



avaya.com

Avaya Aura Contact Center
Release: 7.0
CCMM Outbound WS SDK

Date: December 2015

Avaya Aura Contact Center

Contact Center Multimedia Outbound Web Services Software Development Kit

REVISION HISTORY

Date	Revision #	Summary of Changes
28 th March 2011	Version 1	Initial Creation
1 st December 2015	Version 2	Updated for Contact Center 7.0 Release



OBSOLETE WHEN PRINTED

SECURITY WARNING

Copyright © Avaya Inc. 2011. All Rights Reserved.

Terms of Use

The information and/or products described in this document are subject to change without notice. While the information in this document is believed to be accurate and reliable, Avaya PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OR CONDITION OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OR MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NO INFRINGEMENT. The information in this document could include technical inaccuracies or typographical errors.

Table of Contents

1	DOCUMENT OVERVIEW.....	5
2	Outbound SDK Overview.....	6
	Logging in.....	6
	Understanding the SDK UI.....	7
	Getting Active Campaigns	8
	Getting Campaign Contacts.....	8
	Closing a Contact	9
	Getting the Status of a Contact.....	10
	Get Contact ID by Phone Number	11
	Add Contact	12
	Find a Contact.....	12
3	Available Web Services.....	14
4	Know Issues / Reporting an Issue.....	15
	Known Issues.....	15
	Reporting an Issue.....	15
	Appendix A – Available Web Services	Error! Bookmark not defined.

1 DOCUMENT OVERVIEW

This document is intended to provide an overview of the Avaya Aura Contact Center Outbound WS capabilities for the attached SDK application Implementation.

The Outbound SDK Implementation is designed to provide a starting point for developers and system integrators to customize the outbound application for their own implementation.

The Outbound WS will only ever work with active campaigns; campaign that are closed, cancelled or finished will never be returned or modified. You cannot create / delete campaigns or change the status of any campaigns.

2 Outbound SDK Overview

The Outbound SDK Reference Implementation utilizes the SDK to build a sample application which can be customized for your own requirements. This application requires a CCMM administrator username and password to be present before the user can proceed.

If your building the application or just running the built version, remember to open the .config file and rename the CCMM_SERVER_NAME to the IP of your CCMM so the application knows where to look for the Outbound Web services.

2.1 ACCS 7.0 Outbound Web Service Support

As of Contact Center 7.0 release, ACCS now supports Outbound Web Services

Release	Outbound Web Services Support
ACCS 6.x	No
ACCS 7.x	Yes

Logging in

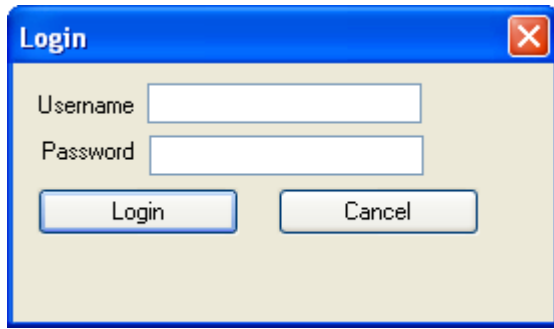
To login you must create a new admin. This admin will be used as a secure manner to access the CCMM database. The result of this login will be a session Key. This key must be used in all forthcoming Web Service calls. To destroy the session Key you must log the user out.

On your CCMA launch CCMM Admin. Navigate to “General Administration” followed by “Administrator Settings”. The standard admin users that are part of the base install will not work with the Outbound Web Services; these have been filtered out for security.

Click New and enter the Admin details as required.

Click Save.

Open the attached SDK and login with the above created details.



The Login Function Call for this window is ...

```
OutboundUtilityWS.OutboundLogin(username, password);
```

The Logout Function Call for this application is...

```
OutboundUtilityWS.OutboundLogoff(username);
```

You will be notified if the user is already logged in, and given the option to log out that user. You will also be notified should you enter invalid credentials.

Understanding the SDK UI

Once successfully logged in, you will see your current active campaigns on the left. Which include campaigns of the status...

- Running
- Waiting
- Paused

It will not show campaigns that are

- Closed
- Completed
- Cancelled

The screenshot displays two overlapping windows from the Avaya Contact Center Outbound application.

Modify Campaign Window:

- Left Panel:** Contains icons for 'Create & Append', 'Modify Campaign', and 'Progress & Results'.
- Campaign List:** A list box showing 'Test_03' (selected), 'Test_02', and 'Test_01'.
- Refresh Campaign Details:** A button below the list.
- Campaign Summary:** A table showing details for 'Test_03' as of 28/03/2011 17:04:16 (Server Time).

Details	Value
Name	Test_03
Status	Waiting
<input type="checkbox"/> Campaign Settings <input type="checkbox"/> Call Settings <input type="checkbox"/> Date/Time Settings <input type="checkbox"/> Dialler Settings <input type="checkbox"/> Disposition Codes	

Outbound Campaign Tool Window:

- List Of Active Campaigns:** A list box showing 'Test_02' and 'Test_03' (selected).
- Summary Row:**

Status	Contacts Open	Contacts Closed	Contacts Processed	Call Rate	Priority	Owner	ID
Waiting	5	0	0	-1	6	webadmin	3
- Contact List Table:**

	Title	First Name	Last Name	International Code	Area Code	Phone
▶						1
						2
						3
						4
						5
*						
- Buttons:** 'Refresh Campaign List', 'Close Contact', 'Get Contact Status', 'Get Contact ID', 'Add Contact', 'Find Contact'.

Getting Active Campaigns

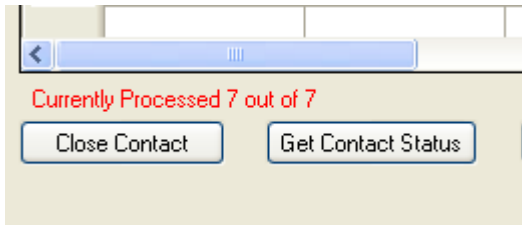
The Refresh Campaign List will pull down the latest data from the CCMM Database.

The Get Campaign Function Call for this application is...
`OutboundCampaignWS.GetCampaigns(MySessionKey) ;`

On the top of the application we have details about the selected campaign. These results are returned as part of the above function.

Getting Campaign Contacts

The middle of the window shows the contact details associated with the selected campaign. The application calls the following function asynchronously. With an update status of how many contacts are loaded and how many need to be loaded.



The web service can only return a max of 200 contacts for each call so it may be necessary to call this function repeatedly for each contact in the campaign.

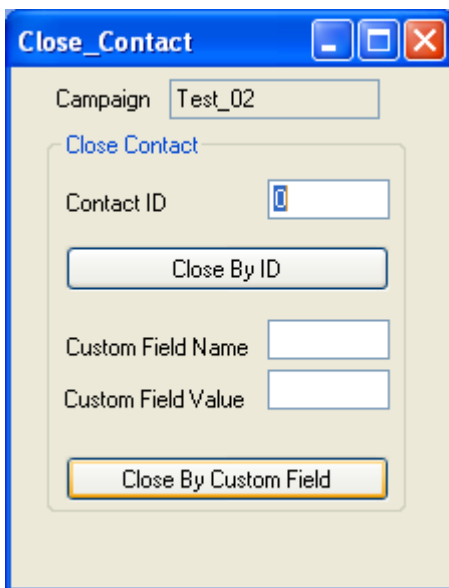
The Get Contacts Function Call for this application is...

```
OutboundCampaignWS. Get200CampaignContacts(CampaignID, nextContactID, MySessionKey);
```

The `nextContactID` should be Zero for the first call to get the first 200 contacts, the second call should be the last received contact id from the last call to get the next 200 contacts and so on until the number of contacts received is equal to that in the Campaign Details.

Closing a Contact

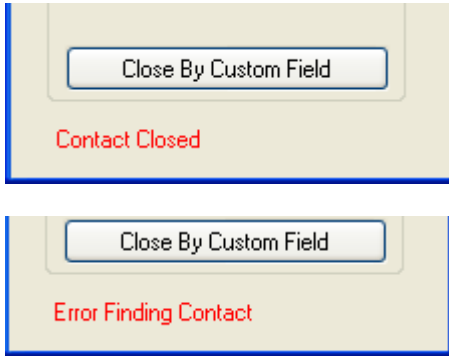
Starting from left to right, The Close Contact button will close a contact in the selected campaign. The user has a choice to close the contact by Contact ID or by Custom Field values (if they are created when the campaign was created.)



The Close Contacts Function Call for this application is...

```
OutboundContactWS.CloseContact(CampaignID, ContactID, CustomField Name, CustomField Value, UseCustomFieldForClosure, ClosureText, DispositionID, DispositionCodeSpecified, MySessionKey);
```

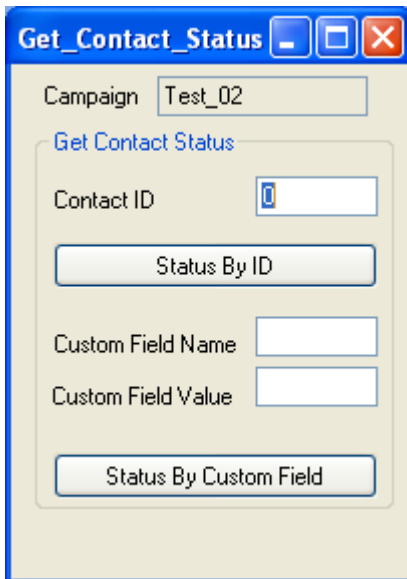
If you choose to close the contact via the custom field values then `UseCustomFieldForClosure` must be set to true, otherwise this should be set to false. The `DispositionID` is part of the data received from the campaign Details call; you will only be able to close the contact with disposition code that is linked to that campaign. It is recommend use always use the “Completed” disposition code. The `ClosureText` is any message you would like to attach for the reason of the closure, this should be human readable text.



Getting the Status of a Contact

The Get Contact Status button will return the contacts status.

The user has a choice to find the contact status by Contact ID or by Custom Field values (if they are created when the campaign was created.)



The Get Contacts Status Function Call for this application is...

```
OutboundContactWS.GetContactStatus(MySessionKey, ContactID, CampaignID,
CustomField Name, CustomField Value, UseCustomField);
```

If you choose to get the contact status via the custom field values then `UseCustomField` must be set to true, otherwise this should be set to false.

Status By Custom Field

Status = New

Status By Custom Field

Status = Closed

Get Contact ID by Phone Number

The Get contact ID by Phone Number will return the contact Id for the sent in phone number. Fields can be left blank if for instance the contact has no international number. You can only search for contacts that are within the selected campaign. If more than one contact has the same number then only the first will be returned. If the contact is not found then “-1” is returned.

Get_Contact_ID

Campaign

Test_02

Get Contact ID

Int Code

Are Code

Phone No

Get ID

The Get Contacts Status Function Call for this application is...

```
OutboundContactWS.GetContactIDByPhoneNumber(ContactNumber, CampaignID, MySessionKey);
```

Get ID

Contact ID = 191

Get ID

Contact not found

Add Contact

The Add Contact Function call is used to add a new contact to a selected campaign. Any Custom Fields that are created with the campaign will be visible on this window. The user must also know the Skillset ID they wish to use prior to adding in the new contact. All fields are options with the exception of the Skillset ID field. The web service does not check for duplicate contacts before adding the new one. The function can only be called once every 1 second

The return value is “1” if the function succeeds.

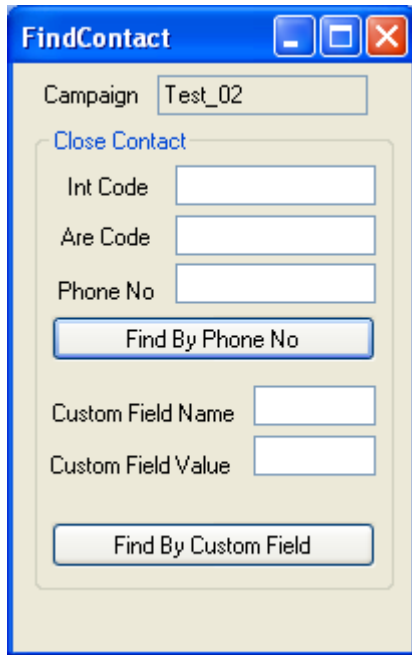
The Get Contacts Status Function Call for this application is...

```
OutboundCampaignWS.AppendContactsToCampaign(CampaignID), ContactDetails, MySessionKey);
```

Find a Contact

The Find Contact button will find a contact in the selected campaign.

The user has a choice to find the contact by Contact ID or by Custom Field values (if they are created when the campaign was created). You have the choice of narrowing down the search by contact status.



The image shows a Windows-style dialog box titled "FindContact". It has a blue title bar with standard minimize, maximize, and close buttons. The main area is light beige. At the top, there is a "Campaign" label followed by a text box containing "Test_02". Below this is a section titled "Close Contact" in blue text. Inside this section, there are three input fields: "Int Code", "Are Code", and "Phone No". Below the "Phone No" field is a button labeled "Find By Phone No". Further down, there are two more input fields: "Custom Field Name" and "Custom Field Value". Below these is a button labeled "Find By Custom Field".

The Find a Contact Function Calls for this application is...

```
OutboundContactWS.FindContactsByPhoneNumber(ContactStatus, CampaignID,
ContactNo, MySessionKey);
```

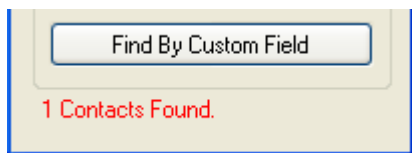
```
OutboundContactWS.FindContactsByCustomField(ContactStatus,CampaignID,
CustomField, MySessionKey);
```

Note the Phone No filed is required and could not be empty.

If you choose to find the contact via the custom field values then use

“FindContactsByCustomField” function call, otherwise use

“FindContactsByPhoneNumber”. The `ContactStatus` is an `OWStatus` status object that can be used to narrow down the search to only new or closed contacts. If you don’t wish to do this then you can pass in the object as `null`.



The image shows a close-up of the "Find By Custom Field" button from the previous dialog box. Below the button, the text "1 Contacts Found." is displayed in red.

3 Available Web Services

[http\(s\)://CCMM_SERVER_NAME/ccmmoutboundwebservices/OutboundContactWS.aspx](http(s)://CCMM_SERVER_NAME/ccmmoutboundwebservices/OutboundContactWS.aspx)

[http\(s\)://CMM_SERVER_NAME/ccmmoutboundwebservices/OutboundUtilityWS.aspx](http(s)://CMM_SERVER_NAME/ccmmoutboundwebservices/OutboundUtilityWS.aspx)

[http\(s\)://CCMM_SERVER_NAME/ccmmoutboundwebservices/OutboundCampaignWS.aspx](http(s)://CCMM_SERVER_NAME/ccmmoutboundwebservices/OutboundCampaignWS.aspx)

4 Know Issues / Reporting an Issue

Known Issues

None

Reporting an Issue

Any issue found must be reproducible on the base plug-in package without any modification. Send the steps to reproduce to the Avaya Support team.