

# Outbound APIs Sample Code READ-ME

- Requirements, Getting Started, and Building the Toolkit
- Introduction
  - Prerequisites
- Campaign Demo
- Import Demo
- Personal CallBack Demo
- Do Not Call Demo
- Reading the CSV file in browser for Import and Personal Callback bulk create (Html/Javascript)
- Handling Time Zones
  - Time Zone API
  - GMTPhone## Column - Dialing List Table
    - Calculation for Time Zones Behind GMT
    - Calculation for Time Zones Ahead of GMT
  - Conversion and Presentation

## Requirements, Getting Started, and Building the Toolkit

For information about requirements, getting started, and building the toolkit, please refer to the following document:

- UnifiedConfig-Example-API-Toolkit-README.pdf

## Introduction

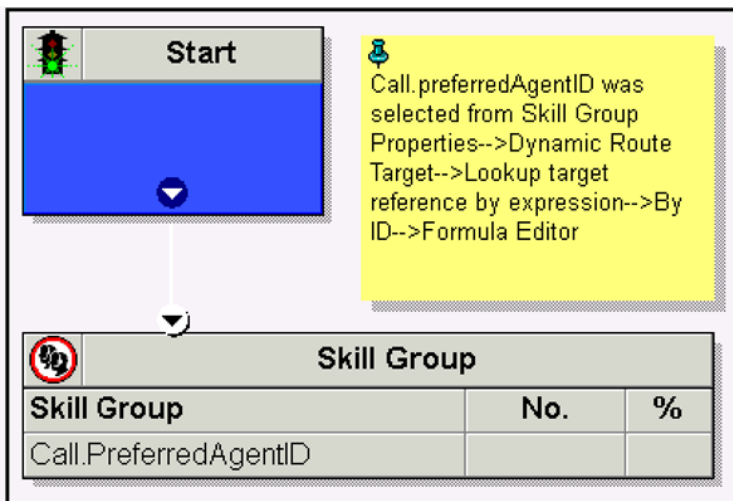
An Outbound Campaign makes outgoing calls to customers for a specific purpose or task and delivers these calls to agents.

The Outbound APIs are REST-based APIs. They are designed for 3rd party Campaign Manager integration and assume that the developer provides much of the dialing list management function. Pre-filtering of dialing lists is done by the application. Because of this, Campaign creation is streamlined. Creating a Campaign automatically creates a single underlying Import Rule and a single associated Query Rule.

The Outbound API is supported for all 11.5 deployments: Unified CCE, Packaged CCE, and HCS for Contact Center.

Admin scripts are not needed. However, if they are added, they will override what is defined as part of the campaign.

Routing scripts for Agent campaigns can be simplified using Dynamic Route Target by ID in the Skill Group node.



## Prerequisites

For Unified CCE and HCS for Contact Center deployments, at least two Skill Groups and one Agent should already exist. The names for these two Skill Groups must be provided as a command-line argument to the CampaignDemo and ImportDemo, and the Agent's username must be provided for PersonalCallbackDemo.

For Packaged CCE deployment, Skill Group names are an optional command-line arguments for CampaignDemo and ImportDemo and the Agent's username is an optional command-line argument for PersonalCallbackDemo. If they are provided, they are used. Otherwise, the java samples will create new Skill Groups or Agent and delete them at the end of the Demo.

## Campaign Demo

Campaign creation via the API requires only four fields to be specified (the rest of the fields will have the default values):

1. Campaign name
2. Dialing mode (Inbound, predictive only, preview only, progressive only, preview direct only)
3. Skill Group information
4. Time Zone

Campaigns can be created, started, stopped, updated, and deleted using the APIs. Campaigns created using the APIs are not visible in the configuration management tools.

A Runtime Status API is provided to get the real-time status of a running campaign. Data is updated every 15-30 seconds.

### Supported Operations:

HTTP Method	Operation
POST	Create a campaign
PUT	Update a campaign
GET	Get a single campaign
GET	Get a list of campaigns
DELETE	Delete a campaign

### Demo:

The CampaignDemo class is in the examples package.

**To launch the program from the command line** (passing in the IP address, username and password for an AW/Data Server, SkillGroup1 name, SkillGroup2 name), **type**:

```
mvn exec:java -Dexec.mainClass="com.cisco.ccbu.cce.unifiedconfig.toolkit.examples.CampaignDemo" -Dexec.args="10.86.135.210 administrator@demo.com abc123= 5001 5002"
```

### The demo performs the following operations:

1. Creates a Campaign with only a Name, Dialing Mode, Skill Group info and Time Zone.
2. Updates a value in the Campaign configuration and ensures that it is updated.
3. Creates one more Campaign.
4. Gets a list of Campaigns created by using a search term common to the two campaigns created.
5. Deletes the Campaigns.
6. Performs a GET on a Campaign with a non-existent CampaignID and prints out the error.

All supported Create, Read, Update, and Delete operations are demonstrated in this demo.

### Troubleshooting:

If there are errors in creating the Campaign, the error message will describe the errors. It is very likely that these errors are because the SkillGroups 1 or 2 or both are already associated with other Campaigns. An exclusive SkillGroup is required for each Campaign.

## Import Demo

The Import API is used to import customer contact information for an Outbound Campaign.

A limited set of data items are allowed for Import:

1. Account number
2. First and last name
3. Phone01 through Phone10
4. TimeZoneBias and DSTObserved (see the [Handling Time Zones](#) section for details on time zone bias values)

Comma (',') and pipe ('|') delimited values are allowed in for the import data.

Up to 10,000 contacts can be imported in one API call.

#### Supported Operations:

HTTP Method	Operation
POST	Create a bulk import of contacts for a campaign
GET	Get a single contact
GET	Get a list of contacts
GET	Get a template for bulk import
DELETE	Delete all imported contacts for a campaign

#### Demo:

The Import Demo class is in the examples package.

**To launch the program from the command line** (passing in the IP address, username and password for an AW/Data Server, and the Skill Group name), **type**:

```
mvn exec:java -Dexec.mainClass="com.cisco.ccbu.cce.unifiedconfig.toolkit.examples.ImportDemo" -Dexec.args="10.86.135.210 administrator@demo.com abc123= 5001"
```

#### This demo performs the following operations:

1. Creates a campaign.
2. Imports information records for four customers using the comma-delimited values.
3. Imports information records for four more customers using the pipe-delimited values.
4. Deletes the imported records and the campaign.

Create and Delete are demonstrated in this Demo. Other operations are similar to Campaign.

#### Troubleshooting:

If there is an error in creating the Campaign, the error message will describe the error. It is very likely that the error is because the Skill Group is already associated with another Campaign. An exclusive SkillGroup is required for each Campaign.

## Personal Callback Demo

The Personal Callback API is used to configure your Outbound Campaign to handle personal callbacks. The Personal Callback feature allows an agent to schedule a callback to a customer for a specific date and time.

There is only one Personal Callback table. Contacts can be added for all campaigns using this API.

#### Supported Operations:

HTTP Method	Operation
POST	Create a single Personal Callback record
POST	Create a bulk import of Personal Callback records
PUT	Update a single Personal Callback record
GET	Get a single Personal Callback record
GET	Get a list of Personal Callback record
GET	Get a template for bulk import of Personal Callback records
DELETE	Delete a Personal Callback Record

#### Demo:

The Personal Callback Demo class is in the examples package.

**To launch the program from the command line** (passing in the IP address, username and password for an AW/Data Server, and the Agent's username), **type**:

```
mvn exec:java -Dexec.mainClass="com.cisco.ccbu.cce.unifiedconfig.toolkit.examples.PersonalCallbackDemo" -Dexec.args="10.86.135.210 administrator@demo.com abc123= agent1001"
```

**The demo performs the following operations:**

1. Creates a Personal Callback record.
2. Deletes all the records.

Create and Delete are demonstrated in this Demo. Bulk create is similar to ImportDemo bulk import. Other operations are similar to the Campaign Demo.

## Do Not Call Demo

The Do Not Call (DNC) API is used to set the DNC import rule configuration for Outbound Option so that the Campaign dialer doesn't dial the numbers in the DNC list.

There is only one DNC list for all Campaigns. The DNC API operates on a DNC Import Rule and not on the DNC list. The DNC API uses the same file detection and import process as DNC import set up via the configuration tool.

**Supported Operations:**

HTTP Method	Operation
POST	Create a DNC import rule
PUT	Update the DNC import rule
GET	Get a single DNC import rule
GET	Get a list of DNC import rules
DELETE	Delete a DNC import rule

**Demo:**

The DNC Demo class is in the examples package.

**To launch the program from the command line** (passing in the IP address, username and password for an AW/Data Server), **type:**

```
mvn exec:java -Dexec.mainClass="com.cisco.ccbu.cce.unifiedconfig.toolkit.examples.DNCDemo" -Dexec.args="10.86.135.210 administrator@demo.com abc123= agent1001"
```

**The demo performs the following operations:**

1. Creates a DNC import rule.
2. Updates the DNC import rule.
3. Deletes the DNC import rule.

Create and Delete are demonstrated in this Demo. Other operations are similar to Campaign.

## Reading the CSV file in browser for Import and Personal Callback bulk create (Html/Javascript)

The Import and Personal Callback APIs bulk create APIs require comma-delimited or pipe-delimited values for the file-content field.

This sample html/javascript page demonstrates how a comma-delimited (csv) file data can be read in a browser. Once this data is read, it can be composed as xml input required by the Import or Personal Callback bulk create APIs.

The CSVReaderDemo.html file is present in the src/main/web directory.

**To launch the sample, perform the following steps:**

1. Create a file on the file system with comma-delimited file content.
2. Open the CSVReaderDemo.html file in a modern web browser.
3. Click on the "Choose File" button and select the file created in Step 1.
4. The contents of the file will appear in the "File Content" text-area.

To see the javascript code, open the CSVReaderDemo.html file in any text editor.

## Handling Time Zones

This section explains differences in the use of time zone bias values in the Time Zone API, the Import API, and Unified CCE Outbound Option database (Dialing List table).

### Time Zone API

The Unified CCE Administration Time Zone API uses the Microsoft Windows Server 2012 time zone information that is stored in the Windows registry. Time zone bias there is expressed as the number of minutes to add to the local time to yield Greenwich Mean Time (GMT) or Universal Coordinated Time (UTC). Therefore, for the United States Eastern Standard Time, which is 5 hours behind UTC, 300 minutes must be added to the local time to get UTC. (For example, 6:00PM EST + 300 minutes = 11:00PM UTC.) For India Standard Time, which is 5 hours, 30 minutes ahead of UTC, -330 minutes must be added to (or 330 minutes subtracted from) local time to get UTC. (For example, 6:00PM IST + -330 minutes = 12:30PM UTC.)

### GMTPhone## Column - Dialing List Table

The Unified CCE Outbound Option database Dialing List (DL) table stores telephone numbers for contacts to be dialed by the Unified CCE Dialer. You may specify up to 10 telephone numbers for each contact. The column names for the telephone numbers are Phone01 through Phone10. The DL table also defines a column for each telephone number that holds the time zone offset for that telephone number; the column names are GMTPhone01 through GMTPhone10 where each GMTPhone## maps to its Phone## counterpart (GMTPhone01 is the time zone offset for Phone01). As with the Time Zone API, the value stored in the GMTPhone## column is a numerical offset from GMT, but the calculation which derives that offset differs from the Time Zone API. The calculation depends on whether the time zone is behind or ahead of GMT.

### Calculation for Time Zones Behind GMT

If the time zone is *behind* GMT/UTC, the formula is:

$$1440 - (\text{minutes behind GMT}) = \text{GMTPhone##}$$

(Note: 1440 is the number of minutes in 24 hours.)

Example: US Eastern Standard Time, which is 300 minutes behind UTC:  **$1440 - 300 = 1140$**

### Calculation for Time Zones Ahead of GMT

If the time zone is *ahead of* GMT/UTC, the offset is the number of minutes that the time zone is ahead of GMT. For example, India Standard Time is 330 minutes ahead of GMT, therefore its GMTPhone## value would be 330.

### Conversion and Presentation

When importing contacts using the Import API into the Outbound Option Dialing List table, the time zone bias value (if specified in the contact data) is converted into the appropriate GMT offset value for storage into the DL table GMTPhone## column(s). This value is presented in the `gmtOffset` XML element in the Import API response.

The formula for converting `timeZoneBias` (from Time Zone API) to `gmtOffset` is:

```
if timeZoneBias is positive:
    gmtOffset = 1440 - timeZoneBias
else if timeZoneBias is negative:
    gmtOffset = -1 * timeZoneBias
else if timeZoneBias is 0
    gmtOffset = timeZoneBias
```

This conversion is done by the Import API handler; it does not need to be done by the caller.

NOTE: If `TimeZoneBias` is not provided in the import content, time zone and daylight saving time information will be assigned by matching telephone numbers to region prefix strings. If a telephone number for a contact does not match a configured region prefix, the import uses the default time zone data for the campaign.