VOICEGENIE

# VoiceGenie 7 CCXML Platform

## *User's 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 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 and renamed to User's Guide | Henry Lum |
| **3** | 2005/06/17 | Final Updates before release | Alex Lee |

# 1.0 Introduction

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.

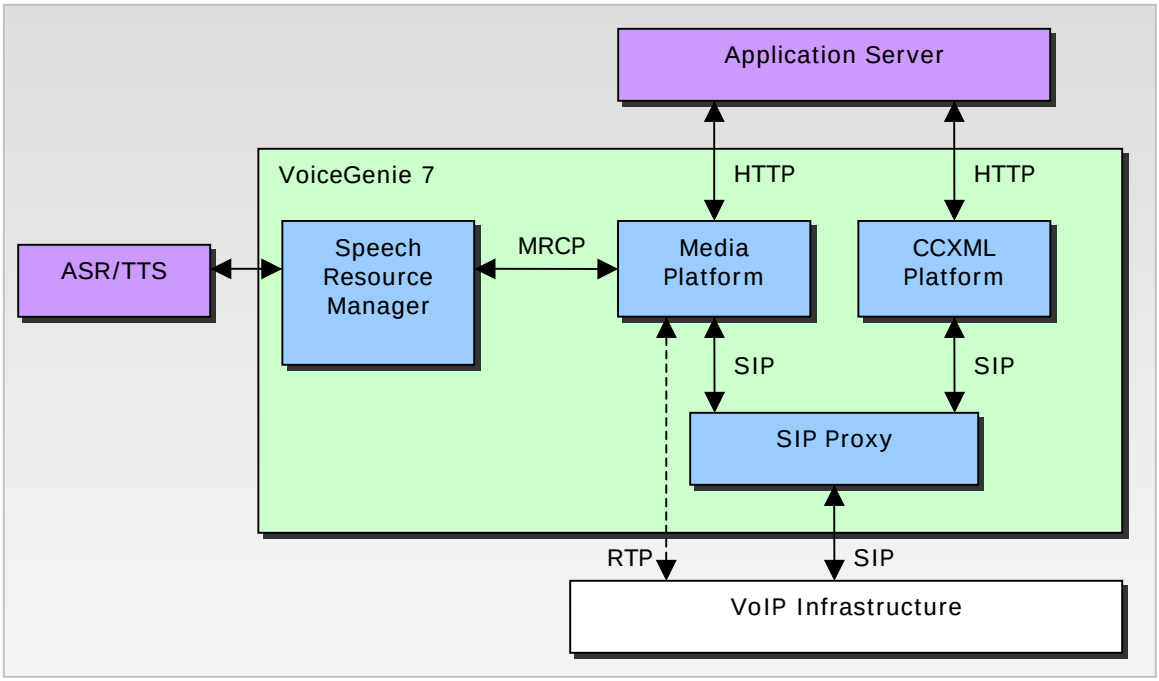
## 1.1 Compatibility and Support

As of writing, 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 CCXML System Architecture

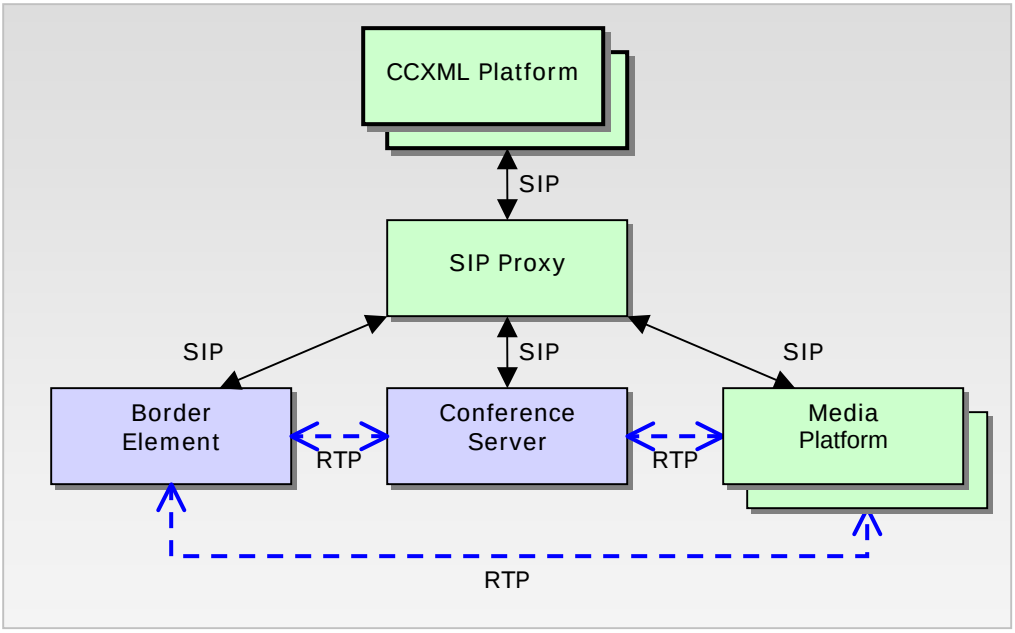
CCXML Platform is a component within the VoiceGenie 7 architecture, which is shown in the diagram below:



CCXML Platform resides within VoiceGenie OA&M framework that allows CCXML Platform to be deployed, configured, monitored, and managed in a consistent manner with other VoiceGenie software components. For detailed information on OA&M functionalities, please refer to section 4.

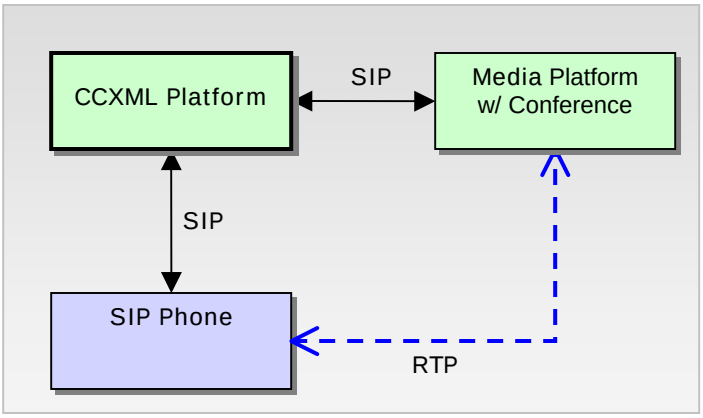
## 2.1 Clustered Environment

In a clustered environment, SIP Proxy manages multiple instances of CCXML Platforms and Media Platforms. External elements send to the SIP Proxy to forward the SIP requests to the appropriate SIP service. The following diagram, similar to the architecture diagram above, 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. The following diagram shows the configuration:




# 3.0 Platform Features

## 3.1 Dialing into CCXML Platform

CCXML Platform accepts incoming SIP connections on port 5060 by default. The port number can be changed if *sip.transport.x* is used to modify the port number.

### *Calling to the default CCXML page*

By default, all incoming connections will start a new CCXML session with a default URI. The default page is located at /usr/local/ccp-ccxml/config/default.ccxml. The default application rejects all inbound connections.

The location of the default page can be changed with configuration parameter *ccpccxml.default_uri*.

### *Starting a non-default CCXML page*

To start a different page with an incoming connection, the CCXML platform follows the netann convention (currently http://www.ietf.org/internet-drafts/draft-burger-sipping-netann-11.txt). The request URI of the incoming request must follow this format:

sip:ccxml@ccxmlplatform.voicegenie.com;ccxml=http://www.voicegenie.com/page.ccxml

where:
1) The userpart of the Request URI must be *ccxml* and it is case-sensitive;
2) *ccxmlplatform.voicegenie.com* is the host or IP address of the CCXML Platform;
3) The Request URI must contain a *ccxml* URI parameter (case-sensitive); the value is the CCXML page to be started. Other URI parameters can be included and the ordering does not matter.

### *Using SIP Proxy to map CCXML applications*

VoiceGenie SIP Proxy can be used to map CCXML applications by translating the SIP request URI to the netann format described in the above sub-section. Here is an example:


With the above mapping, SIP Proxy translates sip:1234@10.0.0.123 into sip:ccxml@10.0.0.124;ccxml=file:///usr/local/phoneweb/samples/application.ccxml

where
10.0.0.123 is the SIP Proxy address;
10.0.0.124 is the CCXML Platform address.

## 3.2 Inbound Connections

### *Passing URI parameters to CCXML applications*

When fetching the initial CCXML page, CCXML Platform adds parameters to the initial URI. These URI parameters are found in the incoming SIP Request URI parameters.

For example, an SIP Request URI looks like this:
sip:ccxml@ccxmlplatform.voicegenie.com;ccxml=http://www.voicegenie.com/page.ccxml;hello=world

The initial URI fetch will be:
GET http://www.voicegenie.com/page.ccxml?hello=world

### *Call Parameters accessible in CCXML applications*

Call Parameter (or SIP headers) can be made accessible in the CCXML application by reading the session variable. The following table shows the connections properties

| Connection properties (shown via Session variable) | Description |
|---|---|
| session.connections[ connectionid ].local | To: header |
| session.connections[ connectionid ].remote | From: header |
| session.connections[ connectionid ].protocol.name | sip |
| session.connections[ connectionid ].protocol.version | 2.0 |
| session.connections[ connectionid ].protocol.sip.callid | Call-ID header |
| session.connections[ connectionid ].protocol.sip.requesturi | Request URI |
| session.connections[ connectionid ].protocol.sip.from | From: header |
| session.connections[ connectionid ].protocol.sip.to | To: header |

#### *183 Session Progressing Response*

CCXML Platform sends 100 Trying response immediate upon receiving an INVITE request. CCXML Platform sends 200 OK when the CCXML application executes the <accept> tag. By default, 183 Session Progressing response is not set.

Setting ccpccxml.sip.send_progressing configuration parameter to 1 allows the CCXML Platform to send 183 Session Progressing along with 200 OK when <accept> tag is executed on an inbound connection.

CCXML application can request the CCXML Platform to send 183 Session Progressing with a <send> tag. Here is an example:
<send target="connectionid" targettype="'x-connection'" data="'connection.alerting'"/>

#### *Rejecting incoming connections*

Rejecting an incoming connection with the <reject> tag will cause CCXML platform to response with a 400 Bad Request response.

Using the reason attribute in the <reject> tag will enable the use of Reason header in the 400 Bad Request response. The header will contain the following:

```
Reason: SIP; cause=400; text="content of the reason attribute"
```

#### *Disconnecting calls*

When a connection is connected, calling <disconnect> tag will send BYE message on the connection to terminate the call. This applies to both inbound and outbound connections.

Using the reason attribute in the <disconnect> tag will enable the use of Reason header in the BYE message. The header will contain the following:

```
Reason: SIP; cause=200; text="content of the reason attribute"
```

### 3.3 Connection Signals

When a SIP INFO message is received on a connection, CCXML platform raises a connection.signal event. There will be two properties set in the *info* property:

*event.info.contenttype* – the value of Content-Type header of the SIP INFO message;
*event.info.content* - the content of the SIP INFO message.

#### *Receiving DTMF digits through SIP INFO*

When a SIP INFO message's Content-Type is "application/dtmf-relay", it implies that it is a DTMF digit. The connection.signal event will also contain the *event.info.dtmf* property that provides the DTMF digit(s).

### 3.4 Outbound Connections

When making an outbound connection, provide the SIP URI in the *dest* attribute of <createcall> tag. CCXML Platform uses the given SIP URI to send an INVITE request. The SIP Request is sent directly to the destination.

#### *Mapping SIP responses to CCXML connection events*

All 1xx response received from the outgoing connection will result in a connection.progressing event.

When 2xx response is received, a connection.connected event will be thrown.

When a non-2xx final response (300-699) response is received, a connection.failled event will be thrown.

***Disconnecting progressing call***

When the <disconnect> tag is used on an outbound progressing call, CCXML Platform will send a CANCEL message on the outgoing call to terminate it.

## 3.5 Call Redirection

***Redirecting an incoming call***

Using <redirect> tag on an incoming call (in ALERTING state) will redirect the call. CCXML Platform will send a 302 Moved Temporarily response. The dest attribute of the <redirect> tag will translate to the Contact header in the 302 response.

***Redirecting a connected call***

Using <redirect> tag on a connected call (applies to both inbound and outbound call) will redirect the call with the REFER message. CCXML Platform will send a REFER message on the call. The dest attribute of the <redirect> tag will translate to the Refer-To header in the REFER message. Once CCXML Platform receives a NOTIFY message with a 200 OK message, the call is considered redirected and the connection will be released. The application will receive a connection.redirected event.

If the redirection fails for any reason, the call will receive an error.connection event.

## 3.6 Call Merge

Two connections can merge at the network level (ie. bridging the calls at the switch) when both of them are in CONNECTED state. CCXML Platform uses REFER message with Replaces as the mechanism to initiate a call merge feature at the switch.

Here an example:

Assume the first call was connected with

```
INVITE sip:hi@10.0.0.1 SIP/2.0
Via: SIP/2.0/UDP
From: sip:bye@10.0.0.2
To: sip:hi@10.0.0.1
Max-Forwards: 70
CSeq: 1 INVITE
Call-ID: DC9D0D00-F5CD-6037-C2A2-6BDBE04CC38E
Contact: sip:bye@10.0.0.2:5060
Content-Length: 147
Content-Type: application/sdp
…
```

Assume the second call was connected with

```
INVITE sip:hello@10.0.0.1 SIP/2.0
Via: SIP/2.0/UDP
From: sip:world@10.0.0.3
To: sip:hello@10.0.0.1
Max-Forwards: 70
CSeq: 1 INVITE
Call-ID: DC9D0D00-F5CD-6037-C2A2-6BDBE04CC123
```

```
Contact: sip:world@10.0.0.3:5060
Content-Length: 147
Content-Type: application/sdp
     …
```

| | | |
|---|---|---|
| <merge> | → | REFER sip:bye@10.0.0.2 SIP/2.0<br>Via: SIP/2.0/UDP 10.0.0.1:5060<br>From: sip:hi@10.0.0.1<br>To: sip:bye@10.0.0.2<br>Cseq: 2 REFER<br>Call-ID: DC9D0D00-F5CD-6037-C2A2-6BDBE04CC38E<br>Refer-To: world@10.0.0.3;Replaces=DC9D0D00-F5CD-6037-C2A2-6BDBE04CC123<br>… |
| | ← | SIP/2.0 202 Accepted<br>…<br>Cseq: 2 REFER<br>… |
| connection.merged | ← | NOTIFY sip:bye@10.0.0.2 SIP/2.0<br>…<br>Cseq: 3 NOTIFY<br>Event: refer<br>Content-Type: message/sipfrag;version=2.0<br>Content-Length: 14<br><br>SIP/2.0 200 OK |
| | → | SIP/2.0 200 OK<br>…<br>Cseq: 3 NOTIFY<br>… |
| | → | BYE sip:bye@10.0.0.2 SIP/2.0<br>…<br>Call-ID: DC9D0D00-F5CD-6037-C2A2-6BDBE04CC38E<br>… |
| | → | BYE sip:world@10.0.0.3 SIP/2.0<br>…<br>Call-ID: DC9D0D00-F5CD-6037-C2A2-6BDBE04CC123<br>… |
| | ← | SIP/2.0 200 OK<br>…<br>Call-ID: DC9D0D00-F5CD-6037-C2A2-6BDBE04CC38E<br>… |
| | ← | SIP/2.0 200 OK<br>…<br>Call-ID: DC9D0D00-F5CD-6037-C2A2-6BDBE04CC123<br>… |

## 3.7 Dialogs

### *Preparing Dialogs*

<dislogprepare> or <dialogstart> creates a new dialog; CCXML platform initiate a new SIP dialog to the dialog server. CCXML Platform sends an INVITE message to the SIP Proxy (configurable with the ccpccxml.sip.proxy parameter) with one of the following request URI types:

Rosenburg syntax (default):
sip:dialog.vxml.http%3F//www.voicegenie.com/page.vxml@sipproxy.voicegenie.com

netann syntax:
sip:dialog@sipproxy.voicegenie.com;voicexml=http%3F//www.voicegenie.com/page.vxml

Where sipproxy.voicegenie.com is a configuration parameter ccpccxml.sip.proxy.

The types of the request URI syntax are selectable in the CCXML interpreter configuration with the ccpccxml.dialog.uri_scheme.

Note that at the time of writing that Media Platform only supports Rosenberg syntax.

Using <dialogprepare> to prepare a dialog will send a NULL SDP to the dialog server to let the dialog server (Media Platform in this case) to prepare the dialog without starting the audio. When the INVITE transaction is ACKed, the dialog is fetched and loaded on the Media Platform.

A null SDP represents an SDP content with 0.0.0.0 in the connection information.

### *Passing dialog results back to CCXML*

VoiceXML pages can return results back to CCXML Platform by adding content to the BYE message. VoiceXML page can use the namelist attribute in the <exit> tag to send dialog results back to the CCXML application.

Example where VoiceXML application ends the call with <exit namelist="hello a"/>:

Media Platform sends BYE to CCXML Platform:

```
BYE sip:10.0.0.1:5060 SIP/2.0
Via: SIP/2.0/UDP 10.0.0.3
Via: SIP/2.0/UDP 10.0.0.2:5060
From: sip:VoiceGenie@10.0.0.2
To: sip:10.0.0.1:5060
Max-Forwards: 69
CSeq: 1 BYE
Call-ID: DC9D0D00-F5CD-6037-C2A2-6BDBE04CC38E
Content-Length: 16
Content-Type: application/text

hello=world
a=b
```

dialog.exit event contains:
values.hello = 'world'
values.a = 'b'
namelist='hello a'

dialog.disconnect is not supported by the CCXML Platform. Whenever a VoiceXML application exits, dialog.exit will be thrown.

### *Dialog User Event*

The VoiceXML dialog may send a user event to the CCXML application by using the <log dest="callmgr"> tag. Here is an example:

<log dest="callmgr">event=hello;a=b;foo=bar</log>

Media Platform sends a SIP INFO message that contains the following headers and content:

```
INFO sip:10.0.0.1:5060 SIP/2.0
Via: SIP/2.0/UDP 10.0.0.2:5060
From: sip:VoiceGenie@10.0.0.2
To: sip:10.0.0.1:5060
Max-Forwards: 70
CSeq: 1 INFO
Call-ID: DC9D0D00-F5CD-6037-C2A2-6BDBE04CC38E
Content-Length: 16
Content-Type: application/text

event=hello;a=b;foo=bar
```

This will raise a dialog.user.hello event to the CCXML session that owns the dialog. The event itself will contain the following properties:

event.values.event = hello
event.values.a = b
event.values.foo = bar
event.namelist = 'event a foo'

CCXML Platform parses the Content of the SIP INFO from the dialog and creates the values object. The property-value will be pairs delimited by semicolons. The event property value exist in the content, the CCXML event raised will be called dialog.user.name, where name is the event property value. Otherwise, the event will be named dialog.user.event.

Limitation: the received event cannot be a multi-level object; only simple name-value pairs are supported as provided in the example above.

### *Dialog-initiated transfer*

CCXML Platform supports one type of dialog-initiated transfer from the Media Platform: blind transfer. To initiate a dialog-initiated transfer, the VoiceXML application must call
<transfer destexpr="number_to_call" bridge="false" type="unsupervised">
The following sequence of events will happen:

1) Media Platform will send a REFER message on the SIP dialog;
2) CCXML application will receive a dialog.transfer event. The *type* attribute is blind and the *uri* attribute is the destexpr in the <transfer> tag;
3) CCXML application executes <redirect> to move the call specified in the dialog.transfer event;
4) If redirection is successful, CCXML application sends telephone.disconnect.transfer event to the dialog;
5) CCXML Platform sends NOTIFY (200 OK) reporting the result of the transfer
6) If redirection fails, CCXML application sends error.transfer.noroute event to the dialog;
7) CCXML Platform sends NOTIFY (500 Server Internal Error) reporting a transfer failure

8) VoiceXML application will receive telephone.disconnect.transfer event to end the transfer and the VoiceXML page. The result is recorded in the metricsfile of the Media Platform.

### *Media Playback dialogs (no VoiceXML)*

CCXML Platform can accept playback dialogs that contains no VoiceXML pages. This feature is useful for playing announcements without the need of using a full VoiceXML page.

When <dialogprepare> or <dialogstart> uses the type attribute that is not application/voicexml+xml, CCXML Platform will initiate the dialog with a Request URI of the following format:

```
sip:annc@sipproxy.voicegenie.com;play=http%3F//www.voicegenie.com/welcome.wav;content-type=audio/wav
```

where
*http://www.voicegenie.com/welcome.wav* is src attribute;
*audio/wav* is the type attribute.

Note that at the time of writing this feature is not available to Media Platform.

### *VoiceXML Session Variables*

As mentioned in Appendix D of the CCXML specification, the Media Platform does not support session.connection.ccxml VoiceXML session variables.

## 3.8 Conferences

When a CCXML application joins to a conference, CCXML Platform sends an INVITE message to the SIP Proxy with a formatted Request URI:

netann format (default):
sip:conf=ABCD1234@10.0.0.1;confinstid=ABCD1234;confreserve=3;confmaxsize=3

Where
conf, confinstid are cluster-wide unique conference identifier
confreserve is the number of conference participants to reserve for this conference
confmaxsize is the maximum size of this conference

simple format (supported by Audiocodes Media Gateways only):
sip:**3**conf**ABCD1234**@sipproxy.voicegenie.com;confinstid=ABCD1234;confreserve=3;confmaxsize=3

Where
The number before conf is the maximum size of this conference
The string after "conf" and before "@" is the cluster-wide unique conference identifier
confinstid are cluster-wide unique conference identifier
confreserve is the number of conference participants to reserve for this conference
confmaxsize is the maximum size of this conference

This is the sum of the reservedtalkers and reservedlisteners attribute in the <createconference> tag represents the number of conference participants to reserve for this conference. The default value can be set in the configuration parameter *ccpccxml.conference.defaultreserve*.

The maximum size of the conference by default equals to confreserve. This value can be set in the hints attribute of the <createconference> tag; it is the *maxsize* property of the hints object.

In a clustered environment where multiple conference servers are available, the CCXML Platform relies on VoiceGenie SIP Proxy to forward the requests for the same instance of the conference to the same conference server. This is a feature of the SIP Proxy. Note that <createconference> will proceed successfully even if the cluster has no conferences available or no conference servers can fit the requested conference size. A <join> operation may fail due to the above reasons and returns error.conference.join event.

***Limitations***

CCXML Platform supports up to **100** conference participants in a conference.

Forward join to a conference running on VoiceGenie's Media Platform will not join properly. A workaround is to always use duplex join to a conference.

## 4.0 HTTP I/O Processor

CCXML Platform supports an external event processor using the HTTP protocol. It is based on an extension as proposed to the CCXML working group in the following HTML document:

ccxml-basichttp-20050226.html

However, CCXML Platform will support only the most basic functions in this proposal, and some of the behaviors are different from the HTML document above. The above document should serve only as a reference. The following sections describe the features to be supported by CCXML Platform.

### 4.1 Session Variable

A new session variable session.ioprocessors is to be exposed to the CCXML session. This is an associative array, with only one member in the array session.ioprocessors["basichttp"] which contains the URI by which events are to be sent to the CCXML platform.

### 4.2 Receiving Events

An external HTTP client may make a GET request to the CCP-CCXML platform. The POST method is not supported by the CCXML platform. The URI that accepts the request is exposed as specified in the session.ioprocessors["basichttp"] session variable above, and will be documented in the User's Guide. The HTTP request is analysed by the basichttp event processor, resulting in

1. an event being injected into an active CCXML session, or
2. no action being taken (an error occurred, operation not permitted, etc)

The basichttp processor then reports its result to the external component in the response of the originating HTTP request.

HTTP request parameters will be used to specify the information of the event to be injected into the session. In particular, the following parameters have special meanings:

| HTTP Parameter Name | Meaning |
|---|---|
| sessionid | This is the Session ID of the session destined to receive the event. This parameter is required. |
| eventname | This is the name of the event to be triggered by the CCXML session. This parameter is required. |
| eventsource | This value specifies a URI to which events may be sent (i.e. it may be used as the value of the target attribute in a <send> element). The parameter is optional. |

Other parameters provided in the HTTP request are treated as the event payload. Payload parameter names must be valid ECMAScript variable names. Reserved and payload parameter values must be valid ECMAScript expressions.

The CCP-CCXML platform will reply to the HTTP request with the following HTTP response codes:

| Response Code | Condition |
|---|---|
| 204 | The sessionid parameters matches an existing CCXML session id, event name and payload parameters are valid |
| 400 | One or more parameters have an invalid name or value |
| 403 | Failure due to other reasons (e.g. session id does not match an existing CCXML session id) |

The following table describes the event attributes of a successfully received event via HTTP request:

| Attribute Name | Description |
|---|---|
| name | the value of the eventname parameter. |
| eventid | a unique string identifier of the event generated by the CCP-CCXML platform. |
| eventsource | the value of the eventsource parameter if provided; otherwise this event attribute is undefined. |
| eventsourcetype | always has the value "basichttp" |
| <*param-name*> | For each *param-name=value* appearing in parameter name in the http request URI, the parameter *param-name* will appear as a property in the event object. Its value is set to *value*. |

In addition, VoiceGenie's HTTP implementation behaves as follows:

- The Session ID of the session to receive the event cannot appear in the Cookie header of the HTTP request. In fact, all HTTP request headers and the HTTP message body will be ignored by the HTTP I/O processor.
- Only the HTTP GET method will be supported.

Limitation: the received event cannot be a multi-level object; only simple name-value pairs are supported in <param-name>.

## 4.3 Sending Events

The CCXML session may send an event by using the <send> tag, as described in section 9.2.3 of the CCXML specification. Inline content for <send> will not be supported, instead only the namelist attribute will be supported. The following describes how the various attributes of <send> will map to the HTTP request:

| Attribute Name | Meaning |
|---|---|
| targettype | If this attribute is set to "basichttp" the message will be routed to the HTTP I/O Processor |
| target | This is the HTTP URL to which a GET request will be made |
| data | This is an ECMA expression evaluating to the event name. This attribute is required for sending to an HTTP I/O processor. This will become the value of the eventname parameter of the HTTP request |
| xmlns | This attribute will **not** be supported, since VoiceGenie will not be supporting the sending of inline contents. |
| namelist | This is an optional parameter, and if it is defined, its variable names and values are mapped to HTTP parameters. |
| hints | This parameter will be ignored by the CCXML interpreter |

In addition, VoiceGenie's HTTP implementation behaves as follows:

- The Session ID (as found in the session.id session variable of the CCXML session) of the session making the <send> request will be one of the HTTP parameters in the GET request. The HTTP parameter name is sessionid. This Session ID will not appear as a Cookie in the HTTP request.
- Inline content of <send> will not be supported.
- Only the HTTP GET method will be supported.

The CCP-CCXML platform will interpret the HTTP response codes in the following way:

| Response Code | Interpretation |
|---|---|
| 204 | The <send> was successfully accepted by the HTTP server and a send.successful event is posted to the session issuing the <send> |
| Any other HTTP response code (include other 2xx codes) | The <send> was not accepted by the HTTP server and a error.send.failed event is posted to the session issuing the <send> |

## 4.4 Example - Receiving Events via HTTP

In the CCXML platform, the value of the session variable session.ioprocessors["basichttp"] is 'http://ccxml.voicegenie.com/ccxml/basichttp'. When the following HTTP request is made to this platform:

```
GET http://ccxml.voicegenie.com/ccxml/basichttp?sessionid=ccxmlsession1&
eventname=basichttp.myevent&eventsource=http://www.example.org/ccxmlext&
agent=agent12&site=Orlando HTTP/1.0
…[other HTTP headers]...
…[other HTTP headers]...
```

If "ccxmlsession1" (value of the "sessionid" parameter in the HTTP request avove) matches the session id of an existing CCXML session, an event with name "basichttp.myevent" is triggered in the "ccxmlsession1". It may be handled as follows:

```
<transition state="'dialogActive'" event="basichttp.*" name="evt">
    <log expr="'Received event'" />
    <log expr="'name=' + evt.name" />
    <log expr="'sourcetype=' + evt.eventsourcetype" />
    <log expr="'source=' + evt.eventsource" />
    <log expr="'agent=' + evt.agent" />
    <log expr="'site=' + evt.site" />
</transition>
```

where evt.name would have the value "basichttp.myevent", evt.eventsourcetype the value "basichttp", evt.eventsource the value "http://www.example.org/ccxmlext", evt.agent the value "agent12" and evt.site the value "orlando".

In addition, the CCXML platform will respond with a 204 HTTP response code:

```
HTTP/1.0 204 No Data
```

## 4.5 Example - Sending Events via HTTP

Consider the following CCXML snippet by CCXML session with session id 'ccxmlsession2':

```
<script>
    var agent='agent21';
    var site='miami';
</script>
<send target="'http://travel.voicegenie.com/travelagent'" data="'myevent'"
targettype="'basichttp'" namelist="agent site"/>
```

With this CCXML snippet, the following HTTP GET request is made:

```
GET http://travel.voicegenie.com/travalagent?sessionid=ccxmlsession2&
eventname=myevent&agent=agent21&site=miami HTTP/1.0
CRLF
```

# 5.0 Device Support

CCXML Platform is supported on the following SIP devices or media gateways:

- X-Lite
- Cisco AS5300
- Audiocodes Media Gateway

### 5.1 Limitations

Media join to multiple listeners (ie. A->B and A->C) will not work on the supported devices. The devices can only send to a single listener and receive from a single listener. If multiple listeners are required, please deploy a conference server and have the parties joined to the conference.

The following list also shows feature limitation on each SIP device:

X-Lite

- REFER (<redirect> in CONNECTED state);
- REFER with replaces (<merge>);
- SIP Session Timer.

Cisco

- REFER with replaces (<merge>). This feature will work on Cisco media gateways that support call merge technologies such as TBCT (Two B-Channel Transfer).

Audiocodes Media Gateway

- REFER with replaces (<merge>);
- SIP Session Timer;
- Conference supports only simple URI scheme. Please change ccpccxml.conference.URI_scheme configuration parameter to simple;
- Audio Codes does not set route correctly when the audio source for the same listener changes. This affects half duplex joins that changes only the source of a connection. For example, if the current state is A--->B and a half duplex join changes to C--->B, B will not hear any audio. A workaround is to make the second join a full duplex join.

## 6.0 CCXML Specification Support Notes

The VoiceGenie platform is does not support some features described in the CCXML specification:

### 6.1 Some optional connection properties are not supported

The following optional attributes (from section 10.2.2 of the CCXML spec) are not supported for the connection object:

- connection.substate
- connection.dialogid
- connection.redirect
- connection.aai

### 6.2 "xmlns" attribute in <ccxml> element is not supported

Additional namespaces specified in the <ccxml> tag is ignored.

### 6.3 "http-equiv" attribute in <meta> element is not supported and ignored

### 6.4 The <metadata> tag is not supported and is simply ignored

### 6.5 "xmlns" and inline content in <send> element are not supported

### 6.6 UTF-16 charset is not supported for CCXML pages and scripts

### 6.7 Runaway sessions cannot be terminated properly

Each CCXML session is essentially a state machine handling events. The CCXML application developer should be very careful when writing their applications, because if a CCXML session does not receive any events, there is no way for it to terminate. To avoid 'runaway sessions', the application developer should ensure that if the session is about to go into a state where it won't receive any further events (e.g. when it no longer has any connections connected to it) it should terminate via an exit tag.

### 6.8 POST method of fetching is not supported for dialogs

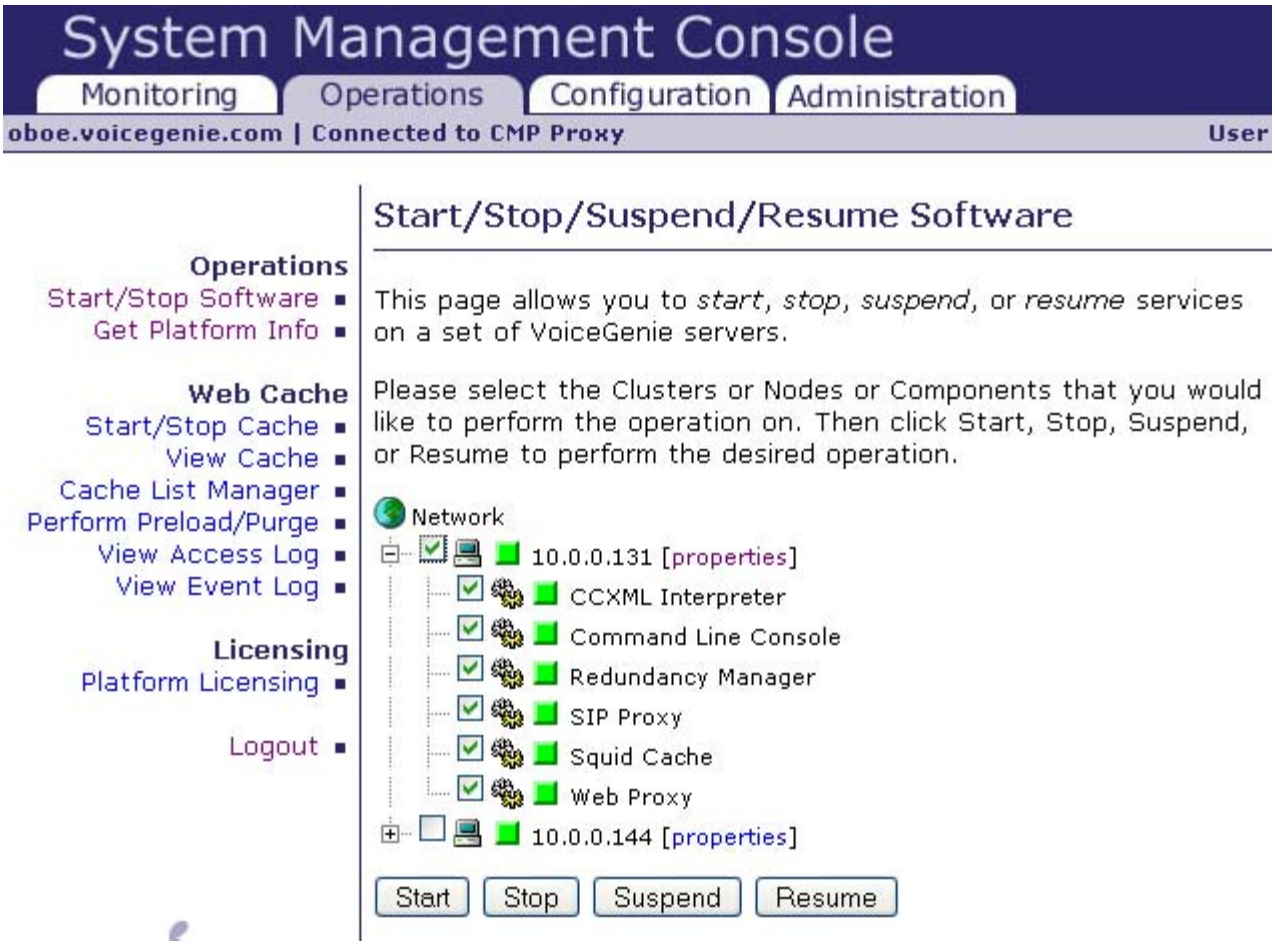
Even when the POST method is specified in <dialogstart> or <dialogprepared>, the VoiceXML page would still be fetched via the GET method.

## 7.0 Operations, Administration, Maintenance

### 7.1 Starting and Stopping

Like all other VoiceGenie 7 components, the System Management Console offers a dedicated page to start or stop the CCXML Platform. 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.

The following is a screenshot of the start/stop page:




## 7.2 Suspend and Resume

Suspend and Resume is available to CCXML Platform. In the "Start/Stop Software" page, select the CCXML Interpreter component and click on Suspend or Resume button to perform the operation.

CCXML Interpreter in Suspend state is shown in blue in the Status Monitor. While the CCXML Interpreter is suspended, all incoming connections will be rejected with 503 Service Unavailable SIP responses, and all outgoing connections will fail. Connected or connecting calls will continue to operate.

| Name | Hostname | CMP Proxy | CPU | Memory | Disk | Call Manager | VoiceXML Interpreter | System Management Console | Command Line Console | Web Proxy | Redundancy Manager | SIP Proxy | CCXML Interpreter | Squid Cache |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10.0.0.131 | 10.0.0.131 | | 1.49% | 104MB | 41% | - | - | - | | | | | | |
| 10.0.0.144 | 10.0.0.144 | | 1.77% | 231MB | 56% | | | | | | - | - | - | |

Online  Suspended  Stopping/Stopped  Offline  Config Not Synchronized

## 7.3 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 ccxml
In SMC, go to "Monitoring" tab and click on the CCXML Interpreter item.

Here is a sample output:

```
CLC> health ccxml

Health for CCXML Interpreter(ccxml) on 10.0.0.131
Started: 2005-03-28/19:00:46.449
Status: ONLINE
Connections: (Current:Peak:Total): 0:0:0
Dialogs: 0:0:0
Conferences: 0:0:0
Conference Participant: 0:0:0
Sessions: 0:0:0
```

The following are explanations of each parameter in the health status summary:
Status: Can be ONLINE or SUSPENDED, depending on whether the CCXML Interpreter is running or in suspended state.

The following fields are shown with 3 values: current number, peak number, or the total number.

**Connections** – shows the number of CCXML connections;

**Dialogs** – shows the number of CCXML dialogs;

**Conference** – shows the number of conferences;

**Conference Participant** – shows the number of conference participants that are joined to any conference. A conference participant is a connection or a dialog that used the <join> tag to join to a conference.

**Sessions** – shows the number of CCXML session.

## 7.4 Logging

CCXML Platform 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.ccpccxml.

Log levels for Notice (LOG_3) and Information (LOG_4) are stored in the log file. LOG_4 entries are logging by the CCXML Interpreter and its SIP platform to record all call operations. Please refer to Appendix C and D for details of the LOG_4 entries.

Trace logs (LOG_5) is disabled by default. Enabling it will trace all SIP messages sent and received by the CCXML Platform. Trace is not recommended for deployment environment as trace will flood the trace files quickly and decreases system performance; it is designed for debugging platform issues.

To enable platform trace, go to CCXML Interpreter Configuration and select true for cmp.trace_flag. Click Update to submit the configuration change.

To enable interpreter trace, set ccxmli.trace_flag to TRUE. Please ensure cmp.trace_flag is also set to true.

## 7.5 License Key

License key must be placed in /usr/local/phoneweb/config/vglicense.txt in order for CCXML Platform to pick up the license key. This key is distributed by VoiceGenie and is required to start and operate the CCXML Platform. The key contains the expiry date and the number of concurrent connections allowed in the CCXML platform.

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 and the key is set to expire in December 31, 2037. This key allows up to 5000 concurrent connections, dialogs, and conference participants.

# 8.0 Directory Structure

CCXML Platform home directory will reside in /usr/local/ccp-ccxml. The following table lists the subdirectories/files and their description:

| File (relative to CCXML platform home): | Description |
|---|---|
| bin/ccpccxml | CCXML interpreter executable |
| bin/pwproxy | Web Proxy executable |
| bin/busy | Helper script |
| bin/clean_shmem.pl | Shared memory cleanup script |
| bin/fm_restart_script.pl | Fetching module restart script |
| bin/fm_start_script.pl | Fetching module start script |
| bin/fm_wait_script.pl | Fetching module wait script |
| bin/get_log | Helper script |
| bin/ks | Helper script |
| bin/squid_start | Squid start script |
| bin/squid_stop | Squid stop script |

| | |
|---|---|
| bin/vgreset | Helper script |
| bin/vg-su | Helper script |
| config/default.ccxml | Default CCXML page |
| config/ccpccxml.cfg | Local CCXML configuration file |
| config/ccpccxml_provision.dat | Local CCXML provisioning file |
| config/ccp-ccxml.xml | CCXML definition file |
| config/iproxy.cfg | Web Proxy local configuration file |
| config/exagent.squid.cfg | Squid external agent local configuration file |
| config/squid.conf.template | Squid configuration template |
| cache/ | Cache directory |
| logs/CMP.log.ccpccxml* | SIP Proxy log files |

## 9.0 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 CCXML.

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

| Name | OID | Type | Description |
|---|---|---|---|
| CCPCCXML-HEALTHDATA-STARTED | .1.3.6.1.4.1.7469.3.9.28.1.1 | Scalar | CCXML Start Time (the amount of time elapsed since the start of CCXML Platform) |
| CCPCCXML-HEALTHDATA-STATUS | .1.3.6.1.4.1.7469.3.9.28.1.2 | Scalar | Status |
| CCPCCXML-HEALTHDATA-TOTALCONNECTIONS | .1.3.6.1.4.1.7469.3.9.28.1.3 | Scalar | Total number of connections |
| CCPCCXML-HEALTHDATA-CURRENTCONNECTIONS | .1.3.6.1.4.1.7469.3.9.28.1.4 | Scalar | Current number of connections in progress |
| CCPCCXML-HEALTHDATA-PEAKCONNECTIONS | .1.3.6.1.4.1.7469.3.9.28.1.5 | Scalar | Peak number of concurrent connections |
| CCPCCXML-HEALTHDATA-TOTALDIALOGS | .1.3.6.1.4.1.7469.3.9.28.1.6 | Scalar | Total number of dialogs |
| CCPCCXML-HEALTHDATA-CURRENTDIALOGS | .1.3.6.1.4.1.7469.3.9.28.1.7 | Scalar | Current number of dialogs in progress |
| CCPCCXML-HEALTHDATA-PEAKDIALOGS | .1.3.6.1.4.1.7469.3.9.28.1.8 | Scalar | Peak number of concurrent dialogs |
| CCPCCXML-HEALTHDATA-TOTALCONFERENCES | .1.3.6.1.4.1.7469.3.9.28.1.9 | Scalar | Total number of conferences |
| CCPCCXML-HEALTHDATA-CURRENTCONFERENCES | .1.3.6.1.4.1.7469.3.9.28.1.10 | Scalar | Current number of conferences in progress |
| CCPCCXML-HEALTHDATA-PEAKCONFERENCES | .1.3.6.1.4.1.7469.3.9.28.1.11 | Scalar | Peak number of concurrent conferences |
| CCPCCXML-HEALTHDATA-TOTALCONFPART | .1.3.6.1.4.1.7469.3.9.28.1.12 | Scalar | Total number of conference participants |
| CCPCCXML-HEALTHDATA-CURRENTCONFPART | .1.3.6.1.4.1.7469.3.9.28.1.13 | Scalar | Current number of conference participants joined to a conference |
| CCPCCXML-HEALTHDATA-PEAKCONFPART | .1.3.6.1.4.1.7469.3.9.28.1.14 | Scalar | Peak number of concurrent conference participants |

| Name | OID | Type | Description |
|---|---|---|---|
| CCPCCXMLI-HEALTHDATA-TOTALSESSIONS | .1.3.6.1.4.1.7469.3.9.28.1.15 | Scalar | Total number of CCXML sessions |
| CCPCCXMLI-HEALTHDATA-CURRENTSESSIONS | .1.3.6.1.4.1.7469.3.9.28.1.16 | Scalar | Current number of CCXML sessions in progress |
| CCPCCXMLI-HEALTHDATA-PEAKSESSIONS | .1.3.6.1.4.1.7469.3.9.28.1.17 | Scalar | Peak number of concurrent CCXML session |

## 10.0 Appendix B – Logging Traps

The OID prefix is ".1.3.6.1.4.1.7469.251.1.317". To get the OID of a trap, just append the prefix with the suffix column. For example, VGLOG-CCXML-WARNING-NOLICENSE has an OID of .1.3.6.1.4.1.7469.251.1.317.158335489.

### 10.1 Severity – Critical (LOG_0)

| Name | OID suffix | Response Code | Impacts | Causes | Recommended Actions | Message |
|---|---|---|---|---|---|---|
| VGLOG-CCXML-ERROR-INIT-FAILURE | 158335232 | CKCFG | Cannot start software | Startup | Check configuration to ensure SIP transports are configured properly (ie. no port conflicts) | Failed to initialize software |

### 10.2 Severity – Error (LOG_1)

| Name | OID suffix | Response Code | Impacts | Causes | Recommended Actions | Message |
|---|---|---|---|---|---|---|
| VGLOG-CCXML-ERROR-INIT-FAILURE | 158335232 | CKCFG | Cannot start software | Startup | Check configuration to ensure SIP transports are configured properly (ie. no port conflicts) | (Component) initialization failure |
| VGLOG-CCXMLI-ERROR-INIT-FAIL | 159383808 | CKCFG | Cannot start software | Startup | Check ccxml interpreter configuration parameters to ensure correctness | CCXML Interpreter initialization fail |

### 10.3 Severity – Warning (LOG_2)

| Name | OID suffix | Response Code | Impacts | Causes | Recommended Actions | Message |
|---|---|---|---|---|---|---|
| VGLOG-CCXML-WARNING-NOLICENSE | 158335489 | CKCFG | Incoming/outgoing connections will fail | Not enough licenses create connections | Review capacity need; contact VoiceGenie to acquire more licenses | Failed to acquire license for connection |

## 11.0 Appendix C – CCXML Platform Log_4 (INFO) events

The following table describes all CCXML Platform events that are related to a connection, dialog, or conference.

| INFO event id | Description |
|---|---|
| 0x9700400 | Inbound connection alerting |
| 0x9700401 | Connection progressing |
| 0x9700402 | Outbound connection requested |
| 0x9700403 | Connection connected |
| 0x9700404 | Connection event |
| 0x9700405 | Redirect connection |
| 0x9700406 | Merge connections |
| 0x9700407 | Merge connection failed |
| 0x9700408 | Connection disconnect |
| 0x9700409 | Dialog perpare |
| 0x970040A | Dialog start |
| 0x970040B | Dialog prepared |
| 0x970040C | Dialog started |
| 0x970040D | Dialog terminate |
| 0x970040E | Dialog event |
| 0x970040F | Dialog transfer request |
| 0x9700410 | Conference created |
| 0x9700411 | Conference terminated |
| 0x9700412 | Conference event |
| 0x9700413 | Media established |
| 0x9700414 | Media modified |
| 0x9700415 | Media terminated |

## 12.0 Appendix D – CCXML Interpreter Log_4 (INFO) events

The following table described all CCXML interpreter events at INFO level:

| INFO event id | Description |
|---|---|
| 0x9800400 | CCXMLI initialized |
| 0x9800401 | CCXMLI uninitialized |

| | |
|---|---|
| 0x9800402 | A new CCXML session created |
| 0x9800403 | A CCXML session terminated |
| 0x9800404 | Failed to fetch document |
| 0x9800405 | Failed to parse document |
| 0x9800406 | Failed to compile document |
| 0x9800407 | Document initialization failed |
| 0x9800408 | Event not caught by application |
| 0x9800409 | Application log (by <log> tag) |
| 0x980040A | Error event generated by application |
| 0x980040B | Exceed maximum session limit |