanced Configuration provides a text box to enter mapping rules.

We are adding 3 rules in order:

1) Forwards SIP requests from CCXML Platform to a VoiceXML service. The SIP Request URI start with sip:dialog*.
2) Forwards SIP requests from CCXML Platform to a conference service. The SIP Request URI starts with sip:conf*.
3) Forwards any SIP requests from any external SIP device to the CCXML Platform. The CCXML Platform will load the page file:///usr/local/ccp-ccxml/config/default.ccxml.

Click Select Target on each rule and target to the SIP Proxy.

The resulting rules should look like the following screenshot:

SIP Service Mapping Entry ID: 14

```
3
external
sip:([^@]*)@*
*
ccxml
*
sipservice=ccxml ccxml=file:///usr/local/ccp-ccxml/config/default.ccxml *
```

Update | Delete | Select Target

SIP Service Mapping Entry ID: 13

```
2
ccxml
sip:conf([^@]*)@*
*
conf\1
*
sipservice=conference *
```

Update | Delete | Select Target

SIP Service Mapping Entry ID: 12

```
1
ccxml
sip:dialog([^@]*)@*
*
dialog\1
*
sipservice=voicexml *
```

Update | Delete | Select Target

The SIP Service Mapping rules can be customized needed. Please talk to your system architect should there be any question about the correct rules which need to be installed.

8.0 Appendix A – CCXML Platform Configuration Parameters

The CCXML Platform configuration can be found in SMC->Configuration->CCXML Interpreter.

8.1 CCXML Platform Configuration Parameters

The following table describes the configuration parameters in the CCXML Platform. These parameters affect the platform behavior regarding SIP messages sent by the CCXML Platform.

CCXML Platform Configuration	
ccpccxml.sip.proxy	The address of SIP Proxy for outbound SIP requests. **This is a required parameter.**
ccpccxml.sip.send_progressing	Determines whether 180 SIP response is sent on <accept> tag for all incoming calls. Possible Values: 0 - 180 response is sent when <accept> is called 1 - 180 response is sent immediately after sending 100 Trying Default: 0
ccpccxml.default_uri	Sets the default URI for new CCXML sessions on incoming calls. This is a required parameter. Default: file:///usr/local/ccp-ccxml/config/default.ccxml
ccpccxml.allow_dialog_transfer	Flag to allow or disallow dialog-initiated transfer using <transfer> tag. Possible Values: 0 - disallow dialog-initiated transfer 1 - allow dialog initiated transfer Default: 0
ccpccxml.dialog.URI_scheme	Select a SIP Request URI scheme where CCXML platform uses to initiate dialogs. Possible values: netann, rosenburg Default: Rosenberg
ccpccxml.conference.URI_scheme	Select a SIP Request URI scheme where CCXML platform uses to initiate conferences. Possible values: netann, simple Default: netann
ccpccxml.conference.defaultreserve	Defines the default size of the conference for reserving the number of participants. This size of the

	conference can be defined in the CCXML application through the tag. Default: 3
ccpccxml.sip.nullsdp	Defines the content of the "c=" line in SDP content that represents media pause. A non-default value may be required on some SIP devices that pauses both directions of the media stream when it receives 0.0.0.0 in this line. For these SIP devices, choose an IP address that is not routable by the external SIP devices, such as "IN IP4 1.1.1.1". Default: IN IP4 0.0.0.0

8.2 CCXML Interpreter Configuration

In the same configuration page, you will find the following configuration parameters that affect interepter behaviour.

CCXML Interpreter Configuration	
ccxmli.max_num_documents	The maximum number of active and cached documents in CCXMLI. Default: 1000
ccxmli.num_session_processing_threads	The number of threads that perform session operations. Default: 10
ccxmli.max_num_sessions	The maximum number of active sessions in CCXML interpreter. Default: 1000
ccxmli.trace_flag	This flag determines tracing/debugging should be logged. Possible values: TRUE, FALSE Default: FALSE
ccxmli.kill_by_other	CCXML applications can kill other CCXML applications by sending ccxml.kill.unconditional or ccxml.kill events. Possible values: TRUE, FALSE Default: FALSE

ccxmli.max_conn_per_session	The maximum number of connections can be owned by a session. Default: 100
ccxmli.max_dialog_per_session	The maximum number of dialogs can be owned by a session. Default: 100
ccxmli.max_conf_per_session	The maximum number of conferences a session can be attached to. Default: 100
ccxmli.ioproc.http.url	The URL to expose to receive HTTP event injections. Note that the portion is ignored by the software, and is to be filled in automatically Default: http://<IP-PORT>/ccxml/ioproc
ccxmli.ioproc.http.port	The port on which the HTTP IO Processor will wait for HTTP connections and receive events. Default: 4892
ccxmli.ioproc.http.maxclients	The maximum number of simultaneous HTTP requests to be handled by the HTTP IO Processor. Default: 200
ccxmli.send_to_external_timeout	The timeout value for an event sent to an external HTTP server, and has nOt been responded to. Default: 60000

8.3 Other CCXML Platform Configuration Parameters

Parameter	Description
CMP Proxy Connection Settings	
cmp.proxy	The IP or hostname of the CMP Proxy that CLC to connect to Default: 127.0.0.1
cmp.proxy_port	The port number of the CMP proxy to connect to Default: 8700
cmp.heartbeat	The interval, in seconds, to send a periodic heartbeat message from the component to the CMP Proxy

	Default: 20
cmp.reconnect	The interval, in seconds, between reconnection attempts to the CMP Server Default: 5
Logging Configuration Settings	
cmp.log_file	This full path to the log file of the CCP-CCXML Default: /usr/local/ccp-ccxml/logs/CMP.log.ccpccxml
cmp.size_option	Rollover all log files by size or by time Possible values: FALSE, TRUE Default: TRUE
cmp.rollover_size	The size limit, in MB, for rollover when rolling over by size Default: 10
cmp.num_rollover_files	The number of files to roll through before they are overwritten when rolling over by size Default: 5
cmp.rollover_mins	The interval of time, in minutes, between rollover when rolling over by time Default: 1440
cmp.rollover_time	The time at which the log files are rolled over when rolling over by time Default: 4:00
Email parameters	
cmp.email	If the EMAIL sink is specified, the email address be used Default: name@domain.com
Logging Service parameters	
cmp.log_sinks	Sinks that will be used by this component,possible sinks are: FILE, UPSTREAM, SYSLOG, SNMP, EMAIL Default: FILE|UPSTREAM
cmp.trace_flag	Enables tracing (Log 5 filter) Possible values: FALSE, TRUE

	Default: FALSE
cmp.pid_option	Appends PID of the process to the name of the trace file so that they are not overwritten when the process restarts Possible values: FALSE, TRUE Default: FALSE
cmp.metrics	log mask for metrics data Default: 0|1
cmp.log_0	Log mask for data logged at log level 0 Default: <too long to display>
cmp.log_1	Log mask for data logged at log level 1 Default: <too long to display>
cmp.log_2	Log mask for data logged at log level 2 Default: <too long to display>
cmp.log_3	Log mask for data logged at log level 3 Default: <too long to display>
cmp.log_4	Log mask for data logged at log level 4 Default: <too long to display>
cmp.log_5	Log mask for data logged at log level 5 Default: <too long to display>
Guaranteed Logs parameters	
cmp.guaranteed_logs_to_file	Specify if logs that are guaranteed to be sent upstream should be logged to a temp file Possible values: FALSE, TRUE Default: true
cmp.unsent_log_file	Specify the name of the temp log file to log to if cmp.guaranteed_logs_to_file Default: /usr/local/ccp-ccxml/logs/guaranteed.log.ccpccxml
cmp.unsent_log_file	Specify the name of the temp log file to log to if cmp.guaranteed_logs_to_file Default: /usr/local/ccp-ccxml/logs/guaranteed.log.ccpccxml
SIP Configuration	

sip.localuser	SIP user presented in outbound calls and at registration This configuration parameter controls the address that will be used in SIP registrations with a registrar/proxy server, and presented in requests that are initiated by the local system. The specified text will be presented in the "From:" field, and must be of the form "sip:user@host[:port]". The default value is "ccxml@(local IP):(default port)" Default: ccxml
sip.localhostname	Similar to sip.localuser, this parameter controls the hostname that will be presented in SIP requests sent by the local system. Note that the local hostname must not include a port number. If this parameter is not specified, then the IP address of the local system will be used. If it is desired to use a hostname or other name instead, then this parameter can be specified. This parameter can also be used to provide the fully qualified domain name in SIP requests.
sip.routeset	Defines a SIP route set for outbound calls. If defined, this route set will be inserted as the Route: header for all outgoing SIP calls. Each element in the routeset must be separated by a comma.
sip.transport.0	Defines transport layer for SIP stack and the network interfaces that are used to process SIP requests sip.transport.x = transport_name type:ip:port [parameters] where transport_name is any string; type is udp; ip is the IP address of the network interface that accepts incoming SIP messages. any (all network interfaces) is the default value. port is the port number where SIP stack accepts incoming SIP messages. 5060 is default value. [parameters] defines any extra SIP transport parameters. mcast is the multicast address which stack will accept multicast SIP messages. This value must be equal to sip.multicast. mcast-if is the network interface which will accept multicast messages. Default: transport0 udp:any:5060
sip.transport.1	Defines transport layer for SIP stack and the network interfaces that are used to process SIP requests sip.transport.x = transport_name type:ip:port [parameters]

	where transport_name is any string; type is udp; ip is the IP address of the network interface that accepts incoming SIP messages. any (all network interfaces) is the default value. port is the port number where SIP stack accepts incoming SIP messages. 5060 is default value. [parameters] defines any extra SIP transport parameters. mcast is the multicast address which stack will accept multicast SIP messages. This value must be equal to sip.multicast. mcast-if is the network interface which will accept multicast messages. Default: transport1 tcp:any:5060
sip.transport.2	Defines transport layer for SIP stack and the network interfaces that are used to process SIP requests sip.transport.x = transport_name type:ip:port [parameters] where transport_name is any string; type is udp; ip is the IP address of the network interface that accepts incoming SIP messages. any (all network interfaces) is the default value. port is the port number where SIP stack accepts incoming SIP messages. 5060 is default value. [parameters] defines any extra SIP transport parameters. mcast is the multicast address which stack will accept multicast SIP messages. This value must be equal to sip.multicast. mcast-if is the network interface which will accept multicast messages.
sip.timer.ci_proceeding	Defines a timer for client transaction in the proceeding state. if a final response is not received within this value in milliseconds, the client transaction is considered terminated. Default value is 120000 (120 seconds). Default: 120000
sip.route.dests	A list of space-delimited entries in a routing table. The entry ID starts from 0 and increments by 1 each time. For example, to specify 4 entries in the routing table, the value would be "0 1 2 3"
sip.route.dest.#	An entry in the routing table. The format is [Destination]

	[Netmask] [Transport] [Metric]. The [Transport] entry corresponds to the index specified in 'sip.transport.x' configuration. The 'x' is the transport interface index. Each transport specified in 'sip.transport.x' must have at least one entry in the routing table, otherwise the interface will never be used. The order of destination does matter as the routing table is linearly searched until none of the rows matches, then the default entry for the specified protocol will be used. To select an interface, take the outgoing IP address. From the list of interfaces with the matching protocol, starting from the top row, mask the IP address with [Netmask] entry and compare with [Destination] entry. If [Destination] entry matches the masked value, then stop and use the interface defined in the [Transport] column. Note that the [Metric] entry is needed but not used at this point.
sip.route.default.udp	Default route for UDP. The number denotes the transport defined in sip.transport.x where x is the value of this parameter and will be used when no UDP routes are found. If this parameter is not set, the first UDP transport found in sip.transport.x becomes the default.
sip.route.default.tcp	Default route for TCP. The number denotes the transport defined in sip.transport.x where x is the value of this parameter and will be used when no TCP routes are found.
sip.threadpoolsize	The size of the thread pool for handling DNS queries Default: 4
sip.mtusize	Defines the Maximum Transmission Unit (MTU) of the network interfaces. If a SIP request size is within 200 bytes of this value, the request will be sent on a congestion controlled transport protocol, such as TCP. Default: 1500
sip.maxtcpconnections	Defines the maximum number of TCP connections established concurrently. if the maximum number of TCP connections has been reached, new SIP requests to establish TCP connections will be rejected. Default: 100
sip.min_se	Defines the Min-SE parameter in seconds. This is the minimum duration of session expiry this SIP stack will accept from a user agent client. Default: 90
sip.sessionexpires	Defines the default session expiry value in seconds. The

	session timer defines the duration of which a SIP session will expire if no re-INVITEs are sent/received within this period. Default: 1800