



Avaya Aura[®] Contact Center CCMA Open Interfaces

Release 7.0.0

Issue 3.0

June 2016

© 2016 Avaya Inc.

All Rights Reserved.

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

"Documentation" means information published by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya may generally make available to users of its products and Hosted Services. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original Published version of documentation unless such modifications, additions, or deletions were performed by Avaya. End User agrees 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 End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked websites referenced within this site or documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on Avaya hardware and software. 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 to Avaya customers and other parties through the Avaya Support website: <http://support.avaya.com> or such successor site as designated by Avaya. Please note that if you acquired the product(s) from an authorized Avaya Channel

Partner outside of the United States and Canada, the warranty is provided to you by said Avaya Channel Partner and not by Avaya.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, [HTTP://SUPPORT.AVAYA.COM/LICENSEINFO](http://support.avaya.com/licenseinfo)

OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AVAYA CHANNEL PARTNER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA CHANNEL PARTNER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE

WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS "YOU" AND "END USER"), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE AVAYA AFFILIATE ("AVAYA").

Avaya grants you a license within the scope of the license types described below, with the exception of Heritage Nortel Software, for which the scope of the license is detailed below. Where the order documentation does not expressly identify a license type, the applicable license will be a Designated System License. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the documentation or other materials available to you. "Designated Processor" means a single stand-alone computing device. "Server" means a Designated Processor that hosts a software application to be accessed by multiple users.

License type(s)

Named User License (NU). You may: (i) install and use the Software on a single Designated Processor or Server per authorized Named User (defined below); or (ii) install and use the Software on a Server so long as only authorized Named Users access and use the Software. "Named User", means a user or device that has been expressly authorized by Avaya to access and use the Software. At Avaya's sole discretion, a "Named User" may be, without limitation, designated by name, corporate function (e.g., webmaster or helpdesk), an e-mail or voice mail account in the name of a person or corporate function, or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software.

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, Hosted Service, or hardware provided by Avaya. All content on this site, the documentation, Hosted Service, and the Product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may

not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

Third Party Components

"Third Party Components" mean certain software programs or portions thereof included in the Software or Hosted Service may contain software (including open source software) distributed under third party agreements ("Third Party Components"), which contain terms regarding the rights to use certain portions of the Software ("Third Party Terms"). As required, information regarding distributed Linux OS source code (for those Products that have distributed Linux OS source code) and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply is available in the Documentation or on Avaya's website at:

<http://support.avaya.com/Copyright> or such successor site as designated by Avaya. You agree to the Third Party Terms for any such Third Party Components.

THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE [HTTP://WWW.MPEGLA.COM](http://www.mpegla.com).

Note to Service Provider

The Product or Hosted Service may use Third Party Components subject to Third Party Terms that do not allow hosting and require a Service Provider to be independently licensed for such purpose. It is your responsibility to obtain such licensing.

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 can 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 Toll Fraud intervention

If you suspect that you are being victimized by Toll Fraud and you need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support website: <http://support.avaya.com> or such successor site as designated by Avaya. Suspected security vulnerabilities with Avaya products should be reported to Avaya by sending mail to: securityalerts@avaya.com.

Trademarks

The trademarks, logos and service marks ("Marks") displayed in this site, the Documentation, Hosted Service(s), and Product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the Documentation, Hosted Service(s) and Product(s) should be construed as granting, by implication, estoppel, or otherwise, any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

Avaya is a registered trademark of Avaya Inc.

All non-Avaya trademarks are the property of their respective owners. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

All non-Avaya trademarks are the property of their respective owners, and "Linux" is a registered trademark of Linus Torvalds.

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support website: <http://support.avaya.com> or such successor site as designated by Avaya.

Contact Avaya Support

See the Avaya Support website: <http://support.avaya.com> for Product or Hosted Service notices and articles, or to report a problem with your Avaya Product or Hosted Service. For a list of support telephone numbers and contact addresses, go to the Avaya Support website: <http://support.avaya.com> (or such successor site as designated by Avaya), scroll to the bottom of the page, and select Contact Avaya Support.

Contents

Chapter 1: Revision History	6
Chapter 2: Overview.....	7
SOAP Style	8
REST Style.....	8
Web Service Security	11
What is the SDK	12
SDK contents	12
SDK support	12
Related Avaya Aura Contact Center documents.....	12
Backward & Forward Compatibility	12
<i>Backward & Forward Compatibility History.....</i>	<i>13</i>
Chapter 3: CCMA Open Interfaces Introduction	14
User Management	15
<i>AddAgent</i>	<i>15</i>
<i>AddSupervisor.....</i>	<i>16</i>
<i>AddUser</i>	<i>16</i>
<i>GetAgentsList.....</i>	<i>18</i>
<i>GetSupervisorsList.....</i>	<i>19</i>
<i>GetAgentDetails</i>	<i>20</i>
<i>GetSupervisorDetails</i>	<i>22</i>
<i>GetSkillsetsList.....</i>	<i>23</i>
<i>GetAgentsAssignedToSkillset.....</i>	<i>24</i>
<i>AddAgentSkillset.....</i>	<i>25</i>
<i>RemoveAgentSkillset.....</i>	<i>26</i>
<i>GetServerList</i>	<i>26</i>
<i>GetCallPresentationClassList</i>	<i>27</i>
<i>GetContactTypeList</i>	<i>28</i>
<i>GetAgentThresholdClassList</i>	<i>29</i>
<i>GetSkillsetThresholdClassList</i>	<i>29</i>
<i>GetActivityCodeList.....</i>	<i>30</i>
<i>GetMultiplicityPresentationClassList.....</i>	<i>30</i>
<i>GetCommunicationManagerServerList.....</i>	<i>31</i>
<i>GetUserDetails.....</i>	<i>32</i>
<i>UpdateUser</i>	<i>34</i>
<i>updateAgentSkillsets.....</i>	<i>36</i>
<i>DeleteUser</i>	<i>36</i>
Skillset Management	38
<i>GetSkillsetConfigurationDetails</i>	<i>38</i>

<i>AddSkillset</i>	39
<i>UpdateSkillset</i>	40
<i>DeleteSkillset</i>	42
Chapter 4: Reference Client	43
Chapter 5: Programming with the CCMA Open Interfaces	45
<i>Prerequisites</i>	45
<i>AACC 7.0 - CCMA Open Interface API HTML Reference</i>	45

Chapter 1: Revision History

Date	Revision #	Summary of Changes
January 2013	<i>Version 1.0</i>	Initial Avaya Aura® Contact Center Release 6.2
December 2015	<i>Version 2.0</i>	Avaya Aura® Contact Center 7.0 Release
June 2016	<i>Version 3.0</i>	Updates for Avaya Aura® Contact Center 7.0 Release

Chapter 2: Overview

Contact Center Manager Administration (CCMA) is a Web browser-based configuration and reporting interface for Avaya Aura Contact Center. Contact Center supervisors use CCMA to configure contact center resources, agents, skillsets, contact flows, components, and activities. CCMA provides historical and real-time reporting about the contact center.

The CCMA Open Interfaces are a set of Web Services that provide access to the following subset of the CCMA operations:

- Add an agent
- Add a supervisor
- Get a list of all agents
- Get a list of all supervisors
- Get agent details (including assigned skillsets)
- Get supervisor details (including assigned agents)
- Get a list of all skillsets
- Get the agents assigned to a specific skillset
- Assign a skillset to an agent
- Remove a skillset from an agent
- Full User Management
 - View, create, update and delete agents, supervisors and supervisor agents.
 - Assign contact types to agents
- Full Skillset Management,
 - Ability to view, create update and delete skillsets
- Configuration
 - Get a list of the configured servers
 - Get a list of the Call Presentation Classes
 - Get a list of the Multiplicity Presentation Classes
 - Get a list of the Activity Codes
 - Get a list of the Contact Types
 - Get a list of the Agent Threshold Classes
 - Get a list of all Skillsets

- Get a list of all the Skillset Threshold Classes
- Get a list of the configured Communication Manager Servers

This CCMA Open Interface Web Services are hosted on the CCMA server and are part of the standard installation.

SOAP Style

When the service is configured without security, the WSDL for the SOAP endpoint can be found at the following URL, where <CCMA FQDN> should be replaced with the fully qualified domain name of the CCMA server.

<https://<CCMA FQDN>/WebServices/OpenInterfaces/soap.svc?wsdl>

This WSDL (Web Service Definition Language) is a machine-readable description of the functionality being offered by this web service. Various technologies can use a WSDL service description to create the relevant proxies to send and receive SOAP messages with the web service.

REST Style

REST is an alternative style of accessing web service interfaces through the sending of simple messages over HTTPS. The REST principle relies on each logical entity being accessible by a URL. Operations that retrieve data use the HTTP GET verb while operations that add or update data use the HTTP POST verb.

The URLs for accessing the CCMA Open Interfaces through REST are summarized in the following table.

Operation	REST URL	HTTP Verb
Add an agent	<a href="https://<CCMA hostname>/WebServices/OpenInterfaces/rest.svc/agents/<CCMS IP Address>/firstName/<firstName>/lastName/<lastName>">https://<CCMA hostname>/WebServices/OpenInterfaces/rest.svc/agents/<CCMS IP Address>/firstName/<firstName>/lastName/<lastName>	POST
Add a supervisor	<a href="https://<CCMA hostname>/WebServices/OpenInterfaces/rest.svc/supervisors/<CCMS IP Address>/firstName/<firstName>/lastName/<lastName>">https://<CCMA hostname>/WebServices/OpenInterfaces/rest.svc/supervisors/<CCMS IP Address>/firstName/<firstName>/lastName/<lastName>	POST
Get agents	<a href="https://<CCMA hostname>/WebServices/OpenInterfaces/rest.svc/agents/<CCMS IP Address>">https://<CCMA hostname>/WebServices/OpenInterfaces/rest.svc/agents/<CCMS IP Address>	GET
Get supervisors	<a href="https://<CCMA hostname>/WebServices/OpenInterfaces/rest.svc/supervisors/<CCMS IP Address>">https://<CCMA hostname>/WebServices/OpenInterfaces/rest.svc/supervisors/<CCMS IP Address>	GET

Get agent	https://<CCMA hostname>/ WebServices/OpenInterfaces/rest.svc/agents/ <CCMS IP Address>/<AgentLocalUserID>	GET
Get supervisor	https://<CCMA hostname>/ WebServices/OpenInterfaces/rest.svc/supervisors/ <CCMS IP Address>/<SupervisorLocalUserID>	GET
Get the list of skillsets	https://<CCMA hostname>/ WebServices/OpenInterfaces/rest.svc/skillsets/<CCMS IP Address>	GET
Get agents assigned to a skillset	https://<CCMA hostname>/ WebServices/OpenInterfaces/rest.svc/agents/<CCMS IP Address>/ skillsets/<SkillsetID>	GET
Assign an agent to a skillset	https:// <CCMA hostname>/ WebServices/OpenInterfaces/rest.svc/agents/<CCMS IP Address>/ <AgentLocalUserID>/skillsets/<SkillsetID>/priority/<priority>	POST
Remove an agent from a skillset	https:// <CCMA hostname>/ WebServices/OpenInterfaces/rest.svc/agents/<CCMS IP Address>/ <AgentLocalUserID>/skillsets/<SkillsetID>	POST
Add a user	https:// <CCMA hostname>/ WebServices/OpenInterfaces/rest.svc/users/<CCMS IP Address> /users/{ccmsIPAddress}	POST
Update a user	https:// <CCMA hostname>/ WebServices/OpenInterfaces/rest.svc/users/{ccmsIPAddress}/{userLoginID}	POST
Delete a user	https:// <CCMA hostname>/ WebServices/OpenInterfaces/rest.svc /users/{ccmsIPAddress}/{userLoginID}	DELETE
Update agents skillsets	https:// <CCMA hostname>/ WebServices/OpenInterfaces/rest.svc /users/{ccmsIPAddress}/{userLoginID}/skillsets	POST
Get User Details	https:// <CCMA hostname>/ WebServices/OpenInterfaces/rest.svc/users/<CCMS IP Address> /{userLoginID}	GET
Get Server List	https:// <CCMA hostname>/ WebServices/OpenInterfaces/rest.svc/configuration/servers	GET
Get list of Call Presentation Classes	https:// <CCMA hostname>/ WebServices/OpenInterfaces/rest.svc/configuration/callPresentationClasses/{ccmsIPAddress}	GET
Get list of	https:// <CCMA hostname>/	GET

Contact Types	WebServices/OpenInterfaces/rest.svc/configuration/contactTypes/{ccmsIPAddress}	
Get list of Agent Threshold Classes	https:// <CCMA hostname>/WebServices/OpenInterfaces/rest.svc/configuration/agentThresholdClasses/{ccmsIPAddress}	GET
Get list of Skillset Threshold Classes	https:// <CCMA hostname>/WebServices/OpenInterfaces/rest.svc/configuration/skillsetThresholdClasses/{ccmsIPAddress}	GET
Get list of Activity Codes	https:// <CCMA hostname>/WebServices/OpenInterfaces/rest.svc/configuration/activityCodes/{ccmsIPAddress}	GET
Get list of Multiplicity Presentation Classes	https:// <CCMA hostname>/WebServices/OpenInterfaces/rest.svc/configuration/multiplicityPresentationClasses/{ccmsIPAddress}	GET
Get list of Communication Manager servers	https:// <CCMA hostname>/WebServices/OpenInterfaces/rest.svc/cm/{CCMSName}	GET
Skillset Management		
Get the Configured Skillset Details	https:// <CCMA hostname>/WebServices/OpenInterfaces/rest.svc/configuration/skillsets/{ccmsIPAddress}	GET
Add a skillset	https:// <CCMA hostname>/WebServices/OpenInterfaces/rest.svc/configuration/skillsets/{ccmsIPAddress}	POST
Delete a skillset	https:// <CCMA hostname>/WebServices/OpenInterfaces/rest.svc/configuration/skillsets/{ccmsIPAddress}/{skillsetID}	DELETE
Update a skillset	https:// <CCMA hostname>/WebServices/OpenInterfaces/rest.svc/configuration/skillsets/{ccmsIPAddress}/{skillsetID}	POST

Table 1 Summary of REST URLs

The URLs in Table 1 contain markers for the input variables for the web service calls. The allowed input values are described in the detail of the interfaces described in the *CCMA Overview* section below. These are:

<CCMA hostname>	The host name or IP address of the CCMA server
<CCMS IP Address>	The IP address of the CCMS server
<CCMSName>	The CCMS server name as configured in CCMA
<firstName>	The first name of a supervisor or agent
<lastName>	The last name of a supervisor or agent
<SkillsetID>	The ID of a CCMA skillset
<AgentLocalUserID>	The local CCMA ID of an agent.
<SupervisorLocalUserID>	The local CCMA ID of a supervisor.
<Priority>	The priority to assign to the skillset for a particular agent.
<userLoginID>	The user's phoneset login ID

Web Service Security

In AACC 7.0, security is enabled by default, so this interface is https enabled on a new install.

For more information about configuring CCMA security and certification, refer to the following Avaya Aura Contact Center documents:

- Avaya Aura Contact Center Commissioning for Avaya Aura Unified Communications
- Avaya Aura Contact Center Commissioning for Avaya Communication Server 1000

What is the SDK

The Avaya Aura® Contact Center CCMA Open Interfaces SDK can be downloaded to your client machine and contains information on how Third Party Applications can call the CCMA Open Interfaces web services.

SDK contents

The SDK contains the following elements:

- Application programming interface documentation
- Reference client
- Source code for reference client implementations
- Simple tutorial for creating a client using the open-source soapUI product

SDK support

Support for the SDK APIs is supplied by through your Developer Partner Program.

Related Avaya Aura Contact Center documents

For more information about configuring CCMA Open Interfaces, https, security and certification, refer to the following Avaya Aura Contact Center documents:

- Avaya Aura Contact Center Overview and Specification
- Avaya Aura Contact Center Commissioning for Avaya Aura Unified Communications
- Avaya Aura Contact Center Commissioning for Avaya Communication Server 1000
- Avaya Aura Contact Center Server Administration

Backward & Forward Compatibility

The open interface provided by CCMA is intended to be backward compatible meaning that a client using the updated interface should still be able to communicate correctly with a CCMA server hosting an older version of the interface.

The CCMA open interfaces in AACC 7.0 are backward compatible with the interface in AACC 6.4 SP15 and SP14.

Backward & Forward Compatibility History

In 6.4.1 (SP13) some of the AACC objects had enumerations extended (new enums added) which resulted in new properties being added to existing objects. Older clients can continue to use the older properties while new clients need to use the “extended” properties to align with 6.4.1 functionality. (E.g. “ContactType.prefix and ContactType.prefix_Ex. The new 6.4.1 contact types are only available on the prefix_Ex property.)

The open interface also supports limited “Forward Compatibility” meaning that an old client can interact with a CCMA server hosting the newer interface, but it will not be able to use the new methods or data that is exposed by the new interface.

The server has a “version tolerant” interface and the client is expected to have a similar approach to versioning. This means that the server may introduce new properties into existing objects between versions or introduce new methods on the interface but it will not modify enums, remove methods or existing mandatory fields.

The XML serializer at the client should allow missing, non-required data and ignore superfluous data when handling the web service responses.

Chapter 3: CCMA Open Interfaces

Introduction

CCMA Open Interface web service operations allow third party applications to retrieve information about agents, supervisors and skillsets from the Contact Center and to add new agents and new supervisors. Both the REST and SOAP styles of interaction are supported.

Before the service can be used the application must first receive a Single Sign On (SSO) token that will be used for all subsequent calls to the service. This token is acquired by a call to the CCMA Authentication web service AuthenticateUser method. To receive this token the developer must supply the required authentication details;

- Username – Any valid CCMA User ID with appropriate privileges
- Password – a password associated with the supplied username.

Once the token is acquired, it must be embedded in a cookie in the HTTP header of each web service invocation. The cookie is a simple key-value pair. The Cookie name must be retrieved by calling the CCMA Authentication web service GetSSOCookieName method.

- Cookie name: The value returned by the GetSSOCookieName method
- Cookie value: The value returned by the CCMA Authentication web service

The retrieval of details for an agent or supervisor is based on the value of their respective Contact Center Local ID. This value is assigned when an agent or supervisor is added to the Contact Center. The Local IDs of all agents or supervisors can be retrieved using the operations for getting the list of agents or the list of supervisors.

User Management

AddAgent

Add a new agent to the Contact Center. The agent is added to the built-in Default Supervisor in CCMS. Each agent is assigned to a default skillset and contact type.

AddAgent requires three base input parameters:

- First name
- Last name
- The IP address of the Contact Center's CCMS server

When an agent is added via the CCMA Open Interface, a CCT user is also added. The following additional CCT specific input parameters are required:

- CCTLogin
- CCTServerName
- Domain
- Enabled: indicate if the new CCT user should be enabled or not
- Password
- Username: the username for the new CCT user including the domain name

It returns one parameter

- The CCMS Local ID of the new agent

The input and output parameters are:

Parameter	Data Type	Direction	min length	max length	Expected values
First name	String	In	1	30	Unicode characters with the exception of <> & " : @ w z C c K k L I M m N n O o P p R r S s W w Z z C c G g h j J w Y y E e I i U u V v Y y and excluding Japanese and Chinese characters This is because the underlying database does not support these characters.
Last name	String	In	1	30	Unicode characters with the exception of <> & " : @ w z C c K k L I M m N n O o P p R r S s W w Z z C c G g h j J w Y y E e I i U u V v Y y and excluding Japanese and Chinese characters This is because the underlying database

					does not support these characters.
CCMS IP address	String	In	7	15	Any valid IP address
agentLocalID	Int	Out			The localID returned by the server.

AddSupervisor

Adds a new supervisor to the Contact Center. It takes the same inputs as AddAgent . The output is the CCMS Local ID of the new supervisor.

The input and output parameters are:

Parameter	Data Type	Direction	min length	max length	Expected values
First name	String	In	1	30	Unicode characters with the exception of <> & " : @ w z Ć ć K k L l M m N n Ō ō P p R r S s V v Ž ž Ć Ć Ğ ğ Ħ ħ W w Ÿ ŷ Ě ě Ĭ ĭ Ů ů V v Ÿ ŷ and excluding Japanese and Chinese characters This is because the underlying database does not support these characters.
Last name	String	In	1	30	Unicode characters with the exception of <> & " : @ w z Ć ć K k L l M m N n Ō ō P p R r S s V v Ž ž Ć Ć Ğ ğ Ħ ħ W w Ÿ ŷ Ě ě Ĭ ĭ Ů ů V v Ÿ ŷ and excluding Japanese and Chinese characters This is because the underlying database does not support these characters.
CCMS IP address	String	In	7	15	Any valid IP address
agentLocalID	Int	Out			The localID returned by the server.

AddUser

Add the details of a Contact Center user (i.e. Agent, Supervisor or Supervisor\Agent). This interface allows you to add the user's basic details (i.e. first name last name etc). If updating an agent this interface also supports the updating of the agent's contact types, skillsets, primary supervisor, call presentation class, threshold class, multiplicity presentation class, agent greeting details and their CCT account details. For supervisor their CCMA and CCT account details can be updated through this interface

It takes three parameters:

- The IP address of the CCMS server
- The phone set login ID of the user to be updated
- The UserDetails object containing the user updates to be applied.

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP Address	String	In	7	15	Any valid CCMS Server IP address
User Login ID	String	In	1	16	The phoneset login ID of the user to be updated
User	UserDetails	in			The user details to be updated

AACC 6.4.1 Changes

The UserDetails object contains an array of ContactTypeAndRelatedSkillsets objects. Each of the ContactTypeAndRelatedSkillsets contains the ContactType attributes including a *prefix* property of type *SkillsetPrefix* which is an enumerated list of prefixes for the contact types (e.g. “EM_” for email). The enum cannot be extended so a new enum has been added, *SkillsetPrefix_Ex*, which contains the additional values:

“PO_”

“EV_”

“SN_”

NOTE: The “EV_” value is not used by AACC.

The ContactType object has been extended to contain a “*prefix_Ex*” property in addition to the “Prefix” property. The “*prefix_Ex*” property should be used when interacting with a 6.4.1 or later CCMA server. The “prefix” property should be populated for all older servers.

The ContactTypeAndRelatedSkillsets contains an array of SkillsetFullDetails objects with a record for each skillset of a given contact type. The SkillsetFullDetails object contains a ContactTypeID property of type int. The range of values that can be returned in this int has been extended to handle the new 6.4.1 contact types.

The new values are:

POM_Outbound = 10013

Elite_Voice = 10014 (NOTE: This is defined but never used)

Social_Networking = 10015

AACC 6.4.2 Changes

The UserDetails object contains the following additional properties:

<i>Type</i>	<i>Name</i>	<i>Description</i>
Int	dnBargeInEnabled	Allow a Supervisor/Agent to barge into a reporting agents DN call. (0=Disabled, 1=Enabled)
Int	dnObserveEnabled	Allow a Supervisor/Agent to observe a reporting agents DN call. (0=Disabled, 1=Enabled)
Int	sipSoftPhoneEnabled	Reserved for future use

The dnBargeInEnabled and dnObserveEnabled fields are only applicable to Supervisor/Agents.

GetAgentsList

Get a list of all agents and supervisors and supervisor agents for a specific CCMS.

It takes one parameter:

- The IP address of the CCMS server

It returns an XML document containing details of all agents for that CCMS. These details contain (for each agent):

- Local ID
- Phone set login ID
- First and last name
- Logged in status
- ID of supervisor
- Contact type ID and name
- For each assigned skillset for a contact type, the status and priority

The input and output parameters are:

Parameter	Data Type	Direction	Expected values
CCMS IP address	String	In	Any valid IP address
Agents	Array of AgentFull Details	Out	An array of agent objects

AACC 6.4.1 Changes

The AgentFullDetails contains a new property “*offsiteAllowed*” of type integer (0 = Disabled, 1 = Enabled).

The AgentFullDetails object contains an array of ContactTypeAndRelatedSkillsets objects. Each of the ContactTypeAndRelatedSkillsets contains the ContactType attributes including a *prefix* property of type *SkillsetPrefix* which is an enumerated list of prefixes for the contact types (e.g. “EM_” for email). The enum cannot be extended so a new enum has been added, *SkillsetPrefix_Ex*, which contains the additional values:

“PO_”

“EV_”

“SN_”

NOTE: The “EV_” value is not used by AACC.

The ContactType object has been extended to contain a “*prefix_Ex*” property in addition to the “Prefix” property. The “prefix_Ex” property should be used when interacting with a 6.4.1 or later CCMA server. The “prefix” property should be used for all older servers.

The ContactTypeAndRelatedSkillsets contains an array of SkillsetFullDetails objects with a record for each skillset of a given contact type. The SkillsetFullDetails object contains a ContactTypeID property of type int. The range of values that can be returned in this int has been extended to handle the new 6.4.1 contact types.

The new values are:

POM_Outbound = 10013

Elite_Voice = 10014 (*NOTE: This is defined but never used*)

Social_Networking = 10015

GetSupervisorsList

Get a list of all supervisors for a specific CCMS.

It takes one parameter:

- The IP address of the CCMS server

It returns an XML document containing details of all agents for that CCMS. These details contain (for each agent):

- Local ID
- Phone set login ID
- First and last name

The input and output parameters are:

Parameter	Data Type	Direction	Expected values
CCMS IP address	String	In	Any valid IP address
Agents	Array of AgentBasic Details	Out	An array of agent objects

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP address	String	In	7	15	Any valid IP address

GetAgentDetails

Get the details of a Contact Center agent identified using the agent's CCMS Local ID

It takes two parameters:

- CCMS Local ID of the agent
- The IP address of the CCMS server

It returns an XML document containing details of the agent. This contains:

- Local ID
- Phone set login ID
- First and last name
- Logged in status
- ID of supervisor
- Contact type ID and name
- For each assigned skillset for a contact type, the status and priority

The input and output parameters for GetAgentDetails are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
agentLocalID	String	In	1	16	A local ID is created for each agent when it is added to the CCMS server. It's a numeric value with a range depending on the server type. min=1, max = 9999999999999999 (16 digits)
AgentFullDetails	AgentFullDetails	Out			This is a serializable class representing an agent. WCF converts this to an XML fragment that corresponds to the XML schema definition in the WSDL description for the Web service operation.

AACC 6.4.1 Changes

The AgentFullDetails contains a new property *“offsiteAllowed”* of type integer (0 = Disabled, 1 = Enabled).

The AgentFullDetails object contains an array of ContactTypeAndRelatedSkillsets objects. Each of the ContactTypeAndRelatedSkillsets contains the ContactType attributes including a *prefix* property of type *SkillsetPrefix* which is an enumerated list of prefixes for the contact types (e.g. “EM_” for email). The enum cannot be extended so a new enum has been added, *SkillsetPrefix_Ex*, which contains the additional values:

“PO_”

“EV_”

“SN_”

NOTE: The “EV_” value is not used by AACC.

The ContactType object has been extended to contain a *“prefix_Ex”* property in addition to the “prefix” property. The “prefix_Ex” property should be used when interacting with a 6.4.1 or later CCMA server. The “prefix” property should be used for all older servers.

The ContactTypeAndRelatedSkillsets contains an array of SkillsetFullDetails objects with a record for each skillset of a given contact type. The SkillsetFullDetails object contains a ContactTypeID

property of type int. The range of values that can be returned in this int has been extended to handle the new 6.4.1 contact types.

The new values are:

POM_Outbound = 10013

Elite_Voice = 10014 (*NOTE: This is defined but never used*)

Social_Networking = 10015

GetSupervisorDetails

Get the details of a Contact Center agent identified using the supervisor's CCMS Local ID.

It takes two parameters:

- CCMS Local ID of the supervisor
- The IP address of the CCMS server

It returns an XML document containing details for the supervisor including:

- Local ID
- Phone set login ID
- First and last name
- Logged in status
- Local IDs of assigned agents
- The number of assigned agents

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP address	String	In	7	15	Any valid IP address
agentLocalID	String	In	1	16	A local ID is created for each agent when it is added to the CCMS server. It's a numeric value with a range depending on the server type. min=1, max = 9999999999999999 (16 digits)
SupervisorFullDetails	Supervisor FullDetails	Out			This is a serializable class representing an agent. WCF converts this to an XML fragment that corresponds to the XML schema definition in the WSDL description for the Web service operation.

GetSkillsetsList

Get a list of all skillsets defined for the specified CCMS.

It takes one parameter:

- The IP address of the CCMS server

It returns an XML document detailing a set of skillsets. This contains:

- Skillset ID
- Skillset Name

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP address	String	In	7	15	Any valid IP address
Skillset array	Array of skillsets	Out			An array of skillsets defined for this CCMS server

AACC 6.4.1 Changes

The Skillset object contains a ContactTypeID property of type int. The range of values that can be returned in this property has been extended.

The new values are:

POM_Outbound = 10013

Elite_Voice = 10014 (NOTE: This is defined but never used)

Social_Networking = 10015

GetAgentsAssignedToSkillset

Get a list of all agents assigned to a skillset.

It takes two parameters:

- The Local ID of the agent
- The IP address of the CCMS server

It returns an XML document detailing a set of agents. This contains (for each agent):

- Local ID
- Phone set login ID
- First and last name
- Logged in status
- ID of supervisor
- Contact type ID and name
- For each assigned skillset for a contact type, the status and priority

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP address	String	In	7	15	Any valid IP address
skillsetID	String	In	1	9	A numeric value between 0 and 999999999
agents	Array of Agent FullDetails	Out			This is a serializable class representing an array of AgentFullDetail objects. WCF converts this to an XML fragment that corresponds to the XML schema definition in the WSDL description for the Web service operation.

AACC 6.4.1 Changes

The AgentFullDetails contains a new property “*offsiteAllowed*” of type integer (0 = Disabled, 1 = Enabled).

The AgentFullDetails object contains an array of ContactTypeAndRelatedSkillsets objects. Each of the ContactTypeAndRelatedSkillsets contains the ContactType attributes including a *prefix* property of type *SkillsetPrefix* which is an enumerated list of prefixes for the contact types (e.g. “EM_” for email). The enum cannot be extended so a new enum has been added, *SkillsetPrefix_Ex*, which contains the additional values:

“PO_”

“EV_”

“SN_”

NOTE: The “EV_” value is not used by AACC.

The ContactType object has been extended to contain a “*prefix_Ex*” property in addition to the “Prefix” property. The “*prefix_Ex*” property should be used when interacting with a 6.4.1 or later CCMA server. The “prefix” property should be used for all older servers.

The ContactTypeAndRelatedSkillsets contains an array of SkillsetFullDetails objects with a record for each skillset of a given contact type. The SkillsetFullDetails object contains a ContactTypeID property of type int. The range of values that can be returned in this int has been extended to handle the new 6.4.1 contact types.

The new values are:

POM_Outbound = 10013

Elite_Voice = 10014 (*NOTE: This is defined but never used*)

Social_Networking = 10015

AddAgentSkillset

Assign a skillset with a specified priority to an agent.

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP address	String	In	7	15	A valid IP address for a CCMS server
agentLocalID	String	In	1	16	Local ID of the agent provided by CCMS. It's a numeric value with a range depending on the server type. range=1 -- 9999999999999999 (16 digits)
skillsetID	String	In	1	9	The ID of the skillset to be assigned. A numeric value between 0 and 999999999
Priority	String	In	1	2	Priority value Valid range for assign is: 0 – 48 0 means the skillset priority is unassigned.

RemoveAgentSkillset

Remove a skillset from an agent

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP address	String	In	7	15	A valid IP address for a CCMS server
agentLocalID	String	In	1	16	Local ID of the agent provided by CCMS. It's a numeric value with a range depending on the server type. range=1 -- 9999999999999999 (16 digits)
skillsetID	String	In	1	9	The ID of the skillset to be removed. A numeric value between 0 and 999999999

GetServerList

Get a list of all servers configured on the CCMA server.

It returns an XML document detailing a set of server. This contains:

- The logged in users access privileges
- CCMA logged in user ID
- CCMA logged in user name
- The servers display name within CCMA
- The servers IP Address
- Boolean value indicating if Basic IVR is enabled
- Boolean value indicating if Hetero Networking is enabled
- Boolean value indicating if the server is a networked
- Boolean value indicating if Open Queue is enabled
- Boolean value indicating if Security is enabled on the CCMA server
- The Server name as configured in CCMA

- The Password used to access the server
- The server type i.e. SIP, Multimedia, CCT etc
- The Server version, only applies to CCMS servers

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS Server array	Array of servers	Out			Array of servers configured on the CCMA server.

AACC 6.4.1 Changes

This method returns an array of *CCMS*Server objects. Each *CCMS*Server object contains a *ServerType* object which in turn contains a *Name* property. The *Name* property is of type *serverTypeName* which is an enumerated list.

New server types have been added in 6.4.1 so a new enum is required: *serverTypeName_Ex*. The new server types and id's are:

CCPO (1005)

AMS (1006)

IPO (1007)

The *ServerType* object now contains a new property *Name_Ex* of type *serverTypeName_Ex*. This will be populated by 6.4.1 and later servers.

GetCallPresentationClassList

Get a list of all call presentation classes defined for the specified CCMS.

It takes one parameter:

- The IP address of the CCMS server

It returns an XML document detailing a set of call presentation classes. This contains:

- Call Presentation Class ID
- Call Presentation Class Name

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
-----------	-----------	-----------	------------	------------	-----------------

CCMS IP address	String	In	7	15	Any valid IP address
CallPresentationClass array	Array of call presentation classes	Out			An array of call presentation classes defined for this CCMS server

GetContactTypeList

Get a list of all contact types defined for the specified CCMS.

It takes one parameter:

- The IP address of the CCMS server

It returns an XML document detailing a set of contact types. This contains:

- Contact Type ID
- Contact Type Name
- Contact Type Prefix

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP address	String	In	7	15	Any valid IP address
ContactType array	Array of contact types	Out			An array of contact types defined for this CCMS server

AACC 6.4.1 Changes

New contact types have been introduced in 6.4.1 to support POM outbound and Social Networking. This method returns an array of *ContactType* objects. The *ContactType* object contains a *prefix* property of type *SkillsetPrefix* which is an enumerated list of prefixes for the contact types (e.g. "EM_" for email). The enum cannot be extended so a new enum has been added, *SkillsetPrefix_Ex*, which contains the additional values:

"PO_"

"EV_"

"SN_"

NOTE: The "EV_" value is not used by AACC.

The *ContactType* object has been extended to contain a "*prefix_Ex*" property in addition to the "Prefix" property.

Clients interacting with 6.4.1 servers should use the "*prefix_Ex*" property. Clients should continue to use the "prefix" property in order to remain backward compatible with older AACC servers.

Older clients interacting with a 6.4.1 server will receive the new contact types in the ContactType array however the “prefix” property will be set to UNDEFINED. These clients must be updated to use the new “prefix_Ex” property.

GetAgentThresholdClassList

Get a list of all agent threshold classes defined for the specified CCMS.

It takes one parameter:

- The IP address of the CCMS server

It returns an XML document detailing a set of contact types. This contains:

- Threshold Class ID
- Threshold Class Name

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP address	String	In	7	15	Any valid IP address
Threshold Class array	Array of agent threshold classes	Out			An array of agent threshold classes defined for this CCMS server

GetSkillsetThresholdClassList

Get a list of all skillset threshold classes defined for the specified CCMS.

It takes one parameter:

- The IP address of the CCMS server

It returns an XML document detailing a set of contact types. This contains:

- Threshold Class ID
- Threshold Class Name

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP address	String	In	7	15	Any valid IP address
Threshold	Array of	Out			An array of skillset threshold classes

Class array	skillset threshold classes				defined for this CCMS server
-------------	----------------------------	--	--	--	------------------------------

GetActivityCodeList

Get a list of all activity codes defined for the specified CCMS.

It takes one parameter:

- The IP address of the CCMS server

It returns an XML document detailing a set of contact types. This contains:

- Activity Code ID
- Activity Code Name

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP address	String	In	7	15	Any valid IP address
ActivityCode array	Array of activity codes	Out			An array of activity codes defined for this CCMS server

GetMultiplicityPresentationClassList

Get a list of all multiplicity presentation classes defined for the specified CCMS.

It takes one parameter:

- The IP address of the CCMS server

It returns an XML document detailing a set of multiplicity presentation classes. This contains:

- Multiplicity Presentation Class ID
- Multiplicity Presentation Class Name
- Multiplicity Presentation Class Delay
- Multiplicity Presentation Class Boolean value indicating if MM can Interrupt Voice
- Multiplicity Presentation Class Boolean value indicating if Voice can Interrupt MM

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP address	String	In	7	15	Any valid IP address
Multiplicity Presentation Class array	Array of Multiplicity Presentation Classes	Out			An array of Multiplicity Presentation Classes defined for this CCMS server

GetCommunicationManagerServerList

Get a list of all communication manager servers defined for the specified CCMS.

It takes one parameter:

- The Name of the CCMS server

It returns an XML document detailing a set of Communication Manager Server types. This contains:

- The CCMS servers associated with this CM server
- The Communication Manager (CM) Server ID
- The logged in users access privileges
- CCMA logged in user ID
- CCMA logged in user name
- The servers display name within CCMA
- The servers IP Address
- Boolean value indicating if Basic IVR is enabled (CCMS Only)
- Boolean value indicating if Hetero Networking is enabled (CCMS Only)
- Boolean value indicating if the server is a networked (CCMS Only)
- Boolean value indicating if Open Queue is enabled (CCMS Only)
- Boolean value indicating if Security is enabled on the CCMA server
- The Server name as configured in CCMA
- The Password used to access the server
- The server type i.e. CM
- The Server version, (CCMS Only)

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS Name	String	In	1	30	Any valid name
ServerDetails array	Array of ServerDetails	Out			An array of Communication Manager Serverdefined for this CCMS server

GetUserDetails

Get the details of a Contact Center user (i.e. Agent, Supervisor or Supervisor\Agent) identified using the user's CCMS phone set login ID

It takes two parameters:

- The IP address of the CCMS server
- The user phone set login ID

It returns an XML document detailing the user's details. This contains:

- The internal system user ID
- The internal system Local ID
- The user's Phoneset Login ID
- The user type (i.e. Agent, Supervisor or Supervisor Agent)
- The Communication Manager server ID the agent is associated with if any.
- The Communication Manager agent ID associated to this user if any
- The user's first and last name
- The user's personal DN
- The logged in status of the user. Set for agents and supervisor agents only.
- The user's position ID
- The user's language. Default is English.
- The user's assigned call presentation class ID (Applies to agents only)
- The user's assigned multiplicity call presentation class ID (Applies to agents only)
- The user's assigned threshold class ID (Applies to agents only)
- A Boolean identifying if the user is a supervisor
- The agents assigned supervisor ID
- The users title, comment and department details

- The user's user name. This field represents the user's CCMA\CCT user name
- The domain details for the user if the user account is a domain account.
- The user account password.
- The Voice and IM Uri's. Applies to SIP systems only.
- A list of the user's assigned contact types and corresponding skillsets. (Set for agents and supervisor agents.)
- A list of the users assigned agents. (Set for supervisor and supervisor agents.)
- The user's agent greeting details (i.e. Agent greeting state, Agent greeting password)
- The offsite user details
 - Boolean value indicating if offsite is allowed for the user
 - The offsite agent mode (i.e. On Demand or Permanent)
 - The agent's offsite phone numbers
 - Boolean value indicating if an alternative number is used

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP Address	String	In	7	15	Any valid CCMS Server IP address
User Login ID	string	In	1	16	The user phoneset login ID
User	UserDetails	Out			The user details

AACC 6.4.1 Changes

The UserDetails object contains an array of ContactTypeAndRelatedSkillsets objects. Each of the ContactTypeAndRelatedSkillsets contains the ContactType attributes including a *prefix* property of type *SkillsetPrefix* which is an enumerated list of prefixes for the contact types (e.g. "EM_" for email). The enum cannot be extended so a new enum has been added, *SkillsetPrefix_Ex*, which contains the additional values:

"PO_"

"EV_"

"SN_"

NOTE: The "EV_" value is not used by AACC.

The ContactType object has been extended to contain a "*prefix_Ex*" property in addition to the "Prefix" property. The "*prefix_Ex*" property should be used when interacting with a 6.4.1 or later CCMA server. The "prefix" property should be used for all older servers.

The ContactTypeAndRelatedSkillsets contains an array of SkillsetFullDetails objects with a record for each skillset of a given contact type. The SkillsetFullDetails object contains a ContactTypeID

property of type int. The range of values that can be returned in this int has been extended to handle the new 6.4.1 contact types.

The new values are:

POM_Outbound = 10013

Elite_Voice = 10014 (*NOTE: This is defined but never used*)

Social_Networking = 10015

AACC 6.4.2 Changes

The UserDetails object contains the following additional properties:

Type	Name	Description
Int	dnBargeInEnabled	Allow a Supervisor/Agent to barge into a reporting agents DN call. (0=Disabled, 1=Enabled)
Int		Allow a Supervisor/Agent to observe a reporting agents DN call. (0=Disabled, 1=Enabled)
Int	sipSoftPhoneEnabled	Reserved for future use

The dnBargeInEnabled and dnObserveEnabled fields are only applicable to Supervisor/Agents.

UpdateUser

Update's the details of a Contact Center user (i.e. Agent, Supervisor or Supervisor\Agent). This interface allows you to update the user's basic details (i.e. first name last name etc). If updating an agent this interface also supports the updating of the agent's contact types, skillsets, primary supervisor, call presentation class, threshold class, multiplicity presentation class, agent greeting details and their CCT account details. For supervisor their CCMA and CCT account details can be updated through this interface

It takes three parameters:

- The IP address of the CCMS server
- The phone set login ID of the user to be updated
- The UserDetails object containing the user updates to be applied.

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP	String	In	7	15	Any valid CCMS Server IP address

Address					
User Login ID	string	In	1	16	The phoneset login ID of the user to be updated
User	UserDetails	in			The user details to be updated

AACC 6.4.1 Changes

The UserDetails object contains an array of ContactTypeAndRelatedSkillsets objects. Each of the ContactTypeAndRelatedSkillsets contains the ContactType attributes including a *prefix* property of type *SkillsetPrefix* which is an enumerated list of prefixes for the contact types (e.g. “EM_” for email). The enum cannot be extended so a new enum has been added, *SkillsetPrefix_Ex*, which contains the additional values:

“PO_”

“EV_”

“SN_”

NOTE: The “EV_” value is not used by AACC.

The ContactType object has been extended to contain a “*prefix_Ex*” property in addition to the “Prefix” property. The “*prefix_Ex*” property should be used when interacting with a 6.4.1 or later CCMA server. The “prefix” property should be used for all older servers.

The ContactTypeAndRelatedSkillsets contains an array of SkillsetFullDetails objects with a record for each skillset of a given contact type. The SkillsetFullDetails object contains a ContactTypeID property of type int. The range of values that can be returned in this int has been extended to handle the new 6.4.1 contact types.

The new values are:

POM_Outbound = 10013

Elite_Voice = 10014 (*NOTE: This is defined but never used*)

Social_Networking = 10015

AACC 6.4.2 Changes

The UserDetails object contains the following additional properties:

Type	Name	Description
Int	dnBargeInEnabled	Allow a Supervisor/Agent to barge into a reporting agents DN call. (0=Disabled, 1=Enabled)
Int	dnObserveEnabled	Allow a Supervisor/Agent to observe a reporting agents DN call. (0=Disabled, 1=Enabled)
Int	sipSoftPhoneEnabled	Reserved for future use

The dnBargelnEnabled and dnObserveEnabled fields are only applicable to Supervisor/Agents.

updateAgentSkillsets

Update's the skillsets assigned to a Contact Center user (i.e. Agent or Supervisor\Agent). It takes three parameters:

- The IP address of the CCMS server
- The phone set login ID of the user to be updated
- The SkillsetFullDetails object containing the skillset changes to be applied.

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP Address	String	In	7	15	Any valid CCMS Server IP address
User Login ID	string	In	1	16	The phoneset login ID of the user to be updated
skillsetList	SkillsetFullDetails	in			An array of SkillsetFullDetails objects where each entry constitutes a change to the skillsets assigned to the user.

AACC 6.4.1 Changes

AACC 6.4.1 supports additional contact types for "POM outbound" and "Social Networking" contacts. The updateAgentSkillsets method returns an array of SkillsetFullDetails objects each of which contains a ContactTypeID property of type int. The range of values that can be returned in this int has been extended to handle the new 6.4.1 contact types.

The new values are:

POM_Outbound = 10013

Elite_Voice = 10014 (*NOTE: This is defined but never used*)

Social_Networking = 10015

DeleteUser

Delete's a user (i.e. agent, supervisor or supervisor agent) from the specified CCMS server. If deleting a supervisor, this interface will delete the CCMA and CCT user account details associated with this user. If deleting an agent the corresponding CCT user account details will be deleted.

It takes two parameters:

- The IP address of the CCMS server
- The phone set login ID of the user to be deleted

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP Address	String	In	7	15	Any valid CCMS Server IP address
User Login ID	string	In	1	16	The phone set login ID of the user to be deleted

Skillset Management

The skillset management interfaces allow you to manage (View, Add, Update, and Delete) the skillsets configured on the CCMS server.

GetSkillsetConfigurationDetails

The GetSkillsetConfigurationDetails interface gets a list of the skillsets configured on a CCMS server. It returns all details for the configured skillsets

It takes one parameter:

- The IP address of the CCMS server

It returns an XML document containing the skillsets details. This contains:

- The skillset ID
- The skillset name
- The skillset prefix value (i.e. EM_,FX_ etc)
- The default activity code ID assigned to the skillset
- The skillset threshold class ID
- The skillset type (Local or Networked)
- The skillset call source preference (i.e. None, Local or Networked)
- The skillset call age preference (i.e. First in queue or Oldest)
- The out of service mode (i.e. N/A, Night or Transition)
- The request queue size
- The flow control threshold
- Boolean value include local node (Networking property)
- The skillset comment details
- The network skillset ID
- The target service level value

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP Address	String	In	7	15	Any valid CCMS Server IP address
Skillset	SkillsetConfigDetails	out			An array of

Configuration Details array					SkillsetConfigDetails objects containing the configured skillset details.
-----------------------------	--	--	--	--	---

AACC 6.4.1 Changes

AACC 6.4.1 supports additional contact types for “POM outbound” and “Social Networking” contacts. The *getSkillsetConfigurationDetails* method returns an array of SkillsetConfigDetails objects each of which contains a ContactTypeID property of type int. The range of values that can be returned in this int has been extended to handle the new 6.4.1 contact types.

The new values are:

POM_Outbound = 10013

Elite_Voice = 10014 (*NOTE: This is defined but never used*)

Social_Networking = 10015

The *SkillsetConfigDetails* object also contains a “prefix” property of type *SkillsetPrefix* which is an enumerated list of prefixes for the contact types (e.g. “EM_” for email). The existing enum cannot be extended so a new enum has been added, *SkillsetPrefix_Ex*, which contains the additional values:

“PO_”

“EV_”

“SN_”

NOTE: The “EV_” value is not used by AACC.

The SkillsetConfigDetails object has been extended to contain a “*prefix_Ex*” property in addition to the “Prefix” property.

Clients interacting with 6.4.1 servers should use the “*prefix_Ex*” property. Clients should continue to use the “prefix” property in order to remain backward compatible with older AACC servers.

AddSkillset

The add skillset interface adds a skillset to the CCMS server.

It takes two parameters:

- The IP address of the CCMS server
- The skillset configuration details of the skillset to be added

It returns the skillset ID of the skillset added:

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP Address	String	In	7	15	Any valid CCMS Server IP address
Skillset Configuration Details array	SkillsetConfigDetails	in			SkillsetConfigDetails objects containing the skillset details to be added.
Skillset ID	String	Out			The ID of the skillset that was added

AACC 6.4.1 Changes

AACC 6.4.1 supports additional contact types for “POM outbound” and “Social Networking” contacts. The *addSkillset* method takes a *SkillsetConfigDetails* objects which contains a *ContactTypeID* property of type *int*. The range of values that can be returned in this *int* has been extended to handle the new 6.4.1 contact types.

The new values are:

POM_Outbound = 10013

Elite_Voice = 10014 (*NOTE: This is defined but never used*)

Social_Networking = 10015

The *SkillsetConfigDetails* object also contains a “prefix” property of type *SkillsetPrefix* which is an enumerated list of prefixes for the contact types (e.g. “EM_” for email). The existing enum cannot be extended so a new enum has been added, *SkillsetPrefix_Ex*, which contains the additional values:

“PO_”

“EV_”

“SN_”

NOTE: The “EV_” value is not used by AACC.

The *SkillsetConfigDetails* object has been extended to contain a “*prefix_Ex*” property in addition to the “Prefix” property.

Clients interacting with 6.4.1 servers should use the “*prefix_Ex*” property. Clients should continue to use the “prefix” property in order to remain backward compatible with older AACC servers.

NOTE: A skillset added to an AACC server with an unrecognized contact type will result in the skillset being added with type Voice.

UpdateSkillset

The update skillset interface updates the skillset properties on the CCMS server.

It takes three parameters:

- The IP address of the CCMS server
- The skillset configuration details of the skillset to be updated
- The skillset ID to be updated

The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP Address	String	In	7	15	Any valid CCMS Server IP address
Skillset Configuration Details array	SkillsetConfigDetails	in			SkillsetConfigDetails objects containing the skillset details to be updated.
Skillset ID	String	Out			The ID of the skillset that was added

AACC 6.4.1 Changes

AACC 6.4.1 supports additional contact types for “POM outbound” and “Social Networking” contacts. The update*Skillset* method takes a *SkillsetConfigDetails* objects which contains a *ContactTypeID* property of type *int*. The range of values that can be returned in this *int* has been extended to handle the new 6.4.1 contact types.

The new values are:

POM_Outbound = 10013

Elite_Voice = 10014 (*NOTE: This is defined but never used*)

Social_Networking = 10015

The *SkillsetConfigDetails* object also contains a “prefix” property of type *SkillsetPrefix* which is an enumerated list of prefixes for the contact types (e.g. “EM_” for email). The existing enum cannot be extended so a new enum has been added, *SkillsetPrefix_Ex*, which contains the additional values:

“PO_”

“EV_”

“SN_”

NOTE: The “EV_” value is not used by AACC.

The *SkillsetConfigDetails* object has been extended to contain a “*prefix_Ex*” property in addition to the “Prefix” property.

Clients interacting with 6.4.1 servers should use the “*prefix_Ex*” property. Clients should continue to use the “prefix” property in order to remain backward compatible with older AACC servers.

NOTE: A skillset added to an AACC server with an unrecognized contact type will result in the skillset being added with type Voice.

DeleteSkillset

The delete skillset interface deletes a skillset from the CCMS server.

It takes two parameters:

- The IP address of the CCMS server
- The skillset ID

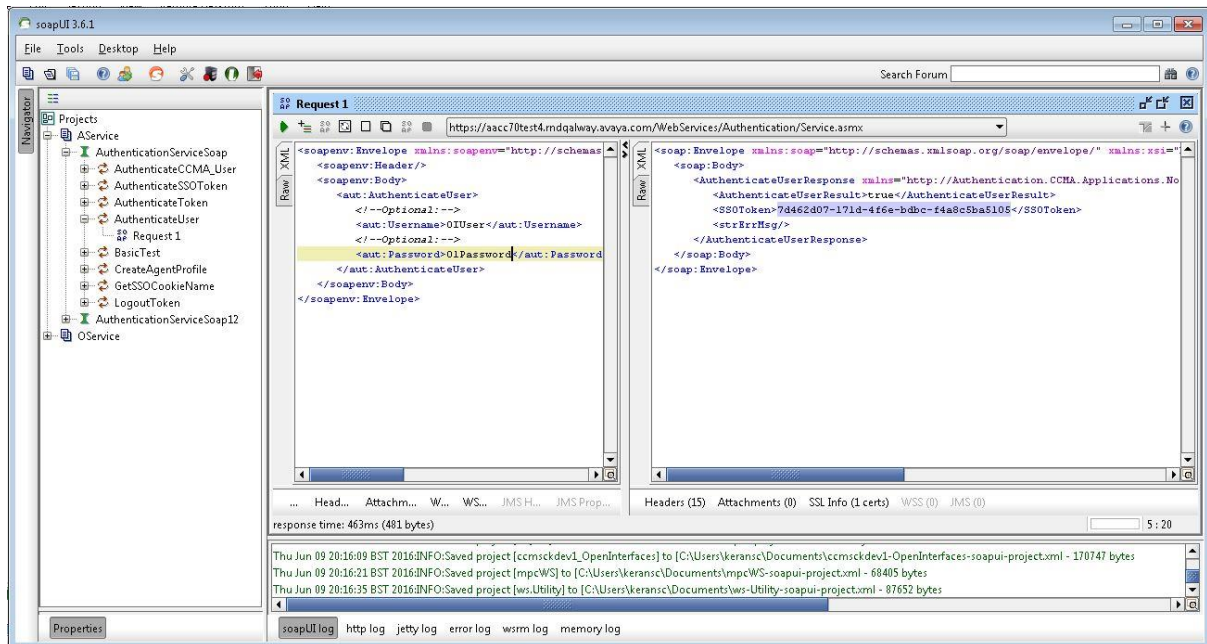
The input and output parameters are:

Parameter	Data Type	Direction	Min length	Max length	Expected values
CCMS IP Address	String	In	7	15	Any valid CCMS Server IP address
Skillset ID	String	in			The ID of the skillset that was added

Chapter 4: Reference Client

A simple test for the operations in the CCMA Open Interface web service can be carried out using the soapUI ¹ open source tool. This provides a GUI that takes the WSDL URL of the CCMA OI service and generates SOAP stubs for all operations. The tool is easy to use and well documented.

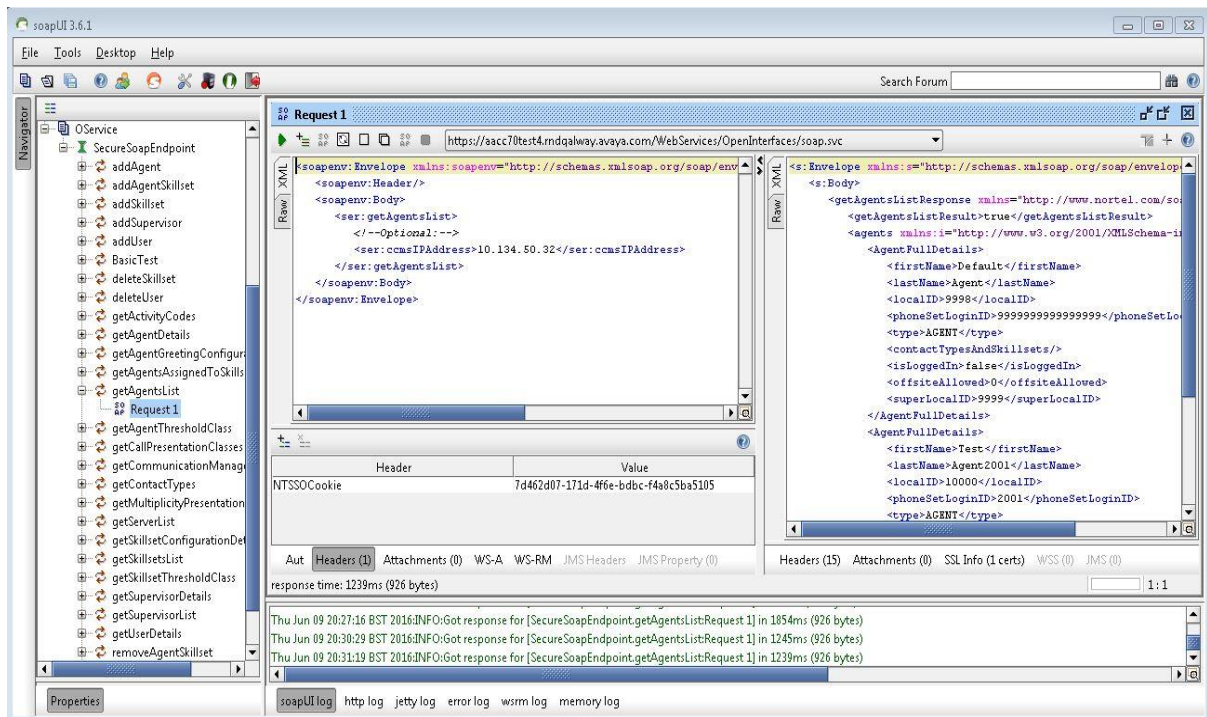
The CCMA OI requires user authentication using the CCMA-provided authentication service and a valid CCMA username and password. The next screenshot shows a sample authentication request. In this case the following user has been created in CCMA (username=OIUser, password=01Password). Any CCMA user could be used.



The screenshot above also shows the response which contains a single sign on (SSO) authentication token which must be inserted as a cookie (named NTSSOCookie) in all subsequent web service calls.

¹ <http://www.soapui.org/>

The next screenshot shows an example of a request to get a list of agents from the CCMA. Note that the SSO token has been added to the HTTP header using the NTSSOCookie.



The above screenshot also shows the response from the web service containing the list of agents retrieved.

A reference client is included with this SDK pack with examples of all the web service features. The reference client is a Microsoft Visual Studio 2008 project.

Chapter 5: Programming with the CCMA Open Interfaces

The CCMA Open Interfaces provide a powerful SOAP and REST based web service interface that allows Third Party developers to integrate their application with Avaya Contact Center CCMA product.

Prerequisites

The following assumptions are made.

1. An AACC 7.0.0 or a later version of CCMA and CCMS are configured and running. Please refer to the section Related Avaya Aura Contact Center documents on how to configure the web services.
2. That the developers are both familiar with their choice of technology and how that technology integrates with web services.

AACC 7.0 - CCMA Open Interface API HTML Reference

The following contains a link to the MSDN-style HTML documentation associated with the implementation of the AACC 7.0- CCMA Open Interface web service.

Service Name	WSDL Location	Doc Location
AACC 7.0 - CCMA OI	<a href="http://<CCMA FQDN or hostname>/WebServices/OpenInterfaces/soap.svc?wsdl">http://<CCMA FQDN or hostname>/WebServices/OpenInterfaces/soap.svc?wsdl	Open Interfaces CCMA 7.0 SDK\docs\help_docs

LAST PAGE