Cisco Unified Contact Center Enterprise Developer Reference, Release 12.6(2)

First Published: 2023-04-28

Last Modified: 2023-08-21

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.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

CHAPTER 1

Working with Unified CCE APIs

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 5.

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 8, Pagination, on page 6, Search, on page 9, and Sort, on page 10.

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 Unified CCE Administration APIs and items is defined by the role for which the administrator is responsible.

The following API is not available for administrators:

- Agent Team API, on page 25

The following APIs allow update with restrictions:

- Agent Call API
    - When updating an agent, administrators can only change the following parameters:
        - skillGroups
        - defaultSkillGroup

        - skillGroupsAdded
        - skillGroupsRemoved
        - agentAttributes
        - agentAttributesAdded
        - agentAttributesRemoved
        - password

- Skill Group API, on page 109
    - When updating a skill group, administrators can only change the following parameters:
        - agents
        - agentsAdded
        - agentsRemoved

The following APIs are only available in the Contact Director deployment:

- Contact Share Group API, on page 55
- Contact Share Rule API, on page 57

Supervisor Access

The following APIs are read only:

- Agent Team API, on page 25
    - Supervisors can only see teams that are on their peripheral.
- Precision Queue API, on page 95

The following APIs allow update with restrictions:

- Attribute API, on page 29
    - 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 Call API

  :

    - 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 109:
    - Supervisors can only see and update skill groups that are on their peripheral.
    - 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 Unified CCE Administration APIs.

Authentication

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

Deployment

In most deployments, all APIs are available except for the Contact Sharing APIs. You can only use the Contact Sharing APIs in a Contact Director deployment.

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>
```

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: Specfies 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 9.
- sortTerm: The value specified in the sort query parameters. See Sort, on page 10.

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 1) operation.
- A changeStamp is a required parameter for the body of a PUT (update, on page 2) 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 built-in character sets, see the latest version of the document *Cisco Unified Contact Center Enterprise Installation and Upgrade Guide* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-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.

Example Get Response

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

Note The default search is a case-insensitive substring search on multiple columns and hence may take a longer time to respond, depending on the number of records and columns to search for.

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.

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
```

CHAPTER 2

Active Directory Domain API

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 List Response

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

CHAPTER 3

Administrator API

Administrator API

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

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 7.
- changeStamp: See Shared Parameters, on page 7.
- description: See Shared Parameters, on page 7.
- 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.

- customer: A reference to a customer, including the refURL.
- 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
• userName • domainName • description	• userName • domainName • description

See Search, on page 9 and Sort, on page 10.

Example Get Response

```
<administrator>
    <changeStamp>0</changeStamp>
    <domainName>domain</domainName>
    <userName>user1</userName>
    <readOnly>false</readOnly>
    <customer>
       <refURL>/unifiedconfig/config/customer/(id)</refURL>
    </customer>
</administrator>
```

CHAPTER 4

Agent API

Agent Call API

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

Note In the Role API, when you enable the ManageAgentAttribute or ReSkillAgents subfeature in the accessList parameter for a custom role then the Agent API is provided with Update Only Access instead of Full Access. With Update Only Access, you cannot create and delete an agent using Agent API.

URL

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

Operations

- 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 2.

- update: Updates one agent.

Parameters

- refURL: The refURL for the agent. See Shared Parameters, on page 7.
- agentId: The unique peripheral number. Maximum length of 11 characters allowed. Default is an autogenerated 7-digit number.
- changeStamp: See Shared Parameters, on page 7.
- description: See Shared Parameters, on page 7.
- department: A reference to the agent's department, including the refURL and name. See References, on page 4.
- agentStateTrace: Indicates if agent state tracing is turned on for the agent. True or false.
- agentDeskSettings: A reference to the agent's agentDeskSettings, including the refURL and name. See References, on page 4.
- 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.

  - digitalChannel:Whether the agent is enabled for digital channel interaction. Email address is mandatory when digitalChannel is set to True.

    The valid values are True or False. The default is False.

- - screenName: The screen name of the agent. Default is null. Maximum characters are 32 .

- emailAddress: The email address of the Agent. Maximum characters are 50.

Note The screenName parameter is applicable only for the ECE agent.

- 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 29) 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 4.

- agentAttributesAdded: A collection of agent attribute references (Attribute API, on page 29) 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 4.

- agentAttributesRemoved: A collection of agent attribute references (Attribute API, on page 29) 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 4.

- 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 109), including the refURL and name of each associated skill group. See References, on page 4.

- 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 4.

- 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 4.

- 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 4.

  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.

- agentServicesEnabled: A collection of Contact Center AI services enabled for an agent.
    - agentService: The type of Contact Center AI service. Supported values are `AgentAnswers` and `Transcript`.
- agentTeam: A reference to the agent's team (Agent Team API, on page 25), including the refURL and name. See References, on page 4.
- 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 4.
-

Search and Sort Values

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

Search parameters	Sort parameters
• agentId • description • person.firstName • person.lastName • person.userName	• agentId • description • supervisor • agentStateTrace • agentTeam.name • person.firstName • person.lastName • person.userName • person.loginEnabled • person.ssoEnabled • datacenter.name • peripheral.name

See Search, on page 9 and Sort, on page 10.

Advanced search parameters

There are several advanced searches you can perform on the Agent API, including supervisor, attributes, skillgroups, team, data centers, 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 & attrt2 & 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.
- **usernames:(username1|username2|username3, ...)** find *all* agents who belong to *any* of the specified username. Up to fifteen usernames can be specified. The username name is fully matched.
- **include: (ID1 & ID2 & ID3, ...)** find *all* specified agents even if they do not meet other search criteria. Each ID is fully matched. Obtain this ID from the refURL. For example, 5017 is the ID in the following refURL `<refURL>/unifiedconfig/config/agent/5017</refURL>`.
- **exclude: (ID1 & ID2 & ID3, ...)** exclude *all* specified agents from the results even if they meet all other search criteria. Each ID is fully matched. Obtain this ID from the refURL. For example, 5017 is the ID in the following refURL `<refURL>/unifiedconfig/config/agent/5017</refURL>`.

Note

If a search string contains both basic and advanced search parameters, only the basic search parameter is considered and displayed. The advanced search parameter is ignored from the search criteria.

If a search string contains either the basic or the advanced search parameter, the respective search parameter is considered in the search criteria.

For example, consider the search string "person usernames:(person1@boston.com)".

The above string contains both search criteria [basic — `person` and advanced — `usernames:(person1@boston.com)`]. So, the advanced search criteria is ignored and the basic search criteria is considered. The search result is displayed based on the basic search criteria, that is the number of instances of agents.

Example Get Response

```
<agent>
         <department>
          <refURL>/unifiedconfig/config/department/5001</refURL>
          <name>debit_card</name>
         </department>
 <changeStamp>2877</changeStamp>
        <refURL>/unifiedconfig/config/agent/5017</refURL>
        <agentId>8006</agentId>
        <agentServicesEnabled>
          <agentService>AgentAnswers</agentService>

          <agentService>Transcript</agentService>
        </agentServicesEnabled>
        <agentStateTrace>false</agentStateTrace>
        <description>an agent</description>
        <person>
             <firstName>Agent2</firstName>
             <lastName>Agent2</lastName>
             <loginEnabled>true</loginEnabled>
```

```
           <userName>Agent2@xyz.com</userName>
           <password>mypassword</password>
           <ssoEnabled>false</ssoEnabled>
         <ecePerson>true</ecePerson>
           <emailAddress>agent@xyz.com</emailAddress>
           <screenName>agentScreenName</screenName>
     </person>
      <agentDeskSettings>
           <name>test2</name>
           <refURL>/unifiedconfig/config/agentdesksetting/5434</refURL>
      </agentDeskSettings>
      11.6  PCCE multi PG support<datacenter>
              <name>Berlin</name>
              <refURL>/unifiedconfig/config/datacenter/5000</refURL>
      </datacenter>
      <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>postgraduate 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>
```

REST Responses

Following are the possible REST responses that can be received for Agent API calls:

- **Success (201 Created or 200 OK)**

  Configuration changes persist in AW DB and synchronized with ECE and CUIC.

- **Partial Success (201 Created or 200 OK)**

  Configuration changes are persist in AW DB, but failed to synchronize with ECE and CUIC. In this case even if the REST response status is a success (201 Created or 200 OK), the response body will include an APIError.

  Examples of API errors:

  ```
  <apiErrors>
   <apiError>
    <errorMessage>Configuration update failed for one or more devices.</errorMessage>
    <errorType>PARTIAL_SUCCESS</errorType>
   </apiError>
  </apiErrors>
  ```

- **Server Busy (503 Service Unavailable)**

  This occurs when data synchronization to a device is in progress.

CHAPTER 5

Agent Security API

Agent Security API

Use the Agent Security API to get and update the status of global switch for generating advanced agent password hashing, and also to clear older agent password hashes from the system.

This API is available for administrators only, applicable for all the deployments, and is not supported in SSO modes.

URL

```
https://unifiedconfig/config/agentsecurity
```

Operations

- get: Returns the secure global switch value.

  ```
  https://unifiedconfig/config/agentsecurity?details=true
  ```

- update: Updates the secure global switch status and clears the older agent password from the system.

Parameters

enforceAdvancedHashing: true means, global switch will be enabled and removes older Agent passwords, false means global switch will be disabled.

Example Get Response

```
<agentSecurity>
     <enforceAdvancedHashing>true</enforceAdvancedHashing>
    <agentsWithoutAdvancedHashing>10</agentsWithoutAdvancedHashing>
  </agentSecurity>
```

Example Update Request

```
<agentSecurity>
    <changeStamp>277</changeStamp>
```

```
    <enforceAdvancedHashing>true</enforceAdvancedHashing>
</agentSecurity>
```

CHAPTER 6

Agent Team API

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 and view the agent teams currently defined in the database.

Note Access to this API is different for supervisors and administrators. For more information, see Access, on page 2.

URL

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

Operations

- 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 2.

Parameters

- refURL: The refURL of the agent team. See Shared Parameters, on page 7.
- name: The name of the agent team. See Shared Parameters, on page 7.
- description: See Shared Parameters, on page 7.

- dialedNumber: A reference to an internal dialed number () for the agent team, including the refURL and dialed number string. See References, on page 4.
- agents: A collection of agent (Agent Call API) references, including the refURL, first name, last name, username, and agent ID for each agent on the team. See References, on page 4.
- agentCount: Read-only field. Number of agents on the team.
- supervisors: A collection of supervisor (Agent Call API) references, including the refURL, first name, last name, username, and agent ID for each supervisor who supervises this team. See References, on page 4.
- supervisorCount: Read-only field. Number of supervisors who supervise this team.
- datacenter: The data center to which the agents on this team belong, 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
• name • description	• name (default) • description • datacenter.name

See Search, on page 9 and Sort, on page 10.

Advanced Search Parameters

datacenters: (dc1|dc2|dc3...) which returns all teams who belong to any of the specified data centers. You can specify up to three data centers. The data center names are fully matched (case-insensitive, no partial matches). Searching for "core" returns all machines in the core data center.

Example Get Response

```
<agentTeam>

<refURL>https://***.***.***.***/unifiedconfig/config/agentteam/(id)</refURL>
    <name>team1</name>
    <datacenter>
        <refURL>/unifiedconfig/config/datacenter/9887</refURL>
        <name>Boston</name>
    </datacenter>
    <dialedNumber>
     <refURL>[https://***.***.***.***/unifiedconfig/config/dialednumber/(id)]</refURL>
       <dialedNumberString>8885551212</dialedNumberString>
    </dialedNumber>
    <description>test agent team1</description>
    <agentCount>1</agentCount>
    <agents>
       <agent>
          <refURL>[https://***.***.***.***/unifiedconfig/config/agent/(id_1)]</refURL>
          <firstName>John</firstName>
          <lastName>Smith</lastName>
          <userName>username</userName>
          <agentId>8006</agentId>
```

```
      </agent>
      <agent>
         <refURL>[https://***.***.***.***/unifiedconfig/config/agent/(id_2)]</refURL>
         <firstName>Jane</firstName>
         <lastName>Doe</lastName>
         <userName>username</userName>
         <agentId>8007</agentId>
      </agent>
   </agents>
   <supervisorCount>2</supervisorCount> <supervisor>
   <supervisors>
      <supervisor>
         <refURL>[https://***.***.***.***/unifiedconfig/config/agent/(id_3)]</refURL>
         <firstName>Mary</firstName>
         <lastName>Hart</lastName>
         <userName>username</userName>
         <agentId>8008</agentId>
      </supervisor>
      <supervisor>
         <refURL>[https://***.***.***.***/unifiedconfig/config/agent/(id_4)]</refURL>
         <firstName>Jack</firstName>
         <lastName>Jones</lastName>
         <userName>username</userName>
         <agentId>8009</agentId>
      </supervisor>
   </supervisors>
   <changeStamp>0</changeStamp>
</agentTeam>
```

CHAPTER 7

Attribute API

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 2.

- 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 7.
- name: The name of the attribute. See Shared Parameters, on page 7.
- changeStamp: See Shared Parameters, on page 7.
- description: See Shared Parameters, on page 7.
- 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 4.
- agentAttributesAdded: A collection of agent attribute references (Attribute API, on page 29) 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 4.

- agentAttributesRemoved: A collection of agent attribute references (Attribute API, on page 29) 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 4.
- 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.

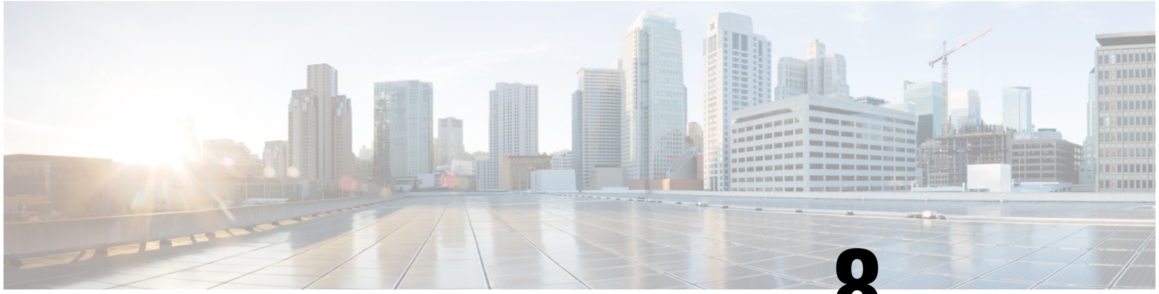
Search parameters	Sort parameters
• name • description	• name (default) • dataType • defaultValue • description

See Search, on page 9 and Sort, on page 10.

Example Get Response

```
<attribute>

   <refURL>/unifiedconfig/config/attribute/5002</refURL>
         <changeStamp>0</changeStamp>
         <dataType>4</dataType>
         <defaultValue>1</defaultValue>
         <description>test</description>
         <name>AttributeA</name>
</attribute>
```



CHAPTER 8

Bucket Interval API

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 7.
- name: The name of the bucket interval. See Shared Parameters, on page 7.
- changeStamp: See Shared Parameters, on page 7.
- 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.

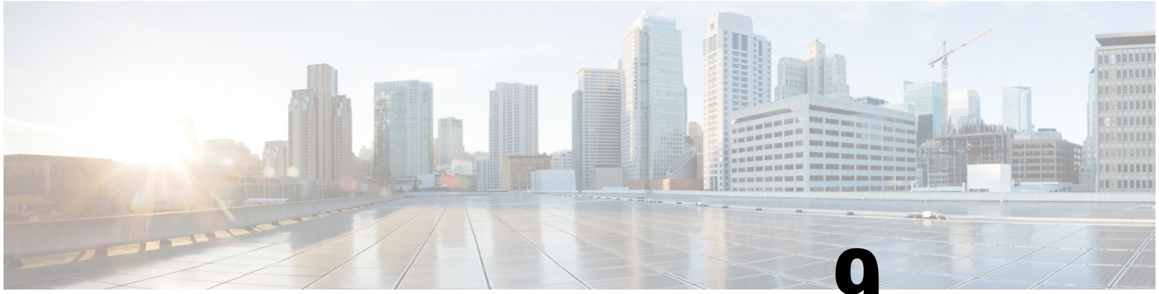
Search parameters	Sort parameters
• name	• name (default) • upperBound 1-9

See Search, on page 9 and Sort, on page 10.

Example Get Response

```
<bucketInterval>
    <refURL>/unified/config/bucketInterval/(id)</refURL>
    <name>test</name>

    <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>
```



CHAPTER 9

Bulk Job API

Bulk Job API

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.

Note Avoid performing a bulk job transaction during a maintenance window.

Note Limit the bulk job import to 1000 records at any given instance.

URL

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

Operations

- create: Creates one bulk job.
  - You can update the existing agent and supervisor usernames to SSO usernames using the **`ssomigration`** jobType.
- 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:
  - ssomigration

Returns the ssomigration CSV template as a file attachment. The template includes the first 12,000 agent and supervisor accounts that are not enabled for SSO.

If the database contains more than 12,000 non-SSO users, get the first 12,000 and update their credentials for SSO. Then get the next 12,000 users and update their credentials for SSO. Repeat until there are no more agents to migrate.

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>.`

- agentservice_association

  The sample CSV template returns the list of agents and the associated Contact Center AI service types.

- list: Retrieves a list of bulk jobs.

Parameters

- refURL: The refURL of the bulk job. See Shared Parameters, on page 7.
- description: See Shared Parameters, on page 7.
- 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 *Manage Bulk Jobs* section in the *Administration Guide for Cisco Unified Contact Center Enterprise* at https://www.cisco.com/c/en/us/support/customer-collaboration/unified-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 to run. 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.

  The valid value is 5, for SSO Migration.

  The valid value is 104, for Contact Center AI service association. You can create a bulk job and update the existing agent to enable or disable Contact Center AI services, using the **`agentservice_association`** jobType. This jobType is available for administrators only when Cloud Connect is added in the inventory.

- description: See Shared Parameters, on page 7.
- 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
• description	• description • jobType • jobState • jobHostName • createDateTime • startDateTime • endDateTime

See Search, on page 9 and Sort, on page 10.

Example Create Request

Example create request for enabling Agent Answers service for the agents:

```
<bulkJob>
<jobType>104</jobType>
<fileContent>agentId,userName,agentServices
1111,Agent1,AgentAnswers
2222,Agent2,AgentAnswers
</fileContent>
</bulkJob>
```

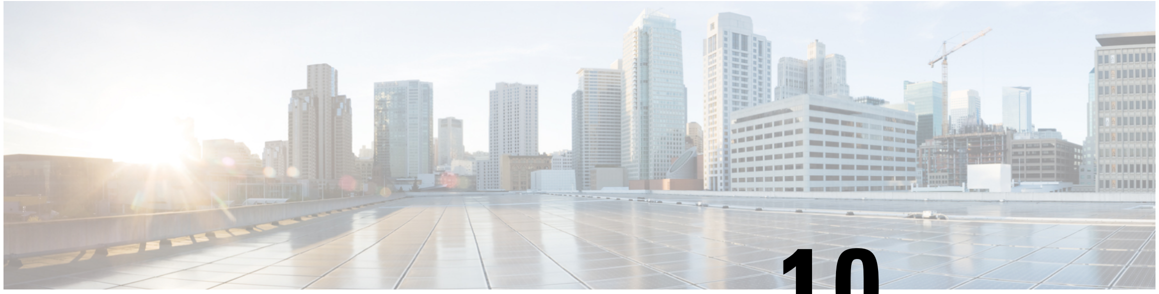
Example create request for enabling Agent Answers and Transcript services for the agents:

```
<bulkJob>
<jobType>104</jobType>
<fileContent>agentId,userName,agentServices
1111,Agent1,AgentAnswers;Transcript
2222,Agent2,AgentAnswers;Transcript
</fileContent>
</bulkJob>
```

Example Get Response

Example response for a SSO migration create job that completed successfully:

```
<bulkJob>
  <changeStamp>2</changeStamp>
  <refURL>/unifiedconfig/config/bulkjob/(id)</refURL>
  <jobHostName>CCE_AW</jobHostName>
  <createDateTime>1461785074000</createDateTime>
  <startDateTime>1461785074000</startDateTime>
  <endDateTime>1461785074000</endDateTime>
  <jobState>3</jobState>
  <jobType>5</jobType>
  <description>sso migration bulk job</description>
  <logFile>
    <refURL>/unifiedconfig/config/bulkjob/(id)/log</refURL>
  </logFile>
  <csvFile>
    <refURL>/unifiedconfig/config/bulkjob/(id)/csv</refURL>
  </csvFile>
</bulkJob>
```



CHAPTER 10

Business Hours

Business Hours API

Use this API to list the currently defined business hours, define new business hours, and view, edit, and delete the existing business hours.

URL

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

Operations

- create: Creates one business hour.
- get: Returns one business hour using the URL `https://<server>/unifiedconfig/config/businesshour/<id>`
- get(template): Returns a sample CSV template for Special Hours and Holidays, which is provided by the API, using the URL `https://<server>/unifiedconfig/config/businesshour/specialdayschedule/template`. The response contains the CSV template as a file attachment.
- list: Retrieves a list of business hours.
- update: Updates one business hour.
- delete: Permanently deletes one business hour.

Parameters

- changeStamp: See Shared Parameters, on page 7.
- refURL: The refURL of the business hour. See Shared Parameters, on page 7.
- name: The name of the business hour. See Shared Parameters, on page 7.

- description: See Shared Parameters, on page 7.
- type: Required. The type of the business hour.
    - 0: 24x7
    - 1: Custom
- configuredStatus: The configured status of the business hour.
    - status: Required.
        - 0: Calendar Schedule
        - 1: Force Close
        - 2: Force Open
    - statusReason: Required if the status is Force Open or Force Close.
- runTimeStatus: The run time status of the business hour. Available only in Get. It cannot be set or updated.
- timezone: Required. The time zone of the business hour.
- weekDaySchedules: The list of schedules on weekdays of business. Required only when business hour type is Custom.
    - dayOfweek: Required.
        - 0: Sunday
        - 1: Monday
        - 2: Tuesday
        - 3: Wednesday
        - 4: Thursday
        - 5: Friday
        - 6: Saturday
    - startTime: Required. Format HH:MM
    - endTime: Required. Format HH:MM
- specialDaySchedules: The list of schedules on special days of business.
    - date: Required. Format: DD-MM-YYYY
    - startTime: Required, if the status is Open. Format: HH:MM
    - endTime: Required, if the status is Open. Format: HH:MM
    - description: Optional. Maximum of 255 characters.
    - status: Required.
        - 0: Closed

  - 1: Open

- statusReason: Required. The refURL to existing status reason.

Search and Sort Values

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

Search parameters	Sort parameters
• name • description	• name • runTimeStatus • runTimeStatusReason • type • timezone.dispName

See Search, on page 9 and Sort, on page 10.

Example Get Response

```
<businessHour>
    <refURL>/unifiedconfig/config/businesshour/1000</refURL>
    <changeStamp>0</changeStamp>
    <runTimeStatus>2</runTimeStatus>
    <runTimeStatusReason>closed</runTimeStatusReason>
    <name>credit_card</name>
    <type>1</type>
    <description>Business Line of Credit</description>
    <timezone>
    <refURL>/unifiedconfig/config/timezone/3001</refURL>
    <displayName>(UTC-05:00) Eastern Time (US & Canada)</displayName>
    </timezone>
    <configuredStatus>
        <status>1</status>
        <statusReason>
            <refURL>/unifiedconfig/config/businesshourstatusreason/5000</refURL>
            <reasonText>Emergency</reasonText>
        </statusReason>
    </configuredStatus>
    <specialDaySchedules>
        <specialDaySchedule>
          <refURL>/unifiedconfig/config/businesshour/1000/specialdayschedule/2001</refURL>

            <changeStamp>0</changeStamp>
            <date>04-02-2019</date>
            <startTime>03:26</startTime>
            <endTime>04:16</endTime>
            <description>Thanksgiving Day</description>
            <status>1</status>
            <statusReason>
                <refURL>/unifiedconfig/config/businesshourstatusreason/5000</refURL>
                <reasonText>Emergency</reasonText>
            </statusReason>
        </specialDaySchedule>
    </specialDaySchedules>
    <weekDaySchedules>
```

```
        <weekDaySchedule>
            <refURL>/unifiedconfig/config/businesshour/1000/weekdayschedule/1001</refURL>
            <changeStamp>0</changeStamp>
            <dayOfWeek>1</dayOfWeek>
            <startTime>06:20</startTime>
            <endTime>07:22</endTime>
        </weekDaySchedule>
    </weekDaySchedules>
</businessHour>
```

Business Hour Status Reason API

Use this API to list the currently defined business hour status reasons, define new status reasons, and view, edit, and delete the existing status reasons.

URL

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

Operations

- create: Creates one business hour status reason.
- get: Returns one business hour status reason using the URL `https://<server>/unifiedconfig/config/businesshourstatusreason/<id>`
- list: Retrieves a list of business hour status reasons.
- update: Updates one business hour status reason.
- delete: Permanently deletes one business hour status reason.

Parameters

- category: Optional. The category of the reason codes: User-defined or System-defined.
- reasonCode: Required. The unique status reason code for business hour. Range is 1001 to 65535.

Note Codes 1 to 1000 are reserved as system defined reason codes. The System-defined reason codes cannot be deleted.

- reasonText: Optional. The reason for the business schedule. Maximum of 255 characters.

Example Create Request

```
<businessHourStatusReason>
 <category>User-defined</category>
<reasonCode>1234</reasonCode>
<reasonText>Emergency</reasonText>
</businessHourStatusReason>
```

Example Get Response

```
<businessHourStatusReason>
 <refURL>/unifiedconfig/config/businesshourstatusreason/5000</refURL>
 <changeStamp>0</changeStamp>
 <category>User-defined</category>
 <reasonCode>2550</reasonCode>
 <reasonText>Open</reasonText>
 </businessHourStatusReason>
```

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 Business Hours API to set the default time zone for a business hour.

This API is read-only.

URL

`https://<server>/unifiedconfig/config/timezone/v2`

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/v2/<id>`.

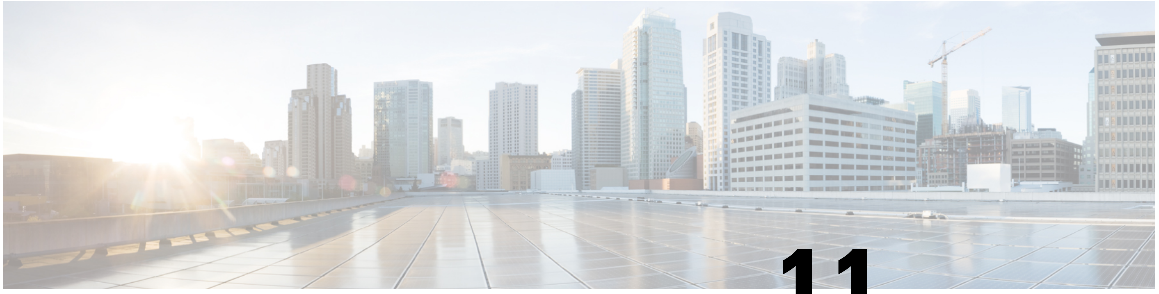
Response Parameters

- 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-05:00) Eastern Time (US & Canada)”
- changeStamp: See Shared Parameters, on page 7.

Example Get Response

```
<timezone>
    <refURL>/unifiedconfig/config/timezone/v2/5000</refURL>
    <changeStamp>0</changeStamp>
    <displayName>(UTC-05:00) Eastern Time (US & Canada)</displayName>
    <name>UTC</name>
</timezone>
```



CHAPTER 11

Call Type

Call Type

Call types categorize calls. Based on call type, the system maps a dialed number 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

- 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 7.
- name: The name of the call type. See Shared Parameters, on page 7.
- changeStamp: See Shared Parameters, on page 7.
- description: See Shared Parameters, on page 7.
- ccaiConfigID: Identifier for the Contact Center AI (CCAI) configuration saved in the Control Hub. Used to map the call type with the CCAI configuration. This parameter is available for administrators only when Cloud Connect is added in the inventory.
- 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.
- 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 33), including the refURL and name.
- survey: Is used to map call type with survey type.

  The two Survey types are

  1. Questionnaire Name: Is used to identify the **Inline Survey** which is Digital Channel (Email and Chat) and Post Call Voice Survey.
  2. Dispatch Id: Is used to identify **Deferred Survey** which is Post Call Email and SMS Survey.

Note You can associate only one survey to a call type.

The above parameters will be available only if you have added **Cloud Connect** in the Inventory page, and configured **Webex Experience Management** in the system.

Search and Sort Values

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

Search parameters	Sort parameters
• name • description • id	• name (default) • description • id • serviceLevelThreshold • serviceLevelType

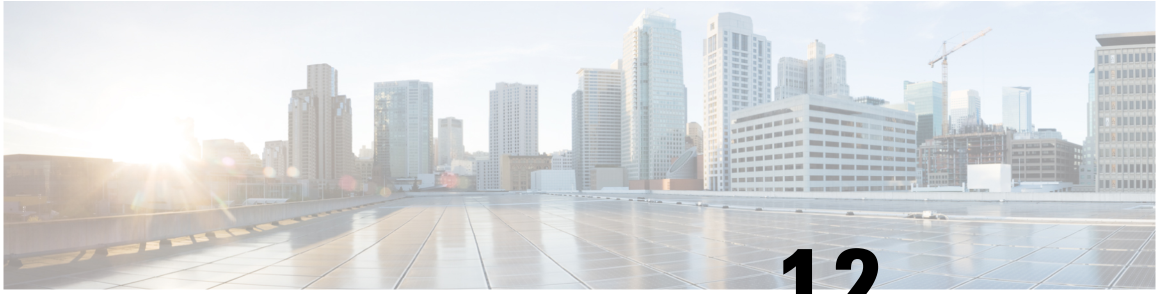
See Search, on page 9 and Sort, on page 10.

Example Get Response

```
 <callType>
<department>
<refURL>/unifiedconfig/config/department/5001</refURL>
<name>Sales</name>
```

```
</department>
<refURL>/unifiedconfig/config/calltype/(id)</refURL>
<name>test</name>
<description>test call type</description>
<id>5002</id>

<ccaiConfigID>Serviceconfig</ccaiConfigID>
<serviceLevelThreshold>10</serviceLevelThreshold>
<serviceLevelType>1</serviceLevelType>
<changeStamp>0</changeStamp>
<bucketInterval>
<refURL>/unifiedconfig/config/bucketinterval/(id)</refURL>
<name>bucket1</name>
</bucketInterval>
<survey>
<questionnaireName>1d222cb3</questionnaireName>
 OR <dispatchId>115097d3-ea65-432b-b90a-08aa7e5de361</dispatchId>
</survey>
</callType>
```



CHAPTER 12

Congestion Control API

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 or edit 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`.
- update: Updates the congestion control parameters.

Parameters

- deploymentType: The type of deployment.
- 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>
```

CHAPTER 13

Contact Center AI API

Global Config API

Use the Contact Center AI (CCAI) Global Config API to retrieve the global configuration for AI services. This API is available for administrators only when Cloud Connect is added in the inventory and is registered.

URL

Operations

- get: returns the global CCAI configuration.

  ```
  https://<server>/unifiedconfig/config/contactcenterai/globalconfig
  ```

- post: syncs the default global configurations with the latest available from the control hub.

  ```
  https://<server>/unifiedconfig/config/contactcenterai/globalconfig/sync
  ```

Parameters

- `name`: Configuration Name.
- `value`: Configuration Value.
- `status`: Indicates if the values are in sync with Control Hub.
- `lastSyncTime`:Time when config data was last synced with Control Hub

Example Get Response

```
<globalConfigDetail>
    <globalConfigs>
        <globalConfig>
            <name>CCAI.GlobalConfigId</name>
            <value>AXgB4em4bwWpdn7vJcLC</value>
        </globalConfig>
        <globalConfig>
            <name>CCAI.GlobalConfigName</name>
```

```
            <value>SS5T1</value>
        </globalConfig>
    </globalConfigs>
    <lastSyncTime>2021-03-05T16:07:43.497+05:30</lastSyncTime>
    <status>IN_SYNC</status>
</globalConfigDetail>
```

Config API

Control Hub API is a proxy to Control Hub CMS service to fetch Contact Center AI configuration.

URL

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

Operations

- get: Returns Contact Center AI config corresponding to <id> using the URL

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

- list: Retrieves list of Contact Center AI configs.

Example Get Response

```
 <ccaiconfig>
<conversationProfileId>sdfsd78bmplj89</conversationProfileId>
<defaultVirtualAgent>false</defaultVirtualAgent>
<connectorId>XK123UIU6887787JLK</connectorId>
<defaultAnswers>true</defaultAnswers>
<name>CCAIConfig2</name>
<description>Sample CCAI Config2</description>
<id>AXQlSeOzECsy_j49EVRz</id>
<type>Cisco</type>
<orgId>6d9069aa-76ce-45d6-a799-d38e60e92788</orgId>
</ccaiconfig>
```

Example List Response

```
 <ccaiconfigs>
<ccaiconfig>
<conversationProfileId>sdfsd78bmplj89</conversationProfileId>
<defaultVirtualAgent>false</defaultVirtualAgent>
<connectorId>XK123UIU6887787JLK</connectorId>
<defaultAnswers>true</defaultAnswers>
<name>CCAIConfig2</name>
<description>Sample CCAI Config2</description>
<id>AXQlSeOzECsy_j49EVRz</id>
<type>Cisco</type>
<orgId>6d9069aa-76ce-45d6-a799-d38e60e92788</orgId>
</ccaiconfig>
<ccaiconfig>
<conversationProfileId>asdf8768mnnb89</conversationProfileId>
<defaultVirtualAgent>true</defaultVirtualAgent>
<connectorId>XK123UIU6123567JLK</connectorId>
<defaultAnswers>false</defaultAnswers>
<name>CCAIConfig3</name>
<description>Sample CCAI Config3</description>
<id>AXQlSjp5ECsy_j49EVSp3</id>
```

```
<type>Google</type>
<orgId>6d9069aa-76ce-45d6-a799-d38e60e92788</orgId>
</ccaiconfig>
<ccaiconfig>
<conversationProfileId>qwewq556sad8asd</conversationProfileId>
<defaultVirtualAgent>false</defaultVirtualAgent>
<connectorId>XK123UIU6778787JLK</connectorId>
<defaultAnswers>false</defaultAnswers>
<name>CCAIConfig</name>
<description>Sample CCAI Config</description>
<id>AXQlSmksECsy_j49EVSs</id>
<type>Google</type>
<orgId>6d9069aa-76ce-45d6-a799-d38e60e92788</orgId>
</ccaiconfig>
</ccaiconfigs>
```

CHAPTER 14

Contact Share Group API

Contact Share Group API

The Contact Share group is a routing entity that operates on a single Contact Share rule with a set of Contact Share queues. It is used to determine to which target Unifed CCE to send the call, given the definition of the rule and the realtime data on each of the queues. This API is only available on the Contact Director Deployment.

Note To learn more about the other Contact Share APIs available in the Contact Director Deployment, see Access, on page 2.

URL

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

Parameters

- changeStamp: See Shared Parameters, on page 7.
- refURL: The refURL of the Contact Share group. See Shared Parameters, on page 7.
- contactShareRule: Required. The refURL of a Contact Share rule to apply to this group.
- name: Required. The name of the Contact Share group.
- acceptQueueIf : Optional. A logical expression that qualifies individual queues to be used in the group.
- description: Optional. The description of the group.
- contactShareQueues: Optional. Information for any Queues in the Contact Share group.

Operations

- create: Creates one Contact Share group.
- delete: Deletes one Contact Share group.

- get: Returns one Contact Share group, using the URL https://<server>/unifiedconfig/config/contactsharegroup<id>.
- list: Retrieves a list of Contact Share group.
- update: Updates one Contact Share group.

Search and Sort Values

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

Search parameters	Sort parameters
• name • description • groupExpression	• name(default) • description • groupExpression

See Search, on page 9 and Sort, on page 10.

Example Get Response

```
<contactShareGroup>
   <changeStamp>0</changeStamp>
   <refURL>/unifiedconfig/config/contactsharegroup/5000</refURL>
   <contactShareRule>
      <name>Cisco_MED</name>
      <refURL>/unifiedconfig/config/contactsharerule/5000</refURL>
   </contactShareRule>
   <name>test</name>
   <acceptQueueIf>Call.PeripheralVariable1==1</acceptQueueIf>
   <description>test something</description>
   <contactShareQueues>
      <contactShareQueue>
         <refURL>/unifiedconfig/config/contactsharequeue/5000</refURL>
         <name>CSQ-1</name>
         <queueType>skillGroup</queueType>
         <targetInstance>
            <id>5000</id>
            <name>cs01</name>
         </targetInstance>
      </contactShareQueue>
   </contactShareQueues>
</contactShareGroup>
```

CHAPTER 15

Contact Share Rule API

Contact Share Rule API

A Contact Share rule is required when configuring a Contact Share group. A group is sent in every routing request to the Contact Share process. Each group has a rule that defines the logic for selecting a skill group or precision queue in that group for a routing request. Use the Contact Share Rules API to create, update, list, and delete rules. This API is only available on the Contact Director deployment.

Note To learn more about the other APIs available in the Contact Director deployment, see Access, on page 2.

URL

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

Parameters

- refURL: The refURL of the Contact Share rule. See Shared Parameters, on page 7.
- name: Required. The name of the Contact Share rule.
- ruleExpression: Required. The logical expression with which the rule operates.
- description: Optional. The description of the rule.
- changeStamp: See Shared Parameters, on page 7.

Operations

- create: Creates one Contact Share rule.
- delete: Permanently deletes one Contact Share rule.
- get: Returns one Contact Share rule, using the URL https://<server>:/unifiedconfig/config/contactsharerule<id>.
- list: Retrieves a list of Contact Share rules.

- update: Updates one Contact Share rule.

Search and Sort Values

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

Search parameters	Sort parameters
• name • description • ruleExpression	• name(default) • description • ruleExpression

See Search, on page 9 and Sort, on page 10.

Example Get Response

```
<contactShareRule>
  <refURL>/unified/config/contactsharerule/(id)</refURL>
  <name>test</name>
  <ruleExpression>Call.PeripheralVariable==1</ruleExpression>
  <description>test something</description>
  <changeStamp>0</changeStamp>
</contactShareRule>
```

CHAPTER 16

Cloud Connect Device Configuration

Cloud Connect Configuration

This is global API for *cloudconnector* (same configuration will be applicable for all the cloud connect present in the system).

The Cloud Connect API is applicable for the following deployments only:

- Unified CCE 2000 Agents.
- PCCE 2000 Agents and
- PCCE LabOnly

URL

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

Operations

- **Get**:

  Gets the configuration information of the cloud connect`https://<server>/unifiedconfig/config/cloudconnectsettings`

- **Put**:

  Updates the proxy server and media type mapping to cloud connect. `https://<server>/unifiedconfig/config/cloudconnectsettings`

  with the following body.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><CloudConnectSettings>
    <registrationStatus>Registered</registrationStatus>

    <proxyAddress>http://10.10.10.10:80</proxyAddress>

    <deploymentID>9de8bdfe-d513-4ab9-a021-2b729efa9c63</deploymentID>
    <deploymentName>CloudConnectSettings</deploymentName>
    <changeStamp>6</changeStamp>
</CloudConnectSettings>
```

Parameters

- dataGenerationEnabled: Enables data generation for Cloud Connect.
- dataSourceAddress: DataSource address for Cloud Connect.
- dataSourceUserName: DataSource user name for Cloud Connect.
- dataSourcePassword: DataSource password for Cloud Connect.
- proxyAddress: Proxy address or hostname for the Cloud Connect to connecct to the Control Hub.
- streamerEnabled: Enables the data flow between Cloud Connect and Control Hub.

Example Get Response

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><CloudConnectSettings>
    <registrationStatus>Registered</registrationStatus>

    <proxyAddress>http://10.10.10.10:80</proxyAddress>

    <deploymentID>9de8bdfe-d513-4ab9-a021-2b729efa9c63</deploymentID>
    <deploymentName>CloudConnectSettings</deploymentName>
    <changeStamp>6</changeStamp>
</CloudConnectSettings>
```

CHAPTER 17

Deployment API

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.
- 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>
```

CHAPTER 18

Deployment Type Info API

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.
- 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>
```

CHAPTER 19

ECC Payload API

ECC Payload API

You use expanded call variables to store values that are associated with a call. These variables are commonly called Expanded Call Context (ECC) variables. While you can define many more ECC variables in the system, you can pass a maximum of 2000 bytes of these variables at any time over any interface. To enable you to pass different ECC variables at different times, the solution has ECC payloads.

An ECC payload is a defined set of ECC variables. You can create ECC payloads to suit the necessary information for a given operation. You can include an ECC variable in multiple ECC payloads. The particular ECC variables in a given ECC payload are called its members.

Note

- ECC payload ID 1 is the default ECC payload. The API prevents the deletion of this ECC payload.
- In solutions that only use the default ECC payload, the Logger does not create an ECC variable that exceeds the 2000-byte limit for an ECC payload or the 2500-byte CTI Message Size limit. The Logger does this because it automatically adds all ECC variables to the default ECC payload if that is the only ECC payload.

  If you create another ECC payload, the Logger no longer checks the 2000-byte limit when creating ECC variables. The Logger creates the ECC variables without assigning them to an ECC payload. Assign the new ECC variable to an appropriate ECC payload yourself, either through the ECC Payload API or through the ECC Payload Tool.
- For create and update operations, the ECC Payload API verifies that the members of the ECC payload do not exceed the 2000-byte limit.

Important

During upgrades, when the system first migrates your existing ECC variables to the Default payload, it does not check the CTI message size limit. The member names might exceed the extra 500 bytes that is allocated for ECC payloads to a CTI client. Manually check the **CTI Message Size** counter in the **Expanded Call Variable Payload List** tool to ensure that the Default payload does not exceed the limit. If the Default payload exceeds the limit, modify it to meet the limit.

URL

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

Operations

- create: Creates one ECC payload and stores it in the database.
  - **Query Parameters**
    - name: The name of the ECC payload (Required)
    - description: A description of the ECC payload (Optional)
    - variables: 0 or more ECC variables (Optional)

Note Specify each variable with the `ecc.refURL` of a valid, non-deleted ECC variable.

Note This API supports synchronous and asynchronous create operations.

- delete: Permanently deletes one ECC payload and all its members from the database.
- get: Returns one ECC payload from the database, using the URL `https://<server>/unifiedconfig/config/eccpayload/<id>`.
- list: Retrieves a list of ECC payloads.
- update: Updates one ECC payload in the database.
  - **Query Parameters**
    - changeStamp: The change stamp for the ECC payload record which the GET returns (Required)
    - refURL: The `refURL` of the ECC payload to update (Required)
    - name: The name of the ECC payload (Optional)
    - description: A description of the ECC payload (Optional)
    - variables: 0 or more ECC variables (Optional)

Note Specify each variable with the `ecc.refURL` of a valid, non-deleted ECC variable.

Parameters

- refURL: The refURL of the ECC payload. See Shared Parameters, on page 7.
- name: The name of the ECC payload. See Shared Parameters, on page 7.

- changeStamp: See Shared Parameters, on page 7.
- description: See Shared Parameters, on page 7.
- variables: The members of the ECC payload.

Search and Sort Values

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

Search parameters	Sort parameters
• name • description	• name (default) • description

See Search, on page 9 and Sort, on page 10.

Note This API does not support advanced search parameters.

Example Get Response

```
<eccpayload>
  <refURL>/unifiedconfig/config/eccpayload/1</refURL>
  <changeStamp>18</changeStamp>
  <description>Initial default data.</description>
  <name>Default</name>
  <variables>
    <variable>
      <refURL>/unifiedconfig/config/expandedcallvariable/8</refURL>
      <name>POD.ID</name>
    </variable>
    <variable>
      <refURL>/unifiedconfig/config/expandedcallvariable/5009</refURL>
      <name>user.A1</name>
    </variable>
    <variable>
      <refURL>/unifiedconfig/config/expandedcallvariable/5010</refURL>
      <name>user.A2</name>
    </variable>
  </variables>
</eccpayload>
```

CHAPTER 20

Expanded Call Variable API

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 view and list the expanded call variables currently defined in the database.

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 7.
- name: The name of the expanded call variable. See Shared Parameters, on page 7.
- changeStamp: See Shared Parameters, on page 7.
- description: See Shared Parameters, on page 7.
- maximumLength: The maximum length of the expanded call variable. The value is 1 to 210.

Note **Cisco Provided** POD.ID variable, **Maximum length** can be modified.

- 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.
- 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
• name • description	• name (default) • description • maximumLength • maximumArraySize • eccArray • enabled • persistent • ciscoProvided

See Search, on page 9 and Sort, on page 10.

Example Get Response

```
<expandedCallVariable>
    <refURL>/unifiedconfig/config/expandedcallvariable/(id)</refURL>
    <name>test</name>
    <maximumLength>9</maximumLength>
    <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>
```

CHAPTER 21

General Setting API

General Setting API

GeneralSetting API is introduced to include the session inactivity timeout field. Both GET and PUT are available in all supported deployments.

URL

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

Operations

- **get**: Returns the general settings configured in the system.
- **put**: To update any general setting applicable in the system.

Parameters

- sessioninactivitytimeout: Timeout after which user is logged out if the user is inactive

Example Get Request

```
<generalSettings>
    <loginSession>
        <sessionInactivityTimeout>
            30
        </sessionInactivityTimeout>
    </loginSession>
</generalSettings>
```

Example Put Request

```
<generalSettings>
    <loginSession>
        <sessionInactivityTimeout>
            60
        </sessionInactivityTimeout>
    </loginSession>
</generalSettings>
```

CHAPTER 22

Internet Script Editor API

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>
```

CHAPTER 23

Machine Inventory API

Machine Inventory API

This API returns the machines in the solution.

URL

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

Operations

- create: Creates a machine by updating the database. See the following table for restrictions per machine type.
- delete: Removes one machine.
- get: Returns one machine and all associated addresses and services based on the machine ID, using the URL `https://<server>/unifiedconfig/config/machineinventory/<id>`.
- list: Retrieves a list of all machines in the inventory. See the following table for more details.
- update: Updates one machine.

Type	Create / Update / Delete operations allowed	Number allowed
CCE_ROUTER	No	0/1 Side A 0/1 Side B
CCE_PROGGER	No	0/1 Side A 0/1 Side B
CCE_CALL_SERVER	No	0/1 Side A 0/1 Side B

Type	Create / Update / Delete operations allowed	Number allowed
CCE_PG	No	0/150 Side A 0/150 Side B
LIVE_DATA	All	0/1 Side A 0/1 Side B
PRIMARY_AW	All	0/2 Primary
SECONDARY_AW	All	0/2 Secondary
CCE_AW	Update only	0/8 Primary
IDS_PUBLISHER	All	0 or more
IDS_SUBSCRIBER	No	0 or more
FINESSE_PRIMARY	All	0 or more
FINESSE_SECONDARY	Update and Delete only	0 or more
CUIC_PUBLISHER	All	0/1
CUIC_SUBSCRIBER	No	0/1
CUIC_PUBLISHER_STANDALONE	All	0 or more
CUIC_SUBSCRIBER_STANDALONE	Update and Delete only	0 or more
EXTERNAL_CUSTOMER_COLLABORATION_PLATFORM	All	0/1
CLOUD_CONNECTOR_PUB	All	0-1
CLOUD_CONNECTOR_SUB	No	0-1

Note When you create a CUIC_PUBLISHER, CUIC_PUBLISHER_STANDALONE, or FINESSE_PRIMARY machine, any other machine in the cluster (CUIC_SUBSCRIBER, CUIC_SUBSCRIBER_STANDALONE, or FINESSE_SECONDARY) is automatically added to the inventory. When you delete a CUIC_PUBLISHER, CUIC_PUBLISHER_STANDALONE, or FINESSE_PRIMARY machine, any other machines in the cluster are also removed from the inventory.

CUIC_PUBLISHER and CUIC_SUBSCRIBER have Unified Intelligence Center, Live Data, and Cisco Identity Service installed on the same VM. These machines can only be used in the following deployment types:

- UCCE: 2000 Agents
- UCCE: Progger (Lab Only)

If you create a CUIC_PUBLISHER on a system that already contains a LIVE_DATA machine, the LIVE_DATA machine is deleted. If the system already contains a CUIC_PUBLISHER_STANDALONE or IDS_PUBLISHER machine, you receive an error message of error type INVENTORY_ONLY_ONE_CUIC_PUB_ALLOWED.

Parameters

Machine parameters:

- refURL: The refURL of the machine. See Shared Parameters, on page 7.
- name: External name of the machine. For example, the VM hostname. 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 7.
- type: The type of machine.
- autogenerated: Indicates if the information was generated automatically.
- hostName: The hostname 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 section.

Network parameters:

- type: Must be public.
- versionInfo: Version info available depending on the product type. Includes the following parameters (by product type):
    - UCCE: version, buildNumber, esNumber, patchVersion

  - VOS: version and buildNumber

- address: The IP address. Must be valid hostname, IPv4, or IPv6 address.
- services: A collection of service parameters. See the Services Parameters section.

Services parameters:

- uri: The service URL, for example MR_PG_A URL.

  For MR_PG_A and MR_PG_B, the machines on which the MR PGs reside must be configured in the Machine Inventory:

  - Most Unified CCE for Contact Center deployments use the CCE_PGs.
  - Progger deployments use the CCE_PROGGERs.

  The URLs must be different, and must follow the pattern `/unifiedconfig/config/machineinventory/<id>.`

- name: The hostname of service. This read-only parameter applies only to the MR_PG_A and MR__PG_B services. It is returned if the uri matches the unifiedconfig machine host (for example `/unifiedconfig/config/machineinventory/1234` where `1234` is a valid id).
- port: The port for this service.

  For MR_PG_A and MR_PG_B, the default port is 38001, and the valid range is 10000 to 65535. If you do not specify a value or specify 0, the default port is used. MR_PG_A and MR_PG_B must use the same port.

- 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.
- description: The description of the service. See Shared Parameters, on page 7.
- pairing: Indicates if services on different machines are related. Related services have a matching value.

  For Contact Share node, if the service type is ACTIVE_MQ, the pairing value contains the valid ApplicationGatewayID.

  For the service type PRINCIPAL_AW, the pairing value is "false" for all AWs.

- type: The service type. Values for type are as follows:
  - LIVE_DATA_API: Connection information for the Live Data Rest API. Requires a username and password.
  - LIVE_DATA_CASSANDRA: Connection information for the Live Data Cassandra database.
  - LIVE_DATA_WEB_SOCKETS: Connection information for the Live Data Socket.IO service.
  - TIP_PG: LiveData connection information for peripheral gateways.
  - TIP_ROUTER: LiveData connection information for the router.
  - TIP_PG_TOS: LiveData test-other-side connection information for peripheral gateways.
  - TIP_ROUTER_TOS: LiveData test-other-side connection information for the router.

- ACTIVE_MQ: LiveData Active MQ connection information.
- STORM_DRPC: LiveData Storm DRPC connection information.
- AW_REST_API: Administration & Data Server (AW) Rest API
- SM_REST_API: Customer Collaboration Platform REST API. Requires a username and password.
- MR_PG_A: Connection information for Side A MR PG. Required for Customer Collaboration Platform machines.
- MR_PG_B: Connection information for the Side B MR PG. Required for Customer Collaboration Platform machines.
- MR_PG_CONNECTION: Connection information for the MR PG. Required for Customer Collaboration Platform machines.
- IDS: Indicates an Identity Server service. IDS contains the following parameters:
    - userName: The username of the IdS for the IDS_PUBLISHER machine.
    - 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.

  When you update the IDS_PRIMARY_REF service type of a machine, the IDS_SECONDARY_REF is automatically set to the other IdS in the cluster, if present.
- 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).
- PUBLISHER_REF: The <uri> parameter for this service refers to the publisher/primary machine's IP Address. This service is only applicable for subscriber/secondary machines.
- CLOUD_CONNECT_CREDINITIALS: Credinitial information for Cloud Connect Services.

Search and Sort Values

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

Search parameters	Sort parameters
• name • description • hostName	• name (default) • description • hostName

See Search, on page 9 and Sort, on page 10.

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_CUSTOMER_COLLABORATION_PLATFORM** returns only external Customer Collaboration Platform 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 Update Request

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

```
<machine>
  <name>myLiveDataMachine</name>
  <hostName>liveDataHostName</hostName>
  <type>LIVE_DATA</type>
  <networks>
    <network>
      <type>PUBLIC</type>
      <address>1.2.3.4</address>
      <services>
        <service>
          <type>ACTIVE_MQ</type>
          <port>61616</port>
          <pairing>1</pairing>
        </service>
        <service>
          <type>STORM_DRPC</type>
          <port>3772</port>
          <pairing>1</pairing>
        </service>
        <service>
          <type>LIVE_DATA_WEB_SOCKETS</type>
          <port>12008</port>
          <pairing>1</pairing>
        </service>
        <service>
          <type>LIVE_DATA_API</type>
          <port>12005</port>
          <pairing>1</pairing>
          <userName>user</userName>
          <password>password</password>
        </service>
        <service>
          <type>LIVE_DATA_CASSANDRA</type>
          <port>12000</port>
          <pairing>1</pairing>
        </service>
      </services>
    </network>
  </networks>
</machine>
```

Example Get Response

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

```
<machine>
  <changeStamp>3</changeStamp>
  <refURL>/unifiedconfig/config/machineinventory/5000</refURL>
  <networks>
    <network>
      <address>1.2.3.4</address>
      <services>
        <service>
          <autoGenerated>true</autoGenerated>
          <pairing>0</pairing>
          <port>40034</port>
          <type>TIP_ROUTER</type>
        </service>
        <service>
          <autoGenerated>true</autoGenerated>
          <pairing>0</pairing>
          <port>40035</port>
          <type>TIP_ROUTER_TOS</type>
        </service>
      </services>
      <type>PUBLIC</type>
    </network>
  </networks>
  <autoGenerated>true</autoGenerated>
  <hostName>routerA</hostName>
  <type>CCE_ROUTER</type>
  <name>routerA</name>
</machine>
```

Example Get Response for IdS

```
<machine>
  <changeStamp>1</changeStamp>
  <refURL>/unifiedconfig/config/machineinventory/5000</refURL>
  <name>IDS-PUB-A</name>
  <type>IDS_PUBLISHER</type>
  <networks>
    <network>
      <type>PUBLIC</type>
      <address>10.10.10.22</address>
      <services>
       <service>
        <type>IDS</type>
        <userName>Admin</userName>
        <components>
           <component>
              <refURL>/unifiedconfig/config/machineinventory/6000</refURL>
              <name>FIN-PUB-A</name>
           </component>
           <component>
              <refURL>/unifiedconfig/config/machineinventory/6001</refURL>
              <name>CUIC-PUB-A</name>
            </component>
         </components>
       </service>
      </services>
    </network>
  </networks>
</machine>
```

Example Update Request for Primary IdS

This example request updates the primary IdS for Finesse:

```
<machine>
  <changeStamp>1</changeStamp>
  <refURL>/unifiedconfig/config/machineinventory/5000</refURL>
  <name>Finesse primary</name>
  <type>FINESSE_PRIMARY</type>
  <networks>
```

```
    <network>
      <type>PUBLIC</type>
      <address>10.10.10.21</address>
      <services>
       <service>
        <type>IDS_PRIMARY_REF</type>
        <uri>/unifiedconfig/config/machineinventory/5001</uri>
       </service>
      </services>
    </network>
  </networks>
</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 updates the IDS_PRIMARY_REF and IDS_SECONDARY_REF of any SSO-capable components.

This operation also deassociates 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>
```

CHAPTER 24

Media Routing Domain API

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 7.
- name: Name of the MRD. See Shared Parameters, on page 7.
- description: See Shared Parameters, on page 7.
- 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
• name • description • id	• name (default) • description • id • interruptible • maxTasksInQueue • maxTaskDuration • maxTimeInQueue • serviceLevelThreshold • taskLife • taskStartTimeout

See Search, on page 9 and Sort, on page 10.

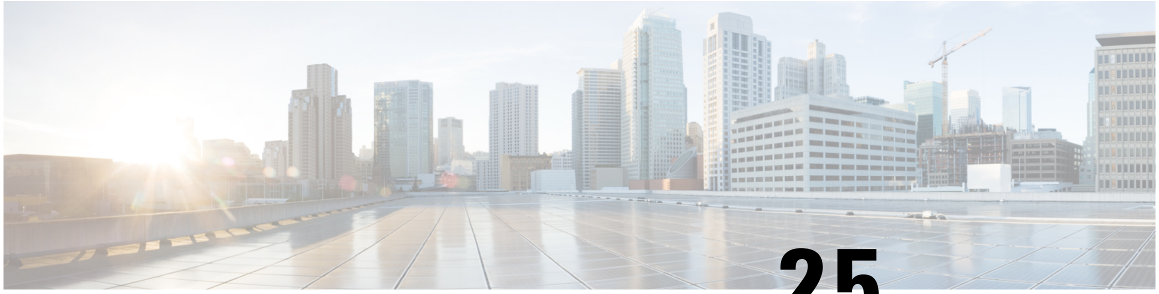
Advanced search parameters

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>
```



CHAPTER 25

Operation API

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 (available for supervisors and administrators).

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 7.
- changeset: Includes the parameters that are changed in an update operation. See Agent Call API.
  - skillGroupsAdded
  - skillGroupsRemoved
  - attributesAdded
  - attributesRemoved
  - agentServicesToEnable: Indicates the Contact Center AI services to be enabled for a set of agents.

- agentService: The type of Contact Center AI service. Supported values are `AgentAnswers` and `Transcript`.

- agentServicesToDisable: Indicates the Contact Center AI service to be disabled for a set of agents.
  - agentService: The type of Contact Center AI service. Supported values are `AgentAnswers` and `Transcript`.

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>

      <agentServicesToEnable>
            <agentService>AgentAnswers<agentService>
            ....
        </agentServicesToEnable>
        <agentServicesToDisable>
            <agentService>Transcript<agentService>
            ....
        </agentServicesToDisable>


      <skillGroupsAdded>
        <skillGroup>
          <refURL>/unifiedconfig/config/skillgroup/6000</refURL>
        </skillGroup>
      </skillGroupsAdded>
      <skillGroupsRemoved>
        <skillGroup>
          <refURL>/unifiedconfig/config/skillgroup/6001</refURL>
        </skillGroup>
      </skillGroupsAdded>
      </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
      /calltype/9999</errorMessage>
      <errorType>operation.resourceErrors</errorType>
    </apiError>
  </apiErrors>
</operationsResult>
```

Request Limit

Maximum 100 items are allowed in a request:

Example Request Limit Response

```
<operationsResult>
<apiErrors>
    <apiError>
     <errorMessage>The number of resources provided (101) exceeds the maximum number allowed
 (100).</errorMessage>
      <errorType>operation.tooManyResources</errorType>
      </apiError>
      </apiErrors>
  <status>failure</status>
</operationsResult>
```

CHAPTER 26

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 API

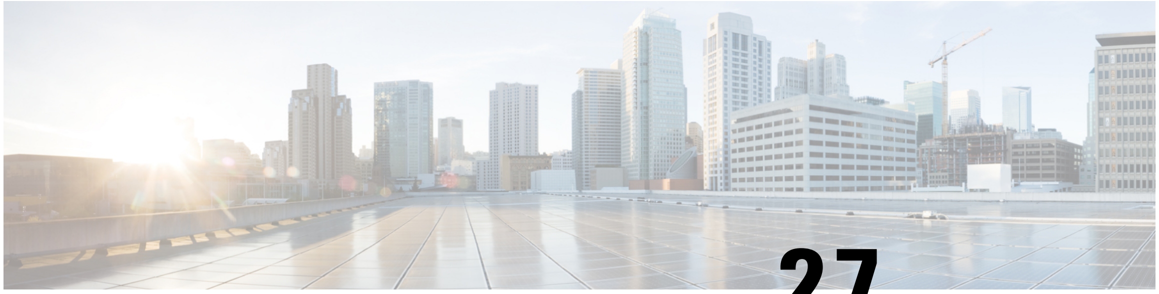
Outbound API allows you to use REST APIs to create, modify, and delete Outbound Option campaigns.

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.



CHAPTER 27

Precision Queue API

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 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 7.
- name: The name of the precision queue. See Shared Parameters, on page 7.
- changeStamp: See Shared Parameters, on page 7.
- description: See Shared Parameters, on page 7.
- mediaRoutingDomain: A reference to the media routing domain (Media Routing Domain API, on page 85) including the name and refURL. See References, on page 4.
  - 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 33), including the refURL and name. See References, on page 4.
- 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
- callOrdering: Determines the order of calls in this precision queue.
  - 1: Top priority in queue. This is the default value.
- 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 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 run 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 29), 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.

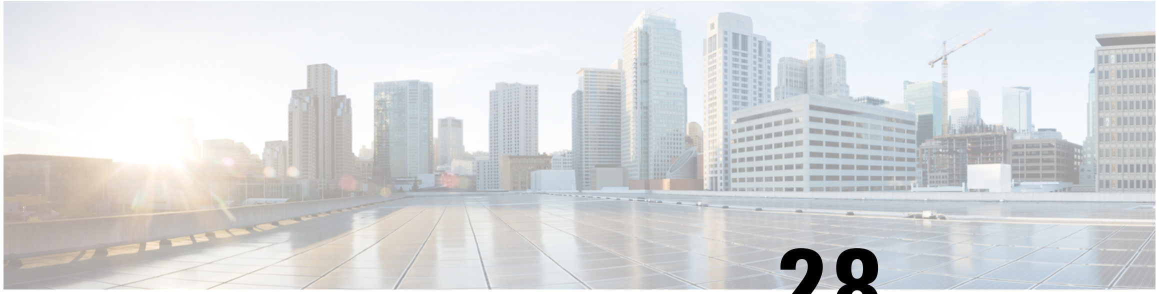
Search parameters	Sort parameters
• name • description	• name (default) • description

See Search, on page 9 and Sort, on page 10.

Example Get Response

```
<precisionQueue>

   <refURL>/unifiedconfig/config/precisionqueue/5002</refURL>
   <changeStamp>4</changeStamp>
   <agentOrdering>1</agentOrdering>
   <callOrdering>1</callOrdering>
   <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>
```



CHAPTER 28

Scan API

Scan API

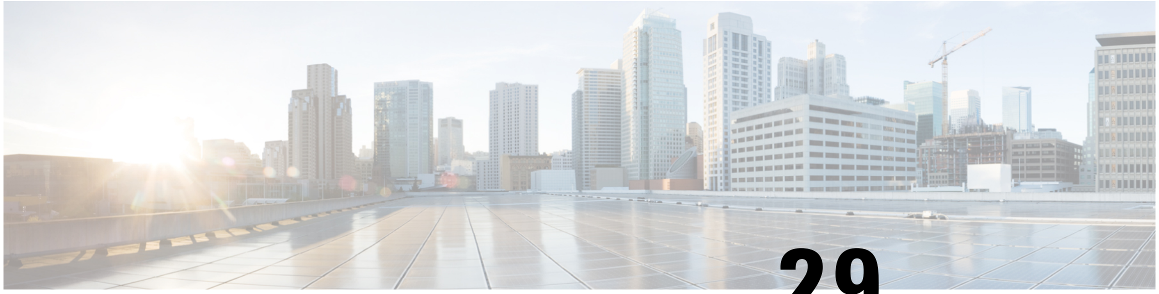
The Scan API forces the validation of the Customer Collaboration Platform external machine status rules.

URL

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

HTTP method

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



CHAPTER 29

Single Sign-On Global State API

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 105.

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 7.
- changeStamp: See Shared Parameters, on page 7.
- 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>
```

CHAPTER 30

Single Sign-On Registration API

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 105.

URL

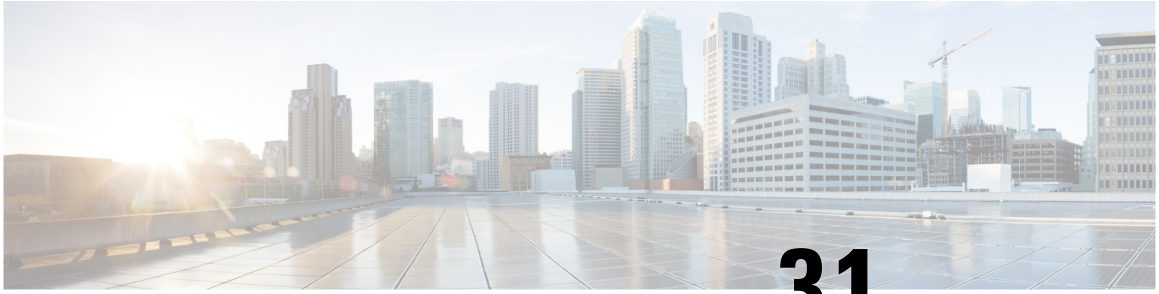
```
https://<server>/unifiedconfig/config/sso/register
```

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 77

Parameters

None



CHAPTER 31

Single Sign-On Status API

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 77. 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 7.
  - 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>
```

CHAPTER 32

Skill Group API

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, view, or edit existing skill groups.

Note Access to this API is different for supervisors and administrators. For more information, see Access, on page 2.

URL

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

Note `unifiedconfig` api does not support skill groups in System PGs

Operations

- 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 2.

- update: Updates one skill group.

Note

- After a skill group has been created, the data center cannot be modified.
- A skill group can only be associated with agents that are on the same data center as that skill group.

Parameters

- refURL: The refURL of the skill group. See Shared Parameters, on page 7.
- name: The name of the skill group. See Shared Parameters, on page 7.
- changeStamp: See Shared Parameters, on page 7.
- description: See Shared Parameters, on page 7.
- peripheral: Contains the name and id of the peripheral.
- mediaRoutingDomain: A reference to the media routing domain (Media Routing Domain API, on page 85) including the name and refURL. See References, on page 4.
    - 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 Call API). References also include firstName, lastName, agentId, and agentTeam (which includes the team name and refURL). See References, on page 4.
    - 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 4.
- 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 4.
- 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 33). Includes the name and refURL. See References, on page 4.
- serviceLevelThreshold: Maximum time in seconds that a caller should wait before being connected with an agent. Positive integers only, or blank.
- 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.
- datacenter: A reference to the data center, 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
• name • description	• name (default) • description • serviceLevelThreshold • serviceLevelType • peripheralNumber • datacenter.name • peripheral.name

See Search, on page 9 and Sort, on page 10.

For more information on search restrictions, see Search, on page 9.

Advanced Search Parameters

Advanced Search Parameters

- **peripherals: (peri1|peri2|peri3, ...)** which returns all SkillGroups that have any of the specified peripherals. Up to ten peripherals can be specified. The peripheral names are fully matched (case-insensitive, no partial matches).
- **datacenters: (dc1|dc2|dc3...)** which returns all the Skill Groups that belong to any of the specified data centers. Up to three data centers can be specified. The data center names are fully matched (case-insensitive, no partial matches). Searching for "core" returns all machines in the core data center.

Example Get Response

```
<skillGroup>
    <refURL>/unifiedconfig/config/skillgroup/(id)</refURL>
    <name>test</name>
    <description>test skill group</description>
    <changeStamp>0</changeStamp>
    <peripheral>
        <id>5001</id>
        <name>CUCM_PG1</name>
    </peripheral>
    <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>
    <agents>
        <agent>
            <refURL>/unifiedconfig/config/agent/5000</refURL>
            <firstName>Jane</firstName>
            <lastName>Doe</lastName>
            <userName>username</userName>
            <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>
```

CHAPTER 33

Status API

Configuration Rules

Unified CCE AW, Unified Intelligence Center, CUIC-LD-IdS, and Finesse Machines

These rules show the potential configuration errors and warnings for Unified CCE AW, Unified Intelligence Center, CUIC-LD-IdS, and Finesse machines.

Table 1: Unified CCE AW, Unified Intelligence Center, CUIC-LD-IdS, and Finesse 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.

Table 2: Unified CCE AW Rules

Rule	Description
TASK_ROUTING_APP_PATHS_EXIST	Each Peripheral Gateway with a Communications Manager PIM must have an associated Task Routing Application Path.

Customer Collaboration Platform Machines

These rules show the potential configuration errors and warnings for Customer Collaboration Platform machines.

Table 3: Customer Collaboration Platform Rules

Rule	Description
SOCIAL_MINER_MR_ENABLED	Multichannel routing must be enabled on Customer Collaboration Platform.

Cloud Connect Machines

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.

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

Table 4: Cloud Connect Rules

Rule	Description
CLOUD_CONNECT_STATUS	Shows the sync status of Cloud Connect Machine.

Operation Rules

Common 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.

These rules show the potential operation errors and warnings for the following machines:

- All Unified CCE components
- Unified Intelligence Center
- CUIC-LD-IdS
- Finesse
- Live Data
- Identity Service
- Customer Collaboration Platform

Table 5: Common Rules

Rule	Description
SERVER_CONNECTION	The Diagnostic Portal, AXL, REST, or SOAP service on this machine must be in service. The status you see varies according to the product type, as follows: • Unified CCE: The Cisco ICM Diagnostic Framework service on the Principal AW must be reachable on the network. • CUIC-LD-IdS: The SOAP service on the Publisher must be reachable on the network. • Finesse: The SOAP service on the Primary Finesse machine must be reachable on the network. • IdS: The REST service on the Primary Identity Server must be reachable on the network • Customer Collaboration Platform : The service must be reachable on the network.
SERVER_CREDENTIALS	The inventory's credentials for the machine must be valid. The status you see varies according to the product type, as follows: • Unified CCE: The Diagnostic Framework credentials entered for the Principal AW must be valid for all CCE components in the solution. • CUIC-LD-IdS: The Unified Intelligence Center Administration credentials entered for the Publisher must be valid • Finesse: The Finesse Administration credentials entered for the Primary Finesse machine must be valid. • IdS: The Identity Service Administration credentials entered for the Primary Identity Server must be valid • Customer Collaboration Platform : The service credentials must be valid. **Note** This rule does not apply to a standalone Live Data machine.

Cloud Connect Machines

These rules show the potential operational errors and warnings for Cloud Connector machines.

Table 6: Cloud Connect Rules

Rule	Description
CLOUD_CONNECTOR_TYPE	Cloud connect is either publisher or subscriber.

Rule	Description
CONTACT_CENTER_AI_CALL_TYPE_CONFIG_NOT_FOUND	Some of the Contact Center AI configurations that are associated with the call types are not available in the Control Hub.