Avaya Communication Manager API Overview

for the connector server and the API

03-300084
Issue 2
May 2004

Notice

Every effort was made to ensure that the information in this document was complete and accurate at the time of printing. However, information is subject to change.

Warranty

Avaya Inc. provides a limited warranty on this product. Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya's standard warranty language as well as information regarding support for this product, while under warranty, is available through the following Web site: http://www.avaya.com/support.

Preventing Toll Fraud

"Toll fraud" is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there may be a risk of toll fraud associated with your system and that, if toll fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya Fraud Intervention

If you suspect that you are being victimized by toll fraud and you need technical assistance or support, in the United States and Canada, call the Technical Service Center's Toll Fraud Intervention Hotline at 1-800-643-2353.

Disclaimer

Avaya is not responsible for any modifications, additions or deletions to the original published version of this documentation unless such modifications, additions or deletions were performed by Avaya. Customer and/or End User agree to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation to the extent made by the Customer or End User.

How to Get Help

For additional support telephone numbers, go to the Avaya support Web site: http://www.avaya.com/support. If you are:

- Within the United States, click the *Escalation Management* link. Then click the appropriate link for the type of support you need.
- Outside the United States, click the *Escalation Management* link. Then click the *International Services* link that includes telephone numbers for the international Centers of Excellence.

Providing Telecommunications Security

Telecommunications security (of voice, data, and/or video communications) is the prevention of any type of intrusion to (that is, either unauthorized or malicious access to or use of) your company's telecommunications equipment by some party.

Your company's "telecommunications equipment" includes both this Avaya product and any other voice/data/video equipment that could be accessed via this Avaya product (that is, "networked equipment").

An "outside party" is anyone who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf. Whereas, a "malicious party" is anyone (including someone who may be otherwise authorized) who accesses your telecommunications equipment with either malicious or mischievous intent.

Such intrusions may be either to/through synchronous (time-multiplexed and/or circuit-based) or asynchronous (character-, message-, or packet-based) equipment or interfaces for reasons of:

- Utilization (of capabilities special to the accessed equipment)
- Theft (such as, of intellectual property, financial assets, or toll facility access)
- Eavesdropping (privacy invasions to humans)
- Mischief (troubling, but apparently innocuous, tampering)
- Harm (such as harmful tampering, data loss or alteration, regardless of motive or intent)

Be aware that there may be a risk of unauthorized intrusions associated with your system and/or its networked equipment. Also realize that, if such an intrusion should occur, it could result in a variety of losses to your company (including but not limited to, human/data privacy, intellectual property, material assets, financial resources, labor costs, and/or legal costs).

Responsibility for Your Company's Telecommunications Security

The final responsibility for securing both this system and its networked equipment rests with you - Avaya's customer system administrator, your telecommunications peers, and your managers. Base the fulfillment of your responsibility on acquired knowledge and resources from a variety of sources including but not limited to:

- Installation documents
- System administration documents
- Security documents
- Hardware-/software-based security tools
- Shared information between you and your peers
- Telecommunications security experts

To prevent intrusions to your telecommunications equipment, you and your peers should carefully program and configure:

- Your Avaya-provided telecommunications systems and their interfaces
- Your Avaya-provided software applications, as well as their underlying hardware/software platforms and interfaces
- Any other equipment networked to your Avaya products

TCP/IP Facilities

Customers may experience differences in product performance, reliability and security depending upon network configurations/design and topologies, even when the product performs as warranted.

Standards Compliance

Avaya Inc. is not responsible for any radio or television interference caused by unauthorized modifications of this equipment or the substitution or attachment of connecting cables and equipment other than those specified by Avaya Inc. The correction of interference caused by such unauthorized modifications, substitution or attachment will be the responsibility of the user. Pursuant to Part 15 of the Federal Communications Commission (FCC) Rules, the user is cautioned that changes or modifications not expressly approved by Avaya Inc. could void the user's authority to operate this equipment.

Product Safety Standards

This product complies with and conforms to the following international Product Safety standards as applicable:

Safety of Information Technology Equipment, IEC 60950, 3rd Edition including all relevant national deviations as listed in Compliance with IEC for Electrical Equipment (IECEE) CB-96A.

Safety of Information Technology Equipment, CAN/CSA-C22.2 No. 60950-00 / UL 60950, 3rd Edition

Safety Requirements for Customer Equipment, ACA Technical Standard (TS) 001 - 1997

One or more of the following Mexican national standards, as applicable: NOM 001 SCFI 1993, NOM SCFI 016 1993, NOM 019 SCFI 1998

The equipment described in this document may contain Class 1 LASER Device(s). These devices comply with the following standards:

- EN 60825-1, Edition 1.1, 1998-01
- 21 CFR 1040.10 and CFR 1040.11.

The LASER devices operate within the following parameters:

- Maximum power output: -5 dBm to -8 dBm
- Center Wavelength: 1310 nm to 1360 nm

Luokan 1 Laserlaite

Klass 1 Laser Apparat

Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposures. Contact your Avaya representative for more laser product information.

Electromagnetic Compatibility (EMC) Standards

This product complies with and conforms to the following international EMC standards and all relevant national deviations:

Limits and Methods of Measurement of Radio Interference of Information Technology Equipment, CISPR 22:1997 and EN55022:1998.

Information Technology Equipment – Immunity Characteristics – Limits and Methods of Measurement, CISPR 24:1997 and EN55024:1998, including:

- Electrostatic Discharge (ESD) IEC 61000-4-2
- Radiated Immunity IEC 61000-4-3
- Electrical Fast Transient IEC 61000-4-4
- Lightning Effects IEC 61000-4-5
- Conducted Immunity IEC 61000-4-6
- Mains Frequency Magnetic Field IEC 61000-4-8
- Voltage Dips and Variations IEC 61000-4-11
- Powerline Harmonics IEC 61000-3-2
- Voltage Fluctuations and Flicker IEC 61000-3-3

Federal Communications Commission Statement

Part 15:

Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Part 68: Answer-Supervision Signaling

Allowing this equipment to be operated in a manner that does not provide proper answer-supervision signaling is in violation of Part 68 rules. This equipment returns answer-supervision signals to the public switched network when:

- answered by the called station,
- answered by the attendant, or
- routed to a recorded announcement that can be administered by the customer premises equipment (CPE) user.

This equipment returns answer-supervision signals on all direct inward dialed (DID) calls forwarded back to the public switched telephone network. Permissible exceptions are:

- A call is unanswered.
- A busy tone is received.
- A reorder tone is received.

Avaya attests that this registered equipment is capable of providing users access to interstate providers of operator services through the use of access codes. Modification of this equipment by call aggregators to block access dialing codes is a violation of the Telephone Operator Consumers Act of 1990.

REN Number

For MCC1, SCC1, CMC1, G600, and G650 Media Gateways:

This equipment complies with Part 68 of the FCC rules. On either the rear or inside the front cover of this equipment is a label that contains, among other information, the FCC registration number, and ringer equivalence number (REN) for this equipment. If requested, this information must be provided to the telephone company.

For G350 and G700 Media Gateways:

This equipment complies with Part 68 of the FCC rules and the requirements adopted by the ACTA. On the rear of this equipment is a label that contains, among other information, a product identifier in the format US:AAAEQ##TXXXX. The digits represented by ## are the ringer equivalence number (REN) without a decimal point (for example, 03 is a REN of 0.3). If requested, this number must be provided to the telephone company.

For all media gateways:

The REN is used to determine the quantity of devices that may be connected to the telephone line. Excessive RENs on the telephone line may result in devices not ringing in response to an incoming call. In most, but not all areas, the sum of RENs should not exceed 5.0. To be certain of the number of devices that may be connected to a line, as determined by the total RENs, contact the local telephone company.

REN is not required for some types of analog or digital facilities.

Means of Connection

Connection of this equipment to the telephone network is shown in the following tables.

For MCC1, SCC1, CMC1, G600, and G650 Media Gateways:

Manufacturer's Port Identifier	FIC Code	SOC/REN/ A.S. Code	Network Jacks
Off premises station	OL13C	9.0F	RJ2GX, RJ21X, RJ11C
DID trunk	02RV2-T	0.0B	RJ2GX, RJ21X
CO trunk	02GS2	0.3A	RJ21X
	02LS2	0.3A	RJ21X
Tie trunk	TL31M	9.0F	RJ2GX
Basic Rate Interface	02IS5	6.0F, 6.0Y	RJ49C
1.544 digital interface	04DU9-BN	6.0F	RJ48C, RJ48M
	04DU9-IKN	6.0F	RJ48C, RJ48M
	04DU9-ISN	6.0F	RJ48C, RJ48M
120A4 channel service unit	04DU9-DN	6.0Y	RJ48C

For G350 and G700 Media Gateways:

Manufacturer's Port Identifier	FIC Code	SOC/REN/ A.S. Code	Network Jacks
Ground Start CO trunk	02GS2	1.0A	RJ11C
DID trunk	02RV2-T	AS.0	RJ11C
Loop Start CO trunk	02LS2	0.5A	RJ11C
1.544 digital interface	04DU9-BN	6.0Y	RJ48C
	04DU9-DN	6.0Y	RJ48C
	04DU9-IKN	6.0Y	RJ48C
	04DU9-ISN	6.0Y	RJ48C
Basic Rate Interface	02IS5	6.0F	RJ49C

For all media gateways:

If the terminal equipment (for example, the media server or media gateway) causes harm to the telephone network, the telephone company will notify you in advance that temporary discontinuance of service may be required. But if advance notice is not practical, the telephone company will notify the customer as soon as possible. Also, you will be advised of your right to file a complaint with the FCC if you believe it is necessary.

The telephone company may make changes in its facilities, equipment, operations or procedures that could affect the operation of the equipment. If this happens, the telephone company will provide advance notice in order for you to make necessary modifications to maintain uninterrupted service.

If trouble is experienced with this equipment, for repair or warranty information, please contact the Technical Service Center at 1-800-242- 2121 or contact your local Avaya representative. If the equipment is causing harm to the telephone network, the telephone company may request that you disconnect the equipment until the problem is resolved.

A plug and jack used to connect this equipment to the premises wiring and telephone network must comply with the applicable FCC Part 68 rules and requirements adopted by the ACTA. A compliant telephone cord and modular plug is provided with this product. It is designed to be connected to a compatible modular jack that is also compliant. It is recommended that repairs be performed by Avaya certified technicians.

The equipment cannot be used on public coin phone service provided by the telephone company. Connection to party line service is subject to state tariffs. Contact the state public utility commission, public service commission or corporation commission for information.

This equipment, if it uses a telephone receiver, is hearing aid compatible.

Canadian Department of Communications (DOC) Interference Information

This Class A digital apparatus complies with Canadian ICES-003.

Cet appareil numérique de la classe A est conforme à la norme NMB-003 du Canada.

This equipment meets the applicable Industry Canada Terminal Equipment Technical Specifications. This is confirmed by the registration number. The abbreviation, IC, before the registration number signifies that registration was performed based on a Declaration of Conformity indicating that Industry Canada technical specifications were met. It does not imply that Industry Canada approved the equipment.

Declarations of Conformity

United States FCC Part 68 Supplier's Declaration of Conformity (SDoC)

Avaya Inc. in the United States of America hereby certifies that the equipment described in this document and bearing a TIA TSB-168 label identification number complies with the FCC's Rules and Regulations 47 CFR Part 68, and the Administrative Council on Terminal Attachments (ACTA) adopted technical criteria.

Avaya further asserts that Avaya handset-equipped terminal equipment described in this document complies with Paragraph 68.316 of the FCC Rules and Regulations defining Hearing Aid Compatibility and is deemed compatible with hearing aids.

Copies of SDoCs signed by the Responsible Party in the U. S. can be obtained by contacting your local sales representative and are available on the following Web site: http://www.avaya.com/support.

All Avaya media servers and media gateways are compliant with FCC Part 68, but many have been registered with the FCC before the SDoC process was available. A list of all Avaya registered products may be found at: http://www.part68.org by conducting a search using "Avaya" as manufacturer.

European Union Declarations of Conformity

CE

Avaya Inc. declares that the equipment specified in this document bearing the "CE" (*Conformité Europeénne*) mark conforms to the European Union Radio and Telecommunications Terminal Equipment Directive (1999/5/EC), including the Electromagnetic Compatibility Directive (89/336/EEC) and Low Voltage Directive (73/23/EEC). This equipment has been certified to meet CTR3 Basic Rate Interface (BRI) and CTR4 Primary Rate Interface (PRI) and subsets thereof in CTR12 and CTR13, as applicable.

Copies of these Declarations of Conformity (DoCs) can be obtained by contacting your local sales representative and are available on the following Web site: http://www.avaya.com/support.

Japan

This is a Class A product based on the standard of the Voluntary Control Council for Interference by Information Technology Equipment (VCCI). If this equipment is used in a domestic environment, radio disturbance may occur, in which case, the user may be required to take corrective actions.

この装置は、情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

Third-party license agreements:

Portions of this product include technology used under license as listed below, and are copyright of the respective companies and/or their licensors.

A) This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

The Apache Software License, Version 1.1

Copyright (c) 2000 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

“This product includes software developed by the Apache Software Foundation (http://www.apache.org/).”

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names “Apache” and “Apache Software Foundation” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called “Apache”, nor may “Apache” appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED ‘‘AS IS’’ AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Portions of this software are based upon public domain software originally written at the National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign.

B) Castor software

This product contains software developed by the Exolab Group (http://www.exolab.org).

Castor Copyright (C) 1999-2001 Intalio, Inc. All Rights Reserved.

Redistribution and use of this software and associated documentation (“Software”), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name “ExoLab” must not be used to endorse or promote products derived from this Software without prior written permission of ExoLab Group. For written permission, please contact info@exolab.org.

4. Products derived from this Software may not be called “ExoLab” nor may “ExoLab” appear in their names without prior written permission of ExoLab Group. Exolab is a registered trademark of ExoLab Group.

5. Due credit should be given to the ExoLab Group (http://www.exolab.org).

THIS SOFTWARE IS PROVIDED BY INTALIO, INC. AND CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL INTALIO, INC. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

C) Java Service Wrapper Copyright (c) 1999, 2003 TanukiSoftware.org

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sub-license, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

To order copies of this and other documents:

Call: Avaya Publications Center
Voice 1.800.457.1235 or 1.207.866.6701
FAX 1.800.457.1764 or 1.207.626.7269

Write: Globalware Solutions
200 Ward Hill Avenue
Haverhill, MA 01835 USA
Attention: Avaya Account Management

E-mail: totalware@gwsmail.com

For the most current versions of documentation, go to the Avaya support Web site: http://www.avaya.com/support.

Contents

About this book

Intended audience

This book is intended for telecom managers, systems engineers, architects, application developers, and IT managers who already have knowledge of the capabilities of Communication Manager and the software needs of their organization that Communication Manager API might fulfill. These users will read this book to decide whether their organizations should use Communication Manager API to develop Communication Manager-based applications and what kinds of applications to develop. Users will also be able to select their connector server machine.

Application developers should also read this book before starting to develop an application, as this book provides an overview of the capabilities of the API and an explanation of the services provided by the Java and XML interfaces.

Summary of information covered in this book

- A description of Communication Manager API, including explanations of its component software. This section also explains the capabilities of the API and the types of applications that can be developed.

  See What is Avaya Communication Manager API? on page 13.
- A further explanation of the device and media control that the API provides, including a call model for device control.

  See Capabilities of the API on page 17.
- An explanation of the services provided by the API.

  See API services on page 25.
- An overview of the software architecture

  See Software architecture on page 29.
- Supported hardware and software configurations, including guidelines and examples.

  See Hardware/software configurations on page 30.
- Prerequisites, including:
  - Hardware requirements
  - Software requirements
  - Communication Manager/media servers supported
  - Required network characteristics
  - Development environment

  See Prerequisites on page 37.

Conventions used in this book

The following typefaces are used in this document:

Matter	Typeface and syntax	Example
SAT commands	• Bold for **literals** • Bold italic for ***variables***	**change signaling-group** *x*
SAT screen input and output	• Bold for **input** • Constant width for `output (screen displays and messages)`	For `DTMF Over IP`, enter **in-band-g711** The message `Command successfully completed` appears.
Linux commands	• Constant-width bold for **`literals`** • Constant-width bold italics for ***`variables`***	As root, execute **`rpm -Uv cmapi-server-`*`release#-`*`1.noarch.rpm`**

Linux output and screen displays	`Constant width`	The following lines are a sample output from the **`top`** command. The **Mem** field equals the memory that is available:
		`Mem:998888K av,986416K used, 12472K free,0K shrd, 116536K buff`
		`Swap:2024180K av,76K used, 2024104K free800708K cached`

Linux interface	Bold for **menu selections, tabs, buttons,** and **field names**	At the **Installation Type** prompt, select **Install** and **Server**.

Related documents

These books comprise the Communication Manager API document set:

- *Avaya Communication Manager API Overview (03-300084)*
- *Avaya Communication Manager API Quick Start (03-300089)*
- *Avaya Communication Manager API Installation and Administration (03-300085)*
- *Avaya Communication Manager API Management (03-300086)*
- *Avaya Communication Manager API Java Programmer's Guide (03-300087)*
- *Avaya Communication Manager API Java Programmer's Reference (Javadoc)*
- *Avaya Communication Manager API XML Programmer's Guide*
- *Avaya Communication Manager API XML Programmer's Reference (XMLdoc)*
- *Avaya Communication Manager API Media Stack Programmer's Reference (Javadoc)*
- *Avaya Communication Manager API Release Notes (03-300088)*

You can find all these documents online on the Avaya Developer Connection Web site (http://www.devconnectprogram.com) and on the Avaya Support Centre Web Site (http://www.avaya.com/support).

For CSTA details not found in the programmer's references or the programmer's guides, see the following CSTA documents. They are in the Publications section of the ECMA Web Site (http://www.ecma-international.org/):

- *ECMA-269: Services for Computer Supported Telecommunications Applications (CSTA) Phase III*
- *ECMA-323: XML Protocol for Computer Supported Telecommunications Applications (CSTA) Phase III*
- *ECMA Technical Report TR/72: Glossary of Definitions and Terminology for Computer Supported Telecommunications Applications (CSTA) Phase III*

The following books from the Communication Manager documentation set provide additional information about administering Communication Manager for Communication Manager API. They are on the Avaya Support Centre Web Site (http://www.avaya.com/support).

- *Administrator's Guide for Avaya Communication Manager*, 555-233-506
  - — Issue 7 for Communication Manager 2.0
  - — Issue 8 for Communication Manager 2.1
- *Administration for Network Connectivity for Avaya Communication Manager*, 555-233-504
  - — Issue 7 for Communication Manager 2.0
  - — Issue 8 for Communication Manager 2.1

Tell us what you think

Let us know what you like or do not like about this book. Although we cannot respond personally to all your feedback, we promise we read each response we receive.

Please email feedback to devconcmapi@avaya.com.

1 What is Avaya Communication Manager API?

Description

Communication Manager API is a software-only connector that provides connectivity between applications and Communication Manager. This connector provides an open, standards-based, Java™ and XML programming interface for developing applications that take advantage of the rich feature set of Avaya Communication Manager software.

In a Communication Manager API application, software objects, called “CMAPI softphones”, are used to represent softphone-enabled, Communication Manager telephones or extensions. Using the API, the application first requests exclusive or shared control of a telephone/extension in order to be given a CMAPI softphone object. The application can then perform telephone operations on the CMAPI softphone object. This, in turn, causes the connector to make requests of Communication Manager to perform those operations as if they were manually occurring on a physical telephone. In addition, Communication Manager asynchronously notifies the connector of any event that occurs on the telephone or extension, such as when a lamp becomes lit. The connector then notifies the application of the event. In this manner, an application can make calls, receive calls, record calls, send announcements, apply tones, detect digits, and redirect the media.

Components of Communication Manager API

Communication Manager API consists of:

- A connector client API library, also referred to as simply “the Java API.”

  The Java API provides a Java class library for Java applications. This interface provides a set of device- and media-control services. It transmits the application's requests to the connector server and then translates the connector server's responses and asynchronous events for the application. The Java library is part of the software development kit (SDK). Sample Java source code, a Java programmer’s guide, and a Javadoc are provided to help the developer in writing to this library.

  For an explanation of the API capabilities, see Capabilities of the API on page 17.

- XML schema definition (XSD) files, referred to as simply “the XML protocol.”

  These XSD files allow applications to be written directly to the XML interface. Applications can be written in a variety of languages. The applications format the XML requests and send them over an IP connection to the server. The applications also process XML responses and events from the server. An XML programmer’s guide and an XML online reference (referred to hereafter as the XMLdoc) are provided to help the developer in writing directly to the XML interface.

- The Communication Manager API server-side runtime software, which communicates with both the application and Communication Manager over an IP connection. We refer to this software as the connector server software,

  NOTE:
  In this document set, we often refer to the connector server software as simply the “connector server.” We refer to the hardware platform on which the software resides as the “connector server machine.”

The connector server software runs on a Linux machine that is separate from the media server that Communication Manager runs on.

> **NOTE:**
> Client applications can reside on the connector server machine (not recommended) or on a separate machine. See Architecture and configurations on page 29 to see where the application resides in different configurations.

Client application capabilities

The connector client API provides a form of device and media control for developing applications that can:

- Instantiate a CMAPI softphone that gains exclusive or shared control of a softphone-enabled, Communication Manager telephone or extension.
- Activate or control a device's physical elements. This includes pressing buttons, going off hook, and going on hook.
- Determine the status of a CMAPI softphone's physical elements. This includes obtaining the status of buttons, displays, lamps, the hookswitch, the message waiting indicator, and the ringer.
- Detect events on a CMAPI softphone's physical elements. This includes detecting events on the display, lamps, and ringer.
- Process the media in any of the following ways:
  - Record media from a call into a Wave file. Wave files can be in pulse code modulation (PCM), G.711, G.729, or G.729A codec Wave file formats.
  - Dub a recording with the contents of another compatible Wave file.
  - Play a voice announcement or tone, which is prerecorded in a Wave file, on a device. A list of Wave files can also be played in succession.
  - Detect in-band or out-of-band dual-tone multi-frequency (DTMF) digits (In-band DTMF detection is not available for G.729/G.729A codecs). Tones can also be buffered for the application.

  For all of the above media processing, the Wave files are located on the connector server machine.
- Direct the media to another IP address to bypass the connector server media processing.
- Collect tones using specified criteria.

The Avaya SDK also provides a light-weight RTP Stack for audio as a library. Applications can use the stack to set up and tear down audio RTP sessions. Applications can also get access to the RTP streams. The stack currently supports G.711 U-Law, G.711 A-Law, G.729 and G.729A codecs. The stack does not include specialized stack features like jitter buffer, gain control, echo control, transcoding, packet loss concealment, and QOS support. The stack also does not support mixing of audio streams from multiple media sources.

For more information about the media stack, see the *Media Stack Programmer's Reference* (Javadoc).

Applications that can be developed

Types of applications

Communication Manager API can be used to create many types of applications, for example:

- Messaging applications
- Specialized console applications for targeted markets such as hospitality, health care, and tenant services
- IP call recording applications
- Interactive voice response (IVR) applications
- Applications that integrate computer-telephone applications
- Call logging applications

Example applications

Some of the possible applications include:

- **IP call recording** applications can provide the ability for end users to record phone conversations by pressing a pre-administered button on their IP phone. The IP call recording application can be developed for various types of recording, for example, multi-user call recording, executive call recording on demand, malicious call recording.
- **IP softphone** applications can be developed that run on users' Windows desktop. An application can take exclusive control of the users' regular telephone extensions so that they can make and receive calls from their PCs and have the audio streamed to/from their PC speaker/microphone. Alternatively, using shared control, an application can allow the user to make and receive calls from their regular telephone using their PC to control that phone.
- **Interactive voice response** (IVR) applications can be created for businesses such as banks or customer service centers. These IVR applications could perform such tasks as detecting button pushes by a caller, playing recordings, and recording conversations. For example, callers might be presented with options such as recording a message or being transferred to another extension. The callers select the preferred option by pressing the specified button on their telephone
- **Click-to-call** applications can be created that perform directory lookups. Applications can provide helpful GUIs, such as a call log GUI to return a call and/or a directory lookup GUI to make a call.

2 Capabilities of the API

This chapter further explains the device and media control that the API provides for client applications. It includes explanations of:

- Device and media control
- Controllable telephone types
- Capacities

It also provides a call model for device and media control.

Device and media control

Both physical devices and CMAPI softphones can be controlled from this API. Calls are made and received on these devices by controlling and observing the physical aspects of the softphone of the device, such as:

- pressing buttons
- going off and on hook
- observing the lamps, ringer, and display

Registering a device gives the application either exclusive or shared control of the device:

- **Exclusive control mode** gives all control of the device to the application including control of the media stream. Exclusive control must be used by applications that need to do any of the following:
  - record media with Voice Unit Services
  - play announcements or messages with Voice Unit Services
  - detect or collect DTMF tones with Tone Detection Services or Tone Collection Services
  - control the media (see Table 2, Media control modes, on page 19).

  An application may take exclusive control of either:
  - a physical telephone's extension
  - an extension that has no physical telephone associated with it

  See also Table 3, Telephone configurations controllable by Communication Manager API, on page 20.
- **Shared control mode** gives control to both the telephone and the application. Shared control must be used by applications that need to monitor and control a physical telephone.

  In shared control mode:
  - An application may take shared control of only a physical telephone, not an extension without hardware.
  - No media is delivered to the CMAPI softphone.

The following table shows what the physical telephone and application can do in each of the control modes.

Table 1: Device control modes

Capabilities	Shared control		Exclusive control	
	Physical telephone	Application	Physical telephone	Application
Initiate action on device (e.g., press buttons)	Yes	Yes	No	Yes
Be notified of status changes to device (e.g., hear ringback or receive ringback event)	Yes	Yes	No	Yes
Receive incoming media stream and send message or voice via outgoing media stream	Yes	No	No	Yes
Be notified of actions taken on device	No[1]	No[2]	N/A	Sometimes[3]

1 In shared control mode, the user of a telephone is not notified of actions initiated by an application except through resulting status changes to the device's lamps and display.

2 In shared control mode, the application is not notified of actions initiated by a user of the telephone except by status changes to the device's hookswitch, lamps, ringer, and display.

3 In exclusive control mode, even though the application is initiating all actions on the device, the application is not always notified when the switch has completed performing those actions except by status changes to the device's hookswitch, lamps, ringer, and display. More specifically, the application *is* notified when its off hook request has been fulfilled, but it is *not* notified when a button press request has been fulfilled.

Media control modes

When a device is registered by an application in exclusive control mode, the application has access to the real-time protocol (RTP) media stream coming into and going out from the softphone. There are three media modes available in exclusive control mode: server media, client media, and telecommuter. When the device is registered in shared control mode, then only the telephone has control of the RTP stream. The following table, Media control modes, explains the different media modes and the media access for the call control modes.

Table 2: Media control modes

Control mode	Media mode	Who handles media	How it works
Exclusive control	Server media	Connector server	The connector server handles the media coming to the softphone and going out from the softphone. In this mode, the application uses the API services called Voice Unit Services to record and play media. The application selects this mode at the time it registers the softphone by letting the local media RTP address default to the connector server.
	Client media	Application	The application handles the media coming to the softphone and going out from the softphone. The application selects this mode at the time it registers the softphone by specifying the local media RTP address to where the RTP stream should be sent.
	Telecommuter	Telecommuter phone	The media is given to the real phone (telecommuter phone) only. This phone may be within the Communication Manager domain or outside of the Communication Manager domain.
Shared control	Telephone media	Telephone	The media stream is given to the telephone only.

The RTP parameters that an application can control or state preferences for at registration time are:

- Local RTP and RTCP addresses
- Coder/decoder (codec): G.711 A–law, G.711 Mu–law, G.729, G.729A

Controllable telephone types

The connector can control the following types of Communication Manager telephones and extensions when they are administered for softphone access:

Table 3: Telephone configurations controllable by Communication Manager API

Telephone configuration	Administered set type	Administered port address	Comments
DCP telephone	Any DCP type that can be administered for softphone access	Physical port address, such as 1B0201.	If an application requests exclusive control, the telephone either goes dead or becomes a TTI set, depending on how it is administered. After the application relinquishes control, control goes back to the DCP telephone. If an application requests shared control, the telephone stays active. (Communication Manager 2.0 and higher)
DCP extension administered without hardware (AWOH)	Any DCP type that can be administered for softphone access	Port address set to “X”.	
IP telephone that is logged in	Any IP type that can be administered for softphone access	Internal software port address, such as S00031, is automatically assigned.	If an application requests exclusive control, the IP telephone is logged off. When the application relinquishes control, the IP telephone must be powered off and on again to be logged back on. If an application requests shared control, the telephone stays active. (An application can gain shared control of an IP phone only if the IP phone is connected to Communication Manager 2.1 or later.)
IP extension without hardware (i.e., no IP telephone logged in using that extension)	Any IP type that can be administered for softphone access	Internal software port address, such as S00031, is automatically assigned.	

Capacities

The number of simultaneous active calls that your application can expect to handle depends on many factors, such as:

- what else is running on your application machine
- what else is running on the connector server
- the processor speed of the application machine and connector server
- the amount of Communication Manager IP traffic and amount of IP resources (such as C-LANs) to handle the traffic
- the amount of other IP network traffic
- the combination and timing of service requests your application makes
- your application's demand for VoIP resources relative to the VoIP resources available on Communication Manager
- the codec used /packet size for media
- media mode used

In lab tests, the following results were obtained for our call recording application and station registration only (no call recording). These results were obtained using a remote client proxy.

Table 4: Performance table

	Call recording (server media)		No recording (no media)[1]
	G7111-Mu 20ms packets	G727 60ms packets	
Maximum # of stations	75	75	1000
Effective BHCC	4500	4500	20000

1 The "no media" case is not truly without media. It was "client media" but the packets were dropped.

Call model for device and media control

In an application, device IDs are used to represent softphone-enabled Communication Manager telephones or extensions. Using the API, the application first registers a device as a Communication Manager telephone/extension. This registration can be exclusive-control mode or shared-control mode. (See Device and media control on page 17 for explanations of these control modes.) The application can then perform telephone operations on the device. This in turn causes the connector to make requests of Communication Manager to perform those operations. In shared-control mode, Communication Manager treats these operations as though they came from the associated physical telephone. In addition, Communication Manager asynchronously notifies the connector of any event that occurs on the telephone or extension, such as when a lamp becomes lit. The connector then notifies the application of the event. In shared-control mode these events are sent to both the connector server/application and the associated physical telephone.

The telephony operations are performed on the softphone. For example, a button press request from the application means that the application wishes to simulate a button press on the softphone, and a display updated event from Communication Manager indicates to the application that the softphone display has changed. An application can:

- make calls from the softphone
- receive calls at the softphone
- record media coming into the softphone (only in exclusive control - server media mode)
- play announcements from the softphone (only in exclusive control - server media mode)
- play tones from the softphone (only in exclusive control - server media mode)
- detect digits coming into the softphone
- redirect the softphone's media (only in exclusive control)

> **NOTE:**
> When performing these operations in shared-control mode, the application affects both the softphone and the physical device.

Figure , , on page 22 shows the various types of messages used within the connector when using the Java API library.

Figure , , on page 23 shows the various types of XML requests used within the connector when using the XML protocol.

Figure 1: Connector message types using the Java API

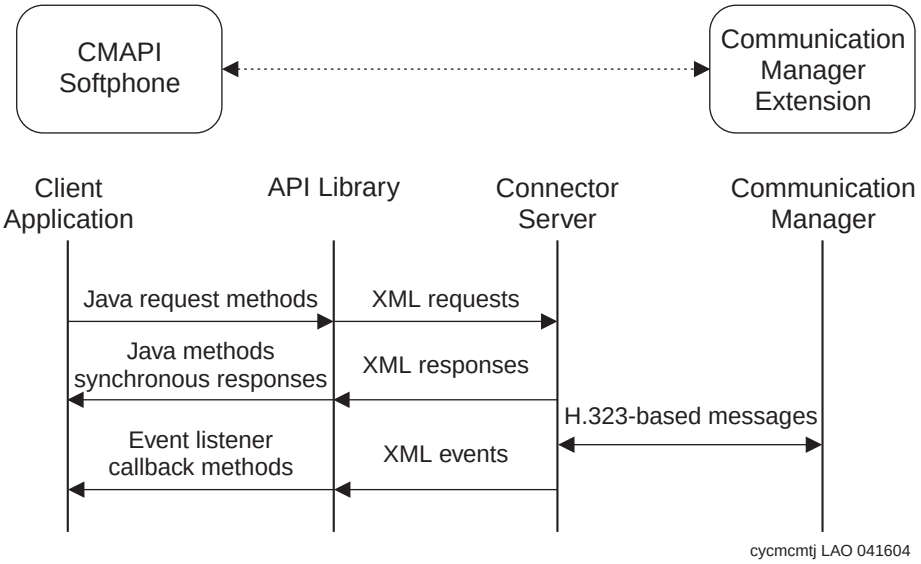


Figure 2: Connector message types using the XML protocol

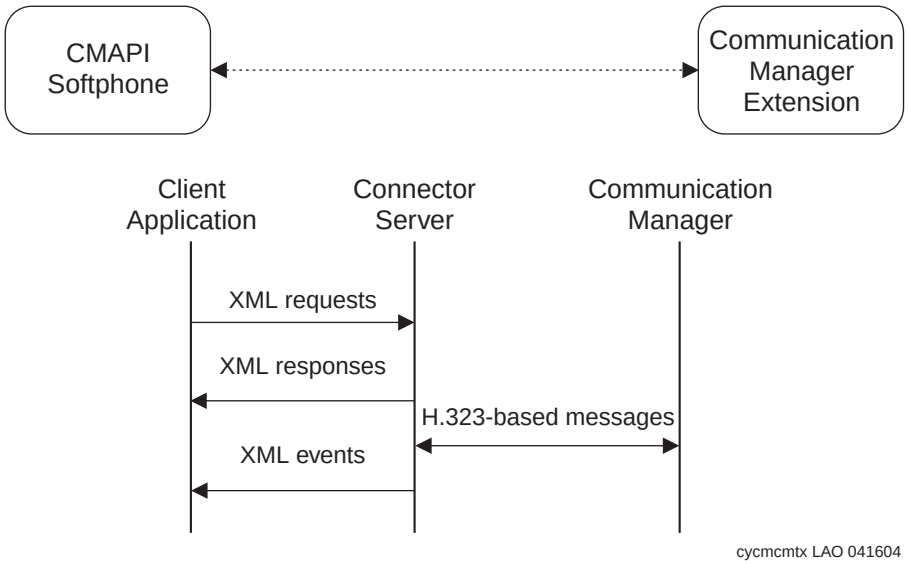
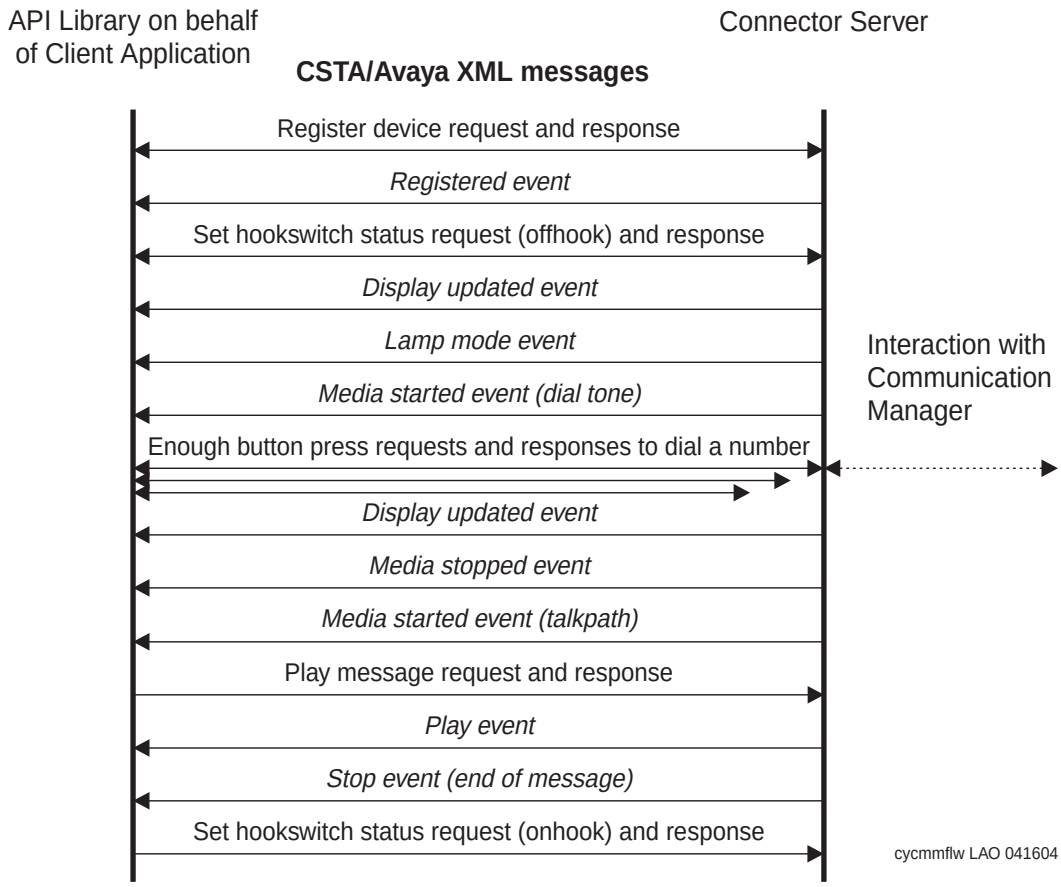


Figure 3, Example message flow (for exclusive control) between the application and the connector server, on page 24, reflects in more detail the message flow used to perform the following simple telephony steps on a softphone:

1. Register a softphone
2. Take the softphone offhook
3. Dial a number from the softphone
4. Play a message to the called party
5. Hang up the softphone

Figure 3: Example message flow (for exclusive control) between the application and the connector server



CSTA standard architecture

Communication Manager API supports the ECMA telephony standard called Computer Supported Telecommunications Applications (CSTA) Phase III. This standard is specified in publication ECMA-269, Services for Computer Supported Telecommunications Applications (CSTA) Phase III, 5th Edition, December 2002 found at http://www.ecma-international.org. The Java API library provides a Java-binding to a subset of the CSTA XML protocol whose specifications can be found in publication ECMA-323, XML Protocol for Computer Supported Telecommunications Applications (CSTA) Phase III, 2nd Edition, December 2002. The XML protocol provides the supported XML subset directly to the application.

CSTA specifies a standard applications interface and XML protocol for OSI Layer 7 communication between a computing platform (such as your application) and a telecommunications network (such as Avaya Communication Manager).

3 API services

This chapter provides an overview of what CSTA services the API supports and what extensions Avaya has implemented. This API supports these telephony services:

- device registration and control
- call recording and message playing
- DTMF digit detection and collection

These services are provided through a Java API interface or the XML protocol. Some of the interfaces conform to the CSTA III standard (ECMA-269) and some are Avaya extensions to the CSTA standard.

CSTA specifies that for any given service some parameters are mandatory and some parameters are optional. To determine which of the optional parameters Avaya supports or which of the field values Avaya supports, refer to the methods or requests detailed in the programmer's references (Javadoc or XMLdoc).

> **NOTE:**
> The ECMA standards body requests that CSTA-compliant implementations reflect conformance to the standard through a *Protocol Implementation Conformance Statement* (PICS). The Communication Manager API PICS is reflected in the programmer's references.

This chapter lists:

For more detailed information about the supported CSTA Services and the Avaya extensions, see the programmer's guides and the programmer's references (Javadoc and XMLdoc).

Supported CSTA services

In CSTA, each service is defined to be a request that either comes from the application to the switch or from the switch to the application. This API, however, is based on a client/server model where the application is the client and the connector server software and the switch together act as the server. Thus, this API:

- allows an application to request services of a switch
- allows an application to request notification of asynchronous events on the switch

The following sets of CSTA services are supported in the Communication Manager API and described in the programmer's guides and the programmer's references (Javadoc and XMLdoc).:

Table 5: Supported CSTA services

Sets of supported CSTA services	Purpose
Physical Device Services	Provides the ability to monitor and control physical aspects of a device
Voice Unit Services	Provides ability to record messages coming to a device and play messages from a device
Monitoring Services	Provides ability to request notification of events that occur on a device

NOTE:
CSTA Data Collection Services were supported in the previous release but are deprecated in this release in favor of the new Avaya Tone Detection Services and Avaya Tone Collection Services. Services in deprecated status may still be used but will not be supported and may be removed in future releases. Applications should move to the new services.

Avaya extensions

The API provides extensions to CSTA that are meant to enhance the capabilities of CSTA and provide higher-level services and useful events that make development of telephony applications easier. The extensions are summarized in this section. More complete descriptions of each extension can be found in the programmer's guides and the programmer's references (Javadoc and XMLdoc).

The Avaya extensions have been implemented per the CSTA guidelines described in ECMA-269, section 28, "Vendor Specific Extensions Services and Events".

The Avaya extensions are listed below. They are further described in the programmer's guides and the programmer's references (Javadoc and XMLdoc).

Table 6: Avaya extensions to CSTA services

Avaya extension	Extends which CSTA service set	Purpose
Service Provider (Java only)	All	Provides a starting point to access all other CSTA and Avaya services.
Device Services	None	Provides an identifier for a given extension on a given switch.
Terminal Services and Events	None	Provides ability to gain exclusive or shared control over switch endpoints - also referred to as *device registration.*
Media Control Events	None	Provides the ability to be notified when the far-end RTP and RTCP parameters for a media stream change.
Extended Voice Unit Services	Voice Unit Services	Provides dubbing of recorded messages and other extensions.
Tone Detection Events	Replaces Data Collection Services	Detects DTMF tones and reports each tone as it is detected.
Tone Collection Services and Events	None	Detects DTMF tones and buffers them as requested before reporting them to the application.

4 Architecture and configurations

This section explains:

- The software architecture, including the interfaces between the connector server and Communication Manager and between the connector server and the client application
- The possible hardware/software configurations, including guidelines and graphics of example configurations

Software architecture

Interface from connector server to Communication Manager

The connector server communicates with Communication Manager over the LAN using an H.323-based protocol for signalling and the Real-Time Protocol (RTP) for media.

> **NOTE:**
> In order for Communication Manager to communicate with the connector server, there must be IP_API_A licenses on Communication Manager - one per registered device. For information on finding out how many IP_API_A licenses your system has, see the *Installation and Administration* guide.

When Communication Manager is on an S8300 media server, the connector server communicates with the processor C-LAN interface (PROCR) and the Voice Over IP (VoIP) media module or other motherboard VoIP. For all other supported Avaya media servers, the connector server connects to the C-LAN module and the media processor interface (MEDPRO).

Interface from connector server to client application

A client application can run:

— on a separate machine, which is then called an “application machine”

or

— (not recommended) on a connector server machine in a separate JVM

The connector server communicates with the client application using the XML messages over a TCP connection as specified in *ECMA-323*, Annex G “CSTA XML over TCP”. Both CSTA and Avaya XML messages are used. The connector server uses TCP port 4721 for its communication with the client application. See Example Configurations on page 31 for some examples of this architecture in different Communication Manager API configurations.

Hardware/software configurations

Configuration guidelines

Communication Manager API 2.1 configurations must follow these guidelines:

- Only one instance of the connector server software can reside on a connector server machine.
- More than one connector server machine can connect to an instance of Communication Manager:
  - For Linux-based media servers (S8300, S8500, S8710, HP380), up to 15 connector servers can connect to an instance of Communication Manager.
  - For non-Linux-based media servers (Csi, Si), one or two connector servers can connect to an instance of Communication Manager.
- One connector server machine can connect to multiple instances of Communication Manager.
- Applications can run either:
  - on an application machine (several applications can run on one machine if the machine has the resources to run these applications).

    or

  - (not recommended) on a connector server machine in a separate JVM

    NOTE:
    This is not a recommended configuration because running co-resident can cause performance problems depending on how CPU-, memory-, and disk-intensive the application is and the number of requests that are made of the connector server software.

- An application can drive several connector servers.

  This configuration might be needed if the application requires more call capacity than the connector server can provide.
- A Communication Manager API connector server can switch between an S8700/S8710's active and standby servers, but it cannot switch to an S8300 LSP.

  NOTE:
  The API cannot automatically switch over to a redundant connector server.

Example Configurations

The following figures show several possible configurations. These examples do not represent a complete list of the possible configurations. Because of the possible combinations of multiple connector server machines, applications, and application machines, many other configurations are possible.

Simple configuration

The following figure shows an application on a separate application machine. This configuration can have more than one application on the application machine.

Figure 4: Java API example

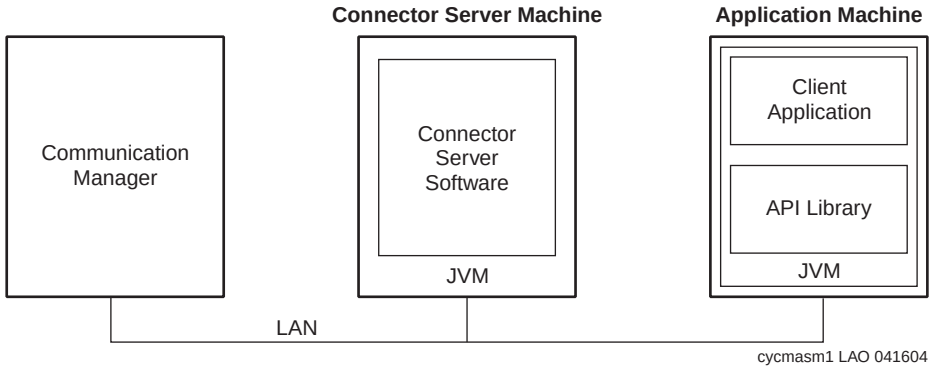
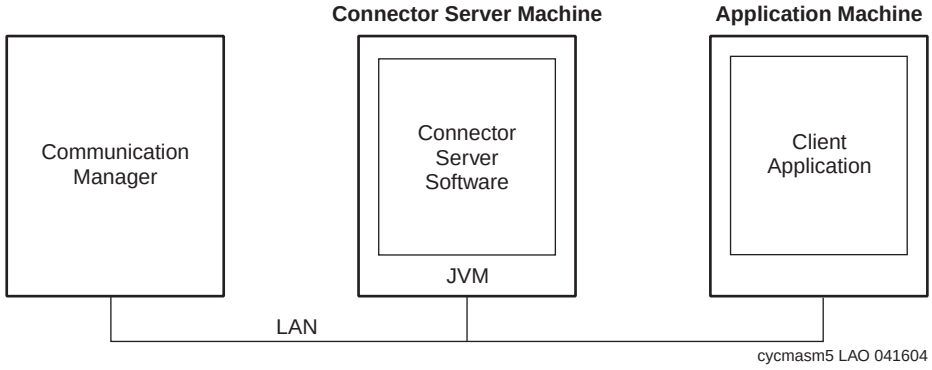


Figure 5: XML protocol example



Several connector server machines and one application machine

The following figures show several connector server machines and one application - on a separate application machine - driving all of the connector servers. This configuration might be used by applications that require more call capacity than a single connector server can provide.

Figure 6: Java API example

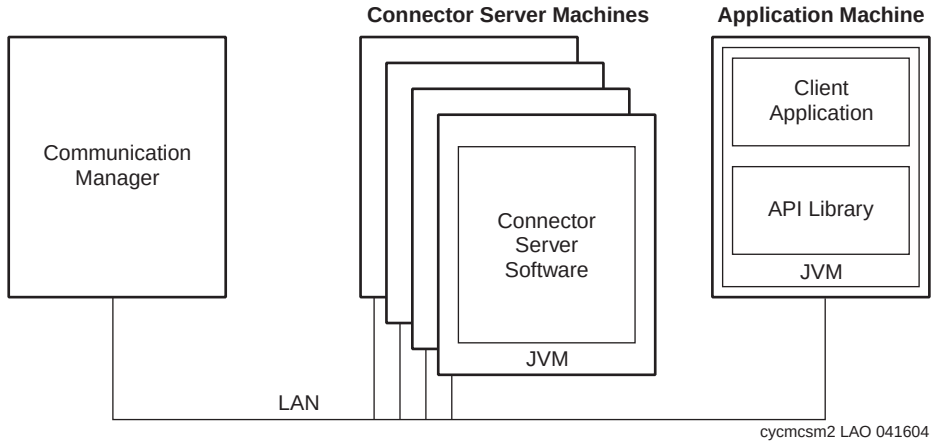
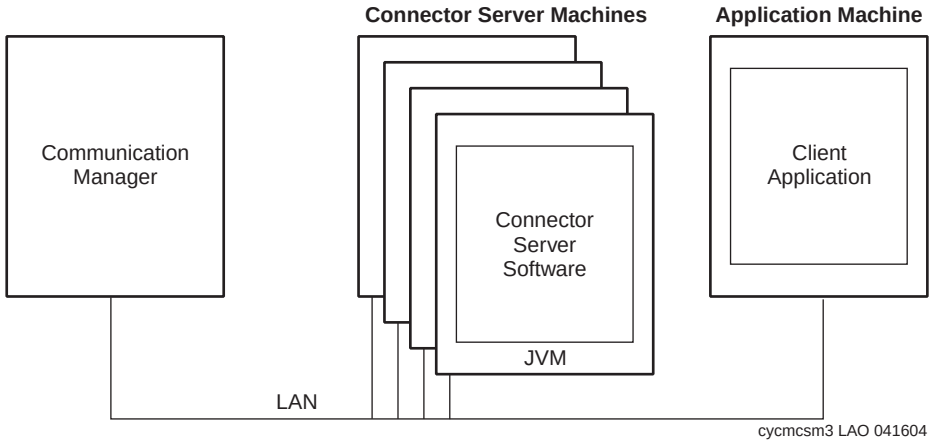


Figure 7: XML protocol example



One connector server machine and several application machines

The following figure shows one connector server machine and multiple applications, each on a separate application machine. Another possible configuration could have multiple application machines and multiple connector server machines, with one connector server machine connected to each application machine.

Figure 8: Java API example

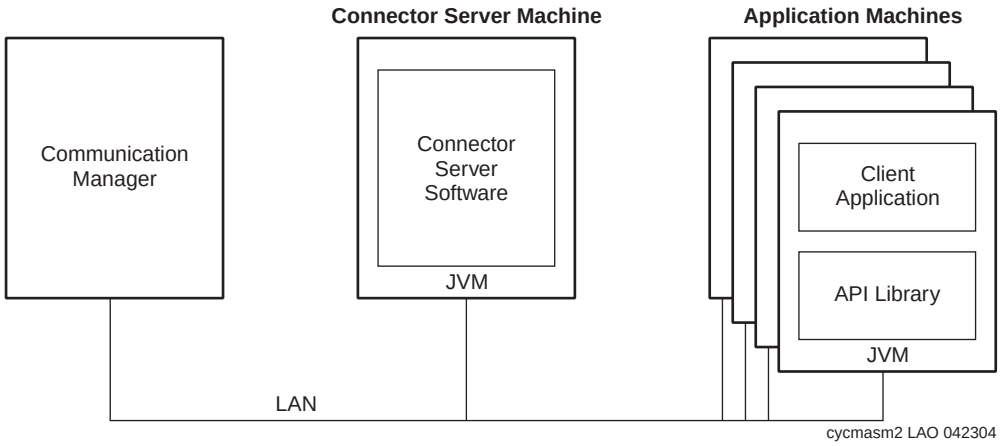
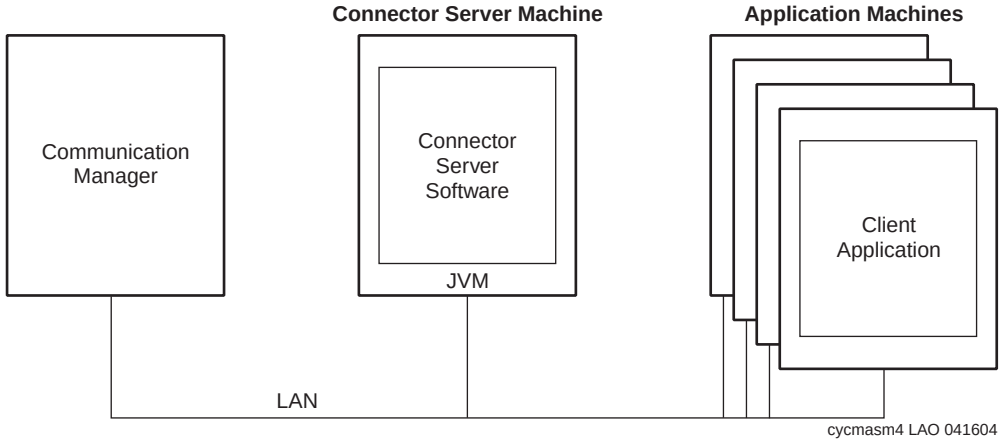


Figure 9: XML protocol example



Several instances of Communication Manager and one connector server machine

The following figure shows one connector server machine connected to multiple instances of Communication Manager. With this configuration, a single Communication Manager API application can control or monitor phones on several Communication Manager systems. This example shows the application on an application machine, but it could also be in a separate JVM on the connector server machine.

Figure 10: Java API example

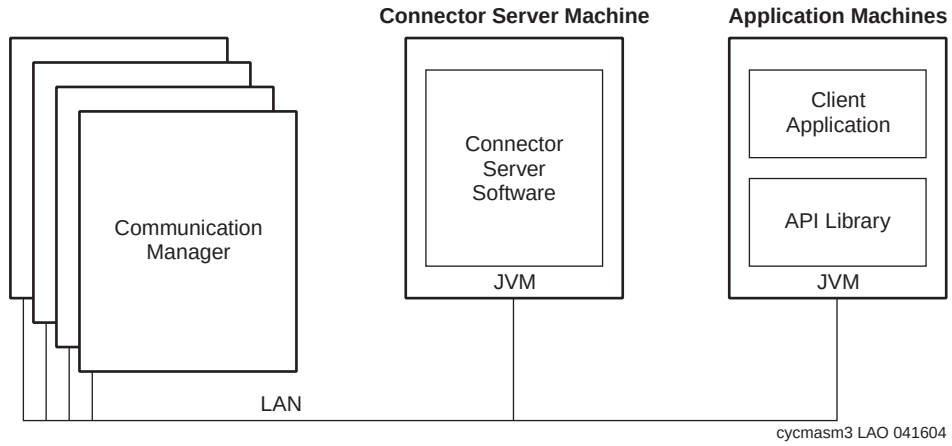
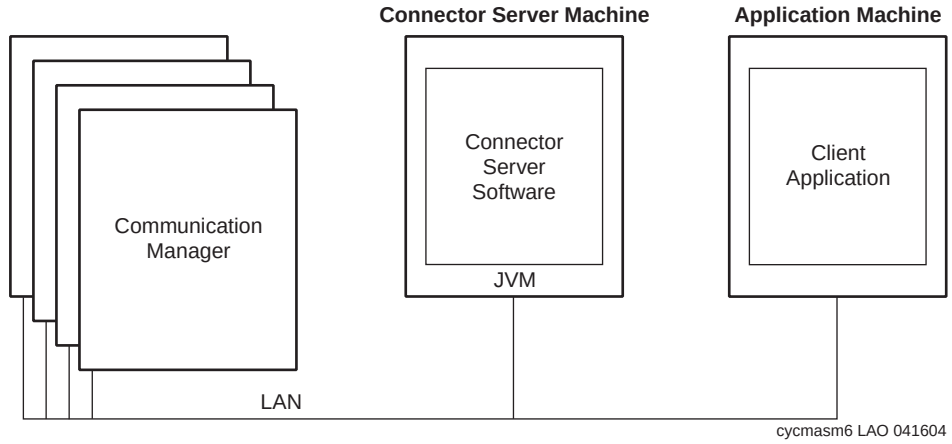


Figure 11: XML protocol example



5 Security considerations

Your application development organization has the responsibility of providing the appropriate amount of security for your particular application and/or recommending appropriate security measures to your application customers for the deployment of your application. Therefore you should be aware of the security measures that Communication Manager API already takes and what risks are known.

Communication Manager API has taken these security measures:

- The station password is required to register a device.
- Filenames specified for recorded files must be relative to the configured directory, their directories must already exist, and recordings cannot overwrite an existing file.

The following are still security risks:

- The signaling and bearer channels are unencrypted.
- The station password is unencrypted.
- The XML messages transmitted between the client application and the connector server software are unencrypted.
- There is no authentication on the JMX™ web console.

Avaya strongly recommends that the application machine, connector server machine, and Communication Manager machines reside on a private LAN.

6 Prerequisites

This chapter explains the prerequisites for running Communication Manager API connector server software. These prerequisites include:

Before installing the connector server software, make sure your system meets the following hardware, software, and network requirements.

Connector server hardware requirements

Recommended specifications	2.4-GHz Single-processor Pentium IV class machine with: • 1GB RAM • Hard Disk with at least 7200 rpm rating • 10/100 ethernet NIC port (should be duplex) **NOTE:** Available disk space is application-specific (the server requires at least 1 GB).

Connector server software requirements

Besides the connector server software package, the following software is needed to run the server.

Operating system	Red Hat Linux 9.0
Other software	Java™ 2 platform (J2SE™ 1.4.2_03 or later)

Instructions for installing this software are found in “Installing Linux” and “Installing Java” in the installation guide.

Application machine requirements

Applications can be developed and executed on the connector server (not recommended by Avaya) or on any machine that is capable of running the latest version of Sun's JVM™. Application development and execution use the Java 2 Platform, Standard Edition (J2SE) 1.4.2_03 or later.

For more information about application machines in the Communication Manager API configuration, see "Hardware/software configurations" in the *Overview.*

Communication Manager and media server requirements

You must have Communication Manager R2.0 or 2.1 running the latest load on your Avaya media server. Check on the latest load at http://www.avaya.com/support. Communication Manager R2.1 or later is required in order to perform shared control of IP telephones.

The Avaya media server running Communication Manager software must be one of the following. This list also explains the additional hardware requirements for these servers:

- S8700/8710 with
  - C-LAN circuit pack: TN799DP
  - One of the following:
    - Media processor circuit pack: TN2302AP
    - G700/G350 media gateway
- S8500 with
  - C-LAN circuit pack: TN799DP
  - One of the following:
    - Media processor circuit pack: TN2302AP
    - G700/G350 media gateway
- HP380 with
  - C-LAN circuit pack: TN799DP
  - One of the following:
    - Media processor circuit pack: TN2302AP
    - G700/G350 media gateway
- S8300 in
  - G700 or G350 media gateway
- DEFINITY® Server CSI with
  - C-LAN circuit pack: TN799DP
  - Media processor circuit pack: TN2302AP

- DEFINITY® Server SI in an MCC/SCC cabinet and with
  - — C-LAN circuit pack: TN799DP
  - — Media processor circuit pack: TN2302AP

    NOTE:
    Each VoIP resource on a gateway or port network has 64 available channels (except a G350 gateway, which has only 32 VoIP channels on its motherboard). For each IP endpoint in the call (including CMAPI endpoints), either one VoIP channel will be used (when using a G.711 codec) or two VoIP channels will be used (when using a G.729 codec).

Required network characteristics

Currently, Communication Manager API testing only covers LAN configurations. WAN configurations are not tested; therefore, WAN configurations are not recommended.

Avaya strongly recommends that the application machine, connector server machine, and Communication Manager machines reside on a private LAN.

Glossary

A

API
Application Programming Interface. A “shorthand” term in this documentation for the Java interface provided by the Communication Manager API. See also connector client API library

application machine
The hardware platform that the connector client API library and the client application are running on

B

BHCC
busy hour call capacity

C

client application
An application created using Communication Manager API

CMAPI softphone
Communication Manager API software objects that represent softphone-enabled, Communication Manager telephones or extensions

Communication Manager API
The product name. This includes the server-side runtime software (see connector server software) and the connector client API library. This term is never used to reference only the client API library.

connector
This describes the function of Communication Manager API

In this context, “connector” means software and communications protocol(s) that allow two disparate systems to communicate. Often used to provide open access to a proprietary system. In the case of Communication Manager API, the connector enables applications running on a computing platform to incorporate telephony functionality through interaction with Communication Manager.

connector client API library
The Communication Manager API Java API, also referred to as the API

connector server machine
The hardware platform that the connector server software is running on. In these books, the term “connector server” by itself never refers to the connector server machine. See connector server software.

connector server software
The Communication Manager API server-side runtime software, often referred to as the “connector server” in these documents

CSTA
Computer-Supported Telecommunications Applications

D

DMA
Direct memory access

E

ECMA
European Computer Manufacturers Association. A European association for standardizing information and communication systems in order to reflect the international activities of the organization.

H

hold time
The total length of time in minutes and seconds that a facility is used during a call

J

JDK
Java Developers Kit

J2SE
Java™ 2 Platform, Standard Edition

JSW
Java Service Wrapper

JVM
Java Virtual Machine

R

RPM
Red Hat Package Manager

S

SAT
System Access Terminal (for Communication Manager)

SDK
Software Development Kit. An SDK typically includes API library, software platform, documentation, and tools.

T

TCP
Transmission Control Protocol. A connection-oriented transport-layer protocol, IETF STD 7. RFC 793, that governs the exchange of sequential data. Whereas the Internet Protocol (IP) deals only with packets, TCP enables two hosts to establish a connection and exchange streams of data. TCP guarantees delivery of data, and also guarantees that packets are delivered in the same order in which the packets are sent.

TTI

Terminal Translation Initialization. This is a feature in Communication Manager that allows administrators, when initially administering new DCP stations, to not initially bind the extension number to a port. When the technician is installing the stations, they then use the TTI feature access code to bind the extension number to the station.

V

VoIP

Voice over IP. A set of facilities that use the Internet Protocol (IP) to manage the delivery of voice information. In general, VoIP means to send voice information in digital form in discrete packets instead of in the traditional circuit-committed protocols of the public switched telephone network (PSTN). Users of VoIP and Internet telephony avoid the tolls that are charged for ordinary telephone service.

XML

Extensible Markup Language

XSD

XML Schema Definition. Specifies how to formally describe the elements in an Extensible Markup Language (XML) document.This description can be used to verify that each item of content in a document adheres to the description of the element in which the content is to be placed.

Index

I

J

L

M

N

O

P

R

S

T

V

W

X