



## **Cisco Packaged Contact Center Enterprise Developer Reference, Release 11.5(1)**

**First Published:** 2016-08-10

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2012-2016 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### CHAPTER 1

#### Working with Cisco Packaged Contact Center Enterprise APIs 1

- Change Log 1
- API Operations 4
- Access 5
- Usage and Behavior 6
- Error Responses 7
- Pagination 8
- Shared Parameters 9
- Permissions 10
- Synchronous vs. Asynchronous Writes 10
- Search 11
- Sort 12

---

### CHAPTER 2

#### Active Directory Domain API 15

---

### CHAPTER 3

#### Administrator API 17

---

### CHAPTER 4

#### Agent API 19

---

### CHAPTER 5

#### Agent Desk Settings API 25

---

### CHAPTER 6

#### Agent State Trace API 29

---

### CHAPTER 7

#### Agent Team API 31

---

### CHAPTER 8

#### Attribute API 35

|                   |   |
|-------------------|---|
| <b>CHAPTER 9</b>  | <b><a href="#">Bucket Interval API</a></b> 39               |
| <b>CHAPTER 10</b> | <b><a href="#">Bulk Job API</a></b> 41                      |
| <b>CHAPTER 11</b> | <b><a href="#">Call Type API</a></b> 45                     |
| <b>CHAPTER 12</b> | <b><a href="#">Congestion Control API</a></b> 47            |
| <b>CHAPTER 13</b> | <b><a href="#">Context Service Configuration API</a></b> 49 |
| <b>CHAPTER 14</b> | <b><a href="#">Context Service Registration API</a></b> 51  |
| <b>CHAPTER 15</b> | <b><a href="#">Department API</a></b> 53                    |
| <b>CHAPTER 16</b> | <b><a href="#">Deployment API</a></b> 55                    |
| <b>CHAPTER 17</b> | <b><a href="#">Deployment Type Info API</a></b> 57          |
| <b>CHAPTER 18</b> | <b><a href="#">Dialed Number API</a></b> 59                 |
| <b>CHAPTER 19</b> | <b><a href="#">Expanded Call Variable API</a></b> 63        |
| <b>CHAPTER 20</b> | <b><a href="#">Facility API</a></b> 67                      |
| <b>CHAPTER 21</b> | <b><a href="#">Global API</a></b> 69                        |
| <b>CHAPTER 22</b> | <b><a href="#">Initialize API</a></b> 73                    |
| <b>CHAPTER 23</b> | <b><a href="#">Instance API</a></b> 75                      |
| <b>CHAPTER 24</b> | <b><a href="#">Internet Script Editor API</a></b> 77        |
| <b>CHAPTER 25</b> | <b><a href="#">Log Collection API</a></b> 79                |

|                   |  |
|-------------------|--|
| <b>CHAPTER 26</b> | <b><a href="#">Machine Inventory API</a></b> 81  |
| <b>CHAPTER 27</b> | <b><a href="#">Media Routing Domain API</a></b> 89   |
| <b>CHAPTER 28</b> | <b><a href="#">Network VRU Script API</a></b> 93   |
| <b>CHAPTER 29</b> | <b><a href="#">Operation API</a></b> 95  |
| <b>CHAPTER 30</b> | <b><a href="#">Outbound API</a></b> 99 <ul style="list-style-type: none"><li><a href="#">Outbound Campaign API</a> 100</li><li><a href="#">Campaign Status API</a> 106</li><li><a href="#">Do Not Call API</a> 108<ul style="list-style-type: none"><li><a href="#">Create Do Not Call List</a> 109</li></ul></li><li><a href="#">Import API</a> 110</li><li><a href="#">Personal Callback API</a> 114</li><li><a href="#">Time Zone API</a> 119</li></ul> |
| <b>CHAPTER 31</b> | <b><a href="#">Peripheral Gateway API</a></b> 121  |
| <b>CHAPTER 32</b> | <b><a href="#">Precision Queue API</a></b> 123   |
| <b>CHAPTER 33</b> | <b><a href="#">Reason Code API</a></b> 127   |
| <b>CHAPTER 34</b> | <b><a href="#">Role API</a></b> 129  |
| <b>CHAPTER 35</b> | <b><a href="#">Routing Type API</a></b> 133  |
| <b>CHAPTER 36</b> | <b><a href="#">Scan API</a></b> 135  |
| <b>CHAPTER 37</b> | <b><a href="#">Serviceability API</a></b> 137 <ul style="list-style-type: none"><li><a href="#">System Validation Rules</a> 139</li></ul>  |
| <b>CHAPTER 38</b> | <b><a href="#">Single Sign-On Global State API</a></b> 143   |

---

|                   |  |
|-------------------|--|
| <b>CHAPTER 39</b> | <b><a href="#">Single Sign-On Registration API</a></b> 145 |
| <hr/>             |  |
| <b>CHAPTER 40</b> | <b><a href="#">Single Sign-On Status API</a></b> 147       |
| <hr/>             |  |
| <b>CHAPTER 41</b> | <b><a href="#">Skill Group API</a></b> 151                 |
| <hr/>             |  |
| <b>CHAPTER 42</b> | <b><a href="#">Stats API</a></b> 155                       |
| <hr/>             |  |
| <b>CHAPTER 43</b> | <b><a href="#">Status API</a></b> 157                      |
|                   | <a href="#">Configuration Rules</a> 158                    |
|                   | <a href="#">Operation Rules</a> 165                        |
|                   | <a href="#">System Health Rules</a> 171                    |
|                   | <a href="#">VM Rules</a> 171                               |
| <hr/>             |  |
| <b>CHAPTER 44</b> | <b><a href="#">Trace Level API</a></b> 173                 |
| <hr/>             |  |
| <b>CHAPTER 45</b> | <b><a href="#">Uninitialize API</a></b> 175                |
| <hr/>             |  |
| <b>CHAPTER 46</b> | <b><a href="#">Version API</a></b> 177                     |



# Working with Cisco Packaged Contact Center Enterprise APIs

---

Cisco Packaged Contact Center Enterprise (Packaged CCE) uses [REST](#)-based API functions accessed over HTTP. Each API Operation is mapped to an HTTP method. For more information, see [API Operations](#), on [page 4](#).

This document explains the operations and parameters for each configurable item in Packaged CCE.

For more information on managing the system, see the *Cisco Packaged Contact Center Enterprise Administration and Configuration Guide* at <http://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-maintenance-guides-list.html>.

To review examples on how to interact with the Unified Config and Outbound REST web services in Java, see the CCE Config Sample REST Toolkit at <https://developer.cisco.com/site/packaged-contact-center/documentation/>.

- [Change Log](#), [page 1](#)
- [API Operations](#), [page 4](#)
- [Access](#), [page 5](#)
- [Usage and Behavior](#), [page 6](#)
- [Error Responses](#), [page 7](#)
- [Pagination](#), [page 8](#)
- [Shared Parameters](#), [page 9](#)
- [Permissions](#), [page 10](#)
- [Synchronous vs. Asynchronous Writes](#), [page 10](#)
- [Search](#), [page 11](#)
- [Sort](#), [page 12](#)

## Change Log

This section notes the new and changed APIs in this Packaged CCE release.

**New APIs**

- [Context Service Configuration API](#), on page 49
- [Context Service Registration API](#), on page 51
- [Do Not Call API](#), on page 108
- [Import API](#), on page 110
- [Internet Script Editor API](#), on page 77
- [Outbound Campaign API](#), on page 100
- [Personal Callback API](#), on page 114
- [Single Sign-On Global State API](#), on page 143
- [Single Sign-On Registration API](#), on page 145
- [Single Sign-On Status API](#), on page 147
- [Stats API](#), on page 155
- [Time Zone API](#), on page 119

**Updated APIs**

| See   | Notes  |
|---|--|
| <a href="#">Agent Desk Settings API</a> , on page 25  | Changed the remoteAgentType parameter. Instead of just two possible values (0: Not allowed, 2: Allowed), there are now two new options (1: Call by Call, 3: Agent Chooses). The 2: Allowed option is now described as Nailed Up.   |
| <a href="#">Agent API</a> , on page 19                | <p>New parameters:</p> <ul style="list-style-type: none"> <li>• ssoEnabled - This parameter is sortable.</li> </ul> <p>The supervisorUserInfo parameter has been removed.</p> <p>For supervisors and for agents with single sign-on (SSO) enabled, the userName parameter now uses the user's Active Directory or SSO account username.</p> <p>For supervisors who are not enabled for single sign-on (SSO), the Active Directory username must be in the user@domain format in the userName parameter .</p> |
| <a href="#">Bulk Job API</a> , on page 41             | New ssomigration jobType.  |
| <a href="#">Deployment Type Info API</a> , on page 57 | Deployment type names have been updated.   |



| See  | Notes  |
|--|--|
| <a href="#">Machine Inventory API, on page 81</a>                | <p>The Machine Inventory API now has SM_REST_API, CONTEXT_SERVICE, IDS, IDS_PRIMARY_REF, IDS_SECONDARY_REF, and PUBLISHER_REF services.</p> <p>Removed the CCE_CALL_SERVER and CCE_DATA_SERVER machine types.</p> <p>Added the CCE_ROGGER, CCE_PG, CCE_AW, CVVB, and ECE machine types.</p> <p>Changed the number of allowed CVP machines on each side from 2 to 1.</p> <p>The hostName parameter is now searchable.</p> <p>The API now has two advanced search parameters: type:&lt;machine type&gt; and serviceType:&lt;service type&gt;.</p>  |
| <a href="#">Media Routing Domain API, on page 89</a>             | <p>The Media Routing Domain API now supports create, update, and delete operations.</p> <p>New parameters:</p> <ul style="list-style-type: none"> <li>• id</li> <li>• type</li> <li>• taskLife</li> <li>• taskStartTimeout</li> <li>• maxTaskDuration</li> </ul> <p>Renamed parameters:</p> <ul style="list-style-type: none"> <li>• maxCallsInQueue has been renamed maxTasksInQueue</li> </ul> <p>Removed parameters:</p> <ul style="list-style-type: none"> <li>• maxCallsInQueuePerCallType</li> <li>• mediaClass</li> </ul>   |
| <a href="#">Precision Queue API, on page 123</a>                 | New mediaRoutingDomain parameter.  |
| <a href="#">Status API &gt; Configuration Rules, on page 158</a> | <p>Configuration rules have been reorganized under new machine names (Unified CCE Rogger, Unified CCE PG, Unified CCE AW-HDS-DDS, CUIC-LD-IdS).</p> <p>New SocialMiner machine rules.</p> <p>New rules for single-sign on for Unified CCE AW-HDS-DDS, CUIC-LD-IdS, Finesse, and external HDS.</p> <p>New rule that hostnames must be in FQDN format, for all machines except gateways and Cisco Virtualized Voice Browser.</p> <p>CUIC_HISTORICAL_REALTIME_DS_CORRECT_CONFIGURATION has been removed from the rules for the Unified Intelligence Center Publisher.</p> <p>New rule that each Peripheral Gateway with a Communications Manager PIM must have an associated Task Routing Application Path.</p> |

| See  | Notes   |
|--|---|
| Status API ><br><a href="#">Operation Rules</a> , on<br>page 165 | Operation rules have been reorganized under new machine names (Unified CCE Rogger, Unified CCE PG, Unified CCE AW-HDS-DDS, and CUIC-LD, IdS).<br><br>New SocialMiner machine rules. |

## API Operations

There are five API operations, and they are invoked by HTTP methods.

Responses are provided using HTTP headers and HTTP body containing XML. For information on XML, see [XML](#), on page 7.

### create

The create operation uses the HTTP POST method to make one new item and return the URL of that item in the HTTP location header. That URL can then be used to perform the get, update, and delete operations. An XML body containing the parameters and values for the new item must be specified.

### delete

The delete operation uses the HTTP DELETE method to delete one item. The item may be marked for deletion or permanently deleted depending on the item type.

To delete more than one item at a time, refer to the Operation API.

You cannot delete **BuiltIn** items (those automatically created by the system, such as the **BuiltIn** bucket interval), items referenced in scripts, or items referenced by other items.

### get

The get operation uses the HTTP GET method to retrieve one item. For example, to return one bucket interval record, perform the get operation using the URL:

```
https://<server>/unifiedconfig/config/bucketinterval/<id> .
```

### list

The list operation uses the HTTP GET method to retrieve a list of items. For example, to retrieve a list of bucket intervals, perform the list operation using the URL:

```
https://<server>/unifiedconfig/config/bucketinterval. See also Permissions, on page 10, Pagination, on page 8, Search, on page 11, and Sort, on page 12.
```

### Query parameters:

- Summary list: Some APIs have parameters that include a large amount of data when returned, such as collections of references. Use this query parameter to reduce the number of parameters returned for each item in the list. For example, in the Skill Group API, if skill groups contain a large number of agents, a large amount of data may be returned. Use this query option to return the basic skill group data along with the number of agents having the skill. Append the query parameter `summary=true` to the URL for the API; for example,

```
https://<server>/unifiedconfig/config/skillgroup?summary=true.
```

### update

The update operation uses the HTTP PUT method to modify one item. An XML body containing the parameters and values to update must be specified. For example, to update the name of a bucket interval, perform the update operation on the URL

`https://<server>/unifiedconfig/config/bucketinterval/(id)` with the following body:

```
<bucketInterval>
  <name>newName</name>
  <changeStamp>0</changeStamp>
</bucketInterval>
```

## Access

### Administrator Access

Administrator access to Packaged CCE Administration APIs and items is defined by the role and by the departments for which the administrator is responsible.

### Supervisor Access

The following APIs are read only:

- [Agent Team API, on page 31](#)
- [Precision Queue API, on page 123](#)

The following APIs allow update with restrictions:

- [Attribute API, on page 35](#)
  - When updating an attribute, supervisors can only modify the collection of agentAttributes by adding, removing, or changing the value of agents who are on their teams.
- [Agent API, on page 19](#)
  - Supervisors can only see and update agents who are on their teams.
  - When updating an agent, supervisors can only change the following parameters:
    - skillGroups
    - defaultSkillGroup
    - skillGroupsAdded
    - skillGroupsRemoved
    - agentAttributes
    - agentAttributesAdded
    - agentAttributesRemoved
    - password
  - The Operation API can also be used to perform updates on agents.

- [Skill Group API, on page 151](#):
  - When updating a skill group, supervisors can only modify the collections of agents by adding or removing agents who are on their teams.

### Agent Access

Agents have no access to the Packaged CCE Administration APIs.

### Authentication

To authenticate administrators and supervisors must provide a fully qualified user name (for example, user@domain.com) and password.

## Usage and Behavior

### Duplicate Parameters

If a parameter is duplicated, the final value that is specified will be used by the API.

### Read-Only Fields

Read-only parameters are ignored on create and update operations.

### References

References are a type of parameter that provide a way to connect one item to another item, defining the relationship between them.

For example, to define which team an agent belongs to, the agent contains a reference to a team. When performing list or get operations, the reference contains the refURL of the item and the name. For example:

```
<agent>
  <team>
    <refURL>/unifiedconfig/team/5000</refURL>
    <name>NameOfTeam</name>
  </team>
  ...
</agent>
```

For items that do not have a name parameter, other parameters such as firstName and lastName are included.

```
<agent>
  <refURL>/unifiedconfig/config/agent/5000</refURL>
  <firstName>Jane</firstName>
  <lastName>Doe</lastName>
  <userName>username</userName>
  <agentId>8007</agentId>
  <canRemove>true</canRemove>
</agent>
```

When doing create or update, only the refURL parameter is required. Additional parameters are ignored. For example:

```
<agent>
  <team>
    <refURL>/unifiedconfig/team/5000</refURL>
  </team>
  ...
</agent>
```

Items can also contain a collection of references. For example, if an agent belongs to multiple skill groups, the skillGroups parameter contains a reference to each associated skill group:

```
<agent>
  <skillGroups>
    <skillGroup>
      <refURL>/unifiedconfig/config/skillgroup/5001</refURL>
      <name>FirstSkill</name>
    </skillGroup>
    <skillGroup>
      <refURL>/unifiedconfig/config/skillgroup/5005</refURL>
      <name>AnotherSkill</name>
    </skillGroup>
  </skillGroups>
  ...
</agent>
```

If the referenced item belongs to a department, then department information is included within that reference:

```
<agentDeskSettings>
  <refURL>/unifiedconfig/config/agentdesksetting/5001</refURL>
  <name>mike</name>
  <department>
    <refURL>/unifiedconfig/config/department/5003</refURL>
    <name>d1</name>
  </department>
</agentDeskSettings>
```

If the referenced item is a global item, then department information is omitted from the reference.

## XML

XML is case sensitive. When XML data is sent to the server, the tag names must match. <Name> and <name> are two different XML elements.

# Error Responses

Operations that fail return an HTTP status code ([HTTP 1.1 Status Codes](#)) indicating if there was a client error or server error. The body of the response contains a collection of API error items to provide additional information about the failure.

## Parameters

- **errorType**: Indicates the type of error. This is the primary identifier for the problem and can be used to map the type to a user readable string. For example, if your application receives an error with the errorType of invalidInput.fieldRequired, then you could display "This field is a required field; it cannot be left blank" to the user.
- **errorData**: The name of the parameter that had the error.
- **errorMessage**: Extra information about the error that is intended for the developer. This information is typically a sentence or other string. It is not localized, so it should not be shown to the user.
- **errorDetail**: Some errors contain additional detail parameters that are included in the errorDetail parameter.
  - If the error type is invalidInput.outOfRange, then errorDetail includes the following parameters:
    - **min**: The minimum value allowed.
    - **max**: The maximum value allowed.

- If you attempt to delete an item that is in use by other items, the `errorType` is `referenceViolation.api` and the `errorDetail` includes the following parameters:
  - `referenceType`: The type of item that references the item you tried to delete.
  - `references`: A collection of references, referencing the item you tried to delete, including the name and `refURL` of each referencing item.
  - `totalCount`: The total number of items referencing the item you attempted to delete.
  - `totalShown`: The total number of items included in the references collection.

### Example Error Response

The following error is returned when attempting to create a call type with a negative value for the `serviceLevelThreshold` parameter:

```
<apiErrors>
  <apiError>
    <errorData>serviceLevelThreshold</errorData>
    <errorDetail>
      <min>1</min>
      <max>2147483647</max>
    </errorDetail>
    <errorMessage>This field must contain a value from 1 to 2147483647</errorMessage>
    <errorType>invalidInput.outOfRange</errorType>
  </apiError>
</apiErrors>
```

## Pagination

Pagination allows you to limit the number of items returned by the list operation and provides information on how to get other pages.

### Query Parameters

- `startIndex`: Specifies the index of the item at which to start. Zero-based: 0 is the first item.
- `resultsPerPage`: Specifies the number of items to retrieve. Minimum: 1. Default: 25. Maximum: 100.

### Returned Parameters

- `totalResults`: Total number of items.
- `resultsPerPage`: Number of items requested per page.
- `startIndex`: The index of the first item returned. If you request a `startIndex` that is greater than total items, a full last page is returned.
- `nextPage`: The URL to get the next page. This parameter is not returned if you are on the last page.
- `prevPage`: The URL to get the previous page. This parameter is not returned if you are on the first page.
- `firstPage`: The URL to get the first page.
- `lastPage`: The URL to get the last page.
- `searchTerm`: The value specified in the search query parameter. See [Search](#), on page 11.

- **sortTerm**: The value specified in the sort query parameters. See [Sort](#), on page 12.

**Note**

Query parameters for search and sort are included in the URL.

**Example Response**

```
<pageInfo>
  <resultsPerPage>2</resultsPerPage>
  <startIndex>0</startIndex>
  <totalResults>10</totalResults>
  <firstPage> http://<server>/bucketIntervals/?resultsPerPage=2</firstPage>
  <lastPage> http://<server>/bucketIntervals/?startIndex=8&resultsPerPage=2</lastPage>

  <prevPage/>
  <nextPage> http://<server>/bucketIntervals/?startIndex=2&resultsPerPage=2</nextPage>
</pageInfo>

<bucketIntervals>
  <bucketInterval/>
  <bucketInterval/>
</bucketIntervals>
```

## Shared Parameters

**changeStamp**

- The version of the item. Initially set during a create ([create](#), on page 4) operation.
- A changeStamp is a required parameter for the body of a PUT ([update](#), on page 5) operation for items. If you do not provide a changeStamp, the update fails. This mechanism is in place so that two clients cannot edit the record at the same time.
- If the update is successful, the changeStamp is incremented.

**description**

- A description for this item.
- Optional parameter.
- No restriction of characters; OEM locale supported characters are allowed. For information on how to configure your system to support native character sets, see the latest version of the document *Cisco Packaged Contact Center Enterprise Installation and Upgrade Guide* at <http://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-installation-guides-list.html>.
- Maximum length of 255 characters.

**name**

- Required parameter.
- Maximum length of 32 characters allowed.

- Valid characters are period (.), underscore (\_), and alphanumeric. The first character must be alphanumeric.
- Does not allow internationalized characters.

#### refURL

- The identifier for an item.
- Read-only parameter.

## Permissions

Permissions information is included in list responses to indicate the write operations that the user is allowed to perform. If the API does not support any write operations, then permissions information is not returned.

#### Parameters

- **canCreate**: Indicates whether a create operation is allowed. Values are true/false. If the create operation is not supported by the API, then this parameter is not returned.
- **canUpdate**: Indicates whether an update operation is allowed. Values are true/false. If the update operation is not supported by the API, then this parameter is not returned.
- **canDelete**: Indicates whether a delete operation is allowed. Values are true/false. If the delete operation is not supported by the API, then this parameter is not returned.
- **role**: Type of role of the user performing the request. Values are administrator/supervisor.
- **departmentAdmin**: Indicates whether or not an administrator is restricted to specific departments only (true), or is a global administrator (false). Values are true/false.

#### Example Get Response

```
<permissionInfo>
  <canCreate>false</canCreate>
  <canUpdate>true</canUpdate>
  <canDelete>false</canDelete>
  <role>Administrator</role>
  <departmentAdmin>>false</departmentAdmin>
</permissionInfo>
```

## Synchronous vs. Asynchronous Writes

Synchronous API calls are blocking calls that do not return until either the change has been completed or there has been an error. For asynchronous calls, the response to the API call is returned immediately with a polling URL while the request continues to be processed. In heavier load conditions, it can be more efficient to submit multiple async calls and periodically check the status than to wait for each call to complete before submitting the next one.

The following examples describe how to use the asynchronous feature to create a call type.



## Performing Asynchronous Operations

The create, update, and delete operations can be performed asynchronously by including the query parameter `async=true`. The request is accepted if the operation is valid and the number of outstanding requests does not exceed the capacity. If the request is accepted, the response includes the following items:

- The response code is HTTP 202, indicating that the request has been accepted for processing.
- The location header specifies a URL that can be polled to receive updated information on the progress of the request.
- The response includes a body. See the next section **Asynchronous result parameters**.

## Asynchronous Result Parameters

- **progress**: Indicates the current state of the request. Values include the following states:
  - **IN\_QUEUE**: The request passed validation and capacity checks and was put in the queue.
  - **IN\_PROGRESS**: The request is being processed.

## Polling the Asynchronous Request Status

Use the URL from the location header of an asynchronous operation request to get updated status. Responses of this request are:

- If the request has not completed yet, the response contains the HTTP 202 response code, a location header with polling URL, and a response body.
- If the request has completed, the response is identical to the responses of synchronous operations, including the following:
  - For a successful create, the response code is HTTP 201 and the location header has the URL of the created item.
  - For a successful update or delete, the response code will be HTTP 200.
  - For an unsuccessful update, a body will provide information about the failure.
- If the request has been in queue for over 30 seconds, then it is removed and an error indicates that the request timed out.

# Search

The list operation can be modified to return data you are looking for by applying the search query parameter.

## Default Search Parameters

Typically, the name and description fields are searched when specifying a search string. Refer to each API section for the default search parameters permitted. For example, a query parameter of `q=abc` causes the list operation to return only entries with a name or description containing **abc**. The search value for default parameters has the following behaviors and restrictions:

Values:

- Are case-insensitive.
- Can be contained anywhere in the parameter value.
- Can match any of the default parameters.
- Cannot include SQL wildcards. They are not supported.
- Must be URL encoded. For example, **&** must be converted to **%26** so that it is not treated as a separator for additional query parameters.

### Advanced Search

Advanced search parameters allow specific parameters to be searched. Refer to each API section for the advanced search parameters permitted. Advanced search parameters can be combined with a default search value. For example, applying the search query parameter of `q=abc routingType:1` to a dialed number list operation returns results where the routingType is set to one, and one of the default search parameters contains **abc**. Advanced search also has the following restrictions:

- Search terms must be separated by a space.
- Search terms can be specified in any order.

### Advanced Departmental Search

For APIs that include a department reference, the following advanced search parameters are allowed:

- **departments:** Search for items that belong to the any of the specified departments. For example: `q=departments:(dept1|dept2)` search for items that belong to a department whose name exactly matches **dept1** or **dept2**. The departments search must follow these guidelines:
  - Department names must match exactly.
  - Departments are separated by the pipe character.
  - Up to 10 departments can be specified.
- **globaldepartment:** Specifies if the search should return items from the global department. Values include:
  - **both:** Return items from the global department and those specified in the departments search.
  - **only:** Return items from the global department only. The departments search term is ignored.
  - **none:** Do not return items from the global department.

## Sort

A sort query parameter can be used to specify the order of the results in a list response.

The query parameter is **sort=<parameterName> order**, where:

- **parameterName:** The name of the parameter that you want to sort on. This is case sensitive, so it must match the parameter in the API exactly.
- **order:** Specifies the order of the sort. Values are as follows:

- asc: Perform an ascending sort. This is the default if no order is specified.
- desc: Perform a descending sort.

**Example**

For example, to find all the CallTypes whose name or description contains *supervisor*, sorted in ascending order by *name*:

```
https://<server>/unifiedconfig/config/calltype?q=supervisor&sort=name
```





## Active Directory Domain API

Use the Active Directory Domain API to list the active directory domains currently defined in your call center environment. It is read-only, and does not require authentication.

### URL

`https://<server>/unifiedconfig/config/activedirectorydomain`

### Operations

- [list](#): Retrieves a list of active directory domains.

### Parameters

- `name`: The name of the domain.
- `systemDomain`: Indicates if the system is a member of this domain. Values are true/false.

### Example Get Response

```
<results>
  <activeDirectoryDomains>
    <activeDirectoryDomain>
      <name>boston.com</name>
      <systemDomain>true</systemDomain>
    </activeDirectoryDomain>
    <activeDirectoryDomain>
      <name>cisco.com</name>
      <systemDomain>>false</systemDomain>
    </activeDirectoryDomain>
  </activeDirectoryDomains>
</results>
```





## Administrator API

An administrator is an Active Directory user who has been provided access to the system.

That access can be controlled by assigning the administrator to roles and departments (see [Role API](#), on page 129 and [Department API](#), on page 53) .

Use the Administrator API to list the administrators currently defined in the database, define new administrators, and view, edit, and delete existing administrators.

### URL

`https://<server>/unifiedconfig/config/administrator`

### Operations

- [create](#): Creates one administrator.
- [delete](#): Permanently deletes one administrator.
- [get](#): Returns one administrator, using the URL  
`https://<server>/unifiedconfig/config/administrator/<id>`.
- [list](#): Retrieves a list of administrators.
- [update](#): Updates one administrator.

### Parameters

- `refURL`: The refURL of the administrator. See [Shared Parameters](#), on page 9.
- `changeStamp`: See [Shared Parameters](#), on page 9.
- `description`: See [Shared Parameters](#), on page 9.
- `userName`: Required. The unique username of an existing Active Directory account. Maximum length of 64 characters.
- `domainName`: The domain for this administrator. If blank, system uses the default domain name. Maximum length of 64 characters.
- `departments`: A collection of department ([Department API](#), on page 53) references associated with this administrator, including the refURL and name. Leave this collection empty to allow the administrator to have access to all departments. See [References](#), on page 6.

- **role:** A reference to a role ([Role API, on page 129](#)), including refURL and name. This parameter sets access to specific features. Automatically creates membership to Active Directory Setup group or Config group. If no role is assigned, the administrator is not placed in either group and does not have access to any of the web configuration tools, the Configuration Manager, or the Script Editor. See [References, on page 6](#).
- **readOnly:** Required. Specifies whether the administrator has read-only access to the APIs and tools. Values are true/false.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• userName</li> <li>• domainName</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• userName</li> <li>• domainName</li> <li>• description</li> </ul> |

See [Search, on page 11](#) and [Sort, on page 12](#).

### Example Get Response

```
<administrator>
  <changeStamp>3</changeStamp>
  <refURL>/unifiedconfig/config/administrator/5000</refURL>
  <domainName>domain</domainName>
  <userName>user1</userName>
  <departments>
    <department>
      <refURL>/unifiedconfig/config/department/5000</refURL>
      <name>dept1</name>
    </department>
    <department>
      <refURL>/unifiedconfig/config/department/5001</refURL>
      <name>dept2</name>
    </department>
  </departments>
  <description>desc</description>
  <readOnly>true</readOnly>
  <role>
    <refURL>/unifiedconfig/config/role/5005</refURL>
    <name>ConfigAdmin</name>
  </role>
</administrator>
```





## Agent API

Agents respond to contacts from customers. Use the Agent API to list the agents currently defined in the database, define new agents, and view, edit, and delete existing agents.

### URL

`https://<server>/unifiedconfig/config/agent`

### Operations

- **create**: Creates an agent.
- **delete**: Marks one agent for deletion, but does not permanently delete the agent.
- **get**: Returns one agent, using the URL  
`https://<server>/unifiedconfig/config/agent/<id>`.
- **list**: Retrieves a list of agents.
  - **Query parameters:**
    - **selectedAttribute**: Use this query parameter to augment the returned agent parameters with an extra parameter called `selectedAttribute`. This parameter indicates if the agent belongs to the attribute with the ID specified in this query parameter. For example, to find out which agents belong to the specified attribute, add `selectedAttribute=5000`.



#### Note

Using `selectedAttribute` automatically sets the summary list query parameter to `true`.

- **selectedSkillGroup**: Use this query parameter to augment the returned agent parameters with an extra parameter called `selectedSkillGroup`. This parameter indicates if the agent belongs to the skill group with the ID specified in this query parameter. For example, to find out which agents belong to the specified skill group, add `selectedSkillGroup=5001`.
- **ignoreRole**: Use this query parameter to allow a supervisor to see a list of all agents in the system, including agents the supervisor does not supervise. For example, to see all agents, add `ignoreRole=true`.
- **Summary list**: See [list](#), on page 4.

- [update](#): Updates one agent.

## Parameters

- `refURL`: The refURL for the agent. See [Shared Parameters, on page 9](#).
- `agentId`: The unique peripheral number. Maximum length of 11 characters allowed. Default is an autogenerated 7-digit number.
- `changeStamp`: See [Shared Parameters, on page 9](#).
- `description`: See [Shared Parameters, on page 9](#).
- `department`: A reference to the agent's department ([Department API, on page 53](#)), including the refURL and name. See [References, on page 6](#).
- `agentStateTrace`: Indicates if agent state tracing is turned on for the agent. True or false. See [Agent State Trace API, on page 29](#).
- `agentDeskSettings`: A reference to the agent's agentDeskSettings ([Agent Desk Settings API, on page 25](#)), including the refURL and name. See [References, on page 6](#).
- `person`: Required. Includes the following parameters:
  - `firstName`: Agent's first name. Maximum of 32 characters. International characters are allowed.
  - `lastName`: Agent's last name. Maximum of 32 characters. International characters are allowed.
  - `password`: Agent's password. Maximum of 256 ASCII characters. Password is case-sensitive. The password can be used when creating or updating, but is not returned.  
If the `ssoEnabled` parameter is set to true, the password is not saved.
  - `userName`: Agent's login name. Maximum of 255 ASCII characters. Must be unique. The login name supports the use of all characters from 33 to 126 in the ASCII character set, except for the following: double quotation mark ("), forward slash (/), backward slash (\), square brackets ([ ]), colon (:), semicolon (;), pipe (|), equal to (=), comma (,), plus sign (+), asterisk (\*), question mark (?), angle brackets (<>), hash (#), percent (%), and SPACE.  
For supervisors and for agents with single sign-on (SSO) enabled, the username is the user's Active Directory or SSO account username.  
For supervisors who are not enabled for single sign-on (SSO), the Active Directory username must be in the user@domain format.
  - `loginEnabled`: Whether the agent can log in. True or false. Default is true.
  - `ssoEnabled`: Whether single sign-on is supported at the agent level. True or false. Default is false. This parameter takes effect only when the global level `SSOEnabled` is mixed.
- `supervisor`: Required. Indicates whether the agent is marked as supervisor. True or false.  
If set to true, the person `userName` is used for the supervisor username and domain.
- `agentAttributes`: A collection of agent attribute ([Attribute API, on page 35](#)) references for this agent, including the refURL, and read-only parameters name, dataType, and description for each associated attribute. Also includes the attributeValue parameter which indicates the value (true or false or 1-10), and description of the attribute for this agent. See [References, on page 6](#).

- **agentAttributesAdded**: A collection of agent attribute references ([Attribute API, on page 35](#)) to be added to the agent, including the agent refURL and the attributeValue of each attribute. If the attributeValue is not specified, it is assigned the default value. Agents that already have this attribute are updated with the specified attributeValue. This parameter is update only, and cannot be used with the agentAttributes parameter. This parameter can be used with the agentAttributesRemoved parameter. See [References, on page 6](#).
- **agentAttributesRemoved**: A collection of agent attribute references ([Attribute API, on page 35](#)) to be removed from the agent, including the refURL of each attribute. This parameter is update only, and cannot be used with the agentAttributes parameter. This parameter can be used with the agentAttributesAdded parameter. See [References, on page 6](#).
- **selectedAttribute**: Indicates if the agent has the specified attribute. Returned only when using the selectedAttribute query parameter. True or false.
- **skillGroups**: A collection of skill group references for this agent ([Skill Group API, on page 151](#)), including the refURL and name of each associated skill group. See [References, on page 6](#).
- **skillGroupsAdded**: A collection of skill group references to be added to the agent, including the refURL of each skill group to be added. This parameter is update only, and cannot be used with the skillGroups parameter. This parameter can be used with the skillGroupsRemoved parameter. See [References, on page 6](#).
- **skillGroupsRemoved**: A collection of skill group references to be removed from the agent, including the refURL of each skill group to be removed. This parameter is update only, and cannot be used with the skillGroups parameter. This parameter can be used with the skillGroupsAdded parameter. See [References, on page 6](#).
- **defaultSkillGroup**: A reference to a skill group, including the refURL and name. Identifies the default skill group associated with this agent. See [References, on page 6](#).
- **selectedSkillGroup**: Indicates if the agent has the specified skill group. Returned only when using the selectedSkillGroup query parameter. True or false.




---

**Note** Using selectedSkillGroup automatically sets the summary list query parameter to true.

---

- **agentTeam**: A reference to the agent's team ([Agent Team API, on page 31](#)), including the refURL and name. See [References, on page 6](#).
- **supervisorTeams**: If this agent has supervisor access, this collection of references is for this supervisor's teams, including the refURL and name of each supervised team. See [References, on page 6](#).

## Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters  | Sort parameters   |
|--|---|
| <ul style="list-style-type: none"> <li>• agentId</li> <li>• description</li> <li>• person.firstName</li> <li>• person.lastName</li> <li>• person.userName</li> </ul> | <ul style="list-style-type: none"> <li>• agentId</li> <li>• description</li> <li>• supervisor</li> <li>• agentStateTrace</li> <li>• person.firstName</li> <li>• person.lastName</li> <li>• person.userName</li> <li>• person.loginEnabled</li> <li>• person.ssoEnabled</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

### Advanced search parameters

There are several advanced searches you can perform on the Agent API, including supervisor, attributes, skillgroups, team, and include and exclude (agentId).

- **supervisor: (true/false)** Find agents that are (or are *not*) supervisors.
  - **q=supervisor:true** Returns all agents who are supervisors.
  - **q=supervisor:false** Returns all agents who are *not* supervisors.
- **attributes: (attr1 & attr2 & attr3, ...)** find *all* agents that have *all* the specified attributes. Up to ten attributes can be specified. The attribute names are fully matched.
- **skillgroups: (skill1 & skill2 & skill3,...)** find *all* agents that have *all* the specified skillgroups. Up to ten skillgroups can be specified. The skillgroup names are fully matched.
- **team: (team1|team2|team3, ...)** find *all* agents who belong to *any* of the specified teams. Up to ten team names can be specified. The team name is fully matched.
- **include: (ID1 & ID2 & ID3, ...)** find *all* specified agents even if they do not meet other search criteria. Each agentId is fully matched.
- **exclude: (ID1 & ID2 & ID3, ...)** exclude *all* specified agents from the results even if they meet all other search criteria. Each agentId is fully matched.

### Example Get Response

```
<agent>
  <changeStamp>2877</changeStamp>
  <refURL>/unifiedconfig/config/agent/5017</refURL>
  <agentId>8006</agentId>
  <agentStateTrace>false</agentStateTrace>
  <description>an agent</description>
  <person>
    <firstName>Agent2</firstName>
    <lastName>Agent2</lastName>
```

```

        <loginEnabled>true</loginEnabled>
        <userName>Agent2@cisco.com</userName>
        <password>mypassword</password>
        <ssoEnalbed>>false</ssoEnabled>
    </person>
    <agentDeskSettings>
        <name>test2</name>
        <refURL>/unifiedconfig/config/agentdesksetting/5434</refURL>
    </agentDeskSettings>

    <supervisor>true</supervisor>
    <agentAttributes>
        <agentAttribute>
            <attribute>
                <refURL>/unifiedconfig/config/attribute/5004</refURL>
                <name>Sales</name>
                <dataType>4</dataType>
                <description>Sales proficiency</description>
            </attribute>
            <attributeValue>8</attributeValue>
            <description>masters certification</description>
        </agentAttribute>
    </agentAttributes>
    <skillGroups>
        <skillGroup>
            <refURL>/unifiedconfig/config/skillgroup/5229</refURL>
            <name>Support</name>
        </skillGroup>
    </skillGroups>

    <defaultSkillGroup>
        <refURL>/unifiedconfig/config/skillgroup/5229</refURL>
        <name>Support</name>
    </defaultSkillGroup>

    <agentTeam>
        <refURL>/unifiedconfig/config/agentteam/5003</refURL>
        <name>theTeam</name>
    </agentTeam>
    <supervisorTeams>
        <supervisorTeam>
            <refURL>/unifiedconfig/config/agentteam/5003</refURL>
            <name>theTeam</name>
        </supervisorTeam>
        <supervisorTeam>
            <refURL>/unifiedconfig/config/agentteam/5006</refURL>
            <name>theBTeam</name>
        </supervisorTeam>
    </supervisorTeams>
</agent>

```





## Agent Desk Settings API

A desk settings is a collection of permissions or characteristics for the agent, such as how and when calls to the agent are redirected, how and when the agent enters various work states, and how requests to the supervisor are handled.

Use the Agent Desk Settings API to list the agent desk settings currently defined in the database, define new agent desk settings, and view, edit, and delete existing agent desk settings.

### URL

`https://<server>/unifiedconfig/config/agentdesksetting`

### Operations

- **create**: Creates one agent desk settings.
- **delete**: Permanently deletes one agent desk settings.
- **get**: Returns one agent desk settings, using the URL  
`https://<server>/unifiedconfig/config/agentdesksetting/<id>`.
- **list**: Retrieves a list of agent desk settings.
- **update**: Updates one agent desk settings.

### Parameters

- **refURL**: The refURL of the agent desk settings. See [Shared Parameters, on page 9](#).
- **name**: The name of the agent desk settings. See [Shared Parameters, on page 9](#).
- **changeStamp**: See [Shared Parameters, on page 9](#).
- **description**: See [Shared Parameters, on page 9](#).
- **department**: A reference to the department ([Department API, on page 53](#)), including refURL and name. See [References, on page 6](#).
- **wrapupDataIncomingMode**: Indicates whether the agent is allowed or required to enter wrap-up data after an inbound call.
  - 0: Required

- 1: Optional (Default)
- 2: Not Allowed
- wrapupDataOutgoingMode: Indicates whether the agent is allowed or required to enter wrap-up data after an outbound call.
  - 0: Required
  - 1: Optional (Default)
  - 2: Not Allowed
- remoteAgentType: Indicates if agents are allowed to login as remote agents.
  - 0: Not Allowed
  - 1: Call by Call
  - 2: Nailed Up
  - 3: Agent Chooses
- logoutNonActivityTime: Number of seconds of non-activity at the desktop after which the software automatically logs out the agent. Value must be between 10 and 7200 seconds (default is NULL).
- workModeTimer: Specifies the auto wrap-up time out. Value must be between 1 and 7200 seconds (default is 7200).
- supervisorAssistCallMethod: Indicates how the supervisor assist request call is made.
  - 0: Consultative Call (Default)
  - 1: Blind Conference
- emergencyCallMethod: Indicates how the emergency call request is made.
  - 0: Consultative Call (Default)
  - 1: Blind Conference
- idleReasonRequired: Indicates whether the agent must enter a reason before entering the Idle state. Values are true/false.
- logoutReasonRequired: Indicates whether or not the agent must enter a reason before logging out. Values are true/false.
- autoAnswerEnabled: Indicates whether or not calls sent to this agent will be answered automatically. Values are true/false.
- agentStateAfterRONA: Indicates the agent state after RONA. Values are notReady or ready.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.



| Search parameters   | Sort parameters  |
|---|--|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> <li>• wrapupDataIncomingMode</li> <li>• wrapupDataOutgoingMode</li> <li>• remoteAgentType</li> <li>• logoutNonActivityTime</li> <li>• workModeTimer</li> <li>• supervisorAssistCallMethod</li> <li>• emergencyCallMethod</li> <li>• idleReasonRequired</li> <li>• logoutReasonRequired</li> <li>• autoAnswerEnabled</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

### Example Get Response

```
<agentDeskSetting>
  <changeStamp>3</changeStamp>
  <refURL>/unifiedconfig/config/agentdesksetting/5000</refURL>
  <agentStateAfterRONA>notReady</agentStateAfterRONA>
  <autoAnswerEnabled>false</autoAnswerEnabled>
  <emergencyCallMethod>0</emergencyCallMethod>
  <idleReasonRequired>false</idleReasonRequired>
  <logoutReasonRequired>false</logoutReasonRequired>
  <name>Default_Agent_Desk_Settings</name>
  <remoteAgentType>0</remoteAgentType>
  <supervisorAssistCallMethod>0</supervisorAssistCallMethod>
  <workModeTimer>7200</workModeTimer>
  <wrapupDataIncomingMode>1</wrapupDataIncomingMode>
  <wrapupDataOutgoingMode>1</wrapupDataOutgoingMode>
</agentDeskSetting>
```





## CHAPTER

# 6

## Agent State Trace API

Enabling agent trace allows you to track and report on every state an agent passes through. Use this feature for short-term tracking of specific agents.



### Note

The maximum number of agents with AgentStateTrace on is 100.

### URL

`https://<server>/unifiedconfig/config/agentstatetrace`

### Operations

- [list](#): Returns a list of agents whose agent state trace is turned on.
- [update](#): Updates the agent state trace in the database.

### Parameters

- `refURL`: The refURL for agent state trace. See [Shared Parameters, on page 9](#).
- `agents`: A collection of agent references ([Agent API, on page 19](#)). Each reference contains person (including `firstName`, `lastName`, `userName`, and `loginEnabled` parameters), agent `refURL`, `agentId`, `supervisor`, and `agentStateTrace`. Agents who are not specified in this collection have `agentStateTrace` turned off. To turn off all the agent state trace, pass in an empty list. See [References, on page 6](#).

### Example Get Response

```
<agentstatetrace>
  <refURL>/unifiedconfig/config/agentstatetrace</refURL>
  <agents>
    <agent xsi:type="agentSummary">
      <refURL>/unifiedconfig/config/agent/10884</refURL>
      <agentId>4294305</agentId>
      <agentStateTrace>true</agentStateTrace>
      <description>Here is a descr</description>
      <person>
        <firstName>John</firstName>
        <lastName>Doe</lastName>
        <loginEnabled>true</loginEnabled>
        <userName>jdoe</userName>
      </person>
    </agent>
  </agents>
</agentstatetrace>
```

```
        </person>
      <supervisor>false</supervisor>
    </agent>
  </agents>
</agentstatetrace>
```



## Agent Team API

You can associate a set of agents to a team with a specific supervisor. The supervisor can run reports on that team and receive Supervisor Assist requests from its members.

You can use the Agent Team API to list the agent teams currently defined in the database, define new agent teams, and view, edit, and delete existing agent teams.

### URL

`https://<server>/unifiedconfig/config/agentteam`

### Operations

- **create**: Creates an agent team.
- **delete**: Permanently deletes one agent team from the database.
- **get**: Returns one agent team, using the URL  
`https://<server>/unifiedconfig/config/agentteam/<id>`.
- **list**: Retrieves a list of agent teams.
  - **Query parameters**:
    - Summary list: See [list](#), on page 4.
- **update**: Updates one agent team.

### Parameters

- **refURL**: The refURL of the agent team. See [Shared Parameters](#), on page 9.
- **name**: The name of the agent team. See [Shared Parameters](#), on page 9.
- **description**: See [Shared Parameters](#), on page 9.
- **department**: A reference to the department ([Department API](#), on page 53), including the name and refURL. See [References](#), on page 6.
- **dialedNumber**: A reference to an internal dialed number ([Dialed Number API](#), on page 59) for the agent team, including the refURL and dialed number string. See [References](#), on page 6.

- **agents:** A collection of agent ([Agent API, on page 19](#)) references, including the refURL, first name, last name, user name, and agent ID for each agent on the team. See [References, on page 6](#).
- **agentCount:** Read-only field. Number of agents on the team.
- **supervisors:** A collection of supervisor ([Agent API, on page 19](#)) references, including the refURL, first name, last name, user name, and agent ID for each supervisor who supervises this team. See [References, on page 6](#).
- **supervisorCount:** Read-only field. Number of supervisors who supervise this team.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> </ul> |

See [Search, on page 11](#) and [Sort, on page 12](#).

### Example Get Response

```
<agentTeam>
  <refURL>/unifiedconfig/config/agentteam/ (id) </refURL>
  <name>team1</name>
  <dialNumber>
    <refURL>/unifiedconfig/config/dialednumber/ (id) </refURL>
    <dialNumberString>8885551212</dialNumberString>
  </dialNumber>
  <description>test agent team1</description>
  <agents>
    <agent>
      <refURL>/unifiedconfig/config/agent/ (id_1) </refURL>
      <firstName>John</firstName>
      <lastName>Smith</lastName>
      <userName>username</userName>
      <agentId>8006</agentId>
    </agent>
    <agent>
      <refURL>/unifiedconfig/config/agent/ (id_2) </refURL>
      <firstName>Jane</firstName>
      <lastName>Doe</lastName>
      <userName>username</userName>
      <agentId>8007</agentId>
    </agent>
  </agents>

  <supervisor>
    <supervisor>
      <refURL>/unifiedconfig/config/agent/ (id_3) </refURL>
      <firstName>Mary</firstName>
      <lastName>Hart</lastName>
      <userName>username</userName>
      <agentId>8008</agentId>
    </supervisor>
    <supervisor>
      <refURL>/unifiedconfig/config/agent/ (id_4) </refURL>
      <firstName>Jack</firstName>
      <lastName>Jones</lastName>
    </supervisor>
  </supervisor>
</agentTeam>
```

```
        <userName>username</userName>  
        <agentId>8009</agentId>  
    </supervisor>  
</supervisors>  
    <changeStamp>0</changeStamp>  
</agentTeam>
```







## Attribute API

Attributes identify a call routing requirement, such as language, location, or agent expertise. You can create two types of attributes: boolean or proficiency. For example, you can create a Boston attribute that specifies that the agent assigned to this attribute must be located in Boston. Then, if a precision queue requires an agent who lives in Boston, then an agent with the attributes Boston = True is a good match. When you create a proficiency attribute, you assign a proficiency level to the agent.

Use the Attribute API to list the attributes currently defined in the database, define new attributes, and view, edit, and delete existing attributes.

### URL

`https://<server>/unifiedconfig/config/attribute`

### Operations

- **create**: Creates an attribute.
- **delete**: Marks one attribute and associated Agent attribute values for deletion, but does not permanently delete them.
- **get**: Returns one attribute, using the URL  
`https://<server>/unifiedconfig/config/attribute/<id>`.
- **list**: Retrieves a list of attributes.

#### ◦ Query parameters:

- **selectedAgentCount**: Use this query parameter to augment attribute information about multiple agents. The selectedAgentCount parameter shows the number of specified agents associated with this attribute. For example, to find out how many of agents 5000, 5001, 5002, and 5003 in the list have this associated attribute, add  
`selectedAgentCount=5000,5001,5002,5003`.



#### Note

Using selectedAgentCount automatically sets the summary list query parameter to **true**.

- Summary list: See [list](#), on page 4.

- [update](#): Updates one attribute.
  - **Query Parameters:**
    - `removeNonMatchingDepartmentalRefs`: Use this query parameter to remove all agent attributes from an attribute when they no longer belong to the department id specified in the query parameter or the global department. For example, to remove all agent attributes that do not belong to department 5000 or the global department, add `removeNonMatchingDepartmentalRefs=5000`. If this parameter is not specified, the agent attributes must belong to the attribute's department or the global department.

## Parameters

- `refURL`: The refURL of the attribute. See [Shared Parameters, on page 9](#).
- `name`: The name of the attribute. See [Shared Parameters, on page 9](#).
- `changeStamp`: See [Shared Parameters, on page 9](#).
- `description`: See [Shared Parameters, on page 9](#).
- `department`: A reference to the department ([Department API, on page 53](#)), including refURL and name. See [References, on page 6](#).
- `dataType`: The data type of the attribute. Values are:
  - 3: Boolean.
  - 4: Proficiency.
- `defaultValue`: Used to specify the default value for the attribute when assigned to an agent, if no explicit value is provided. Values are:
  - Boolean: true/false.
  - Proficiency: 1-10.
- `agentAttributes`: A collection of agent attribute references for this attribute, including the description, refURL, and read-only parameters `agentId`, `userName`, `firstName`, and `lastName`. Also includes the `attributeValue` parameter which indicates the value (true/false or 1-10) of the attribute for this agent. See [References, on page 6](#).
- `agentAttributesAdded`: A collection of agent attribute references ([Attribute API, on page 35](#)) to be added to the attribute, including the agent refURL and the `attributeValue` of each agent. If the `attributeValue` is not specified, it is assigned the default value. Agents that already have this attribute are updated with the specified `attributeValue`. This parameter is update only, and cannot be used in conjunction with the `agentAttributes` parameter. This parameter can be used with the `agentAttributesRemoved` parameter. See [References, on page 6](#).
- `agentAttributesRemoved`: A collection of agent attribute references ([Attribute API, on page 35](#)) to be removed from the attribute, including the refURL of each agent. This parameter is update only, and cannot be used in conjunction with the `agentAttributes` parameter. This parameter can be used with the `agentAttributesAdded` parameter. See [References, on page 6](#).
- `agentCount`: Read-only field. Number of agents associated with the attribute.

- selectedAgentCount: Read-only field. Indicates the number of specified agents associated with this attribute. Returned only when using the selectedAgentCount query parameter.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• dataType</li> <li>• defaultValue</li> <li>• description</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

### Example Get Response

```
<attribute>
  <changeStamp>0</changeStamp>
  <refURL>/unifiedconfig/config/attribute/5002</refURL>
  <agentCount>1</agentCount>
  <dataType>4</dataType>
  <defaultValue>1</defaultValue>
  <description>test</description>
  <name>AttributeA</name>
  <agentAttributes>
    <agentAttribute>
      <attributeValue>5</attributeValue>
      <description>test</description>
      <agent>
        <refURL>/unifiedconfig/config/agent/8467</refURL>
        <agentId>123</agentId>
        <firstName>Jane</firstName>
        <lastName>Smith</lastName>
        <userName>jsmith</userName>
      </agent>
    </agentAttribute>
  </agentAttributes>
</attribute>
```





## Bucket Interval API

Configure bucket intervals to report how many calls are handled or abandoned during specific, incremental time slots. Each bucket interval has a maximum of nine configurable time slots, called Upper Bounds. Upper Bounds are ranges measured in seconds to segment and capture call-handling activity. You can run reports that show calls answered and calls abandoned for these intervals.

Use the Bucket Intervals API to add new bucket intervals, edit the name of an existing bucket interval, get a list of all of the configured bucket intervals, and delete existing bucket intervals.

### URL

`https://<server>/unifiedconfig/config/bucketinterval`

### Operations

- **create**: Creates one bucket interval.
- **delete**: Deletes one bucket interval from the database.
- **get**: Returns one bucket interval, using the URL  
`https://<server>/unifiedconfig/config/bucketinterval/<id>`.
- **list**: Retrieves a list of bucket intervals.
- **update**: Updates the name of one bucket interval.

### Parameters

- **refURL**: The refURL of the bucket interval. See [Shared Parameters, on page 9](#).
- **name**: The name of the bucket interval. See [Shared Parameters, on page 9](#).
- **changeStamp**: See [Shared Parameters, on page 9](#).
- **department**: A reference to the department ([Department API, on page 53](#)), including the refURL and name. See [References, on page 6](#).
- **upperBound1**: Required. The first Bucket Interval value, in seconds. Must be greater than 0. This parameter cannot be updated.

- upperBound2 to upperBound 9: Optional. The next Bucket Interval values, in seconds. Each must be greater than the previous upperBound field or be left blank (if blank, all remaining upperBound fields must also be blank). These parameters cannot be updated.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters  | Sort parameters  |
|--|--|
| <ul style="list-style-type: none"> <li>• name</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• upperBound 1-9</li> </ul> |

See [Search, on page 11](#) and [Sort, on page 12](#).

### Example Get Response

```
<bucketInterval>
  <refURL>/unified/config/bucketInterval/(id)</refURL>
  <name>test</name>
  <department>
    <name>sales</name>
    <refURL>/unifiedconfig/config/department/5003</refURL>
  </department>
  <upperBound1>10</upperBound1>
  <upperBound2>20</upperBound2>
  <upperBound3>30</upperBound3>
  <upperBound4>40</upperBound4>
  <upperBound5>50</upperBound5>
  <upperBound6>60</upperBound6>
  <upperBound7>70</upperBound7>
  <upperBound8>80</upperBound8>
  <upperBound9>90</upperBound9>
  <changeStamp>0</changeStamp>
</bucketInterval>
```



## Bulk Job API

Bulk jobs are a fast and efficient way to enter data at initial setup and to incorporate large-scale changes, such as changing agent skill groups between shifts and hiring multiple new agents.

You can use the Bulk Job API to list the bulk jobs currently defined in the database, define new bulk jobs, and view or delete records of existing bulk jobs.

### URL

`https://<server>/unifiedconfig/config/bulkjob`

### Operations

- **create**: Creates one bulk job.
- **delete**: Permanently deletes one bulk job.
- **get**: Returns one bulk job using the URL  
`https://<server>/unifiedconfig/config/bulkjob/<id>`.
- **get (template)**: Returns a sample CSV template for the given bulk job type. The response contains the CSV template as a file attachment. The form of the request is  
`https://<server>/unifiedconfig/config/bulkjob/templates/<jobType>`,  
where `<jobType>` can be one of the following:
  - `dialednumber`
  - `agent`
  - `calltype`
  - `skillgroup`
  - `inventory` (for lab mode inventory)
  - `ssomigration`

This template returns records for agent and supervisor accounts that are not enabled for single sign-on. You can use the template to migrate these user accounts to SSO.

Non-SSO users can be filtered by `peripheralId`, `agentTeamId`, or both.

To filter by `peripheralId`, use the following URL:

```
https://<server>/unifiedconfig/config/bulkjob/templates/ssomigration?q=peripheralId:<peripheralId>.
```

To filter by `agentTeamId`, use the following URL:

```
https://<server>/unifiedconfig/config/bulkjob/templates/ssomigration?q=agentTeamId:<agentTeamId>.
```

To filter by both `peripheralId` and `agentTeamId`, use the following URL:

```
https://<server>/unifiedconfig/config/bulkjob/templates/ssomigration?q=peripheralId:<peripheralId>agentTeamId:<agentTeamId>.
```

- `list`: Retrieves a list of bulk jobs.

## Parameters

- `refURL`: The refURL of the bulk job. See [Shared Parameters, on page 9](#).
- `description`: See [Shared Parameters, on page 9](#).
- `fileContent`: The content of the bulk CSV file. The size of the file must not exceed 3MB. For information about the CSV file data, see the "Manage Bulk Jobs" section of the *Cisco Packaged Contact Center Enterprise Administration and Configuration Guide* at <http://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-maintenance-guides-list.html>.
- `createDateTime`: The time the bulk job was submitted. It indicates the time in milliseconds elapsed from the zero epoch value of January 1, 1970, 00:00:00 GMT. Read-only.
- `jobHostName`: The Windows computer name of the AW that initiated the bulk job. Read-only.
- `startDateTime`: The time the bulk job began executing. Read-only.
- `endDateTime`: The time the bulk job completed or failed. Read-only.
- `jobState`: The current state of the job. Read-only.
  - 1: Queued
  - 2: Processing
  - 3: Succeeded
  - 4: Failed
  - 5: Cancelled
  - 6: Partially succeeded
- `jobType`: The job type. Optional. If this parameter is not provided, the value is determined automatically based on the header in the bulk job file. If the parameter is provided, the header must match the job type specified.
  - 1: Dialed Number
  - 2: Agent
  - 3: Call Type



- 4: Skill Group
  - 5: SSO Migration
  - 102: Inventory
- logFile: A URL to download the log file for the bulk job. Read-only.
  - csvFile: A URL to download the CSV file that was originally uploaded in the fileContent parameter. Read-only.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• description</li> <li>• jobType</li> <li>• jobState</li> <li>• jobHostName</li> <li>• createDateTime</li> <li>• startDateTime</li> <li>• endDateTime</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

### Example Get Response

Example response for a dialed number create job that completed successfully:

```
<bulkJob>
  <changeStamp>0</changeStamp>
  <refURL>/unifiedconfig/config/bulkjob/(id)</refURL>
  <jobHostName>CCE-AW-1</jobHostName>
  <createDateTime>1330441858360</createDateTime>
  <startDateTime>1330441858361</startDateTime>
  <endDateTime>1330441858368</endDateTime>
  <jobState>3</jobState>
  <jobType>1</jobType>
  <description>dn create bulk job</description>
  <logFile><refURL>/unifiedconfig/config/bulkjob/(id)/log</refURL></logFile>
  <csvFile><refURL>/unifiedconfig/config/bulkjob/(id)/csv</refURL></csvFile>
</bulkJob>
```





# Call Type API

Call types categorize calls. Based on call type, the system maps a dialed number (see [Dialed Number API, on page 59](#)) to a routing script that ultimately sends the call to the appropriate destination.

Use the Call Type API to list the call types currently defined in the database, define new call types, and view, edit, or delete records of existing call types.

## URL

`https://<server>/unifiedconfig/config/calltype`

## Operations

- [create](#): Creates one call type.
- [delete](#): Marks one call type for deletion, but does not permanently delete it.
- [get](#): Returns one call type, using the URL  
`https://<server>/unifiedconfig/config/calltype/<id>`.
- [list](#): Retrieves a list of call types.
- [update](#): Updates one call type.

## Parameters

- `refURL`: The refURL of the call type. See [Shared Parameters, on page 9](#).
- `name`: The name of the call type. See [Shared Parameters, on page 9](#).
- `changeStamp`: See [Shared Parameters, on page 9](#).
- `description`: See [Shared Parameters, on page 9](#).
- `department`: A reference to the department ([Department API, on page 53](#)), including the refURL and name. See [References, on page 6](#).
- `id`: The database id of the call type. Read-only field. Used in scripting.
- `serviceLevelThreshold`: Maximum time in seconds that a caller should wait before being connected with an agent. Leave blank to use the system default (set in the [Global API, on page 69](#)).
- `serviceLevelType`: This value indicates how the system calculates the service level.

- blank: Use the system default.
  - 1: Ignore Abandoned Calls.
  - 2: Abandoned Calls have Negative Impact.
  - 3: Abandoned Calls have Positive Impact.
- bucketInterval: A reference to the bucket interval ([Bucket Interval API, on page 39](#)), including the refURL and name.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters  |
|---|--|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> <li>• id</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> <li>• id</li> <li>• serviceLevelThreshold</li> <li>• serviceLevelType</li> </ul> |

See [Search, on page 11](#) and [Sort, on page 12](#).

### Example Get Response

```
<callType>
  <refURL>/unifiedconfig/config/calltype/(id)</refURL>
  <name>test</name>
  <description>test call type</description>
  <id>5002</id>
  <serviceLevelThreshold>10</serviceLevelThreshold>
  <serviceLevelType>1</serviceLevelType>
  <changeStamp>0</changeStamp>
  <bucketInterval>
    <refURL>/unifiedconfig/config/bucketinterval/(id)</refURL>
    <name>bucket1</name>
  </bucketInterval>
</callType>
```



## Congestion Control API

Congestion control parameters determine how calls are treated by the system when too many calls are received at one time. Use the Congestion Control API to list the current congestion control parameters in the database.

### URL

`https://<server>/unifiedconfig/config/congestioncontrol`

### Operations

- [get](#): Returns the congestion control parameters, using the URL `https://<server>/unifiedconfig/config/congestioncontrol`.

### Parameters

- `deploymentType`: The type of deployment. See [Deployment Type Info API](#), on page 57.
- `congestionEnabled`: Indicates if congestion control is enabled. Value is true/false.
- `congestionTreatmentMode`: Mode to handle congestion. Values are:
  - 1: Dialed Number default label is used for call treatment.
  - 2: Treat call with Routing client default label.
  - 3: Treat call with System default label.
  - 4: Terminate with Dialog Fail/RouteEnd.
  - 5: Release message to the Routing client.
- `systemDefaultLabel`: Default label string to treat the calls subjected to congestion control. Only used if `congestionTreatmentMode` is set to 3 (Treat call with System default label).
- `cpsCapacity`: The maximum number of calls per second allowed.
- `cpsCapacityDefault`: The default value for the `cpsCapacity` parameter for the current deployment type. Read-only.

### Example Get Response

```
<congestionControl>
  <deploymentType>0</deploymentType>
  <congestionTreatmentMode>1</congestionTreatmentMode>
  <congestionEnabled>true</congestionEnabled>
  <systemDefaultLabel></systemDefaultLabel>
  <cpsCapacity>100</cpsCapacity>
  <cpsCapacityDefault>150</cpsCapacityDefault>
</congestionControl>
```



## Context Service Configuration API

Use the Context Service Configuration API to configure Context Service settings for SocialMiner.

### URL

`https://<server>/unifiedconfig/config/contextservice`

### Operations

- **get**: Gets the Context Service configuration data, using the URL `https://<server>/unifiedconfig/config/contextservice`.
- **update**: Updates the Context Service configuration data.

### Parameters

- **proxyUrl**: URL of the proxy server. The default is an empty string. Optional.
- **maxRetries**: The maximum number of times that the system attempts the operation. This parameter must be set to 1.
- **timeout**: The amount of time, in milliseconds, that the system waits before abandoning the attempt to perform the operation. The default is 1200 ms. The values are 200 - 15000 ms. Optional.
- **labMode**: Whether the Context Service is in lab mode. The default is false. Optional.

### Example Get Response

```
<contextServiceConfig>
  <proxyUrl>http://example.com:80</proxyUrl>
  <maxRetries>1</maxRetries>
  <timeout>10000</timeout>
  <labMode>false</labMode>
</contextServiceConfig>
```







## Context Service Registration API

Cisco Context Service is a cloud-based omnichannel solution for Cisco Contact Center Enterprise Solutions. It enables you to capture your customer's interaction history by providing flexible storage of customer-interaction data across any channel.

Various components in the CCE Solution provide out of the box integration with Context Service. Context Service also provides an API for integration with your own applications or third-party applications to capture end-to-end customer-interaction data.

For more information about Context Service and to check service availability, see <http://cisco.com/go/contextservice>.

Use the Context Service Registration API to register SocialMiner with Context Service and get the current registration status.

### URL

`https://<server>/unifiedconfig/config/contextserviceregistration`

### Operations

- **get**: Returns the current Context Service registration state using the URL `https://<server>/unifiedconfig/config/contextserviceregistration?callbackUrl=(url)`.
- **update**: Updates Context Service connection data in SocialMiner and the CCE database.

### Parameters

- **callbackUrl**: This parameter is optional in the Get URL. The browser is redirected to this URL to complete the registration or deregistration process.
- **state**: The Context Service registration state. Values are as follows:
  - **notRegistered**
  - **registered**
  - **invalid**: The connection data used to register with Context Service is no longer valid. Reregister to recover from this state.

- **managementUrl**: Navigate to this URL to manage the Context Service. This parameter is returned if the state is registered.
- **registrationUrl**: Navigate to this URL to begin Context Service registration. This parameter is returned if you include the **callbackUrl** in the Get URL and the state is either **notRegistered** or **invalid**.
- **deregistrationUrl**: Navigate to this URL to begin Context Service deregistration. This parameter is returned if you include the **callbackUrl** in the Get URL and the state is registered.
- **refURL**: The refURL. See [Shared Parameters](#), on page 9.
- **connectionData**: This parameter is required and the maximum size is 4 KB. **connectionData** is an opaque string used to connect to the Context Service. Use an empty string to clear the **connectionData** when deregistering.

**Note**

You must deregister before reregistering to Context Service. If the current **connectionData** is not empty, you can set the value either to an empty string or to the current value.

**Sample API Workflow**

- 1 The user navigates to a custom Context Service web page.
- 2 The page calls the Get API to get the current registration status from Context Service. The page displays either a Register or Deregister button.
- 3 The user clicks Register or Deregister.
- 4 The page calls the Get API with a callback URL for a custom web page. This page can be the custom Context Service web page from step 1. The API returns either the Context Service registration or deregistration URL.
- 5 The browser redirects to the Context Service registration or deregistration URL.
- 6 The user completes the registration or deregistration wizard.
- 7 The browser redirects back to the callback URL with a **connectionData** query parameter appended to it.
- 8 The callback page calls the Update API with the connection data from step 7 to send the data to SocialMiner and the CCE database.
- 9 Go to step 2.

**Example Get Response**

```
<contextServiceRegistration>
  <state>registered</state>
  <managementUrl>(url)</managementUrl> <!-- Included if state is registered -->
  <registrationUrl>(url)</registrationUrl> <!-- Included if state is notRegistered
    and callbackUrl is provided -->
  <deregistrationUrl>(url)</deregistrationUrl> <!-- Included if state is registered
    and callbackUrl is provided -->
  <refURL>/unifiedconfig/config/contextservice</refURL>
</contextServiceRegistration>
```

**Example Update Request**

```
<contextServiceRegistration>
  <connectionData>(data)</connectionData>
</contextServiceRegistration>
```



## Department API

Packaged CCE allows you to create departments, add configuration items to departments, and assign administrators to departments to limit the scope of their control. For example, the call center for a hospital might have departments for Radiology, Surgery, and Cardiology. Use of departments is optional.

For more information on how departments work, see the *Cisco Packaged Contact Center Enterprise Administration and Configuration Guide* at <http://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-maintenance-guides-list.html>.

Use the Department API to list the departments currently defined in the database, define new departments, and view, edit, and delete existing departments.

### URL

`https://<server>/unifiedconfig/config/department`

### Operations

- **create**: Creates one department.
- **delete**: Marks one department for deletion.
- **get**: Returns one department, using the URL  
`https://<server>/unifiedconfig/config/department/<id>`.
- **list**: Retrieves a list of departments.
- **update**: Updates one department.

### Parameters

- **refURL**: The refURL of the department. See [Shared Parameters, on page 9](#).
- **name**: The name of this department. See [Shared Parameters, on page 9](#).
- **changeStamp**: See [Shared Parameters, on page 9](#).
- **description**: See [Shared Parameters, on page 9](#).
- **administrators**: A collection of administrator ([Administrator API, on page 17](#)) references associated with this department, including the refURL, user name, and domain name. See [References, on page 6](#).

## Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

## Example Get Response

```
<department>
  <changeStamp>0</changeStamp>
  <refURL>/unifiedconfig/config/department/(id)</refURL>
  <name>department1</name>
  <description>test department1</description>
  <administrators>
    <administrator>
      <refURL>/unifiedconfig/config/administrator/(id_1)</refURL>
      <userName>JohnSmith</userName>
      <domainName>BOSTON.COM</domainName>
    </administrator>
    <administrator>
      <refURL>/unifiedconfig/config/administrator/(id_2)</refURL>
      <userName>JaneDoe</userName>
      <domainName>BOSTON.COM</domainName>
    </administrator>
  </administrators>
</department>
```



## Deployment API

---

The Deployment API is used to view the deployment type of the installation. It is read-only, and does not require authentication. To change the deployment type, use the Deployment Type Info API.

### URL

`https://<server>/unifiedconfig/config/deployment`

### Parameters

- `deploymentType`: The type of deployment. See [Deployment Type Info API](#), on page 57.
- `supervisorLoginAllowed`: Indicates whether the current deployment type allows supervisor login.

### Operations

- `get`: Returns the deployment type of the installation using the URL  
`https://<server>/unifiedconfig/config/deployment`.

### Example Get Response

```
<deployment>
  <deploymentType>7</deploymentType>
  <supervisorLoginAllowed>true</supervisorLoginAllowed>
</deployment>
```





## Deployment Type Info API

Use the Deployment Type Info API to view or edit the current system deployment type.

### URL

`https://<server>/unifiedconfig/config/deploymenttypeinfo`

### Operations

- **get**: Returns the current deployment type and the results of the capacity and system validation tests, using the URL `https://<server>/unifiedconfig/config/deploymenttypeinfo`.
- **update**: Sets the specified deployment type if the system validation check, capacity check, and VM Validation for that deployment type pass and are required.

### Parameters

- **changeStamp**: See [Shared Parameters](#), on page 9.
- **vmHosts**: vmHost information, including name, address, username, and password parameters of Side A and Side B. Only required when switching to Packaged CCE, to allow access to the ESX servers for VM validation.
- **permissionInfo**: See [Permissions](#), on page 10.
- **systemValidationStatus**: See [Serviceability API](#), on page 137.
- **capacityInfo**: See [Serviceability API](#), on page 137.
- **vmLayoutType**: Represents the current deployment type and hardware.
- **vmValidationLogURL**: The URL to download a file about VM layout validation.
- **deploymentType**: The type of deployment. The following types are supported:
  - 0: No deployment type specified. Initial type set at installation. Once set to another deployment type, you cannot switch back to 0.
  - 1: NAM (Deprecated)
  - 2: Contact Director
  - 3: NAM Rogger (Deprecated)

- 4: ICM Router/Logger
- 5: UCCE: 8000 Agents Router/Logger
- 6: UCCE: 12000 Agents Router/Logger
- 7: Packaged CCE: 2000 Agents
- 8: ICM Rogger
- 9: UCCE: 4000 Agents Rogger
- 10: Packaged CCE: Lab Mode
- 11: HCS-CC: 2000 Agents
- 12: HCS-CC: 500 Agents (Deprecated)
- 13: UCCE: Progger (Lab Only)
- 14: HCS-CC: 4000 Agents
- 15: HCS-CC: 12000 Agents

**Note**

You can only use Live Data on deployment types UCCE: 8000 Agents Router/Logger, UCCE: 12000 Agents Router/Logger, Packaged CCE: 2000 Agents, UCCE: 4000 Agents Rogger, and Packaged CCE: Lab Mode.

- targetDeploymentType: Indicates which deployment type is being initialized.

**Example Get Response**

```
<deploymentTypeInfo>
  <changeStamp>59</changeStamp>
  <deploymentType>7</deploymentType>
  <vmLayoutType>PCCE_C240M3_Full</vmLayoutType>
  <vmHosts>
    <vmHost>
      <name>sideA</name>
      <address>10.86.141.10</address>
      <userName>root</userName>
    </vmHost>
    <vmHost>
      <name>sideB</name>
      <address>10.86.141.29</address>
      <userName>root</userName>
      <password>pwexample</password>
    </vmHost>
  </vmHosts>
</deploymentTypeInfo>
```





## Dialed Number API

Dialed numbers are string values used to select the appropriate routing script so that a voice call or a non-voice task (such as an email or a request for a web chat) can be delivered to an agent.

Use the Dialed Number API to list the dialed numbers currently defined in the database, define new dialed numbers, and view, edit, and delete existing dialed numbers.

### URL

`https://<server>/unifiedconfig/config/dialednumber`

### Operations

- **create**: Creates one dialed number.
- **delete**: Marks one dialed number for deletion.
- **get**: Returns one dialed number, using the URL  
`https://<server>/unifiedconfig/config/dialednumber/<id>`.
- **list**: Retrieves a list of dialed numbers.
  - **Query parameters**:
    - Summary list: See [list](#), on page 4.
- **update**: Updates one dialed number.

### Parameters

- **dialedNumberString**: Required. Value used to route the call or direct the non-voice task. A unique string for the routing type. Maximum of 25 characters.
- **changeStamp**: See [Shared Parameters](#), on page 9.
- **description**: See [Shared Parameters](#), on page 9.
- **department**: A reference to the department ([Department API](#), on page 53), including the refURL and name. See [References](#), on page 6.
- **routingType**: Specifies where a call or non-voice task request originates.

Refer to the [Routing Type API, on page 133](#) to determine the routing types on which you can create a dialed number.

- 1: External Voice. Calls come from Unified CVP. When creating a Dialed Number using this type, a dialed number database record is created for each Unified CVP routing client.
  - 2: Internal Voice. Calls come from a Unified CM phone.
  - 3: Outbound. Calls that come from the Outbound Option Dialer.
  - 4: Multichannel 1. Requests that come from an Enterprise Chat and Email, SocialMiner, or third party.
  - 5: Multichannel 2. Requests that come from an Enterprise Chat and Email, SocialMiner, or third party.
  - 6: Multichannel 3. Requests that come from an Enterprise Chat and Email, SocialMiner, or third party.
- **callType:** A reference to a call type ([Call Type API, on page 45](#)) for this dialed number, including a refURL and name. See [References, on page 6](#).
  - **dialedNumberRecords:** A collection of dialed number record entries each containing the id and name of a dialed number database record. Read-only.
  - **mediaRoutingDomain:** A reference to the media routing domain ([Media Routing Domain API, on page 89](#)) for the dialed number. See [References, on page 6](#).

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• dialedNumberString</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• dialedNumberString (default)</li> <li>• description</li> </ul> |

See [Search, on page 11](#) and [Sort, on page 12](#).

### Advanced search parameters

The Dialed Number API also supports advanced search parameters, such as routing type ([Routing Type API, on page 133](#)) and data center ([Data Center API](#)).

- **routingType:<type>** Finds all dialed numbers with the specified routing type value. Valid types match those in the routingType parameter.
  - **routingType:1** Returns all dialed numbers with an external voice routing type.
- 

### Example Get Response

```
<dialedNumber>
  <refURL>/unifiedconfig/config/dialedNumber/(id)</refURL>
```

```
<description>test dialed number</description>
<dialedNumberString>8885551212</dialedNumberString>
<routingType>1</routingType>
<changeStamp>0</changeStamp>
<mediaRoutingDomain>
  <refURL>/unifiedconfig/config/mediaroutingdomain/1</refURL>
  <name>Cisco_Voice</name>
</mediaRoutingDomain>

  <callType>
    <refURL>/unifiedconfig/config/calltype/{id}</refURL>
    <name>calltype name</name>
  </callType>
<dialedNumberRecords>
  <dialedNumberRecord>
    <id>10</id>
    <name>cvplrc.8885551212</name>
  </dialedNumberRecord>
  <dialedNumberRecord>
    <id>11</id>
    <name>cvp2rc.8885551212</name>
  </dialedNumberRecord>
</dialedNumberRecords>
</dialedNumber>
```





## Expanded Call Variable API

Calls carry data with them as they move through the system. This data, called expanded call variable data, is embedded with the call and is visible on the agent desktop.

Use the Expanded Call Variable API to list the expanded call variables currently defined in the database, define new expanded call variables, and view, edit, and delete existing expanded call variables.

### URL

`https://<server>/unifiedconfig/config/expandedcallvariable`

### Operations

- **create**: Creates one expanded call variable.
- **delete**: Marks one expanded call variable for deletion, but does not permanently delete it.
- **get**: Returns one expanded call variable, using the URL  
`https://<server>/unifiedconfig/config/expandedcallvariable/<id>`.
- **list**: Retrieves a list of expanded call variables.
- **update**: Updates one expanded call variable.

### Parameters

- **refURL**: The refURL of the expanded call variable. See [Shared Parameters, on page 9](#).
- **name**: The name of the expanded call variable. See [Shared Parameters, on page 9](#).
- **changeStamp**: See [Shared Parameters, on page 9](#).
- **description**: See [Shared Parameters, on page 9](#).
- **maxLength**: The maximum length of the expanded call variable. The value is 1 to 210.
- **eccArray**: Indicates whether the expanded call variable is an array. Values are true/false.
- **maximumArraySize**: The maximum number of elements in the array is 1 to 255. Required if eccArray is true; must be blank or not specified if eccArray is false.
- **enabled**: Indicates whether the expanded call variable is enabled. Values are true/false.

- The total bytesRequired of all enabled expanded call variables cannot exceed 2000 bytes.
- The total bytesRequiredInCtiServer of all enabled expanded call variables cannot exceed 2500 bytes.
- persistent: Specifies whether the expanded call variable is written to the historical database with each Termination Call Detail and Route Call Detail record. Values are true/false.
  - No persistent, enabled arrays are allowed.
  - The maximum number of persistent, scalar, enabled variables is 20.
- ciscoProvided: Indicates whether the expanded call variable is provided by Cisco. Values are true/false. Read-only.
- bytesRequired: The number of bytes required to store the expanded call variable in the system. Read-only.  
The size is calculated using the following formula:
  - If eccArray is false, the size is 5+Maximum Length.
  - If eccArray is true, the size is 5+(1+Maximum length)\*Maximum Array size.
- bytesRequiredInCtiServer: The number of bytes required to send this variable to CTI Server. Read-only.  
The size is calculated using the following formula:
  - If eccArray is false, the size is Length of name+Maximum length+4.
  - If eccArray is true, the size is (Length of name+Maximum length+5)\*Maximum array size.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters  |
|---|--|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> <li>• maximumLength</li> <li>• maximumArraySize</li> <li>• eccArray</li> <li>• enabled</li> <li>• persistent</li> <li>• ciscoProvided</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

### Example Get Response

```
<expandedCallVariable>
  <refURL>/unifiedconfig/config/expandedcallvariable/(id)</refURL>
  <name>test</name>
  <maxLength>9</maxLength>
  <maximumArraySize>10</maximumArraySize>
  <eccArray>true</eccArray>
  <enabled>true</enabled>
  <ciscoProvided>false</ciscoProvided>
  <description>test expanded call variable</description>
  <persistent>false</persistent>
  <changeStamp>0</changeStamp>
  <bytesRequired>105</bytesRequired>
  <bytesRequiredInCtiServer>180</bytesRequiredInCtiServer>
</expandedCallVariable>
```







## Facility API

Use the Facility API to retrieve a list of every facility, including associated instances, from the Active Directory domain. This API is read-only.

### URL

`https://<server>/unifiedconfig/config/facility`

### Operations

- [list](#): Returns a list of facilities.

### Parameters

- `facilities`: A collection of facility items, including a list of each facility's instances. Includes a name parameter.

### Example Get Response

```
<results>
  <facilities>
    <facility>
      <instances>
        <instance>
          <name>bos01</name>
        </instance>
        <instance>
          <name>pra01</name>
        </instance>
        <instance>
          <name>bos02</name>
        </instance>
      </instances>
      <name>bos</name>
    </facility>
    <facility>
      <instances>
        <instance>
          <name>test</name>
        </instance>
      </instances>
      <name>Cisco_test_fac</name>
    </facility>
  </facilities>
</results>
```





# Global API

---

The Global API returns global settings from the following categories:

- Call
- Agent
- Reporting
- Script
- Labels

## URL

`https://<server>/unifiedconfig/config/globalsetting`

## Operations

- [list](#): Retrieves a list of global settings.
- [update](#): Updates global settings.

## Parameters

### callReporting

- `defaultBucketInterval`: Required. A reference to a bucket interval ([Bucket Interval API, on page 39](#)), including refURL and name. See [References, on page 6](#).
- `defaultCallType`: Required. A reference to a call type ([Call Type API, on page 45](#)), including refURL and name. A call is categorized against this call type unless it comes into the system on a dialed number that is associated with another call type. See [References, on page 6](#).
- `serviceLevelType`: Required. This value indicates how the system calculates the service level.
  - 1: Ignore Abandoned Calls.
  - 2: Abandoned Calls have Negative Impact.
  - 3: Abandoned Calls have Positive Impact.

- **serviceLevelThreshold**: Required. Maximum time in seconds that a caller should wait before being connected with an agent. Maximum is 86,400 seconds (1 day).
- **abandonCallWaitTime**: Required. Configures the minimum time an incoming call must be queued before the call is considered abandoned if the caller hangs up. Maximum is 14400 seconds (4 hours).
- **answeredShortCallThreshold**: Configures the maximum duration for a short call. Calls with a duration below that value are considered short. Value is between 0 and 14400 seconds (4 hours).

### agent

- **agentPhoneLineControl**: Indicates whether all agents supported on the agent peripheral can have one or more than one line configured.
  - 0: Single Line.
  - 1: All Lines.
- **nonACDLineImpact**: Specifies how the agent state is set when the agent is on a call on a secondary line and **agentPhoneLineControl** is set to All Lines.
  - 0: Available agent stays available.
  - 1: Available agent goes not ready.
- **defaultDeskSetting**: A reference to a desk setting ([Agent Desk Settings API, on page 25](#)), including refURL and name.
- **loginNameCaseSensitivity**: Identifies whether usernames are case-sensitive. Values are true/false.
- **minimumPasswordLength**: Changing this value affects new passwords only and does not apply to existing ones. Value is between 0 and 32.

### reporting

- **reportingInterval**: Configures the system to store historical information in 15-minute or 30-minute summaries. The 15-minute interval requires a larger amount of database space than the 30-minute interval. Values are 15 or 30.

### script

- **retainScriptVersion**: Defines the maximum number of versions of each routing script to maintain in the database. The system automatically deletes the oldest version when the limit is exceeded. Maximum is 100.

### labels

- **cmLabel**: Pattern that matches the Unified CM route pattern. Must be a 10-digit string.
- **cvpLabels**: A collection of labels that include the pattern and routingClientName. The patterns should match the CVP Dialed Number patterns. Must be a 10-digit string.
- **outboundLabel**: Pattern that matches IOS Voice Gateway dial-peer. Must be a 10-digit string.

### Example Get Request

```
<globalSettings>
  <callReporting>
```

```

    <serviceLevelType/>
    <serviceLevelThreshold/>
    <abandonCallWaitTime/>
    <answeredShortCallThreshold/>
    <defaultCallType>
      <refURL/>
      <name/>
    </defaultCallType>
    <defaultBucketInterval>
      <refURL/>
      <name/>
    </defaultBucketInterval>
  </callReporting>
  <agent>
    <agentPhoneLineControl/>
    <nonACDLineImpact/>
    <defaultDeskSetting>
      <refURL/>
      <name/>
    <defaultDeskSetting/>
    <loginNameCaseSensitivity/>
    <minimumPasswordLength/>
  </agent>
  <reporting>
    <reportingInterval/>
  </reporting>
  <script>
    <retainScriptVersion/>
  </script>
</globalSettings>
<globalSettings>
  <callReporting>
    <abandonCallWaitTime>5</abandonCallWaitTime>
    <answeredShortCallThreshold>0</answeredShortCallThreshold>
    <defaultBucketInterval>
      <refURL>/unifiedconfig/config/bucketinterval/1</refURL>
      <name>BuiltIn</name>
    </defaultBucketInterval>
    <defaultCallType>
      <refURL>/unifiedconfig/config/calltype/1</refURL>
      <name>BuiltIn</name>
    </defaultCallType>
    <serviceLevelThreshold>20</serviceLevelThreshold>
    <serviceLevelType>1</serviceLevelType>
  </callReporting>
  <agent>
    <agentPhoneLineControl>0</agentPhoneLineControl>
    <defaultDeskSetting>
      <refURL>/unifiedconfig/config/agentdesksetting/5000</refURL>
      <name>Default_Agent_Desk_Settings</name>
    </defaultDeskSetting>
    <loginNameCaseSensitivity>false</loginNameCaseSensitivity>
    <minimumPasswordLength>0</minimumPasswordLength>
    <nonACDLineImpact>0</nonACDLineImpact>
  </agent>
  <reporting>
    <reportingInterval>30</reportingInterval>
  </reporting>
  <script>
    <retainScriptVersion>5</retainScriptVersion>
  </script>
  <labels>
    <cmLabel>8881111000</cmLabel>
    <cvpLabels>
      <cvpLabel>
        <pattern> 7777777777 </pattern>
        <routingClientName>CVP_PG_1A</routingClientName>
      </cvpLabel>
      <cvpLabel>
        <pattern> 7777777777</pattern>
        <routingClientName>CVP_PG_1B</routingClientName>
      </cvpLabel>
    </cvpLabels>
  </labels>

```

```
        <pattern>777777777</pattern>
        <routingClientName>CVP_PG_2A</routingClientName>
      </cvpLabel>
      <cvpLabel>
        <pattern> 777777777 </pattern>
        <routingClientName>CVP_PG_2B</routingClientName>
      </cvpLabel>
    </cvpLabels>
    <outboundLabel> 6661111000 </outboundLabel>
  </labels>
</globalSettings>
```

### Example Update Request



## Initialize API

The initialize API serves as an entry point for Packaged CCE system setup. It allows you to:

- Start setup tasks for system configuration.
- Check the initialization status of the system.

### URL

`https://<server>/unifiedconfig/config/initialize`

### Operations

- [list](#): Lists information about the system initialization status.
- [update](#): Starts a system initialization.

### Parameters

- name: The name of the startup task.
- state: The state of the task. Values are:
  - NOT\_STARTED
  - PROCESSING
  - FAILED\_NEEDS\_RETRY: Occurs when an initialization task that does not require uninitialization fails. Correct the errors then invoke the Initialize API again.
  - FAILED: Occurs when a task that requires uninitialization fails. Correct the errors, and then invoke the Uninitialize API. After a successful uninitialization, the initialization request can be made again.
  - SUCCEEDED

### Example Get Response

```
<results>
  <state>FAILED</state>
  <initializationStatuses>
    <initializationStatus>
```

```
<state>NOT_STARTED</state>
<name>JtapiClientTask</name>
</initializationStatus>
<initializationStatus>
  <state>FAILED</state>
  <name>PGUserTask</name>
  <apiErrors>
    <apiError>
      <errorType>errorConnectAXL</errorType>
      <errorMessage>There is no UCM AXL service defined yet</errorMessage>
    </apiError>
  </apiErrors>
</initializationStatus>
</initializationStatuses>
</results>
```





## Instance API

The Instance API is used during a Packaged CCE installation to select the name of the facility and instance for the deployment.

### URL

`https://<server>/unifiedconfig/config/instance`

### Operations

- **create**: Creates one instance. Only one instance can be created.
- **delete**: Deletes one instance.
- **get**: Returns one instance, using the URL `https://<server ip address>/unifiedconfig/config/instance/1`.
- **list**: Retrieves the instance from the database.
- **update**: Updates one instance.



### Note

The user performing the operation must be in the setup group for the instance Organizational Unit that is specified.

### Parameters

- **refURL**: The refURL of the instance. See [Shared Parameters](#), on page 9.
- **facilityName**: Name of an existing facility Organizational Unit in Active Directory. See [Facility API](#), on page 67.
- **instanceName**: Name of an existing instance Organizational Unit in Active Directory. See [Facility API](#), on page 67.

### Example Get Response

```
<instance>
  <refURL>/unifiedconfig/config/instance/1</refURL>
```

```
<facilityName>Lab</facilityName>  
<instanceName>pcce</instanceName>  
</instance>
```



## Internet Script Editor API

---

The Internet Script Editor API indicates whether Internet Script Editor is enabled. If Internet Script Editor is enabled in Web Setup, the API displays the download link in the format `https://<server>/install/iScriptEditor.exe`.

### URL

`/internetscripteditor`

### Operations

- [get](#): Returns whether Internet Script Editor is enabled and a download link.

### Parameters

- `enabled`: Indicates whether Internet Script Editor is enabled in Web Setup. True or false.
- `downloadLink`: The download link for Internet Script Editor. This link appears only when the `enabled` parameter is true.

### Example Get Response

```
<internetScriptEditor>
  <enabled>true</enabled>
  <downloadLink>https://10.10.10.207/install/iScriptEditor.exe</downloadLink>
</internetScriptEditor>
```





## Log Collection API

Use the Log Collection API to collect the log files, or request a download of the log files in one collected zipped file.

### Operations

- **create**: Creates one request to collect the log files, and begins collecting them.
  - Only one log collection may be performed at a time.
  - The maximum number of saved log collections is 3.
  - Inventory status errors must be cleared before log collection starts.
- **delete**: Deletes one log collection.
- **get**: Returns the log collection item, using the URL  
`https://<server>/unifiedconfig/config/logcollection/<id>`.
- **list**: Retrieves a list of collection requests.

### Parameters

- **refURL**: The refURL of the log collection. See [Shared Parameters, on page 9](#).
- **startDateTime**: The start date of the logs collected.
- **endDateTime**: The end date of the logs collected.
- **description**: See [Shared Parameters, on page 9](#).
- **components**: A list of components for which logs are collected. Defaults to all components if the list is blank or not provided. Possible component values include:
  - 1: Unified Contact Center Enterprise (CCE)
  - 2: Unified Customer Voice Portal (CVP)
  - 3: Unified Communications Manager (CM)
  - 4: Cisco Unified Intelligence Center (Intelligence Center)
  - 5: Finesse

- **status.state:** The status of the collection request: IN\_PROGRESS, DONE, or ERROR.
- **status.apiErrors:** The error indicating why the collection request failed. Returned when status.state is ERROR.
- **resultsFile:** Zipped file containing logs collected. Includes the following parameters:
  - **refURL:** The URL of the zipped file which is used for download.
  - **size:** The size of the zipped file (in bytes).

### Example Get Response

```
<logCollection>
  <refURL>/unifiedconfig/config/logcollection/1</refURL>
  <status>
    <state>IN_PROGRESS</state>
  </status>
  <components>
    <component>1</component>
    <component>2</component>
  </components>
  <description>this is a log collection to see if ____</description>
  <startTime>1368564152000</startTime>
  <endTime>1368564156000</endTime>
  <resultsFile>
    <refUrl>/unifiedconfig/config/logcollection/(id)/log</refUrl>
    <size>450</size>
  </resultsFile>
</logCollection>
```



## Machine Inventory API

This API returns the machines in the solution. Machines include VMs, VM Hosts, external machines, and gateways.

For information on how to perform a machine inventory scan, see [Scan API](#), on page 135.

### URL

`https://<server>/unifiedconfig/config/machineinventory`

### Operations

- **create**: Creates a machine by updating the database. See table below for restrictions per machine type. Create is allowed for external machines types only.
- **delete**: Removes one machine.
- **get**: Returns one machine and all associated addresses and services based on machine ID, using the URL `https://<server>/unifiedconfig/config/machineinventory/<id>`.
- **status**: Returns any alerts indicating errors in the state of the inventory, using the URL `https://<server>/unifiedconfig/config/machineinventory/status`.
- **list**: Retrieves a list of all machines in the inventory. See table below for more details.
- **update**: Updates one machine.

| Type       | Create / Update / Delete operations allowed | Number allowed       |
|------------|---|----------------------|
| VM_HOST    | No  | 1 Side A<br>1 Side B |
| CCE_ROGGER | No  | 1 Side A<br>1 Side B |
| CCE_PG     | No  | 1 Side A<br>1 Side B |

| Type                                      | Create / Update / Delete operations allowed | Number allowed   |
|---|---|--|
| CCE_AW                                    | Side A: Update only<br>Side B: No           | 1 Side A<br>1 Side B   |
| CVP                                       | No  | 1 Side A<br>1 Side B   |
| CM  | Update only                                 | 0 - Must be changed after initial scan.                                      |
| CM_PUBLISHER                              | Update only                                 | 1 Side A, 0 Side B for on box CM Deployments<br>0 for off box CM deployments |
| CM_SUBSCRIBER                             | Update only                                 | 1 Side A, 1 Side B for on box CM Deployments<br>0 for off box CM deployments |
| CVP_REPORTING                             | No  | 0 - 1 Side A   |
| CUIC_PUBLISHER                            | Update only                                 | 1 Side A   |
| CUIC_SUBSCRIBER                           | No  | 1 Side B   |
| CVP_OPS                                   | Update only                                 | 1 Side A   |
| FINESSE                                   | Side A: Update only<br>Side B: No           | 1 Side A<br>1 Side B<br>0 - 1 Side B   |
| ECE                                       | No  |  |
| CVVB<br>(Cisco Virtualized Voice Browser) | No  | 0 - 8  |
| GATEWAY                                   | No  | 0 - 8  |
| EXTERNAL_SOCIAL_MINER                     | All   | 0 - 1  |
| EXTERNAL_CM_PUBLISHER                     | All   | 0 for on box CM deployments<br>1 for off box CM Deployments                  |



| Type                              | Create / Update / Delete operations allowed | Number allowed  |
|-----------------------------------|---|---|
| EXTERNAL_CM_SUBSCRIBER            | No  | Not allowed for on box CM Deployments<br><br>A minimum of 2 for off box CM Deployments. External subscribers cannot be created, updated, or deleted as this automatically occurs when the external publisher is created, updated, or deleted. |
| EXTERNAL_CVP_REPORTING            | All   | 0 - 1   |
| EXTERNAL_HDS                      | Update only                                 | 0 - 2   |
| EXTERNAL_MEDIA_SENSE              | All   | 0 - 1   |
| EXTERNAL_ECE                      | All   | 0 - 1   |
| EXTERNAL_THIRD_PARTY_MULTICHANNEL | All   | 0 - 1   |

## Parameters

### Machine parameters:

- refURL: The refURL of the machine. See [Shared Parameters, on page 9](#).
- name: External name of the machine. For example, the VM host name. Valid characters are period (.), hyphen (-), underscore (\_), and alphanumeric. The first character must be alphanumeric. Maximum length is 128 characters.
- changeStamp: See [Shared Parameters, on page 9](#).
- type: The type of machine.
- versionInfo: Version info available depending on product type. Includes the following parameters (by product type):
  - UCCE: version, buildNumber, esNumber, patchVersion
  - CVVB: version and buildNumber
  - CVP: version, buildNumber, buildDate, esNumber, srNumber, dropNumber
  - VOS: version and buildNumber
  - Gateway: version
- vmhost: A reference to a machine of type VM\_HOST, including refURL and name. See [References, on page 6](#).

- autogenerated: Indicates if the information was generated automatically.
- hostName: The host name of the machine.
  - If you provide a hostname, a lookup is performed to find the FQDN for that host. If the FQDN is found, the hostName field is then set to that FQDN value and the Machine Address is set to the IP address corresponding to the hostname.
  - If you do not provide a hostname, a lookup is performed using the address field in the API request. If the FQDN is found, the hostName field is set to that FQDN value.
- networks: A collection of network parameters. See the network parameters below.
- vmInstanceUuid: A unique identifier for the virtual machine.

#### Network parameters:

- type: Public or private. Private networks are only specified for CCE\_PG and CCE\_ROGGER.
- address: The IP address. Must be valid hostname, IPV4, or IPV6 address.
- services: A collection of service parameters. See the services parameters below.

#### Services parameters:

- uri: The service URL, for example, Diagnostic Portal URL or management console link.
- port: The port for this service.
- username: The username used to access the service. Username maximum is 128 characters.
- password: The password used to access the service. The password can be used when creating or updating, but is not returned.
- enablePassword: Used for gateways.
- description: The description of the service. See [Shared Parameters, on page 9](#).
- pairing: Indicates if services on different machines are related. Related services have a matching value.
- type: The service type. Values for type are as follows:
  - ESXI: VM Host running ESXi server for connection to interrogate for VMs.
  - AXL: AXL connection information for Unified CM. Required for UCM Publisher.
  - DIAGNOSTIC\_PORTAL: For CCE and Unified CVP, this is the Diagnostic Portal API. For Finesse and Unified Intelligence Center, this is the SOAP API. Required for CCE\_AW on side A, CVP OAMP, FINESSE on side A, and CUIC Publisher.
  - GATEWAY: Connection information for gateways.
  - MANAGEMENT\_LINK: Provides a URL of the management console for the machine.
  - TIP\_PG: LiveData connection information for peripheral gateways.
  - TIP\_ROUTER: LiveData connection information for the router.
  - ADMINISTRATION: Connection information for administrative access. Required for CVP OAMP.
  - SM\_REST\_API: SocialMiner REST API.

- **CONTEXT\_SERVICE**: Contains the last-written Context Service connection data.
- **IDS**: Indicates an Identity Server service. IDS contains the following parameters:
  - **components**: Information about the component, including the component refURL and component name.
- **IDS\_PRIMARY\_REF**: When present, the <uri> parameter for this service refers to the primary Identity Server (IdS) to which this component is associated.
- **IDS\_SECONDARY\_REF**: Read only. The <uri> parameter for this service refers to the backup IdS to which this component is associated. For a create or update request, this field is automatically filled with the refURL of the secondary IdS (if present).

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> <li>• hostName</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

### Advanced search parameters

You can perform a machine type search or a service type search on the Machine Inventory API:

- **type:<machine type>**: Returns only machines of the specified type. For example, **type:EXTERNAL\_SOCIAL\_MINER** returns only external SocialMiner machines. The machine type is case-insensitive.
- **serviceType:<service type>**: Returns only machines associated with the specified service type. For example, **serviceType:IDS\_PRIMARY\_REF** returns only SSO-capable machines.

### Example Inventory Status Response

`https://<server>/unifiedconfig/config/machineinventory/status`

```
<status>
  <alerts>
    <alert>
      <apiErrors>
        <apiError>
          <errorData>CM_PUBLISHER</errorData>
          <errorMessage>CM_PUBLISHER not found on vmhost sideA</errorMessage>
          <errorType>inventory.MissingMachine</errorType>
        </apiError>
      </apiErrors>
      <machine>
        <host>sideA</host>
        <type>CM_PUBLISHER</type>
      </machine>
    </alert>
```

```

    </alerts>
    <scanInfo>
      <lastScanDateTime>1374842924017</lastScanDateTime>
      <scanState>Idle</scanState>
    </scanInfo>
  </status>

```

### Example Get Response

`https://<server>/unifiedconfig/config/machineinventory/<id>`

```

<machine>
  <changeStamp>4</changeStamp>
  <refURL>/unifiedconfig/config/machineinventory/12073</refURL>
  <networks>
    <network>
      <address>10.10.10.21</address>
      <services>
        <service>
          <autoGenerated>>false</autoGenerated>
          <port>7890</port>
          <type>DIAGNOSTIC_PORTAL</type>
          <userName>user@domain</userName>
        </service>
      </services>
      <type>PUBLIC</type>
    </network>
  </networks>
  <autoGenerated>>false</autoGenerated>
  <hostName>CCE-AW-1-21</hostName>
  <type>CCE_AW</type>
  <versionInfo>
    <buildNumber>03297</buildNumber>
    <version>11.0 (1)</version>
  </versionInfo>
  <name>WJ2-AW-1</name>
  <vmHost>
    <refURL>/unifiedconfig/config/machineinventory/12071</refURL>
    <name>sideA</name>
  </vmHost>
  <vmInstanceUuid>50290e07-fa3d-2667-b84b-1c35989000e2</vmInstanceUuid>
</machine>

```

### Example Updating SSO-Capable Components Associated to an IdS

SSO-capable components can be associated to an IdS with the `<components>` element. For example, a PUT request like this will update the `IDS_PRIMARY_REF` and `IDS_SECONDARY_REF` of any SSO-capable components. Note that refURLs of any on-box PCCE machines will be ignored as these cannot be changed. An example of an off-box PCCE machine is the `EXTERNAL_HDS` machine.

This operation will also de-associate any existing component references to that IdS that aren't in the list. For example, if there were three components associated to this IdS before the operation, only two components would be associated after the operation was complete.

```

<machine>
  <changeStamp>1</changeStamp>
  <refURL>/unifiedconfig/config/machineinventory/5000</refURL>
  <networks>
    <network>
      <type>PUBLIC</type>
      <address>10.10.10.22</address>
      <services>
        <service>
          <type>IDS</type>
          <components>
            <component>
              <refURL>/unifiedconfig/config/machineinventory/6000</refURL>
            </component>
            <component>

```

```
        <refURL>/unifiedconfig/config/machineinventory/6001</refURL>
      </component>
    </components>
  </service>
</services>
</network>
</networks>
</machine>
```





## Media Routing Domain API

A media routing domain is a collection of skill groups associated with a common media class. It is used to organize how requests for different media are routed.

Use the Media Routing Domain (MRD) API to list the MRDs currently defined in the database, define new MRDs, and view, edit, and delete existing MRDs.

The built-in Cisco\_Voice MRD and legacyMultichannel MRDs are read-only; they cannot be created, updated, or deleted. You can perform all API operations on multichannel MRDs.

### URL

`https://<server>/unifiedconfig/config/mediaroutingdomain`

### Operations

- **create**: Creates an MRD.
- **delete**: Permanently deletes one MRD.
- **list**: Retrieves a list of MRDs.
- **get**: Returns one MRD using the URL  
`https://<server>/unifiedconfig/config/mediaroutingdomain/<id>`.
- **update**: Updates one MRD.

### Parameters

- **refURL**: The refURL of the MRD. See [Shared Parameters, on page 9](#).
- **name**: Name of the MRD. See [Shared Parameters, on page 9](#).
- **description**: See [Shared Parameters, on page 9](#).
- **id**: The database id of the MRD. Read-only field.
- **type**: The type of MRD. Values are as follows:
  - **voice**: Used only for the built-in Cisco\_Voice MRD. These MRDs are read-only.
  - **legacyMultichannel**: Used for MRDS for the Enterprise Chat and Email application. These MRDs are read-only.

◦ multichannel: (Default) Used for MRDs for Task Routing APIs.

- taskLife: If the connection goes down, the amount of time, in seconds, that the system waits before ending all tasks. Default is 1200.
- taskStartTimeout: The amount of time, in seconds, that the system waits between an agent being selected for a task and an agent being offered or beginning the task. When this time is reached, the system makes the agent Not Routable. Default is 30.
- maxTaskDuration: The maximum duration for a task, in seconds. Default is 28800.
- serviceLevelThreshold: Maximum time in seconds that a customer should wait before being connected with an agent. Default is 30.
- interruptible: Indicates if an agent can be interrupted by assigned tasks from another MRD. Values are true/false.
- maxTasksInQueue: The maximum number of tasks allowed to be queued at one time.
- maxTimeInQueue: The maximum amount of time, in seconds, a task can be queued.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> <li>• id</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> <li>• id</li> <li>• interruptible</li> <li>• maxTasksInQueue</li> <li>• maxTaskDuration</li> <li>• maxTimeInQueue</li> <li>• serviceLevelThreshold</li> <li>• taskLife</li> <li>• taskStartTimeout</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

### Advanced search parameters

You can perform a nonVoiceOnly search on the Media Routing Domain API:

- **nonVoiceOnly**: Set this attribute to true in the search query parameter to make the API return only media routing domains other than the Cisco\_Voice MRD. For example, **q=nonVoiceOnly:true**.



**Example Get Response**

```

<mediaRoutingDomains>
  <mediaRoutingDomain>
    <changeStamp>0</changeStamp>
    <refURL>/unifiedconfig/config/mediaroutingdomain/5001</refURL>
    <description>Media channel for routing Chat tasks</description>
    <id>5001</id>
    <type>multichannel</type>
    <interruptible>false</interruptible>
    <taskLife>1200</taskLife>
    <taskStartTimeout>30</taskStartTimeout>
    <maxTaskDuration>28800</maxTaskDuration>
    <maxTasksInQueue>1000</maxTasksInQueue>
    <maxTimeInQueue>1000</maxTimeInQueue>
    <name>Chat_Task_MRD</name>
    <serviceLevelThreshold>30</serviceLevelThreshold>
  </mediaRoutingDomain>
  <mediaRoutingDomain>
    <changeStamp>0</changeStamp>
    <refURL>/unifiedconfig/config/mediaroutingdomain/1</refURL>
    <description>Default Media Routing Domain for Cisco_Voice</description>
    <id>1</id>
    <type>voice</type>
    <interruptible>false</interruptible>
    <taskLife>1200</taskLife>
    <taskStartTimeout>30</taskStartTimeout>
    <maxTaskDuration>28800</maxTaskDuration>
    <name>Cisco_Voice</name>
    <serviceLevelThreshold>30</serviceLevelThreshold>
  </mediaRoutingDomain>
</mediaRoutingDomains>

```





## Network VRU Script API

Calls may be sent to a Voice Response Unit (VRU) instead of or before they are sent to an agent. In the Packaged CCE deployment, the VRU is Customer Voice Portal (Unified CVP). You must configure network VRU scripts to direct Unified CVP on how to handle the treatment of individual calls.

Use the Network VRU Script API to list, create, edit and delete network VRU scripts.

### URL

`https://<server>/unifiedconfig/config/networkvruscript`

### Operations

- **create**: Creates one network VRU script.
- **delete**: Deletes one network VRU script from the database.
- **get**: Returns one network VRU script, using the URL  
`https://<server>/unifiedconfig/config/networkvruscript/<id>`.
- **list**: Retrieves a list of network VRU scripts.
- **update**: Updates one network VRU script.

### Parameters

- **refURL**: The refURL of the network VRU script. See [Shared Parameters, on page 9](#).
- **name**: The name of the network VRU as seen by CCE. See [Shared Parameters, on page 9](#).
- **changeStamp**: See [Shared Parameters, on page 9](#).
- **description**: See [Shared Parameters, on page 9](#).
- **routingType**: This field is optional and defaults to 1. Options are:
  - 1: Voice. Used by Unified CVP.
  - 2: Multichannel. Used by Email and Web Collaboration.
- **vruscriptName**: Required. The name of the script as it is known on the Unified CVP. Maximum length of 39 characters allowed.

- **timeout:** Number of seconds for the system to wait for a response from the routing client after directing it to run the script. Must be an integer that is 1 or higher. Default is 180.
- **configParam:** Optional string used by Unified CVP to pass additional parameters to the IVR Service. Maximum length is 255 characters.
- **interruptible:** Indicates whether the script can be interrupted. Values are true/false.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> <li>• vruScriptName</li> <li>• timeout</li> <li>• configParam</li> <li>• interruptible</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

### Example Get Response

```
<networkVruScript>
  <refURL>/unifiedconfig/networkvruscript/(id)</refURL>
  <routingType>1</routingType>
  <name>test</name>
  <vruScriptName>GS,Server,V</vruScriptName>
  <timeout>180</timeout>
  <configParam>Y</configParam>
  <interruptible>true</interruptible>
  <description>CVP VXML Server script</description>
  <changeStamp>0</changeStamp>
</networkVruScript>
```



## Operation API

Use the Operation API to save changes to several items of the same type in a single request. The following changes are allowed in an operation:

- delete: Multiple items of the same type. Any item that supports the delete operation can be deleted using the Operation API.
- Agent update: Update multiple agents.

### URL

`https://<server>/unifiedconfig/config/operation`

### HTTP Method

Use HTTP POST to submit a request to the Operation API.

### Parameters

- operationType: Indicates if the items specified in the refURLs should be updated or deleted. Values are update/delete.
- refURLs: A collection of refURL parameters indicating which items are included in the request. See [Shared Parameters](#), on page 9.
- changeset: Includes the parameters that are changed in an update operation. See [Agent API](#), on page 19.
  - skillGroupsAdded
  - skillGroupsRemoved
  - agentAttributesAdded
  - agentAttributesRemoved
  - description
  - agentTeam
  - agentDeskSettings

### Example Delete Request

```
<operation>
  <operationType>delete</operationType>
  <refURLs>
    <refURL>/unifiedconfig/config/calltype/5000</refURL>
    <refURL>/unifiedconfig/config/calltype/5001</refURL>
  </refURLs>
</operation>
```

### Example Update Request

```
<operation>
  <operationType>update</operationType>
  <refURLs>
    <refURL>/unifiedconfig/config/agent/5000</refURL>
    <refURL>/unifiedconfig/config/agent/5001</refURL>
  </refURLs>
  <changeSet>
    <agent>
      <description>New description</description>
      <agentDeskSettings>
        <refURL>/unifiedconfig/config/agentdesksetting/5434</refURL>
      </agentDeskSettings>
      <agentTeam>
        <refURL>/unifiedconfig/config/agentteam/5000</refURL>
      </agentTeam>
      <skillGroupsAdded>
        <skillGroup>
          <refURL>/unifiedconfig/config/skillgroup/6000</refURL>
        </skillGroup>
      </skillGroupsAdded>
      <skillGroupsRemoved>
        <skillGroup>
          <refURL>/unifiedconfig/config/skillgroup/6001</refURL>
        </skillGroup>
      </skillGroupsRemoved>
      <agentAttributesAdded>
        <agentAttribute>
          <attribute>
            <refURL>/unifiedconfig/config/attribute/5004</refURL>
          </attribute>
          <attributeValue>8</attributeValue>
        </agentAttribute>
      </agentAttributesAdded>
      <agentAttributesRemoved>
        <agentAttribute>
          <attribute>
            <refURL>/unifiedconfig/config/attribute/5005</refURL>
          </attribute>
        </agentAttribute>
      </agentAttributesRemoved>
    </agent>
  </changeSet>
</operation>
```

### Response Parameters

- status: Indicates the state of the operation.
  - success: The operation succeeded for all items.
  - partialSuccess: The operation succeeded for some items, but other items had errors.
  - failure: The operation failed for all items.
- apiErrors: Errors indicate which items had errors and the cause of the error.

### Example Success Response

The following example shows the response when the delete operation is successful:

```
<operationsResult>
  <status>success</status>
</operationsResult>
```

### Example Partial Success Message

The following example shows a partial success response for a request to delete several agents:

```
<operationsResult>
  <apiErrors>
    <apiError>
      <errorDetail xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type=
        "resourceErrorDetail">
        <refURL>agent/1</refURL>
        <apiErrors>
          <apiError>
            <errorMessage>The specified ID does not exist
              in the database.</errorMessage>
            <errorType>notFound.dbData</errorType>
          </apiError>
        </apiErrors>
      </errorDetail>
      <errorMessage>There were one or more errors processing the following
        request: delete agent/1</errorMessage>
      <errorType>operation.resourceErrors</errorType>
    </apiError>
    <apiError>
      <errorDetail xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type=
        "resourceErrorDetail">
        <refURL>agent/2</refURL>
        <apiErrors>
          <apiError>
            <errorMessage>The specified ID does not exist
              in the database.</errorMessage>
            <errorType>notFound.dbData</errorType>
          </apiError>
        </apiErrors>
      </errorDetail>
      <errorMessage>There were one or more errors processing the following
        request: delete agent/2</errorMessage>
      <errorType>operation.resourceErrors</errorType>
    </apiError>
  </apiErrors>
  <status>partialSuccess</status>
</operationsResult>
```

### Example Failure Response

The following example shows a failure response for a request to delete a call type that does not exist:

```
<operationsResult>
  <status>failure</status>
  <apiErrors>
    <apiError>
      <errorDetail xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type=
        "resourceErrorDetail">
        <refURL>/unifiedconfig/config/calltype/9999</refURL>
        <apiErrors>
          <apiError>
            <errorMessage>The specified ID does not exist in the database.</errorMessage>
            <errorType>notFound.dbData</errorType>
          </apiError>
        </apiErrors>
      </errorDetail>
      <errorMessage>There were one or more errors processing the following request:
        delete /unifiedconfig/config
      </errorMessage>
    </apiError>
  </apiErrors>
```

```
        /calltype/9999</errorMessage>
        <errorType>operation.resourceErrors</errorType>
    </apiError>
</apiErrors>
</operationsResult>
```





## Outbound API

---

Outbound API allows you to use REST APIs to create, modify, and delete Outbound Option campaigns.

Outbound API provides a streamlined mechanism for creating campaigns with a single associated query rule and import rule. As such, if you use the API to create a campaign, that campaign is not available in the Outbound Campaign tool. If a campaign was created with the API, you must use the API to view, edit, or delete it. If a campaign was created with the Outbound Campaign tool, you must use the Outbound Campaign tool to view, edit, or delete it.

Administrative scripts are not required for Outbound Option campaigns created with the Outbound API. If an administrative script is provided, the information in the script overrides the information defined in the API.

Outbound API consists of the following APIs:

- **Outbound Campaign API:** Use this API to define new Outbound Option campaigns, and to view, edit, or delete existing campaigns.
  - **Campaign Status API:** Use this API to get the real-time status of running Outbound Option campaigns.
  - **Do Not Call API:** Use this API to set the Do Not Call (DNC) import rule configuration for Outbound Option. This prevents the dialer from dialing numbers on the DNC list.
  - **Import API:** Use this API to import customer contact information for an Outbound Option campaign.
  - **Personal Callback API:** Use this API to configure your Outbound Option campaign to handle personal callbacks. You can create personal callback records individually or in bulk. You can also update or delete personal callback records.
  - **Time Zone API:** Use this API to list all available time zones and to get information about a specified time zone.
- 
- [Outbound Campaign API, page 100](#)
  - [Campaign Status API, page 106](#)
  - [Do Not Call API, page 108](#)
  - [Import API, page 110](#)
  - [Personal Callback API, page 114](#)
  - [Time Zone API, page 119](#)

# Outbound Campaign API

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

Use the Outbound Campaign API to define new outbound campaigns, and to view, edit, or delete existing outbound campaigns.

## URL

`https://<server>/unifiedconfig/config/campaign`

## Operations

- **create**: Creates one campaign and stores it in the database.
- **delete**: Deletes one campaign from the database. The campaign is saved under **Configuration Manager > Miscellaneous Tools > Deleted Objects**.

You can only delete campaigns that were created using the API. You cannot use the API to delete campaigns that were created in the Outbound Option Campaign tool.

- **list**: Retrieves a list of campaigns from the database. Only campaigns that were created using the Outbound Option API are retrieved. Campaigns created using the Outbound Option Campaign tool do not appear in the list.
- **get**: Returns one campaign from the database using the URL  
`https://<server>/unifiedconfig/config/campaign/<id>`.

You must specify the ID for a campaign that was created using the API. If you specify the ID for a campaign created using the Outbound Option Campaign tool, the request returns `CceDBDataNotFoundException`.

- **update**: Updates one campaign in the database using the URL  
`https://<server>/unifiedconfig/config/campaign/<id>`.

You can only update campaigns that were created using the API. You cannot use the API to update campaigns that were created in the Outbound Option Campaign tool.

## Parameters

- **name**: Required. The name of the campaign. See [Shared Parameters, on page 9](#).
- **description**: Optional description for the campaign. See [Shared Parameters, on page 9](#).
- **linesPerAgent**: The number of lines dedicated to each agent in the campaign. Range is 1 to 100. Default is 1.5. This parameter performs as follows in the Outbound Option dialing modes:
  - Preview mode: Ignored (always 1).
  - Progressive mode: Used as defined.
  - Predictive mode: Used as an initial value.
- **maximumLinesPerAgent**: The upper bound for the number of customers the dialer dials for a reserved agent when a campaign is running in predictive mode. Range is 1 to 100. Default is 2.

- **abandonEnabled**: True or false. Default is true.
- **abandonPercent**: When enabled (**abandonEnabled** is set to true), you can set the abandoned calls limit for the percentage of abandoned calls in the campaign. When disabled (**abandonEnabled** is set to false), the campaign dials without regard to the abandon limit. Range is 1 to 100. The granularity is to one 10th of a percent. Default is 3.
- **predictiveCorrectionPace**: A count of the number of live voice connections that must occur before the Dialer adjusts. Increasing this number results in less frequent adjustments based on larger sample size. Decreasing this number results in more frequent adjustments using a smaller sample size. Range is 10 to 5000. Default is 70.
- **predictiveGain**: The size of the adjustment to lines per agent each time an adjustment is made. Increasing this number results in larger lines per agent adjustments. Decreasing this number results in smaller lines per agent adjustments. Range is 0.1 to 3.0. Default is 1.
- **noAnswerRingLimit**: The number of times the software allows a dialed phone number to ring. Range is from 2 to 10. Default is 4.
- **maxAttempts**: The maximum number of attempts, including callbacks and retries. Range is from 1 to 100. Default is 3.
- **minimumCallDuration**: Minimum duration (in seconds) of an outbound call. If the outbound call is less than the specified value, Outbound Option considers the call to be customer abandoned and schedules the record for a retry. To disable this feature, set the parameter to 0. Range is 0 to 10. Default is 1.
- **noAnswerDelay**: The time (in minutes) that the software waits before calling back a no-answer call. Range is 1 to 99999. Default is 60.
- **busySignalDelay**: The time (in minutes) that the software waits before calling back a busy phone number. Range is 1 to 99999. Default is 60.
- **customerAbandonedDelay**: If a customer abandons a call, the time (in minutes) that the Dialer waits before calling the customer back. Range is 1 to 99999. Default is 30.
- **dialerAbandonedDelay**: If the Dialer abandons a call, the time (in minutes) that the Dialer waits before calling the customer back. Range is 1 to 99999. Default is 60.
- **answeringMachineDelay**: If an answering machine answers a call, the time (in minutes) that the Dialer waits before calling the customer back. Range is 1 to 99999. Default is 60.
- **customerNotHomeDelay**: If a customer is not home, the time (in minutes) that the Dialer waits before calling the customer back. Range is 1 to 99999. Default is 60.
- **personalizedCallbackEnabled**: If enabled, this parameter allows an agent to schedule a callback to a customer for a specific date and time. A personal callback connects the same agent who initiated the callback to the customer. True or false. Default is false.
- **rescheduleCallbackMode**: Determines how Outbound Option handles a personal callback if the agent is not available. Default is **useCampaignDN**. Options are as follows:
  - Use campaign DN
  - Same time next business day
  - Abandon
- **campaignPurposeType**: The type of campaign. Default is **agentCampaign**. The options are as follows:

- Agent Campaign: This type of campaign uses an outbound mode that causes the Dialer to transfer every customer call associated with a specific skill group to an agent.
- IVR Campaign: This type of campaign uses an outbound mode that causes the Dialer to transfer every customer call associated with a specific skill group to a service control-based IVR instead of an agent. This option allows a contact center to run unassisted outbound campaigns using prerecorded messages in the IVR.
- ipAmdEnabled: When enabled, directs the Dialer to perform a specific action if it detects an answering machine. True or false. Default is true.
- amdTreatmentMode: If enabled (ipAmdEnabled is set to true), when the Dialer detects an answering machine, it does one of the following:
  - Abandon call (default)
  - Transfer to agent
  - Transfer to IVR route point
  - Terminate tone detect
- ipTerminatingBeepDetect: When this parameter is set to true, the Dialer transfers the call after detecting the answering machine beep. True or false. Default is false.
- timeZone: Required. The refURL and display name for the selected time zone. The default time zone is UTC (Universal Coordinated Time). The display name is the text that may be displayed in a user interface and can be localized.

**Note**

If time zone information changed due to periodic updates and the campaign's configured time zone is no longer valid, the following information is returned:

```
<timeZone>
  <refURL>/unifiedconfig/config/timezone/INVALID</refURL>
  <displayName>INVALID Time Zone</displayName>
</timeZone>
```

- startTime: The time the campaign starts dialing customer numbers. The format for this parameter is hours:minutes. Range is from 00:00 to 23:59. The default value is taken from the Blended\_Agent\_Options table column values for DialStartHours and DialStartMinutes.
- endTime: The time the campaign stops dialing customer numbers. The format for this parameter is hours:minutes. Range is from 00:00 to 23:59. The default value is taken from the Blended\_Agent\_Options table column values for DialEndHours and DialEndMinutes.
- enabled: Whether the dialer is available to call contacts. True or false. Default is false.
- startDate: The date that the campaign starts. The format is YYYY-MM-DD.
- endDate: The date that the campaign ends. The format is YYYY-MM-DD.
- campaignPrefix: Digits to prefix to each customer number dialed from this campaign. Maximum length of 15 digits.
- dialingMode: The dialing mode to use for the campaign skill groups. Valid values are as follows:
  - INBOUND

- PREDICTIVEONLY
  - PREVIEWONLY
  - PROGRESSIVEONLY
  - PREVIEWDIRECTONLY
- reservationPercentage: The percentage of agents to reserve within the skill groups associated with the campaign. Range is from 0 to 100. Default is 100.
  - callProgressAnalysis: A collection of parameters for Call Progress Analysis (CPA). Any combination of parameters within this collection can be set. If none of the parameters are provided and CPA is enabled for the campaign by default, CPA recording is set to false and default parameter values are set from the Blended\_Agent\_Options table.
    - enabled: When set to false, CPA for all calls made from this Dialer is disabled on a campaign-by-campaign basis, including voice detection, fax/modem detection, and answering machine detection. True or false. Default is true.
    - record: If enabled is set to true, you can specify this parameter. If you set it to true, the gateway provides a media stream and the Dialer records .wav files. True or false. Default is false.
    - minSilencePeriod: The minimum silence period (in milliseconds) required to classify a call as voice detected. If many answering machine calls are being passed through to agents as voice, then increasing this value accounts for longer pauses in answering machine greetings. Range is from 100 to 1000. Default is 608.
    - analysisPeriod: The number of milliseconds spent analyzing this call. If there is a short agent greeting on an answering machine, then a longer value categorizes that answering machine call as voice. If the call is to a business where the operator has a longer scripted greeting, a shorter value categorizes the long, live greeting as an answering machine call. Range is from 1000 to 10000. Default is 2500.
    - minimumValidSpeech: Minimum number of milliseconds of voice required to classify a call as voice detected. Range is 50 to 500. Default is 112.
    - maxTimeAnalysis: The maximum number of milliseconds allowed for analysis before identifying a problem analysis as dead air/low volume. Range is 1000 to 10000. Default is 3000.
    - maxTermToneAnalysis: The maximum number of milliseconds the dialer analyzes an answering machine voice message looking for a termination tone. If the message has an odd tone and the analysis does not recognize it, the call is not transferred or dropped until this timeout occurs. Range is 1000 to 60000. Default is 30000.
  - skillGroupInfos: A collection of information about the skill groups associated with the campaign.
    - skillGroupInfo: A collection of information about one skill group.
      - skillGroup: The name and the refURL of the skill group assigned to the campaign.
      - overflowAgents: This parameter ensures that at least one extra agent is reserved before the dialer begins dialing for a Progressive campaign. If the parameter is set to 1, at least two agents must be reserved before the dialer begins dialing. Range is 0 to 100. Default is 0.
      - dialedNumber: The digits that are dialed to reserve an agent in the configured skill group. This parameter can contain letters, numbers, periods (.), and underscores (\_) and can be up to ten characters in length.

- **recordsToCache**: The minimum number of dialing numbers that each dialer caches for each of the Outbound Option skill groups. Range is 1 to 100. Default is 1.
  - **ivrPorts**: The total number of IVR ports allocated for the skill group. This parameter indicates how many ports are available for the dialer to transfer customer calls. Because this value indicates the total number of ports supported by the IVR for the current skill group, multiple skill groups can make transfer to IVR calls. One IVR can also be used to play different messages based on the route point where the contact is transferred. If multiple dialers are associated with the skill group, each dialer dials a fraction of the total number of ports. Default is 0.
  - **ivrRoutePoint**: If the campaign is a Transfer to IVR campaign or is configured to transfer AMD calls to an IVR, this parameter indicates the route point required to run the transfer to IVR routing script. This parameter must coincide with a route point configured on Unified Communications Manager and be assigned to a PGUser. Contacts are transferred to the route point, which points to a routing script. The script transfers the call to an IVR. Maximum length of 32 characters.
  - **abandonedRoutePoint**: If the campaign is a Transfer to IVR campaign or is configured to transfer AMD calls to an IVR, this number allows the dialer to play a message to calls about to be disconnected because no agents are available. This number must coincide with a route point configured on Unified Communications Manager and be assigned to the agent PG's CTI application (for example, PGUser). Contacts are transferred to this route point, which points to a routing script. The script transfers the call to an IVR. Maximum length of ten characters.
- **skillGroupInfosAdded**: A collection of skill group references to be added to the campaign, including the skill group refURL. This parameter is update only. This parameter can be used with the skillGroupInfosRemoved parameter. See [References](#), on page 6.
  - **skillGroupInfosRemoved**: A collection of skill group references to be removed from the campaign, including the skill group refURL of each skill group. This parameter is update only. This parameter can be used with the skillGroupInfosAdded parameter. See [References](#), on page 6.

### Search and Sort Parameters

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

### Disable All Campaigns (Emergency Stop)

The Outbound Campaign API provides a method to immediately disable all outbound campaigns that are currently enabled. To disable all of the campaigns, perform a PUT on the following URL:

`https://<server>/unifiedconfig/config/campaign/disable`

This request retrieves a list of all campaigns created using the Outbound Campaign API that are currently enabled and sets the enabled parameter to false. The Campaign Manager stops sending out records to the Dialer for these campaigns. All active records in the Dialer's memory are removed.

### Example Get Response

```
<campaign>
  <changeStamp>48</changeStamp>
  <refURL>/unifiedconfig/config/campaign/5168</refURL>
  <import>
    <refURL>/unifiedconfig/config/import/5000</refURL>
  </import>
  <abandonEnabled>true</abandonEnabled>
  <abandonPercent>3.0</abandonPercent>
  <amdTreatmentMode>abandonCall</amdTreatmentMode>
  <campaignPrefix>978</campaignPrefix>
  <campaignPurposeType>agentCampaign</campaignPurposeType>
  <dialingMode>PREVIEWONLY</dialingMode>
  <enabled>true</enabled>
  <endDate>2016-01-15</endDate>
  <endTime>17:00</endTime>
  <ipAMDEnabled>true</ipAmdEnabled>
  <ipTerminatingBeepDetect>false</ipTerminatingBeepDetect>
  <linesPerAgent>1.5</linesPerAgent>
  <maxAttempts>3</maxAttempts>
  <maximumLinesPerAgent>100.0</maximumLinesPerAgent>
  <minimumCallDuration>1</minimumCallDuration>
  <name>APIoct1</name>
  <noAnswerRingLimit>4</noAnswerRingLimit>
  <personalizedCallbackEnabled>false</personalizedCallbackEnabled>
  <predictiveCorrectionPace>70</predictiveCorrectionPace>
  <predictiveGain>1.0</predictiveGain>
  <rescheduleCallbackMode>useCampaignDN</rescheduleCallbackMode>
  <reservationPercentage>100</reservationPercentage>
  <retries>
    <answeringMachineDelay>60</answeringMachineDelay>
    <busySignalDelay>60</busySignalDelay>
    <customerAbandonedDelay>30</customerAbandonedDelay>
    <customerNotHomeDelay></customerNotHomeDelay>
    <dialerAbandonedDelay>60</dialerAbandonedDelay>
    <noAnswerDelay>60</noAnswerDelay>
  </retries>
  <skillGroupInfos>
    <skillGroupInfo>
      <ivrPorts>0</ivrPorts>
      <overflowAgents>0</overflowAgents>
      <recordsToCache>1</recordsToCache>
      <abandonedRoutePoint>12345</abandonedRoutePoint>
      <dialedNumber>123</dialedNumber>
      <ivrRoutePoint>91234</ivrRoutePoint>
      <skillGroup>
        <refURL>/unifiedconfig/config/skillgroup/5004</refURL>
        <name>errorDetailsRouteScript</name>
      </skillGroup>
    </skillGroupInfo>
  </skillGroupInfos>
  <startDate>2016-01-14</startDate>
  <startTime>09:00</startTime>
  <timeZone>
    <displayName>(UTC-05:00) Eastern Time (US & Canada)</displayName>
    <refURL>/unifiedconfig/config/timezone/Eastern%20Standard%20Time</refURL>
  </timeZone>
  <callProgressAnalysis>
    <enabled>true</enabled>
    <record>false</record>
    <minSilencePeriod>608</minSilencePeriod>
    <analysisPeriod>2500</analysisPeriod>
    <minimumValidSpeech>112</minimumValidSpeech>
    <maxTimeAnalysis>3000</maxTimeAnalysis>
  </callProgressAnalysis>
</campaign>
```

```

    <maxTermToneAnalysis>30000</maxTermToneAnalysis>
  </callProgressAnalysis>
</campaign>

```

### Example Create Request

```

<campaign>
  <name>APIOct1</name>
  <description>APIOct1</description>
  <dialingMode>PREVIEWONLY</dialingMode>
  <skillGroupInfos>
    <skillGroupInfo>
      <ivrPorts>0</ivrPorts>
      <overflowAgents>0</overflowAgents>
      <recordsToCache>1</recordsToCache>
      <skillGroup>
        <refURL>/unifiedconfig/config/skillgroup/5001</refURL>
        <name>sgcampaign</name>
      </skillGroup>
    </skillGroupInfo>
  </skillGroupInfos>
  <timeZone>
    <refURL>/unifiedconfig/config/timezone/UTC</refURL>
  </timeZone>
  <callProgressAnalysis>
    <enabled>true</enabled>
    <record>false</record>
    <minSilencePeriod>608</minSilencePeriod>
    <analysisPeriod>2500</analysisPeriod>
    <minimumValidSpeech>112</minimumValidSpeech>
    <maxTimeAnalysis>3000</maxTimeAnalysis>
    <maxTermToneAnalysis>30000</maxTermToneAnalysis>
  </callProgressAnalysis>
</campaign>

```

## Campaign Status API

Use the Campaign Status API to get the real-time status of running Outbound Option campaigns.

### URL

`https://<server>/unifiedconfig/config/campaign/<campaign-id>/runtimestats`



#### Note

You must specify the ID for a campaign that was created using the Outbound Campaign API (see [Outbound Campaign API](#), on page 100). If you specify the ID for a campaign created using the Outbound Option Campaign tool, the request returns `CceDBDataNotFoundException`.

### Operations

- `get`: Returns the runtime status of a campaign.

### Response Parameters

- `abandonDetectCount`: The number of calls abandoned by the dialer.
- `abandonToIvrCount`: The number of calls that were abandoned by the dialer and transferred to IVR.
- `agentClosedCount`: The number of preview and callback calls that were closed by the agent.



- **agentRejectedCount:** The number of preview and callback calls that were rejected by the agent.
- **answeringMachineCount:** The number of calls that detected an answering machine.
- **attemptedCount:** Summary total of the number of calls attempted.
- **busyCount:** The number of calls that detected a busy signal.
- **callBackCount:** The number of callback contacts.
- **cancelledDetectCount:** The number of calls where the dialer canceled a ringing customer call.
- **closedCount:** The number of contacts closed for any reason other than reaching a live customer.
- **customerAbandonDetectCount:** The number of calls where the customer hung up immediately after picking up the phone.
- **customerNotHomeCount:** The number of contacts where the party who answered the phone was not the customer.
- **dateTime:** The date and time when this data was updated last. The format is yyyy-MM-ddTHH:mm:ss (for example, 2016-03-13T04:50:31).
- **faxDetectCount:** The number of calls that detected a fax machine.
- **networkAnsMachineCount:** The number of calls that detected a network answering machine.
- **noAnswerDetectCount:** The number of calls that were not answered.
- **noDialToneDetectCount:** The number of calls that did not detect a dial tone.
- **noRingBackDetectCount:** The number of calls that did not detect a ring back.
- **personalCallbackCount:** The number of callback contacts scheduled.
- **sitToneDetectCount:** The number of calls that detected a special information tone (SIT).
- **talkTimeCount:** The total number of seconds that agents spent talking on the phone since midnight.
- **totalCount:** The total number of records available to dial for the current campaign.
- **totalVoiceCount:** The number of live customers that have been reached for this campaign since the last time the imported dialing list was overwritten.
- **voiceCount:** The number of calls for the day that ended in successful customer contact.
- **wrapupTimeCount:** The number of seconds that agents spent in wrap-up mode since midnight.
- **wrongNumberCount:** The number of contacts where the party who answered the phone indicated that the customer did not live there.

### Example Get Response

```
<runtimeStatus>
  <abandonDetectCount>0</abandonDetectCount>
  <abandonToIvrCount>0</abandonToIvrCount>
  <agentClosedCount>0</agentClosedCount>
  <agentRejectedCount>0</agentRejectedCount>
  <answeringMachineCount>0</answeringMachineCount>
  <attemptedCount>1</attemptedCount>
  <busyCount>0</busyCount>
  <callBackCount>0</callBackCount>
  <cancelledDetectCount>0</cancelledDetectCount>
  <closedCount>0</closedCount>
```

```

<customerAbandonDetectCount>0</customerAbandonDetectCount>
<customerNotHomeCount>0</customerNotHomeCount>
<dateTime>2016-01-15T13:43:00</dateTime>
<faxDetectCount>0</faxDetectCount>
<networkAnsMachineCount>0</networkAnsMachineCount>
<noAnswerDetectCount>0</noAnswerDetectCount>
<noDialToneDetectCount>0</noDialToneDetectCount>
<noRingBackDetectCount>0</noRingBackDetectCount>
<personalCallbackCount>0</personalCallbackCount>
<sitToneDetectCount>0</sitToneDetectCount>
<talkTimeCount>1</talkTimeCount>
<totalCount>0</totalCount>
<totalVoiceCount>1</totalVoiceCount>
<voiceCount>1</voiceCount>
<wrapupTimeCount>0</wrapupTimeCount>
<wrongNumberCount>0</wrongNumberCount>
</runtimeStatus>

```

## Do Not Call API

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

The DNC import is automatically scheduled to start when the file is available. After the import is complete, the file is RENAMED or DELETED based on the boolean value for the renameFileAfterImport field.

### URL

`https://<server>/unifiedconfig/config/dnc`

### Operations

- **create**: Sets the configuration in the database. Returns the refURL if the configuration is set.
- **delete**: Deletes the specific DNC configuration using the URL `https://<server>/unifiedconfig/config/dnc/<id>`. This does not delete the DNC import file that is present at the location of the filePath.
- **list**: Retrieves a list available DNC import rules.
- **get**: Returns the DNC import configuration from the database using the URL `https://<server>/unifiedconfig/config/dnc/<id>`.
- **update**: Updates the specific DNC import configuration with the new values using the URL `https://<server>/unifiedconfig/config/dnc/<id>`.

### Parameters

- **name**: Name of the DNC import. See [Shared Parameters](#), on page 9.
- **filePath**: Path to a file on the logger or accessible from the logger. Maximum length of 255 characters.
- **overwrite**: Whether the new DNC import overwrites the phone numbers from the previous DNC import. True or false. Default is false.
- **renameFileAfterImport**: Whether to delete the DNC import file after the import is complete. True or false. Default is true. If the parameter is set to true, the DNC file is renamed. If the parameter is set to false, the DNC import file is deleted.

### Search and Sort Parameters

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters  | Sort parameters  |
|--|--|
| <ul style="list-style-type: none"> <li>• name</li> </ul> | <ul style="list-style-type: none"> <li>• name</li> </ul> |

See [Search, on page 11](#) and [Sort, on page 12](#).

### Example Get Response

```
<dnc>
  <refURL>/unifiedconfig/config/dnc/unifiedconfig/config/dnc/5001</refURL>
  <name>dnc1</name>
  <filePath>\\CCE-ROGGER-A\C$\Users\boston\Desktop\dnc.txt</filePath>
  <overwrite>>false</overwrite>
  <renameFileAfterImport>true</renameFileAfterImport>
</dnc>
```

## Create Do Not Call List

When creating a Do\_Not\_Call list file, format it correctly using the following instructions.

### Procedure

- 
- Step 1** Using a text editor, create a text file that contains all the do-not-call phone numbers.
- Step 2** Enter a phone number for each Do Not Call entry on a new line.
- Step 3** Observe the following characteristics for each Do Not Call entry:
- Each phone number can be a maximum of 20 characters long.
  - The Do Not Call table can support up to 60 million entries, but note that the information is stored in memory in the Campaign Manager process.
- Step 4** Save the text file to the local server.
- 

The following is an example of a Do\_Not\_Call list:

2225554444

2225556666

2225559999

To add a customer to this list, import a Do Not Call list.

The Campaign Manager reads the Do Not Call import files. Dialing List entries are marked as Do Not Call entries only when the Campaign Manager fetches the Dialing List entry *and only when there is an exact, digit-for-digit match*. This allows Do Not Call imports to happen while a Campaign is running without rebuilding the Dialing List.

**Note**

If the Dialing List includes a base number plus extension, this entry must match a Do Not Call entry for that same base number and same extension. The dialer will not dial the extension.

When the Campaign Manager starts it automatically imports from the DoNotCall.restore file that is stored in the `<drive>\icm\<instance>\la\bin` directory. When reading Do Not Call import files, the Campaign Manager appends the data to the DoNotCall.restore file. This restore file allows recovery of Do Not Call records after the Campaign Manager stops unexpectedly or for planned maintenance, such as a Service Release installation.

The restore file can grow to approximately 1 GB if 60 million DNC records are imported, each having ten-digit numbers plus five-digit extensions. Sufficient disk space must be available on LoggerA to store the DoNotCall.restore file.

**Note**

To clear the Do Not Call list, import a blank file with the Overwrite table option enabled.

## Import API

Use the Import API to import customer contact information for an outbound campaign.

You can import up to 10,000 records in one create request.

**Note**

You can queue up to 30 requests at a time. If you have 30 requests in queue and submit another request, the response is HTTP 503 Service Unavailable with the boundary error condition "Request processing queue is at capacity."

If you receive this error, wait a few seconds and then retry the request.

### URL

`https://<server>/unifiedconfig/config/campaign/<id>/import`

**Note**

You must specify the ID for a campaign that was created using the Outbound Campaign API. If you specify the ID for a campaign created using the Outbound Option Campaign tool, the request returns `CceDBDataNotFoundException`.

### Operations

- **create**: Imports customer contacts for a specific campaign.
- **delete**: Deletes the imported contacts for a specific campaign.

**Note**

The delete operation deletes all imported contacts for the campaign. You cannot delete an individual record.

- **get** (template): Returns a sample CSV template for contacts, which is provided by the API, using the URL `https://<server>/unifiedconfig/config/campaign/import/template`. The response contains the CSV template as a file attachment.
- **get**: Retrieves the details of a single import record using the URL `https://<server>/unifiedconfig/config/campaign/<campaignId>/import/<id>`. The `<id>` is the ID of the imported contact.
- **list**: Retrieves a list of all imported contacts for the campaign.
  - **Query parameters**
    - Summary list: See [list](#), on page 4.

## Parameters

### Parameters for get:

- **accountNumber**: The customer's account number.
- **callsMade**: The number of calls made.
- **callStatus**: The call status of the last call placed for this record. Possible values include the following:
  - active (A)
  - callbackRequested (B)
  - closed (C)
  - agentRejected (J)
  - maxAttemptsReached (M)
  - pending (P)
  - retry (R)
  - personalCallbackRequested (S)
  - unknown (U)

For more information about these values, see the CallStatusZone Values section of the *Outbound Option Guide for Unified Contact Center Enterprise* at <http://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-user-guide-list.html>.

- **callResultOverall**: The call result from the last call placed for this record.
- **firstName**: The customer's first name.
- **lastName**: The customer's last name.
- **importDate**: The date and time that the record was imported.
- **phone01 through phone10**: A collection of information about the customer's phones. Each phone contains the following parameters:
  - **number**: The phone number.
  - **callResult**: The call result from the last call placed for this phone number.

For more information about possible callResult values, see the CallResult Codes and Values section of the *Outbound Option Guide for Unified Contact Center Enterprise* at <http://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-user-guide-list.html>.

- dstObserved: Whether DST is observed for this phone's location. True or false.
- gmtOffset: The number of minutes that the customer's time zone is offset from GMT. When you create import records, you can optionally provide a TimeZoneBias. The TimeZoneBias parameter represents the way information about time zones is stored in the Windows registry. You can use the following formula to convert TimeZoneBias to gmtOffset:

```
if timeZoneBias is postive:
    gmtOffset = 1440 - timeZoneBias (where 1440 is the number of minutes in 24
    hours)
else if timeZoneBias is negative:
    gmtOffset = -1 * timeZoneBias
else if timeZoneBias is 0
    gmtOffset = timeZoneBias
```

#### Parameters for create:

- fileContent: Required. Comma-separated or pipe-separated list of data embedded within a CDATA section.
- overwriteData: True or false. If set to true, the existing import data in the database is overwritten by the new import data. If set to false, the new import data is appended to the existing data. Default is false.




---

**Note** You cannot modify or remove existing fields or append a new field to existing data.

---

- delimiter: Comma (,) or pipe (|). Default is comma (,).
- header fields: The following header fields are allowed:
  - AccountNumber: Can contain any characters, including internationalized characters, except the delimiter specified in the input XML. Maximum length of 30 bytes.
  - FirstName: Can contain any characters, including internationalized characters, except the delimiter specified in the input XML. Maximum length of 50 characters.
  - LastName: Can contain any characters, including internationalized characters, except the delimiter specified in the input XML. Maximum length of 50 characters.
  - Phone01 through Phone10: Can contain digits 0-9, pound sign (#), and asterisk (\*). Maximum length of 20 bytes. At least one Phone field is required.
  - TimeZoneBias: Specifies the offset of the customer's time zone from UTC in minutes. Integer. Range is from -780 to 720 (-13 to 12 hours from UTC).
  - DstObserved: Specifies whether DST is observed for the customer's location. True or false. Default is false.

**Note**

If the TimeZoneBias parameter is provided but the DstObserved parameter is not, DstObserved is set to false. If the DstObserved parameter is provided but not TimeZoneBias, a validation error is returned.

If TimeZoneBias is not provided, time zone and Daylight Saving Time information is assigned by matching phone numbers to region prefix strings. If a phone number for a contact does not match a configured region prefix, the default time zone for the campaign is used.

If a customer record contains multiple phone numbers, the value for TimeZoneBias and DstObserved are applied to all of that customer's phone numbers.

You can use the Time Zone API to look up the values for TimeZoneBias and DstObserved for the target time zone. For more information, see [Time Zone API](#), on page 119.

**Search and Sort**

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters  | Sort parameters  |
|--|--|
| <ul style="list-style-type: none"> <li>• accountNumber</li> <li>• firstName</li> <li>• lastName</li> </ul> | <ul style="list-style-type: none"> <li>• accountNumber (default)</li> <li>• firstName</li> <li>• lastName</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

**Example Get Response**

```
<importContact>
  <refURL>/unifiedconfig/config/campaign/5000/import/1</refURL>
  <accountNumber>4019</accountNumber>
  <callsMade>1</callsMade>
  <callStatus>pending</callStatus>
  <callResultOverall>0</callResultOverall>
  <firstName>Mir</firstName>
  <lastName>Ali</lastName>
  <importDate>2016-03-28T01:09:40</importDate>
  <phone01>
    <callResult>0</callResult>
    <dstObserved>true</dstObserved>
    <gmtOffset>720</gmtOffset>
    <number>9789360001</number>
  </phone01>
  <phone02>
    <callResult>0</callResult>
    <dstObserved>true</dstObserved>
    <gmtOffset>720</gmtOffset>
    <number>9789360002</number>
  </phone02>
  <phone03>...</phone03>
  <phone04>...</phone04>
```

```

<phone05>...</phone05>
<phone06>...</phone06>
<phone07>...</phone07>
<phone08>...</phone08>
<phone09>...</phone09>
<phone10>
  <callResult>0</callResult>
  <dstObserved>true</dstObserved>
  <gmtOffset>330</gmtOffset>
  <number>9789360010</number>
</phone10>
</importContact>

```

### Example Create Request

```

<import>
  <fileContent>
    <![CDATA[
      AccountNumber,FirstName,LastName,Phone01,Phone02,TimeZoneBias,DstObserved
      6782, Henry, Martin, 2225554444, 2225556262, 720, false
      3456, Michele, Smith, 2225559999, 2225551234, 540, true
      4569, Walker, Evans, 2225552000, 2225557890, -600, true
    ]]>
  </fileContent>
  <delimiter>,</delimiter>
</import>

```

## Personal Callback API

Use the Personal Callback API to configure your Outbound Option 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.

### URL

<https://<server>/unifiedconfig/config/personalcallback>

### Operations

- **create**: Creates a single PersonalCallback record and stores it in the BA database.
- **create (bulk)**: Creates PersonalCallback records in bulk and stores them in the BA database using the URL <https://<server>/unifiedconfig/config/personalcallback/import>.
- **delete**: Deletes one or more PersonalCallback records from the PersonalCallback list using the URL <https://<server>/unifiedconfig/config/personalcallback/<id>>.
- **get (template)**: Retrieves a sample CSV template for contacts, which is provided by the API, using the URL <https://<server>/unifiedconfig/config/personalcallback/template>.
- **get (record)**: Retrieves one PersonalCallback record from the database using the URL <https://<server>/unifiedconfig/config/personalcallback/<id>>.
- **list**: Retrieves a list of PersonalCallback records from the database.
  - **Query parameters:**
    - agentId
    - accountNumber



- firstName
- lastName

- **update**: Updates one PersonalCallback record in the database using the URL `https://<server>/unifiedconfig/config/personalcallback/<id>`.

## Parameters

### Parameters for create (single record), get, and update operations:

- **campaign**: A reference to a specific campaign, including the refURL and campaign ID.
- **agent**: A reference to a specific agent, including the refURL and agent ID.
- **campaignId**: Read only. The ID for the campaign.
- **peripheralId**: Read only. ID for the peripheral on which the agent would be available.
- **agentId**: Read only. ID of the agent to whom to connect the call.
- **campaignDn**: The Dialed Number (DN) to use if the original agent is not available.
- **phone**: Required. The phone number to call back. Can contain digits 0 to 9, pound sign (3), and asterisk (\*). Maximum length of 20 bytes.
- **accountNumber**: The customer's account number. Internationalized characters are allowed. Maximum length of 30 bytes.
- **maxAttempts**: Required. The maximum number of times to attempt a call (decrements at each attempt). An attempt is the Dialer's attempt to reserve the agent and call the customer. Because the Dialer places multiple customer call attempts (such as busy, no answer), individual call attempts are not tracked here; only the result at the end of the callback time range. After this parameter is set to 0, no more attempts are made. Must be a positive integer value.
- **callbackDateTime**: Required. The time to attempt the customer callback is normalized to the logger GMT zone. For example, the Campaign Manager is in Boston and the customer is in California. The customer wants to be called back at 3 p.m. The time in this column is 6 p.m. The format for this parameter is yyyy-MM-ddTHH:mm:ss (for example, 2016-03-13T04:50:31).

The following rules apply to the callbackDateTime parameter:

- The callbackDateTime that is used is local to the time zone of the Logger. You need to consider this if the client machine on which you are using this API is in a different time zone than the Logger machine.
- The setting of the callbackDateTime by this API reads the CallbackDateTimeLimit in the registry. For more information, see the Registry Settings section of the *Outbound Option Guide for Unified Contact Center Enterprise* at <http://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-user-guide-list.html>.
- The Campaign Manager uses the CallbackDateTimeLimit registry value to select Personal Callback records to queue to the Dialer.
- The default CallbackDateTimeLimit in the registry is 15 minutes. For create and update operations, you cannot set the callbackDateTime to 15 minutes or more before the current time. For example, if it is currently 4:00, you cannot set the callbackDateTime to 3:30.

- For update operations, the new `callbackDateTime` value can only be set if the current `callbackDateTime` is 15 minutes or more after the current Logger time. For example, if the current Logger time is 4:00 and the current `callbackDateTime` is 4:30, the new `callbackDateTime` can be set to 4:10. However, if the current `callbackDateTime` is 4:10, it cannot be set to 4:00.

- `callStatus`: The status of the personal callback. Possible values include the following:
  - `active` (A)
  - `callbackRequested` (B)
  - `closed` (C)
  - `agentRejected` (J)
  - `maxAttemptsReached` (M)
  - `pending` (P)
  - `retry` (R)
  - `personalCallbackRequested` (S)
  - `unknown` (U)
  - `agentNotAvailable` (X)

For more information about these values, see the *CallStatusZone Values* section of the *Outbound Option Guide for Unified Contact Center Enterprise* at <http://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-user-guide-list.html>.

- `callResult`: Read only. Corresponds to the `CallResult` value in the `Dialer_Detail` table.  
For more information about possible `callResult` values, see the *CallResult Codes and Values* section of the *Outbound Option Guide for Unified Contact Center Enterprise* at <http://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-user-guide-list.html>.
- `lastName`: The customer's last name. Internationalized characters are allowed. Maximum length of 50 characters.
- `firstName`: The customer's first name. Internationalized characters are allowed. Maximum length of 50 characters.

#### Bulk parameters:

- `fileContent`: Comma-separated or pipe-separated list of data embedded within a CDATA section.
- `overwriteData`: True or false. If set to true, the existing personal callback data in the database is overwritten by the new personal callback data. If set to false, the new personal callback data is appended to the existing data. Default is false.




---

**Note** You cannot modify or remove existing fields or append a new field to existing data. If `overwriteData` is set to true, any agent scheduled personal callback records are also overwritten.

---

- `delimiter`: Comma (,) or pipe (|). Default is comma (,).

**Bulk fileContent parameters:****Note**

The descriptions for these parameters are the same as the descriptions for the parameters for create (single record), get, and update operations. Only the case is different.

- CampaignId (must be a valid campaign ID in the database)
- AgentSkillTargetId (must be a valid SkillTargetID in the database)
- CampaignDn
- Phone
- AccountNumber
- MaxAttempts
- CallbackDateTime
- LastName
- FirstName

**Search and Sort Values**

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• agentId</li> <li>• accountNumber</li> <li>• firstName</li> <li>• lastName</li> </ul> | <ul style="list-style-type: none"> <li>• agentId</li> <li>• accountNumber</li> <li>• firstName</li> <li>• lastName</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

**Advanced search parameters**

There are several advanced searches you can perform on the Personal Callback API, including agentId, accountNumber, firstName, and lastName. All search terms are case-insensitive.

- **agentId:<ID>** finds all personal callback records where the agent ID contains a specified ID. For example, **agentId:123** returns all records where the agent ID contains the string "123".
- **lastName:<name>** finds all personal callback records where the last name contains the specified name. For example, **lastName:smith** returns all records where the last name includes "smith".
- **firstName:<name>** finds all personal callback records where the firstName contains the specified name. For example, **firstName:John** returns all records where the first name includes "john".

- **accountNumber:<number>** finds all personal callback records where the accountNumber contains the specified number. For example, **accountNumber:456** returns all records where the account number includes "456".

### Example Create Request (Single Record)

```
<personalCallback>
  <campaign>
    <refURL>/unifiedconfig/config/campaign/5000</refURL>
  </campaign>
  <agent>
    <refURL>/unifiedconfig/config/agent/5050</refURL>
  </agent>
  <campaignDn>22222</campaignDn>
  <phone>999333</phone>
  <accountNumber>23334343334</accountNumber>
  <maxAttempts>1</maxAttempts>
  <callbackDateTime>2016-01-15T11:37:00</callbackDateTime>
  <callStatus>pending</callStatus>
  <lastName>Kumar</lastName>
  <firstName>Akshaya</firstName>
</personalCallback>
```

### Example Create Request (Bulk)

```
<personalCallback>
  <fileContent>
    <![CDATA[
      AccountNumber,FirstName,LastName,Phone,AgentSkillTargetId,CampaignId,CampaignDn,
      CallbackDateTime,MaxAttempts
      6782, Henry, Martin, 2225554444, 1004, 5000, 222222221, 2016-01-25T05:12:00, 1
      3444, Thomas, Edison, 2225554555, 1004, 5000, 222222222, 2016-01-25T05:23:00, 1
      5444, Tom, Hilfiger, 2225554666, 1004, 5000, 222222223, 2016-01-25T05:37:00, 1
    ]]>
  </fileContent>
</personalCallback>
```

### Example Get (Record) Response

```
<personalCallback>
  <changeStamp>48</changeStamp>
  <refURL>/unifiedconfig/config/campaign/personalcallback/2</refURL>
  <campaign>
    <refURL>/unifiedconfig/config/campaign/5000</refURL>
  </campaign>
  <agent>
    <refURL>/unifiedconfig/config/agent/5050</refURL>
  </agent>
  <campaignId>5000</campaignId>
  <peripheralId>5000</peripheralId>
  <agentId>1001</agentId>
  <campaignDn>22222</campaignDn>
  <phone>999333</phone>
  <accountNumber>23334343334</accountNumber>
  <maxAttempts>1</maxAttempts>
  <callbackDateTime>2016-01-15T11:37:00</callbackDateTime>
  <callStatus>pending</callStatus>
  <callResult>0</callResult>
  <lastName>Kumar</lastName>
  <firstName>Akshaya</firstName>
</personalCallback>
```

# Time Zone API

Use the Time Zone API to list all available time zones and to get time zone information for a specified zone. Time zone information is stored in the registry of the Windows operating system.

**Important**

Microsoft periodically releases cumulative time zone updates. These updates include worldwide changes to time zone names, bias (the amount of time in minutes that a time zone is offset from Coordinated Universal Time (UTC)), and observance of daylight saving time. These patches update the information in the Windows registry. When these updates are available, apply them to all virtual machines in the deployment that are running a Microsoft Windows operating system.

Use this API with the Outbound Campaign API to set the default time zone for an Outbound Option campaign. An Outbound Option campaign uses the time zone when the location of the customer number being dialed is unknown.

This API is read-only.

**URL**

`https://<server>/unifiedconfig/config/timezone`

**Operations**

- **list**: Retrieves a list of available time zones. The list is sorted by UTC offset from the International Date Line from west to east.
- **get**: Returns information for a specific time zone using the URL `https://<server>/unifiedconfig/config/timezone/<id>`, where `<id>` is the URL-encoded version of the name parameter.

**Response Parameters**

- **name**: The name of the time zone.
- **displayName**: Specific bias and location information about the time zone, such as the offset from UTC and one or more places located within the time zone.  
Example: "(UTC+5:30) Chennai, Kolkata, Mumbai, New Delhi"
- **stdName**: The time zone name during standard time.  
Example: Malay Peninsula Standard Time
- **dstName**: The time zone name during daylight saving time.  
Example: Malay Peninsula Daylight Time
- **dstObserved**: Indicates whether daylight saving time is observed for the time zone. True or false.
- **bias**: The current bias for local time translation on the server (in minutes). That is, the number of minutes to add to the local time to yield UTC.

### Example Get Response

```
<timeZone>
  <refURL>
    /unifiedconfig/config/timezone/Pacific%20Standard%20Time%20%28Mexico%29
  </refURL>
  <name>Pacific Standard Time (Mexico)</name>
  <displayName>(UTC-08:00) Baja California</displayName>
  <stdName>Pacific Standard Time (Mexico)</stdName>
  <dstName>Pacific Daylight Time (Mexico)</dstName>
  <dstObserved>true</dstObserved>
  <bias>480</bias>
</timeZone>
```



## CHAPTER 31

# Peripheral Gateway API

Use the Peripheral Gateway (PG) API to retrieve peripheral gateway information.

### URL

`https://<server>/unifiedconfig/config/peripheralgateway`

### Operations

- [list](#): Retrieves a list of peripheral gateways.
- [get](#): Returns one peripheral gateway using the URL  
`https://<server>/unifiedconfig/config/peripheralgateway/<id>`.

### Parameters

- `refURL`: The refURL of the peripheral gateway. See [Shared Parameters](#), on page 9.
- `name`: The name of the peripheral gateway. See [Shared Parameters](#), on page 9.
- `logicalControllerId`: The ID of the logical controller.
- `peripherals`: A collection of peripheral information, including client type, name, peripheralID, routingClientID, and routingType (see [Dialed Number API](#), on page 59 for routingType values).
  - The client type values are:
    - 13: VRU
    - 30: CUCM
    - 42: Generic PG
    - 47: MediaRouting

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search Parameters | Sort Parameters  |
|-------------------|--|
| name              | <ul style="list-style-type: none"> <li>name (Default)</li> </ul> |

See [Search, on page 11](#) and [Sort, on page 12](#).

### Advanced search parameters

The Peripheral Gateway API has an advanced search for datacenters.

### Example Get Response

```
<peripheralGateway xsi:type="peripheralGateway">
  <changeStamp>0</changeStamp>
  <refURL>/unifiedconfig/config/peripheralgateway/5001</refURL>
  <name>MR_PG</name>
  <logicalControllerID>5001</logicalControllerID>

  <peripherals>
    <peripheral>
      <changeStamp>824</changeStamp>
      <clientType>47</clientType>
      <name>Multichannel</name>
      <peripheralID>5005</peripheralID>
      <routingClientID>5005</routingClientID>
      <routingType>3</routingType>
    </peripheral>
    <peripheral>
      <changeStamp>822</changeStamp>
      <clientType>47</clientType>
      <name>Outbound</name>
      <peripheralID>5007</peripheralID>
      <routingClientID>5007</routingClientID>
      <routingType>4</routingType>
    </peripheral>
  </peripherals>
</peripheralGateway>
```





## Precision Queue API

Precision queues help direct incoming callers to appropriate agents, as they match specific agent attributes with caller requirements. If a precision queue requires an agent who lives in Boston and who speaks fluent Spanish, then an agent with the attributes **Boston = True** and **Spanish = True** is a good match.

Use the Precision Queue API to list the precision queues currently defined in the database, define new precision queues, and view, edit, and delete existing precision queues.

### URL

`https://<server>/unifiedconfig/config/precisionqueue`

### Operations

- **create**: Creates one precision queue.
- **delete**: Marks one precision queue for deletion, but does not permanently delete it. Deleting a precision queue that is referenced dynamically in a script is allowed. No new calls are queued against it, but the precision queue remains operational until calls are no longer in the queue.
- **get**: Returns one precision queue, using the URL  
`https://<server>/unifiedconfig/config/precisionqueue/<id>`.
  - **Query parameters**:
    - **agentcount**: Use this query parameter to have the agent count parameter included in the response; for example, `agentcount=true`.
    - **attributes**: Use this query parameter to have the attribute parameter included in the response; for example, `attributes=true`.
    - **skillgroups**: Use this query parameter to augment the returned precision queue attributes with an id listing of all of the skillgroups that are associated with the precision queue; for example, `skillgroups=true`.
- **list**: Retrieves a list of precision queues. Query parameters described above for the get operation are also allowed for list.
- **update**: Updates one precision queue.

## Parameters

### Precision queue parameters:

- refURL: The refURL of the precision queue. See [Shared Parameters, on page 9](#).
- name: The name of the precision queue. See [Shared Parameters, on page 9](#).
- changeStamp: See [Shared Parameters, on page 9](#).
- description: See [Shared Parameters, on page 9](#).
- department: A reference to the department ([Department API, on page 53](#)), including the refURL and name. See [References, on page 6](#).
- mediaRoutingDomain: A reference to the media routing domain ([Media Routing Domain API, on page 89](#)) including the name and refURL. See [References, on page 6](#).
  - Defaults to Cisco\_Voice MRD if this parameter is not provided.
  - This reference cannot be updated.
- bucketInterval: A reference to a bucket interval ([Bucket Interval API, on page 39](#)), including the refURL and name. See [References, on page 6](#).
- agentCount: Returns agent count for the precision queue. Returned only when using the agentcount query parameter.
- agentOrdering: Determines the order in which agents receive calls from this queue.
  - 1: LAA (Agent availability time)
  - 2: Most skilled agent
  - 3: Least skilled agent
- id: The database id of the precision queue. Read-only field. Used in scripting.
- attributes: A collection of attribute names (attribute1, attribute2, and so on) indicating all of the attributes used in this precision queue. Returned only when the query parameter attributes=true.
- serviceLevelThreshold: Maximum time in seconds that a caller should wait before being connected with an agent.
- serviceLevelType: This value indicates how the system calculates the service level.
  - 1: Ignore abandoned calls.
  - 2: Abandoned call has negative impact.
  - 3: Abandoned call has positive impact.
- skillgroups: A collection of skill groups associated with this precision queue, including the id of each skill group. Returned only when the query parameter skillgroups=true.
- steps: Required. A collection of steps for this precision queue. You can have 1-10 steps. Returned only for get operation. See the Step parameters below.

### Step parameters:

- waitTime: Time in seconds to wait before proceeding to the next step.

- **considerIf:** A Consider If expression which must be met to execute a particular step. Items used in the expression are case sensitive. You cannot add an expression to the last step.
- **terms:** Required. A collection of terms for this step. Each step can have 1-10 terms. See the Term parameters below.

#### Term parameters:

- **attribute:** A reference to the attribute ([Attribute API, on page 35](#)), including the refURL, name, description, and dataType. Multiple unique attributes can be used across all terms in a precision queue.
- **parenCount:** Denotes a parenthesis before or after this term. A value of 1 means a parenthesis before the current term, and a value of -1 means a parenthesis after the current term. The sum of all parenCount for all terms in a step must be equal to zero, meaning that all parenthesis in the expression are matched. For example, a step to check for agents that have (sales > 7 or expertSales = true) and english = true requires 3 terms with the parenCount set to 1 on the first term, -1 on the second term, and 0 on the last term.
- **termRelation:** Indicates the relationship of this term to the preceding term, using the following values:
  - 0: None. Valid only on the first term in a step.
  - 1: AND
  - 2: OR
- **attributeRelation:** Indicates what kind of comparison is done on the attribute, using the following values:
  - 1: Equal
  - 2: Not equal
  - 3: Less than
  - 4: Less than or equal
  - 5: Greater than
  - 6: Greater than or equal
- **value1:** The value that the attribute is tested against. For boolean attributes, this value must be true/false. For proficiency attributes, this value must be 1-10.

#### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> </ul> |

See [Search, on page 11](#) and [Sort, on page 12](#).

## Example Get Response

```

<precisionQueue>
  <changeStamp>4</changeStamp>
  <refURL>/unifiedconfig/config/precisionqueue/5002</refURL>
  <agentOrdering>1</agentOrdering>
  <bucketInterval>
    <refURL>/unifiedconfig/config/bucketinterval/1</refURL>
    <name>Default_Bucket_Intervals</name>
  </bucketInterval>
  <mediaRoutingDomain>
    <name>Cisco_Voice</name>
    <refURL>/unifiedconfig/config/mediaroutingdomain/1</refURL>
  </mediaRoutingDomain>
  <description>This is a practice precision queue</description>
  <name>Practice_Queue</name>
  <serviceLevelThreshold>3</serviceLevelThreshold>
  <serviceLevelType>1</serviceLevelType>
  <steps>
    <step>
      <terms>
        <term>
          <attribute>
            <refURL>/unifiedconfig/config/attribute/5698</refURL>
            <name>test</name>
            <dataType>4</dataType>
          </attribute>
          <attributeRelation>5</attributeRelation>
          <parenCount>0</parenCount>
          <termRelation>0</termRelation>
          <value1>2</value1>
        </term>
      </terms>
      <waitTime>0</waitTime>
    </step>
  </steps>
</precisionQueue>

```



## Reason Code API

Agents enter reason codes on their agent desktops when they go Not Ready. Reason codes appear in Unified Intelligence Center reports and help identify agent behavior.

Use the Reason Code API to define new reason codes, and edit and delete records of existing reason codes.

### URL

`https://<server>/unifiedconfig/config/reasoncode`

### Operations

- **create**: Creates one reason code.
- **delete**: Marks one reason code for deletion.
- **get**: Returns one reason code, using the URL  
`https://<server>/unifiedconfig/config/reasoncode/<id>`.
- **list**: Retrieves a list of reason codes.
- **update**: Updates one reason code.

### Parameters

- **refURL**: The refURL of the reason code. See [Shared Parameters, on page 9](#).
- **changestamp**: See [Shared Parameters, on page 9](#).
- **description**: See [Shared Parameters, on page 9](#).
- **code**: Required. The unique reason code. Integers between 0 and 65535. Value cannot be updated after the reason code has been created.
- **text**: Required. The text that describes the reason code. Maximum length of 40 characters allowed.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters  | Sort parameters   |
|--|---|
| <ul style="list-style-type: none"><li>• text</li><li>• description</li></ul> | <ul style="list-style-type: none"><li>• text (default)</li><li>• description</li><li>• code</li></ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

### Example Get Response

```
<reasonCode>
  <changeStamp>0</changeStamp>
  <refURL>/unifiedconfig/config/reasoncode/5001</refURL>
  <code>1</code>
  <description>This is a reason code.</description>
  <text>Reason Code1</text>
</reasonCode>
```



## Role API

Roles specify the APIs that an administrator can use, and which menus and tools an administrator can see and use.

Use the Role API to list the roles currently defined in the database, or edit, delete, or create new custom roles.

### Roles

Cisco Packaged CCE database has four built-in roles for administrators that set the access to specific features (APIs and tools). The built-in roles are hierarchical in that each role identified in the table below contains the access rights of the roles above it. You cannot alter the feature access for the built-in roles, but you can create additional roles to define customized sets of feature access.

| This role   | Allows full access to   |
|-------------|---|
| AgentAdmin  | Agent APIs and tools only.  |
| ScriptAdmin | Agent, Call, and Scripting tools - including Bulk Jobs.   |
| ConfigAdmin | All APIs and tools except administrator, departments, and roles.  |
| SystemAdmin | All APIs and tools. Only this role provides access to the Administrator, Department, and Role features. |

For more information on these roles, see the *Cisco Packaged Contact Center Enterprise Administration and Configuration Guide* at <http://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-maintenance-guides-list.html>.

### URL

`https://<server>/unifiedconfig/config/role`

### Operations

- **create:** Creates one role, given the specified name, description, and featureList.
- **delete:** Deletes one role permanently from the database.

- **get**: Returns one role using the URL  
`https://<server>/unifiedconfig/config/role/<id>`.
  - Auxiliary get method returns the accessList parameter containing a collection of all features that can be used to define a role. Use the URL  
`https://<server>/unifiedconfig/config/role/available_features`.
- **list**: Retrieves a list of roles.
- **update**: Gets the role item and updates the new configuration entered in the <accessList> fields.

### Parameters

- refURL: The refURL of the role. See [Shared Parameters, on page 9](#).
- name: The role name, such as ConfigAdmin. See [Shared Parameters, on page 9](#).
- changeStamp: See [Shared Parameters, on page 9](#).
- description: See [Shared Parameters, on page 9](#).
- accessList: A collection of features that indicates which APIs and tools the administrators assigned to this role can access.
  - feature: Element of the access list that represents access to a feature (API / tool). Includes the following parameters:
    - name: Feature name. Maps to a valid API / tool.
    - category: Feature category, such as Agent, Call, System, and Access. This parameter is used only for get and list operations.
- administrators: A collection of administrator references ([Administrator API, on page 17](#)), including the user name, domain name, and refURL of the administrator. See [References, on page 6](#).
- systemDefined: Read-only parameter. Indicates if this role is a system-defined role. Values are true/false.

### Search and Sort Values

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters   |
|---|---|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> </ul> |

See [Search, on page 11](#) and [Sort, on page 12](#).

### Advanced search parameters

You can perform a system defined role search on the Role API:



- **systemDefined:(both/only/none)**: Allows searching of roles with the specified value of the systemDefined parameter.
  - **q=systemDefined:only** Returns roles with the systemDefined parameter set to true.
  - **q=systemDefined:none** Returns roles with the systemDefined parameter set to false.
  - **q=systemDefined:both** Returns roles with the systemDefined parameter set to true or false.

### Example Get Response

```
<role>
  <changeStamp>15</changeStamp>
  <refURL>/unifiedconfig/config/role/5003</refURL>
  <administrators>
    <administrator>
      <refURL>/unifiedconfig/config/administrator/5002</refURL>
      <domainName>BOSTON.COM</domainName>
      <userName>userAttribute</userName>
    </administrator>
  </administrators>
  <description>testAttribute</description>
  <accessList>
    <feature>
      <category>agent</category>
      <name>attribute</name>
    </feature>
    <feature>
      <category>agent</category>
      <name>reasoncode</name>
    </feature>
    <feature>
      <category>call</category>
      <name>bucketinterval</name>
    </feature>
  </accessList>
  <name>testAttribute</name>
  <systemDefined>false</systemDefined>
</role>
```





## Routing Type API

Use the Routing Type API to retrieve the list of routing types in the solution, including whether each routing type allows dialed number creation ([Dialed Number API](#), on page 59). This API is read-only.

### URL

`https://<server>/unifiedconfig/config/routingtype`

### Operations

- [list](#): Returns routing type information.

### Parameters

- **type**: Indicates the value of this routing type (see [Dialed Number API](#), on page 59).
- **state**: Indicates the state of the routing type:
  - **UNKNOWN**: Indicates that the system cannot determine the state of the routing type due to inventory or status errors.
  - **UNUSED**: Indicates that the routing type has not been used in Peripheral Gateway Setup.
  - **MISCONFIGURED**: Indicates that the routing type is used in Peripheral Gateway Setup, but either the Application Host Name associated with this routing type has not yet been added to the inventory, or there are status rule violations for the configuration.
  - **OK**: The routing type is configured correctly and dialed numbers can be created on this routing type (see [Dialed Number API](#), on page 59).
- **machineType**: For routing types 4, 5, and 6, indicates the type of machine in the inventory that is used for this routing type. Potential machine types include `EXTERNAL_SOCIAL_MINER`, `EXTERNAL_ECE`, and `EXTERNAL_THIRD_PARTY_MULTICHANNEL`.

### Example Get Response

```
<results>
  <routingTypes>
    <routingType>
      <type>1</type>
      <state>OK</state>
```

```
</routingType>
<routingType>
  <type>2</type>
  <state>OK</state>
</routingType>
<routingType>
  <type>3</type>
  <state>OK</state>
</routingType>
<routingType>
  <type>4</type>
  <machineType>EXTERNAL_ECE</machineType>
  <state>OK</state>
</routingType>
<routingType>
  <type>5</type>
  <state>MISCONFIGURED</state>
</routingType>
<routingType>
  <type>6</type>
  <state>UNUSED</state>
</routingType>
</routingTypes>

</results>
```



## Scan API

---

The Scan API forces an update of multiple types of system checks, including:

- Machine Inventory: Validate the VM Host configuration, VM layout, and required inventory configuration.
- Status: Validate the status rules. See [Status API](#), on page 157.

### URL

`https://<server>/unifiedconfig/config/status/scan`

### HTTP method

PUT: Invoke the PUT method to force a scan to start immediately.





## Serviceability API

---

Use the Serviceability API to view information about the system, such as API statistics and version information.

### URL

`https://<server>/unifiedconfig/config/serviceability`

### Operations

- [get](#): Returns serviceability information.
  - **Query parameters:**
    - **category**: Use this query parameter to reduce the number of parameters returned. The values allowed match the names of the parameters. For example,  
`https://<server>/unifiedconfig/config/serviceability?category=systemValidationStatus&category=capacityInfo`.

### Parameters

- **currentTime**: The time at which this web request was made.
- **instanceName**: The name of the active Unified CCE instance.
- **version**: Version information for Packaged CCE. Includes the following parameters:
  - **buildDate**: The date the application was built.
  - **buildVersion**: The build number of the application.
  - **esVersion**: The engineering special (ES) version.
  - **maintenanceVersion**: The maintenance version.
  - **majorVersion**: The major version.
  - **minorVersion**: The minor version.
  - **srVersion**: The SR version.

- **ucceVersion**: Version information for Unified CCE. Includes the same parameters listed for the version parameter, above, as well as:
  - **versionString**: Textual representation of the Unified CCE version.
  - **patchInfos**: A collection of patch information, including **majorVersion**, **minorVersion**, **maintenanceVersion**, **srVersion**, and **esVersion** parameters.
- **capacityInfo**: A collection of **capacityRules** indicating if the capacity limits are valid. Each rule contains the following parameters:
  - **name**: The name of the capacity rule.
  - **max**: The maximum number of items allowed for the rule.
  - **actual**: The current number of items configured for the rule.
- **systemValidationStatus**: A collection of **validationRules** that show the potential errors regarding system configuration. For more information on the rules, see [System Validation Rules](#), on page 139. Each rule contains the following parameters:
  - **name**: The name of the rule.
  - **isValid**: Indicates if the rule is passing. Values are true/false.
  - **min**: The minimum number of items required to match for this rule.
  - **max**: The maximum number of items required to match for this rule.
  - **actual**: The current number of items configured that match this rule.

### Example Get Response

```
<Serviceability>
  <currentTime>Tue Nov 29 04:00:45 EST 2011</currentTime>
  <instanceName>instance</instanceName>
  <version>
    <majorVersion>9</majorVersion>
    <minorVersion>0</minorVersion>
    <maintenanceVersion>0</maintenanceVersion>
    <srVersion>0</srVersion>
    <esVersion>0</esVersion>
    <buildVersion>1</buildVersion>
    <buildDate>1969-12-31T19:00:00-05:00</buildDate>
  </version>
  <ucceVersion>
    <majorVersion>9</majorVersion>
    <minorVersion>5</minorVersion>
    <maintenanceVersion>3</maintenanceVersion>
    <srVersion>0</srVersion>
    <esVersion>0</esVersion>
    <buildVersion>375</buildVersion>
    <versionString>9.5.3.0.0.375</versionString>
    <patchInfos>
      <patchInfo>
        <majorVersion>9</majorVersion>
        <minorVersion>5</minorVersion>
        <maintenanceVersion>3</maintenanceVersion>
        <srVersion>0</srVersion>
        <esVersion>0</esVersion>
      </patchInfo>
      <patchInfo>
        <majorVersion>9</majorVersion>
        <minorVersion>5</minorVersion>
      </patchInfo>
    </patchInfos>
  </ucceVersion>
</Serviceability>
```



```

        <maintenanceVersion>2</maintenanceVersion>
        <srVersion>0</srVersion>
        <esVersion>0</esVersion>
    </patchInfo>
</patchInfos>
</ucceVersion>
</Serviceability>

```

- [System Validation Rules, page 139](#)

## System Validation Rules

The system validation rules show the potential errors regarding system configuration.

| Rule                                       | Explanation  |
|--|--|
| ECC_VARIABLES_CTI_SIZE                     | ECC Variables: Total bytes required for enabled variables in CTI Server must not exceed 2500.                                      |
| CMS_NODE_DISABLED                          | CMS Node: Configuration Management Service (CMS) Node and Agent Re-skilling Web Tool must be disabled using Unified CCE Web Setup. |
| ENTERPRISE_SERVICE_COUNT                   | Enterprise Services: No Enterprise Services may be configured.   |
| MR_PIM_COUNT                               | Peripheral: Exactly 2 MR Peripherals must be configured on the MR PG in the PG Explorer tool.                                      |
| DESK_SETTING_WITH_RING_NO_ANSWER_SET_COUNT | Agent Desk Settings: Ring No Answer Timer must not be set.   |
| ENT_SG_COUNT                               | Skill Groups: A maximum of 1 Enterprise Skill Group can be configured.   |
| ENT_SG_MEMBER_COUNT                        | Skill Groups: No Enterprise Skill Group Members may be configured.   |
| ENT_ROUTE_COUNT                            | Enterprise Routes: A maximum of 1 Enterprise Route must be configured.   |
| ENT_ROUTE_MEMBER_COUNT                     | Enterprise Routes: No Enterprise Route Members may be configured.  |
| PCCE_CUCM_PG_COUNT                         | Peripheral Gateway: Exactly 1 Unified CM Peripheral Gateway must be configured.  |
| VRU_PG_COUNT                               | Peripheral Gateway: Exactly 1 VRU Peripheral Gateway must be configured.   |
| MR_PG_COUNT                                | Peripheral Gateway: Exactly 1 Media Routing Peripheral Gateway must be configured.   |

| Rule                             | Explanation   |
|----------------------------------|---|
| MULTICHANNEL_COUNT               | Peripheral: Exactly 1 MR PIM must be configured with the Enterprise Name of <b>Multichannel</b> .                                     |
| OUTBOUND_COUNT                   | Peripheral: Exactly 1 MR PIM must be configured with the Enterprise Name of <b>Outbound</b> .   |
| PG_COUNT                         | Peripheral Gateway: Exactly 3 Peripheral Gateways must be configured.   |
| SERVICE_MEMBER_COUNT             | Service Members: No Service Members may be configured.  |
| TYPE10_NETWORK_VRU_COUNT         | VRU: Exactly 1 Type 10 Network VRU must be configured in the Network VRU Explorer.  |
| TYPE10_NETWORK_VRU_MAP_COUNT     | Peripheral Gateway: Both VRU Peripherals must be configured on the VRU PG and associated with the Type 10 Network VRU.                |
| UCM_PIM_COUNT                    | Peripheral: Exactly 1 Unified CM Peripheral must be configured on the Unified CM PG in the Peripheral Explorer tool.                  |
| VRU_PIM_COUNT                    | Peripheral: Exactly 2 VRU Peripherals must be configured.   |
| NOT_SKILL_GROUP_ROUTE_NAME_COUNT | Skill Groups: All Skill Group records must have a corresponding Route record with the same Enterprise Name as the Skill Group record. |
| ECC_VARIABLES_ENABLED_COUNT      | ECC Variables: ECC variables must be enabled in the System Information tool.  |
| SERVICE_COUNT                    | Services: No Services may be configured.  |
| TRANSLATION_ROUTE_COUNT          | Translation Routes: No Translation Routes may be configured.  |
| NIC_COUNT                        | NICs: No NICs may be configured.  |
| MRD_COUNT                        | Max Media Routing Domains: The maximum number of Media Routing Domains is 20.   |
| MEDIA_CLASS_COUNT                | Max Media Classes: The maximum number of Media Classes is 10.   |
| NOT_PARTITIONED_COUNT            | Partitioning: Partitioning must be disabled in the System Information tool.   |

| Rule                                 | Explanation   |
|--------------------------------------|---|
| NON_NULL_SERVICE_LEVEL_COUNT         | Service Level Threshold: The default service level must not be set in the Peripheral Explorer tool. The default is set in the System Information tool.                      |
| DEVICE_TARGET_COUNT                  | Device Targets: No Device Targets can be configured.  |
| CVP_LABEL_COUNT                      | VRU: Each VRU PIM associated with the VRU PG in the PG Explorer tool must have exactly 1 label with a length of 10 digits.  |
| CUCM_LABEL_COUNT                     | CUCM Routing Label: Exactly 1 label with length of 10 digits must be configured and associated with the Unified CM routing client.  |
| CORRELATION_ID_RANGE_COUNT           | Correlation ID: The minimum and maximum correlation number in the VRU section of the System Information tool must be 1001 and 9999 respectively.                            |
| NULL_FEATURE_SET_ID_COUNT            | Feature Control Set: The Feature Control Set in the Customer Definition of the ICM Instance Explorer must set to NONE.  |
| ECC_FOR_CVP_COUNT                    | ECC Variables: Exactly 9 Expanded Call Variables are required for CVP.  |
| NETWORK_VRU_SCRIPT_COUNT             | VRU: There must be a Network VRU Script with the Enterprise Name of <b>VXML_Server</b> and the Script Name of <b>GS_Server_V</b> configured in the Network VRU Script tool. |
| DEFAULT_DESK_SETTING_COUNT           | Agent Desk Settings: Default_Agent_Desk_Setting must be set as the default Agent Desk Settings for the CUCM PIM in the PG Explorer tool.                                    |
| PCCE_APP_INSTANCE_MULTICHANNEL_COUNT | Multichannel Application Instance: An Application Instance must be defined for Multichannel.  |
| TYPE2_NETWORK_VRU_COUNT              | VRU: Exactly one Type 2 Network VRU must be configured in the Network VRU Explorer tool.  |
| TYPE2_NETWORK_VRU_MAP_COUNT          | Peripheral: Both MR PIMs must be associated with a Type 2 Network VRU in the PG Explorer tool.  |
| DIALED_NUMBER_EXTERNAL_VOICE_COUNT   | Dialed Numbers: For each External Voice Dialed Numbers, there must be exactly 2 Dialed Number records for each Dialed Number String, with one for each VRU PIM.             |

| Rule                                       | Explanation   |
|--|---|
| DIALED_NUMBER_MAP_COUNT                    | Dialed Numbers: All Dialed Number records must not have an associated Region, ANI, and must have a maximum of 1 Call Type associated in the Call Type Map.  |
| AGENT_REAL_TIME_ENABLED_COUNT              | Peripheral: Agent Reporting must be enabled on the Unified CM Peripheral in the PG Explorer tool.   |
| CUSTOMER_DEFINITION_COUNT                  | Customer Definition: Exactly 1 Customer Definition must be configured in the ICM Instance Explorer.   |
| CUSTOMER_DEFINITION_HAS_TYPE10_NETWORK_VRU | Customer Definition: Exactly 1 Customer Definition must have a Type 10 Network VRU selected in the ICM Instance Explorer.   |
| DIALED_NUMBERS_REQUIRE_CUSTOMER_DEFINITION | Dialed Numbers: No Dialed Number records can have the Customer set to None.   |
| SCRIPT_VERSIONS_TO_RETAIN                  | Script Versions to Retain: The number of script versions to retain must be between 1 and 100, inclusively.  |
| APP_GATEWAY_COUNT                          | Application Gateway: No Application Gateways may be configured.   |
| DATABASE_LOOKUP_COUNT                      | Database Lookups: No Database Lookups may be configured.  |
| SCRIPT_VERSIONS_ALLOWED                    | Max Script Versions: The maximum number of script versions allowed is 100 (minimum is 1).   |
| OEM_CP_MATCHES_DB_COLLATION                | The system locale must be compatible with the database collation. The valid pairs are: cp437 / iso_1 (US English) and cp850 / iso_1 (Other Latin derivatives). All other valid pairs must be identical strings: cp936 / cp936 (Chinese), cp866/cp866 (Russian), cp 932/cp932 (Japanese), and so on. For more information, see the Collation and Locale Settings for Localization section in the <i>Cisco Packaged Contact Center Enterprise Installation and Upgrade Guide</i> in the <a href="http://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-installation-guides-list.html">http://www.cisco.com/c/en/us/support/customer-collaboration/packaged-contact-center-enterprise/products-installation-guides-list.html</a> . |
| AGENT_TARGETING_RULE                       | There must be at exactly one agent targeting rule defined in CCE.   |



## Single Sign-On Global State API

Use the Single Sign-On (SSO) Global State API to view or update the global status of SSO.

To retrieve the overall status of setting the SSO state or the status for a single component, see the [Single Sign-On Status API](#), on page 147.

### URL

`https://<server>/unifiedconfig/config/sso/globalstate`

### Operations

- [get](#): Returns the current global state of SSO in the database.
- [update](#): Updates the global state of SSO in the database.

### Parameters

- refURL: The RefURL. See [Shared Parameters](#), on page 9.
- changeStamp: See [Shared Parameters](#), on page 9.
- permissionInfo: Information about permissions.
  - canUpdate: Whether ssoState can be updated. True or false.
  - role: The role of the user.
- state: Required for update. Valid values are NON\_SSO (SSO is disabled for all users), SSO (SSO is enabled for all users), and HYBRID (mix of enabled and disabled).

### Example Get Response

```
<ssoState>
  <refURL>/ssostate</refURL>
  <changeStamp>227</changeStamp>
  <permissionInfo>
    <canUpdate>true</canUpdate>
    <role>Administrator</role>
  </permissionInfo>
  <state>SSO</state>
</ssoState>
```





## Single Sign-On Registration API

---

Use the Single Sign-On (SSO) Registration API to register SSO-compatible components with the Cisco Identity Service. These components include AW, Finesse, and Unified Intelligence Center machines.

To retrieve the overall registration status or the status for a single component, see the [Single Sign-On Status API](#), on page 147.

### URL

`https://<server>/unifiedconfig/config/sso/register`

### Operations

- **update:** Registers all SSO-compatible components in the Machine Inventory with the Cisco Identity Service, using the URL `https://<server>/unifiedconfig/config/sso/register`. (See [Machine Inventory API](#), on page 81)

### Parameters

None







## Single Sign-On Status API

Use the Single Sign-On (SSO) Registration API to get the current status of registering components with the Cisco Identity Service (IdS) and setting SSO state on the SSO-compatible components. These components include AW, Finesse, and Unified Intelligence Center machines.

### URL

`https://<server>/unifiedconfig/config/sso/status`

### Operations

- **get** component status: Returns the status of registering a specific component with the Cisco IdS and setting SSO state, using the URL `https://<server>/unifiedconfig/config/sso/status/<id>`. The `<id>` is the `machine_id` of the component.
- **list**: Retrieves a list with the overall and individual component statuses of registering SSO-compatible components with the Cisco Identity Service and setting SSO state, using the URL `https://<server>/unifiedconfig/config/sso/status`.

### Parameters

- **globalSsoState**: The current SSO state as set in the AW database. The values are NON\_SSO (SSO is disabled for all users), SSO (SSO is enabled for all users), and HYBRID (mix of enabled and disabled).
- **registrationState**: The overall status of registering components with the Cisco IdS. Values are the following:
  - **SUCCEEDED**: All of the components were successfully registered.
  - **FAILED**: Registration failed on one or more components. Error detail is set on each failed component.
  - **PROCESSING**: Registration started and is not complete.
  - **NOT\_STARTED**: Registration has not started.
- **modeState**: The overall status of registering components with the Cisco IdS. Values are the following:
  - **SUCCEEDED**: The SSO state was successfully set on all of the components.

- FAILED: The SSO state failed to be set on one or more components. Error detail is set on each failed component.
- PROCESSING: The SSO state change has started and is not complete.
- NOT\_STARTED: The SSO state change has not started.
- idSConfigurationState: Whether the Cisco IdS has been configured and is in service.
  - STATE\_NOT\_CONFIGURED : The Cisco IdS is not been configured.
  - STATE\_IN\_SERVICE : The Cisco IdS is configured and is in service.
  - STATE\_OUT\_OF\_SERVICE : The Cisco IdS is configured and is out of service.
  - STATE\_PARTIAL\_SERVICE: The Cisco IdS is configured and is partially in service.
  - STATE\_UNREACHABLE: The Cisco IdS cannot be reached.
- hasIdsCredentials: Whether the Machine Inventory has the necessary IdS credentials to register components (see [Machine Inventory API](#), on page 81. Values are true or false. The default is false.
- idsBaseUrl: The base URL for accessing the Identity Service.
- ssoComponentStatuses: A collection of registration and SSO state status information for all of the individual components. Returned on a list operation.
- ssoComponentStatus: Registration and SSO state status information for an individual component. Includes the following parameters:
  - registrationState: The status of registering the component with the IdS. See the values for this parameter above.
  - modeState: The status of setting the SSO state for the component. See the values for this parameter above.
  - refURL: The refURL for the component machine. See [Shared Parameters](#), on page 9.
  - name: The name of the component machine.

### Example Get Response

Example URL: `https://<server>/unifiedconfig/config/sso/status/21`

```
<ssoComponentStatus>
  <registrationState>FAILED</registrationState>
  <modeState>NOT_STARTED</modeState>
  <refURL>/unifiedconfig/config/machineinventory/21</refURL>
  <name>FINESSE-A.boston.com</name>
</ssoComponentStatus>
```

### Example List Response

```
<ssoStatus>
  <globalSsoState>HYBRID</globalSsoState>
  <registrationState>FAILED</registrationState>
  <modeState>NOT_STARTED</modeState>
  <idSConfigurationState>STATE_IN_SERVICE</idSConfigurationState>
  <hasIdsCredentials>true</hasIdsCredentials>
  <idsBaseUrl>https://<server>:<serverport></idsBaseUrl>
  <ssoComponentStatuses>
```

```
<ssoComponentStatus>
  <registrationState>FAILED</registrationState>
  <modeState>NOT_STARTED</modeState>
  <refURL>/unifiedconfig/config/machineinventory/21</refURL>
  <name>FINESSE-A.boston.com</name>
</ssoComponentStatus>
<ssoComponentStatus>
  <registrationState>FAILED</registrationState>
  <modeState>NOT_STARTED</modeState>
  <refURL>/unifiedconfig/config/machineinventory/22</refURL>
  <name>FINESSE-B.boston.com</name>
</ssoComponentStatus>
<ssoComponentStatus>
  <registrationState>FAILED</registrationState>
  <modeState>NOT_STARTED</modeState>
  <refURL>/unifiedconfig/config/machineinventory/23</refURL>
  <name>CUIC-A.boston.com</name>
</ssoComponentStatus>
</ssoComponentStatuses>
</ssoStatus>
```





## Skill Group API

A skill group is a collection of agents who share a common set of competencies that equip them to handle the same types of requests. Some examples of skill groups are a collection of agents who speak a specific language or who can assist callers with billing questions.

Use the Skill Group API to list the skill groups currently defined in the database, define new skill groups, and view, edit, or delete existing skill groups.



### Note

Access to this API is different for supervisors and administrators. For more information, see [Access](#), on [page 5](#).

### URL

`https://<server>/unifiedconfig/config/skillgroup`

### Operations

- **create**: Creates one skill group.
- **delete**: Marks one skill group for deletion, but does not permanently delete it.
- **get**: Returns one skill group, using the URL  
`https://<server>/unifiedconfig/config/skillgroup/<id>`.
- **list**: Retrieves a list of skill groups.
  - **Query Parameters:**
    - **selectedAgentCount**: Use this query parameter to augment skill group information about multiple agents. The **selectedAgentCount** parameter shows the number of specified agents belonging to that skill group. For example, to find out how many of agents 5000, 5001, 5002, and 5003 belong to each of the skill groups in the list, add **selectedAgentCount=5000,5001,5002,5003**.



### Note

Using **selectedAgentCount** automatically sets the summary list query parameter to **true**.

◦ Summary list: See [list](#), on page 4.

- [update](#): Updates one skill group.

## Parameters

- `refURL`: The refURL of the skill group. See [Shared Parameters](#), on page 9.
- `name`: The name of the skill group. See [Shared Parameters](#), on page 9.
- `department`: A reference to the department ([Department API](#), on page 53), including the name and refURL. See [References](#), on page 6.
- `changeStamp`: See [Shared Parameters](#), on page 9.
- `description`: See [Shared Parameters](#), on page 9.
- `mediaRoutingDomain`: A reference to the media routing domain ([Media Routing Domain API](#), on page 89) including the name and refURL. See [References](#), on page 6.
  - Defaults to `Cisco_Voice MRD` if this parameter is not provided.
  - This reference cannot be updated.
- `agents`: A collection of agents assigned to the skill group (See [Agent API](#), on page 19). References also include `firstName`, `lastName`, `agentId`, and `agentTeam` (which includes the team name and refURL). See [References](#), on page 6.
  - `canRemove`: This parameter only appears for supervisors. It indicates whether or not the supervisor has permission to remove the agent from this skill group. The supervisor can remove the agent from the skill group if the agent belongs to a team of this supervisor.
- `agentsAdded`: A collection of agent references to be added to the skill group, including the refURL of each agent to be added. This parameter is update only, and cannot be used in conjunction with the `agents` parameter. This parameter can be used with the `agentsRemoved` parameter. See [References](#), on page 6.
- `agentsRemoved`: A collection of agent references to be removed from the skill group, including the refURL of each agent to be removed. This parameter is update only, and cannot be used in conjunction with the `agents` parameter. This parameter can be used with the `agentsAdded` parameter. See [References](#), on page 6.
- `agentCount`: Read-only parameter containing the number of agents having the skill.
- `selectedAgentCount`: Read-only field. Indicates the number of specified agents belonging to the skill group. Returned only when using the `selectedAgentCount` query parameter.
- `bucketInterval`: A reference to the bucket interval ([Bucket Interval API](#), on page 39). Includes the name and refURL. See [References](#), on page 6.
- `serviceLevelThreshold`: Maximum time in seconds that a caller should wait before being connected with an agent. Positive integers only, or blank.  
Blank means use the value from the specified `mediaRoutingDomain`.
- `serviceLevelType`: This value indicates how the system calculates the service level.

- 1: Ignore Abandoned Calls (default).
  - 2: Abandoned Calls have Negative Impact.
  - 3: Abandoned Calls have Positive Impact.
- peripheralNumber: Read-only parameter. Automatically generated when using the create operation.

**Note**

A route record is maintained seamlessly by the Skill Group API; that is, a single route record is generated for each skill group created and the process is hidden from the user. The route records are updated and deleted via the Skill Group API.

**Search and Sort Values**

The following table shows the parameters that are searched and the parameters that are sortable.

| Search parameters   | Sort parameters  |
|---|--|
| <ul style="list-style-type: none"> <li>• name</li> <li>• description</li> </ul> | <ul style="list-style-type: none"> <li>• name (default)</li> <li>• description</li> <li>• serviceLevelThreshold</li> <li>• serviceLevelType</li> <li>• peripheralNumber</li> </ul> |

See [Search](#), on page 11 and [Sort](#), on page 12.

For more information on search restrictions, see [Search](#), on page 11.

**Example Get Response**

```
<skillGroup>
  <refURL>/unifiedconfig/config/skillgroup/(id)</refURL>
  <name>test</name>
  <description>test skill group</description>
  <changeStamp>0</changeStamp>

  <mediaRoutingDomain>
    <name>Cisco_Voice</name>
    <refURL>/unifiedconfig/config/mediaroutingdomain/1</refURL>
  </mediaRoutingDomain>
  <bucketInterval>
    <name>bucketIntervalName</name>
    <refURL>/unifiedconfig/config/bucketinterval/1</refURL>
  </bucketInterval>
  <serviceLevelThreshold>20</serviceLevelThreshold>
  <serviceLevelType>1</serviceLevelType>
  <peripheralNumber>1234567</peripheralNumber>
  <agents>
    <agent>
      <refURL>/unifiedconfig/config/agent/5000</refURL>
      <firstName>Jane</firstName>
      <lastName>Doe</lastName>
      <userName>username</userName>
    </agent>
  </agents>
</skillGroup>
```

```
<agentId>8007</agentId>
<canRemove>true</canRemove>
</agent>
<agent>
  <refURL>/unifiedconfig/config/agent/5001</refURL>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <userName>username2</userName>
  <agentId>8008</agentId>
  <agentTeam>
    <refURL>/unifiedconfig/config/agentteam/5000</refURL>
    <name>someTeam</name>
  </agentTeam>
  <canRemove>false</canRemove>
</agent>
<agent>...</agent>
<agent>...</agent>
</agents>
<agentCount>4</agentCount>
</skillGroup>
```





## Stats API

---

Use the Stats API to get statistical information about your deployment, such as the number of logged in agents.

This API is read-only.

### URL

`https://<server>/unifiedconfig/config/stats`

### Operations

- [get](#): Returns statistical information about your deployment.

### Response Parameters

- `numberOfAgentsLoggedIn`: The number of agents logged in.

### Example Get Response

```
<stats>
  <numberOfAgentsLoggedIn>10</numberOfAgentsLoggedIn>
</stats>
```





## Status API

---

The Status API retrieves information about the state of rules that collectively determine if the system is working correctly.

- [Configuration Rules, on page 158](#)
- [Operation Rules, on page 165](#)
- [System Health Rules, on page 171](#)
- [VM Rules, on page 171](#)

The system periodically checks the state of the rules. To force the status to be updated, refer to the [Scan API, on page 135](#).

### URL

`https://<server>/unifiedconfig/config/status`

### Operations

- [get](#): Returns the status rule results, using the URL  
`https://<server>/unifiedconfig/config/status`.

### Response Parameters

- **name**: The name of the status. See the [Configuration Rules, on page 158](#), [Operation Rules, on page 165](#), [System Health Rules, on page 171](#), and [VM Rules, on page 171](#).
- **category**: The status category of Configuration, Operation, System Health, or VM.
- **level**: The severity of the condition. Values include: OK, INFO, WARNING, ERROR, and BLOCKED. The BLOCKED level indicates that the rule has not been processed yet or that the failure of another rule prevents this rule from running.
- **machine**: A collection of machines ([Machine Inventory API, on page 81](#)) including each machine's name, type, and refURL. See [References, on page 6](#).

**Example Get Response**

```

<results>
  <statuses>
    <status>
      <name>CPU_USAGE</name>
      <category>SYSTEM_HEALTH</category>
      <level>OK</level>
      <detail xsi:type="metricDataDetail"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <actual>2</actual>
      </detail>
      <machines>
        <machine>
          <refURL>/unifiedconfig/config/machineinventory/7507</refURL>
          <machineType>CVP</machineType>
          <name>CVP-A</name>
          <side>sideA</side>
        </machine>
      </machines>
    </status>
    <status>
      <name>MEMORY_USAGE</name>
      <category>SYSTEM_HEALTH</category>
      <level>OK</level>
      <detail xsi:type="metricDataDetail"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <actual>17</actual>
      </detail>
      <machines>
        <machine>
          <refURL>/unifiedconfig/config/machineinventory/7507</refURL>
          <machineType>CVP</machineType>
          <name>CVP-A</name>
          <side>sideA</side>
        </machine>
      </machines>
    </status>
  </statuses>
</results>

```

- [Configuration Rules, page 158](#)
- [Operation Rules, page 165](#)
- [System Health Rules, page 171](#)
- [VM Rules, page 171](#)

## Configuration Rules

These rules show the potential configuration errors and warnings for all of the machines, ESX hosts, and Gateways. Each section has a rule table that applies to all machines listed in that category, as well as a rule table for each type of machine in that category.

**CCE Machines**

These rules show the potential configuration errors and warnings for CCE machines (Unified CCE Rogger, Unified CCE PG, Unified CCE AW-HDS-DDS, Unified CCE External HDS).

**Table 1: Rules for All CCE Machines (Unified CCE Rogger, Unified CCE PG, Unified CCE AW-HDS-DDS, Unified CCE External HDS)**

| Rule                        | Description   |
|-----------------------------|---|
| TRACE_LEVEL_NORMAL          | The trace level must be set to normal to ensure performance.                    |
| CAN_RESOLVE_ADDRESS_TO_FQDN | The machine address must be resolvable to a fully qualified domain name (FQDN). |

**Table 2: Unified CCE Rogger Rules**

| Rule                               | Description   |
|------------------------------------|---|
| LOGGER_INSTALLED                   | The logger service must be installed.                     |
| LOGGER_AUTOMATIC                   | The logger service startup type must be set to automatic. |
| NO_EXTRA_SERVICES_INSTALLED_ROGGER | Only required services can be installed.                  |
| ROUTER_INSTALLED                   | The router service must be installed.                     |
| ROUTER_AUTOMATIC                   | The router service startup type must be set to automatic. |

**Table 3: Unified CCE PG Rules**

| Rule  | Description  |
|---|--|
| CTI_SVR_INSTALLED                                 | The CTI Server service must be installed.  |
| CTI_SVR_AUTOMATIC                                 | The CTI server service startup type must be set to automatic.  |
| DIALER_INSTALLED                                  | If installed, the dialer service must be installed on both sides.  |
| DIALER_AUTOMATIC                                  | If installed, the dialer service startup type must be set to automatic.  |
| DIALER_INSTALLED_OUTBOUND_ENABLED                 | The dialer services must be installed when outbound is enabled.  |
| MR_PG_MR_PIM_CONFIGURED_FOR_MULTICHANNEL_MACHINES | <p>A Media Routing Peripheral must be configured in Peripheral Gateway Setup for each multichannel machine in the solution inventory. Multichannel machine types include:</p> <ul style="list-style-type: none"> <li>EXTERNAL_SOCIAL_MINER</li> <li>EXTERNAL_ECE</li> <li>EXTERNAL_THIRD_PARTY_MULTICHANNEL</li> </ul> |

| Rule   | Description  |
|--|--|
| MR_PG_APPLICATION_SERVER_HOSTNAME_NO_DUPLICATES                    | Application Server Host Name for each peripheral must be unique.   |
| MR_PG_MULTICHANNEL_APPLICATION_SERVER_HOSTNAME_BOTH_SIDES_MATCH    | Application Server Host Name for Media Routing Peripheral Multichannel must be the same on both sides.   |
| MR_PG_MULTICHANNEL2_APPLICATION_SERVER_HOSTNAME_BOTH_SIDES_MATCH   | Application Server Host Name for Media Routing Peripheral Multichannel2 must be the same on both sides.  |
| MR_PG_MULTICHANNEL3_APPLICATION_SERVER_HOSTNAME_BOTH_SIDES_MATCH   | Application Server Host Name for Media Routing Peripheral Multichannel3 must be the same on both sides.  |
| MR_PG_MULTICHANNEL_APPLICATION_SERVERS_HOSTNAME_FOUND_IN_INVENTORY | <p>Application Server Host Names used in Peripheral Gateway Setup for all multichannel peripherals must be created as external machines in the solution inventory. The machine type must be one of the following:</p> <ul style="list-style-type: none"> <li>• EXTERNAL_SOCIAL_MINER</li> <li>• EXTERNAL_ECE</li> <li>• EXTERNAL_THIRD_PARTY_MULTICHANNEL</li> </ul> |
| MR_PG_MR_PIM_COUNT   | The number of media routing PG MR_PIM processes (mr_pim.exe) on Side A and Side B must match. Valid if 0 to 4 MR_PIMS are enabled.   |
| MR_PG_INSTALLED  | The Media Routing PG service must be installed.  |
| MR_PG_AUTOMATIC  | The Media Routing PG service must be set to automatic.   |
| NO_EXTRA_SERVICES_INSTALLED_PG                                     | Only required and optional services can be installed (extra services such as an additional PG or CTI Server are not permitted).  |
| UCM_PG_APPUSER_BOTH_SIDES_MATCH                                    | The Communications Manager PIM application user configured on Side A must be identical to the user configured on Side B.   |
| UCM_PG_APPUSER_FOUND_ON_CUCM                                       | The Communications Manager PIM application user must be configured on the Communications Manager as an application user.   |
| UCM_PG_JTAPI_MATCHES_CM_SUB  | The Communications Manager PIM service address configured in PG Setup must match the Communications Manager Subscriber address on the same side  |
| UCM_PG_JTAPI_CLIENT_VERSION_MATCH_UCM                              | The JTAPI Client version installed must match the JTAPI Client version available on Unified Call Manager.  |

| Rule             | Description   |
|------------------|---|
| UCM_PG_INSTALLED | The UCM PG service must be installed.                     |
| UCM_PG_AUTOMATIC | The UCM PG service startup type must be set to automatic. |
| VRU_PG_INSTALLED | The VRU PG service must be installed.                     |
| VRU_PG_AUTOMATIC | The VRU PG service startup type must be set to automatic. |

**Table 4: Unified CCE AW-HDS-DDS Rules**

| Rule                                | Description  |
|-------------------------------------|--|
| SSO_COMPONENT_STATUS_MATCHES_GLOBAL | The global SSO status must be the same as component SSO status.  |
| SSO_VALID_IDS_REFERENCE             | If single sign-on is enabled, this machine must be associated with a valid Cisco Identity Service.                       |
| DISTRIBUTOR_CONAPI_MUST_BE_DISABLED | Configuration Management Service (CMS) Node and Agent Re-skilling Web Tool must be disabled using Unified CCE Web Setup. |
| DISTRIBUTOR_INSTALLED               | The distributor service must be installed.   |
| DISTRIBUTOR_AUTOMATIC               | The distributor service startup type must be set to automatic.   |
| NO_EXTRA_SERVICES_INSTALLED_AW      | Only required services can be installed.   |
| TASK_ROUTING_APP_PATHS_EXIST        | Each Peripheral Gateway with a Communications Manager PIM must have an associated Task Routing Application Path.         |

**Table 5: Unified CCE External HDS Rules**

| Rule                                | Description  |
|-------------------------------------|--|
| SSO_COMPONENT_STATUS_MATCHES_GLOBAL | The global SSO status must be the same as component SSO status.  |
| SSO_VALID_IDS_REFERENCE             | If single sign-on is enabled, this machine must be associated with a valid Cisco Identity Service.                       |
| DISTRIBUTOR_CONAPI_MUST_BE_DISABLED | Configuration Management Service (CMS) Node and Agent Re-skilling Web Tool must be disabled using Unified CCE Web Setup. |
| TASK_ROUTING_APP_PATHS_EXIST        | Each Peripheral Gateway with a Communications Manager PIM must have an associated Task Routing Application Path.         |

## Gateways

These rules show the potential configuration errors and warnings for gateways.

**Table 6: Gateway Rules**

| Rule            | Description   |
|-----------------|---|
| GW_CODEC        | <p>The dial peers for each CVP call server on each gateway are configured with the supported codec:</p> <ul style="list-style-type: none"> <li>• voice-class codec #num</li> <li>• dial-peer voice #num voip (codec). The supported voice codecs are g711alaw, g711ulaw, g729r8, mp4a-latm and g722-64 for the above four dial peers to CVP call servers (if the dial peers are configured on the voice gateway). The supported video codec is h264. The dial peer is identified via "session target ipaddress xxxxx". The IP address must point to the IP address of CVP call server.</li> </ul> |
| GW_SIP_PROTOCOL | The dial peers for each CVP call server on each gateway must be configured with the supported session protocol, sipv2.  |

## CUIC-LD-IdS Machines

These rules show the potential configuration errors and warnings for CUIC-LD-IdS machines.

**Table 7: Rules for all CUIC-LD-IdS Machines**

| Rule                                | Description  |
|-------------------------------------|--|
| SSO_COMPONENT_STATUS_MATCHES_GLOBAL | The global SSO status must be the same as component SSO status.                                    |
| SSO_VALID_IDS_REFERENCE             | If single sign-on is enabled, this machine must be associated with a valid Cisco Identity Service. |
| CAN_RESOLVE_ADDRESS_TO_FQDN         | The machine address must be resolvable to an FQDN.   |

**Table 8: Rules for the CUIC-LD-IdS Publisher**

| Rule                          | Description   |
|-------------------------------|---|
| CUIC_REALTIME_DS_CORRECT_HOST | The realtime datasource must be configured with the correct hosts: either the Side A and B Unified CCE AW-HDS-DDS Servers, or, if used, the External HDS. |



| Rule                            | Description   |
|---------------------------------|---|
| CUIC_HISTORICAL_DS_CORRECT_HOST | The historical datasource must be configured with the correct hosts: either the Side A and B Unified CCE AW-HDS-DDS Servers, or, if used, the External HDS. |

### Finesse Machines

These rules show the potential configuration errors and warnings for Finesse machines.

**Table 9: Rules for All Finesse Machines**

| Rule                                | Description  |
|-------------------------------------|--|
| SSO_COMPONENT_STATUS_MATCHES_GLOBAL | The global SSO status must be the same as the component SSO status.                                |
| SSO_VALID_IDS_REFERENCE             | If single sign-on is enabled, this machine must be associated with a valid Cisco Identity Service. |
| CAN_RESOLVE_ADDRESS_TO_FQDN         | The machine address must be resolvable to an FQDN.   |

### Unified CVP Machines

These rules show the potential configuration errors and warnings for CVP machines.

**Table 10: Rules for All Unified CVP Machines (Unified CVP, Unified CVP OPS, Unified CVP Reporting)**

| Rule                        | Description  |
|-----------------------------|--|
| TRACE_LEVEL_NORMAL          | The trace level must be set to normal to ensure performance. |
| CAN_RESOLVE_ADDRESS_TO_FQDN | The machine address must be resolvable to an FQDN.           |

**Table 11: Unified CVP Rules**

| Rule                                       | Description   |
|--|---|
| CVP_CUCM_SIP_SERVER_GROUPS_MATCH_CUCM_FQDN | The CVP SIP Server Group names that contain Communication Manager addresses must match the Communication Manager Cluster Fully Qualified Domain Name. |

| Rule  | Description   |
|---|---|
| CVP_CUCM_SIP_SERVER_GROUPS_ONLY _CONTAIN_CUCM_HOSTS | The CVP SIP Server Groups that contain Communication Manager addresses cannot contain non-Communication Manager addresses (which include Communication Manager hosts that are not part of the inventory). |
| CVP_RING_TONE_DN_PATTERN                            | The Ring Tone Dialed Number configured on the CVP Call Server should match the pattern 91*.   |
| CVP_ERROR_LABEL_DN_PATTERN                          | The Error Tone Dialed Number configured on the CVP Call Server should match the pattern 92*.  |

### Unified CM Machines

These rules show the potential configuration errors and warnings for Unified CM machines.

**Table 12: Rules for All Unified CM Machines (Unified CM Publisher, Unified CM Subscriber)**

| Rule                        | Description  |
|-----------------------------|--|
| TRACE_LEVEL_NORMAL          | The trace level must be set to normal to ensure performance. |
| CAN_RESOLVE_ADDRESS_TO_FQDN | The machine address must be resolvable to an FQDN.           |

**Table 13: Unified CM Publisher Rules**

| Rule                                | Description  |
|-------------------------------------|--|
| CUCM_FQDN_DEFINED                   | The Communication Manager Cluster Fully Qualified Domain Name must be defined.                   |
| CVP_SERVER_MUST_HAVE_A_CM_SIP_TRUNK | Each CVP Server must be referenced by at least one Communications Manager SIP Trunk destination. |

### ESX Hosts

These rules show the potential configuration errors and warnings for ESX hosts.

**Table 14: Rules for All ESX Hosts**

| Rule                      | Description  |
|---------------------------|--|
| VMHOST_ESXI_VERSION_MATCH | Side A and Side B VM Hosts must be on the same ESXi version. |

| Rule         | Description   |
|--------------|---|
| VM_DATASTORE | Virtual machines must be deployed on the correct datastore. |

### SocialMiner Machines

These rules show the potential configuration errors and warnings for SocialMiner machines.

**Table 15: SocialMiner Rules**

| Rule                        | Description  |
|-----------------------------|--|
| SOCIAL_MINER_MR_ENABLED     | Multichannel routing must be enabled on SocialMiner. |
| CAN_RESOLVE_ADDRESS_TO_FQDN | The machine address must be resolvable to an FQDN.   |

## Operation Rules

These rules show the potential operation errors and warnings for Unified CCE, Unified CM, Unified CVP, Gateways, Unified Intelligence Center, Finesse, and Enterprise Chat and Email. Each section has a rule table that applies to all machines listed in that category, as well as a rule table for each type of machine in that category.

### Unified CCE Machines

These rules show the potential operation errors and warnings for Unified CCE machines (Rogger, PG, and AW-HDS-DDS).

**Table 16: Unified CCE Rogger Rules**

| Rule                         | Description  |
|------------------------------|--|
| LOGGER_CAMPAIGN_MGR_RUNNING  | The logger campaign manager process (campaignmanager.exe) must be running on the AW 1 when the dialer is installed on the Roggers. |
| LOGGER_CONFIG_LOGGER_RUNNING | The logger configuration logger process (configlogger.exe) must be running.  |
| LOGGER_CSFS_RUNNING          | The logger customer support forwarding service process (csfs.exe) must be running.   |
| LOGGER_HIST_LOGGER_RUNNING   | The logger historical logger process (histlogger.exe) must be running.   |
| LOGGER_BA_IMPORT_RUNNING     | The logger import process (baimport.exe) must be running on the AW 1 when the dialer is installed on the Roggers.                  |

| Rule                                     | Description   |
|--|---|
| LOGGER_RECOVERY_RUNNING                  | The logger recovery process (recovery.exe) must be running.   |
| LOGGER_REPLICATION_RUNNING               | The logger replication process (replication.exe) must be running.   |
| LOGGER_RUNNING                           | The logger service must be running.   |
| SERVER_CREDENTIALS                       | The Diagnostic Framework credentials entered for the Side A Unified AW-HDS-DDS must be valid.                         |
| SERVER_CONNECTION                        | The Cisco ICM Diagnostic Framework service on this machine must be reachable on the network.                          |
| ROUTER_CCAGENT_INSVL_ACTIVE_ENABLE_COUNT | The router central controller agent process (ccagent.exe) must be in service for both PGs.                            |
| ROUTER_CCAGENT_RUNNING                   | The router central controller agent process (ccagent.exe) must be running.  |
| ROUTER_DBAGENT_RUNNING                   | The router database agent process (dbagent.exe) must be running.  |
| ROUTER_LIVE_DATA_ACTIVE_IDLE             | The router Live Data connection must be active on one side and idle on the other side.                                |
| ROUTER_MDSPROC_IN_SVC_PR_ENB_DSB         | The router message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side. |
| ROUTER_MDSPROC_RUNNING                   | The router message delivery service process (mdsproc.exe) must be running.  |
| ROUTER_ROUTER_RUNNING                    | The router process (router.exe) must be running.  |
| ROUTER_RUNNING                           | The router service must be running.   |

Table 17: Unified CCE PG Rules

| Rule                        | Description  |
|-----------------------------|--|
| CTI_SVR_CTI_SVR_ACTIVE_IDLE | The CTI server process (ctisvr.exe) must be active on one side and idle on the other side. |
| CTI_SVR_CTI_SVR_RUNNING     | The CTI server process (ctisvr.exe) must be running.                                       |
| CTI_SVR_RUNNING             | The CTI Server service must be running.  |

| Rule                             | Description   |
|----------------------------------|---|
| DIALER_RUNNING                   | If dialer is installed, then the dialer service must be running.  |
| DIALER_BA_DIALER_SIP_ACTIVE_IDLE | The dialer process (badialer_sip.exe) must be active on one side and idle on the other side.                                    |
| DIALER_BA_DIALER_SIP_RUNNING     | The dialer process (badialer_sip.exe) must be running.  |
| MR_PG_MR_PIM_ACTIVE_IDLE         | Each MR_PIM process (mr_pim.exe) must be active on one side and idle on the other side.   |
| MR_PG_MDSPROC_IN_SVC_PR_ENB_DSB  | The media routing PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side. |
| MR_PG_MDSPROC_RUNNING            | The media routing PG message delivery service process (mdsproc.exe) must be running.  |
| MR_PG_PG_AGENT_ACTIVE_IDLE       | The media routing PG PG agent process (pgagent.exe) must be active on one side and idle on the other side.                      |
| MR_PG_PG_AGENT_RUNNING           | The media routing PG PG agent process (pgagent.exe) must be running.  |
| MR_PG_RUNNING                    | The media routing PG service must be running.   |
| SERVER_CREDENTIALS               | The Diagnostic Framework credentials entered for the Side A Unified CCE AW-HDS-DDS must be valid.                               |
| SERVER_CONNECTION                | The Cisco ICM Diagnostic Framework service on this machine must be reachable on the network.                                    |
| UCM_PG_JTAPI_ACTIVE_IDLE         | The UCM PG jtapi process (jtapigw.exe) must be active on one side and idle on the other side.                                   |
| UCM_PG_JTAPI_RUNNING             | The UCM PG jtapi process (jtapigw.exe) must be running.   |
| UCM_PG_LIVE_DATA_ACTIVE_IDLE     | The UCM PG Live Data connection must be active on one side and idle on the other side.  |
| UCM_PG_MDSPROC_IN_SVC_PR_ENB_DSB | The UCM PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.           |
| UCM_PG_MDSPROC_RUNNING           | The UCM PG message delivery service process (mdsproc.exe) must be running.  |

| Rule                             | Description  |
|----------------------------------|--|
| UCM_PG_PIM_COUNT                 | The number of UCM PG's PIM processes (eagtpim.exe) on Side A and Side B must match. Valid if 1 PIM is enabled.           |
| UCM_PG_PG_AGENT_ACTIVE_IDLE      | The UCM PG PG agent process (pgagent.exe) must be active on one side and idle on the other side.                         |
| UCM_PG_PG_AGENT_RUNNING          | The UCM PG PG agent process (pgagent.exe) must be running.   |
| UCM_PG_RUNNING                   | The UCM PG service must be running.  |
| VRU_PG_VRU_PIM_ACTIVE_IDLE       | Each VRU PIM process (vrupim.exe) must be active on one side and idle on the other side.                                 |
| VRU_PG_MDSPROC_IN_SVC_PR_ENB_DSB | The VRU PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.    |
| VRU_PG_MDSPROC_IN_SVC_PR_ENB_DSB | The VRU PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.    |
| VRU_PG_MDSPROC_RUNNING           | The VRU PG message delivery service process (mdsproc.exe) must be running.   |
| VRU_PG_PIM_COUNT                 | The number of VRU PG's PIM processes (vrupim.exe) on Side A and Side B must match. Valid if 0 to 2 VRU PIMs are enabled. |
| VRU_PG_PG_AGENT_ACTIVE_IDLE      | The VRU PG PG agent process (pgagent.exe) must be active on one side and idle on the other side.                         |
| VRU_PG_PG_AGENT_RUNNING          | The VRU PG PG agent process (pgagent.exe) must be running.   |
| VRU_PG_RUNNING                   | The VRU PG service must be running.  |

Table 18: Unified CCE AW-HDS-DDS and External HDS Rules

| Rule                              | Description  |
|-----------------------------------|--|
| DISTRIBUTOR_CONFIG_LOGGER_RUNNING | The distributor configuration logger process (configlogger.exe) must be running. |
| DISTRIBUTOR_RT_CLIENT_RUNNING     | The distributor real-time client process (rtclient.exe) must be running.         |

| Rule  | Description   |
|---|---|
| DISTRIBUTOR_RT_DIST_RUNNING   | The distributor real-time distributor process (rtdist.exe) must be running.                   |
| DISTRIBUTOR_RUNNING<br><b>Note</b> This rule is not applicable for the external HDS | The distributor service must be running.  |
| DISTRIBUTOR_UPDATE_AW_RUNNING   | The distributor update process (updateaw.exe) must be running.                                |
| DEPLOYMENT_TASKS_PASSING  | The deployment tasks must all be passing.   |
| SERVER_CREDENTIALS  | The Diagnostic Framework credentials entered for the Side A Unified AW-HDS-DDS must be valid. |
| SERVER_CONNECTION   | The Cisco ICM Diagnostic Framework service on this machine must be reachable on the network.  |

### Unified CM Machines

These rules show the potential operation errors and warnings for Unified CM machines.

**Table 19: Unified CM Publisher and Unified CM Subscriber Rules**

| Rule               | Description  |
|--------------------|--|
| SERVER_CREDENTIALS | The AXL service credentials entered for the Publisher must be valid. |
| SERVER_CONNECTION  | The AXL service on the Publisher must be reachable on the network.   |

### Unified CVP Machines

These rules show the potential operation errors and warnings for CVP machines.

**Table 20: Rules for All Unified CVP Machines (Unified CVP, Unified CVP OPS, and Unified CVP Reporting)**

| Rule               | Description  |
|--------------------|--|
| SERVER_CREDENTIALS | The Cisco CVP WebServicesManager credentials entered for the CVP Ops Console Server must be valid.   |
| SERVER_CONNECTION  | The Cisco CVP WebServicesManager service on CVP Ops Console Server must be reachable on the network. |

### Gateways

These rules show the potential operation errors and warnings for gateways.

**Table 21: Gateway Rules**

| Rule               | Description   |
|--------------------|---|
| SERVER_CREDENTIALS | The service credentials entered for the gateway in Unified CVP Ops Console must be valid. |
| SERVER_CONNECTION  | The service must be reachable on the network.   |

### Unified Intelligence Center Machines

These rules show the potential operation errors and warnings for Unified Intelligence Center machines.

**Table 22: Rules for Unified Intelligence Center Machines (Unified Intelligence Center Publisher and Unified Intelligence Center Subscriber)**

| Rule               | Description   |
|--------------------|---|
| SERVER_CREDENTIALS | The Unified Intelligence Center Administration credentials entered for the Publisher must be valid. |
| SERVER_CONNECTION  | The SOAP service on the Publisher must be reachable on the network.                                 |

### Finesse

These rules show the potential operation errors and warnings for Finesse.

**Table 23: Finesse Rules**

| Rule                           | Description   |
|--------------------------------|---|
| FINESSE_SYSTEM_STATUS          | Finesse must be in service.   |
| SERVER_CREDENTIALS             | The Finesse Administration credentials entered for the Primary Finesse machine must be valid. |
| SERVER_CONNECTION              | The SOAP service on the Primary Finesse machine must be reachable on the network.             |
| FINESSE_TOMCAT_SERVICE_STARTED | The Tomcat service must be started.   |

### SocialMiner Machines

These rules show the potential operation errors and warnings for SocialMiner machines.



Table 24: SocialMiner Rules

| Rule               | Description                                   |
|--------------------|---|
| SERVER_CREDENTIALS | The service credentials must be valid.        |
| SERVER_CONNECTION  | The service must be reachable on the network. |

## System Health Rules

System health data is collected on a 20 second interval. The values represent the 95th percentile calculated over the preceding 10 minute period.

### Usage Errors and Warnings

The following system health rules show the usage errors and warnings for all virtual machines. The DISK\_USAGE rule applies to all disks in the system.

| Rule         | Description  | Machine Type |
|--------------|--|--------------|
| CPU_USAGE    | CPU usage must be below 80%.   | All          |
| MEMORY_USAGE | Memory usage must be below 80%.  | All          |
| DISK_USAGE   | Disk usage must be below 90%. If any disk value is over the threshold, then the highest error level is reported. | All          |

### Datastore Metrics

The following system health rules apply to all VM Hosts. Each rule output in the status detail identifies the datastore affected and the value for that particular metric.

| Rule                                      | Description                                  |
|---|--|
| DATASTORE_DISK_COMMANDS_ABORTED_SUMMATION | Number of aborted commands per datastore.    |
| DATASTORE_DISK_BUS_RESETS_SUMMATION       | Number of bus resets per datastore.          |
| DATASTORE_DISK_TOTAL_LATENCY_AVERAGE      | Total latency in milliseconds per datastore. |

## VM Rules

These rules show the potential errors and warnings for virtual machines.

| Rule         | Description                      | Machine Type |
|--------------|----------------------------------|--------------|
| VMWARE_TOOLS | VMware tools must be up-to-date. | VM Host      |

| Rule            | Description   | Machine Type |
|-----------------|---|--------------|
| VMWARE_GUEST_OS | The operating system setting on each Virtual Machine (VM) must match the operating system installed on the Guest. | All          |



## Trace Level API

---

Use the Trace Level API to set the trace levels for the following components:

- Unified Contact Center Enterprise (CCE)
- Unified Customer Voice Portal (CVP)
- Unified Communications Manager (UCM)

### URL

`https://<server>/unifiedconfig/config/tracelevel`

### Operations

- [list](#): Returns the trace level for each component type.
  - **Query parameters**
    - Summary list: See [list](#), on page 4. Summary defaults to true on the Trace Level API.
- [update](#): Sets the trace level for each component type.

### Parameters

- component: A list of components for which the trace level is set. Includes the following parameters:
  - type:
    - CCE
    - CVP
    - UCM
  - level:
    - NORMAL
    - DETAILED

- **CUSTOM:** Not set by user. This level appears when one or more trace levels have been set by an outside program, as the component does not match the normal detailed definitions for the trace levels.

- **traceMachines:** A collection of trace information about each machine containing the refURL of the machine and a collection of process level trace values. Only available when the summary query parameter is false.

### Example List Response

```
<traceLevels>
  <component>
    <type>CCE</type>
    <level>DETAILED</level>
  </component>
  <component>
    <type>CVP</type>
    <level>NORMAL</level>
  </component>
  <component>
    <type>UCM</type>
    <level>CUSTOM</level>
  </component>
</traceLevels>
```



## Uninitialize API

Use this API to roll back the changes made by the Initialization API to the system only if an initialization task is in the FAILED state.

- Start reversion process of initialization tasks which are in a state of SUCCEEDED or FAILED.
- Check the initialization status of the system using Initialization API.

### URL

`https://<server>/unifiedconfig/config/uninitialize`

### Operations

- **list**: Lists information about the system uninitialization status.
- **update**: Starts a system uninitialization.

### Parameters

- **initializationStatus**: The name of the startup task.
- **state**: The state of the task. Values include PROCESSING, FAILED, FAILED\_NEEDS\_RETRY, or SUCCEEDED.

Tasks in the FAILED\_NEEDS\_RETRY state do not require roll-back using the Uninitialize API.

### Example GET Response

```
<results>
  <state>SUCCEEDED</state>
  <initializationStatuses>
    <initializationStatus><state>SUCCEEDED</state><name>SetProcessingStateTask</name>
  </initializationStatus>
  <initializationStatus><state>SUCCEEDED</state><name>SqlUserTask_sideA</name>
</initializationStatus>
  <initializationStatus><state>SUCCEEDED</state><name>SqlUserTask_sideB</name>
</initializationStatus>
  <initializationStatus><state>SUCCEEDED</state><name>DatabaseTask_sideA</name>
</initializationStatus>
  <initializationStatus><state>SUCCEEDED</state><name>JtapiClientTask_sideA</name>
</initializationStatus>
  <initializationStatus><state>SUCCEEDED</state><name>InstanceTask</name>
```

```
</initializationStatus>
<initializationStatus><state>SUCCEEDED</state><name>PGUserTask</name>
</initializationStatus>
<initializationStatus><state>SUCCEEDED</state><name>DatabaseTask_sideB</name>
</initializationStatus>
<initializationStatus><state>SUCCEEDED</state><name>JtapiClientTask_sideB</name>
</initializationStatus>
</initializationStatuses>
</results>
```



## Version API

The Version API is used to get the Unified Contact Center Enterprise version information from the system.

### URL

`https://<server>/unifiedconfig/config/version`

### Parameters

- `ucceVersion`: Information about the UCCE version installed on the system, including maintenance releases, engineering specials, and the schema version.

### Operations

- `get`: Returns UCCE version information using the URL `https://<server>/unifiedconfig/config/version`.

### Example Get Response

```
<versionInfo>
  <ucceVersion>
    <majorVersion>11</majorVersion>
    <minorVersion>0</minorVersion>
    <maintenanceVersion>1</maintenanceVersion>
    <srVersion>0</srVersion>
    <esVersion>0</esVersion>
    <buildVersion>3086</buildVersion>
    <versionString>11.0.1.0.0.3086</versionString>
    <schemaVersion>
      <major>181</major>
      <ccMinor>3</ccMinor>
      <awMinor>3</awMinor>
      <cceMinor>0</cceMinor>
    </schemaVersion>
  </ucceVersion>
</versionInfo>
```







## INDEX

### C

creating [109](#)  
Do Not Call list [109](#)

### D

Do Not Call list [109](#)  
creating [109](#)

