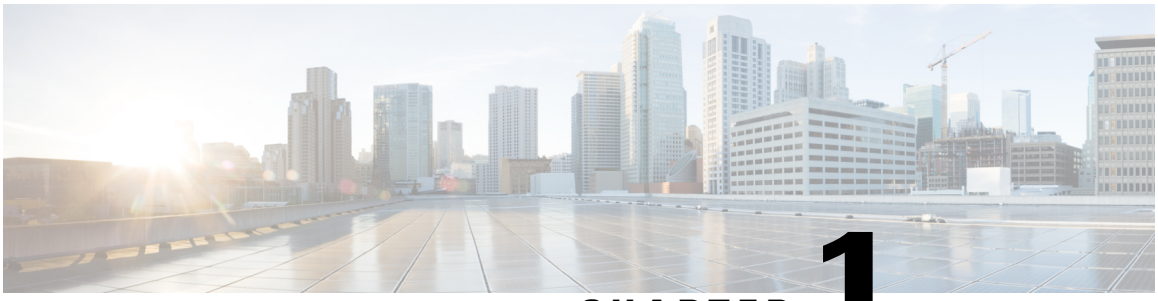
Cisco Packaged Contact Center Enterprise Developer Reference Release 10.0(x)

First Published: December 12, 2013

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

CONTENTS



CHAPTER 1

Working with Cisco Packaged Contact Center Enterprise APIs

Cisco Packaged Contact Center Enterprise (Packaged CCE) uses REST-based API functions accessed over HTTP. Each API Operation is mapped to an HTTP method. For more information, see API operations, on page 3.

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

For more information on managing the system, see the Packaged CCE Administration Guide.

To review examples on how to interact with the REST web service in Java, see the CCE Config Sample REST Toolkit.

Change log

This section notes the new and changed APIs in the Packaged CCE 10.0(x) release.

New APIs

- Administrator API, on page 15
- Department API, on page 43
- Global API, on page 57
- Operation API, on page 73
- Status API, on page 101
- Peripheral Gateway API, on page 77
- Role API, on page 85
- Log Collection API, on page 61
- Machine Inventory API, on page 63

Updated APIs

See	Notes
Serviceability API, on page 91	Added SCRIPT_VERSIONS_ALLOWED and OEM_CP_matches_db_collation rule to System Configuration Validation Output section.

Deployment Type Info API, on page 47	The VMHost XML changed for 10.0(x).
	If you are a 9.0(x) customer updating to 10.0(x) and use the Deployment Type Info API directly, you will need to make the following changes to the XML:
	`<deploymentTypeInfo>`
	`<changeStamp>59</changeStamp>`
	`<deploymentType>7</deploymentType>`
	`<vmHosts>`
	`<vmHost>`
	`<name>sideA</name>`
	`<address>10.86.141.10</address>`
	`<userName>root</userName>`
	`<password>ciscoABC</password>`
	`</vmHost>`
	`<vmHost>`
	`<name>sideB</name>`
	`<host>10.86.141.29</address>`
	`<userName>root</userName>`
	`<password>ciscoABC</password>`
	`</vmHost>`
	`</vmHosts>`
	`</deploymentTypeInfo>`

Bulk Job API, on page 35	Department information added in bulk job requests in 10.0(x). This column must be added, but can be left blank. **Note** If you saved 9.0(x) copies of the CSV files on your system, download new copies for 10.0(x). If you have an existing 9.0(x) .CSV file and attempt to load it on a 10.0(x) system, an error will appear regarding the missing department column. New call type and skill group bulk operations also available.

See	Notes
Dialed Number API, on page 49	The dialedNumberString parameter now allows additional characters.
Agent API, on page 17	The skillGroupsAdded and skillGroupsRemove parameters were added.

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 6.

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 (Operation API, on page 73).

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>:<serverport>/unifiedconfig/config/bucketinterval`. See also Permissions, on page 9, Pagination, on page 7, Search, on page 10, and Sort, on page 11.

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

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

Access

Administrator access to Packaged CCE Administration APIs and items is defined by role and by the departments for which the administrator is responsible. Agents have no access to the Packaged CCE Administration APIs.

Supervisor access

The following APIs are read only:

- Agent Team API, on page 27
- Attribute API, on page 31
- Precision Queue API, on page 79

The following APIs allow update with restrictions:

- Agent API, on page 17:
  - Supervisors that call the Agent list API only see the agents on their team.
  - When updating an agent, supervisors can only change the following parameters:
    - Skill Groups
    - Default Skill Group
    - Attributes
    - Password
- Skill Group API, on page 97:
  - Supervisors are only allowed to update the list of agents that are members of the skill group. When updating a skill group, supervisors can only change the agent parameter to add or remove agents that are part of his team.

The Operation API can also be used to perform updates on Agents.

Authentication

To authenticate:

- Administrators must provide a fully qualified user name (for example, user@cisco.com) and password.

- Supervisors must provide their agent username and password.

Usage and behavior

Duplicate parameters

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

Read-only fields

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

References

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

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

```
<agent>
   <team>
     <refURL>/config/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>https://10.10.10.5/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>/config/team/5000</refURL>
   </team>
  ...
</agent>
```

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

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

```
  ...
</agent>
```

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

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

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

XML

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

Error responses

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

Parameters

- errorType: Indicates the type of error. This is the primary identifier for the problem and can be used to map the type to a user readable string. For example, if your application receives an error with the errorType of invalidInput.fieldRequired, then you could display "This field is a required field; it cannot be left blank" to the user.
- errorData: The name of the parameter that had the error.
- errorMessage: Extra information about the error that is intended for the developer. This information is typically a sentence or other string. It is not localized, so it should not be shown to the user.
- errorDetail: Some errors contain additional detail parameters that are included in the errorDetail parameter.
    - If the error type is invalidInput.outOfRange, then errorDetail includes the following parameters:
        - min: The minimum value allowed.
        - max: The maximum value allowed.
    - If you attempt to delete an item that is in use by other items, the errorType is referenceViolation.api and the errorDetail includes the following parameters:
        - referenceType: The type of item that references the item you tried to delete.
        - references: A collection of references, referencing the item you tried to delete, including the name and refURL of each referencing item.
        - totalCount: The total number of items referencing the item you attempted to delete.
        - totalShown: The total number of items included in the references collection.

Example error response

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

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

Pagination

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

Query parameters

- startIndex: 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 10.
- sortTerm: The value specified in the sort query parameters. See Sort, on page 11.

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 3) operation.
- A changeStamp is a required parameter for the body of a PUT (update, on page 4) operation for items. If you do not provide a changeStamp, the update fails. This mechanism is in place so that two clients cannot edit the record at the same time.
- If the update is successful, the changeStamp is incremented.

description

- A description for this item.
- Optional parameter.
- No restriction of characters; OEM locale supported characters are allowed. For information on how to configure your system to support native character sets, see the latest version of the document Installing and Configuring Cisco Packaged Contact Center Enterprise.
- Maximum length of 255 characters.

name

- Required parameter.
- Maximum length of 32 characters allowed.
- Valid characters are period (.), underscore (_), and alphanumeric. The first character must be alphanumeric.
- Does not allow internationalized characters.

refURL

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

Permissions

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

Parameters

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

Example get response

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

Synchronous vs. Asynchronous Writes

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

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

Performing asynchronous operations

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

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

Asynchronous result parameters

- progress: Indicates the current state of the request. Values include the following states:
  - IN_QUEUE: The request passed validation and capacity checks and was put in the queue.
  - IN_PROGRESS: The request is being processed.

Polling the asynchronous request status

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

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

Search

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

Default search parameters

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

Values:

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

Advanced search

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

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

Advanced departmental search

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

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

Sort

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

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

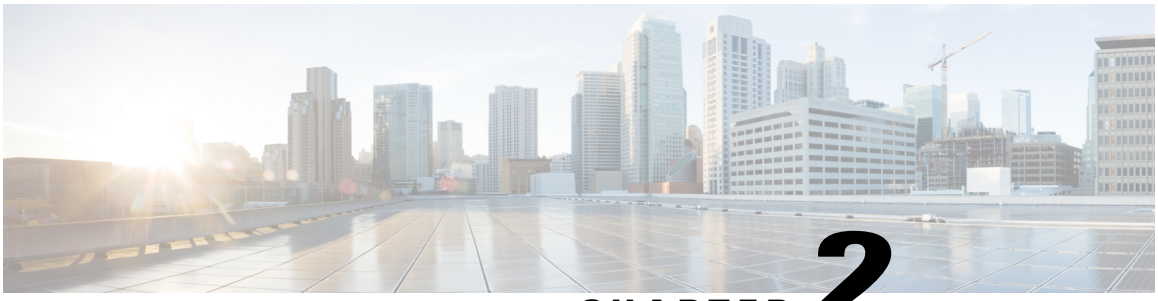
- parameterName: The name of the parameter that you want to sort on. This is case sensitive, so it must match the parameter in the API exactly.
- order: Specifies the order of the sort. Values are as follows:
  - asc: Perform an ascending sort. This is the default if no order is specified.
  - desc: Perform a descending sort.

Example

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

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

Sort



CHAPTER 2

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

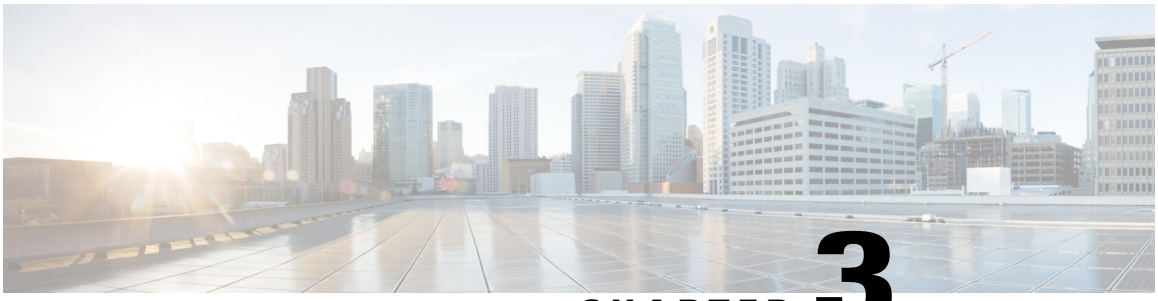
- list: Retrieves a list of active directory domains.

Parameters

- activeDirectoryDomains: A collection of activeDirectoryDomain items, including a name parameter.

Example get response

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



CHAPTER 3

Administrator API

An administrator is an Active Directory user who has been provided access to the system. That access can be controlled by assigning the administrator to roles and departments (see Role API, on page 85 and Department API, on page 43) .

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>:<serverport>/unifiedconfig/config/administrator`

Operations

- create: Creates one administrator.
- delete: Permanently deletes one administrator.
- get: Returns one administrator, using the URL `https://<server>:<serverport>/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 8.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- userName: Required. The unique username of an existing Active Directory account. Maximum length of 64 characters.
- domainName: The domain for this administrator. If blank, system uses the default domain name. Maximum length of 64 characters.
- departments: A collection of department (Department API, on page 43) references associated with this administrator, including the refURL and name. Leave this collection empty to allow the administrator to have access to all departments. See References, on page 5.

- role: A reference to a role (Role API, on page 85), including refURL and name. This parameter sets access to specific features. Automatically creates membership to Active Directory Setup group or Config group. If no role is assigned, the administrator is not placed in either group and does not have access to any of the web configuration tools, the Configuration Manager, or the Script Editor. See References, on page 5.
- readOnly: Required. Specifies whether the administrator has read-only access to the APIs and tools. Values are true/false.

Search and sort values

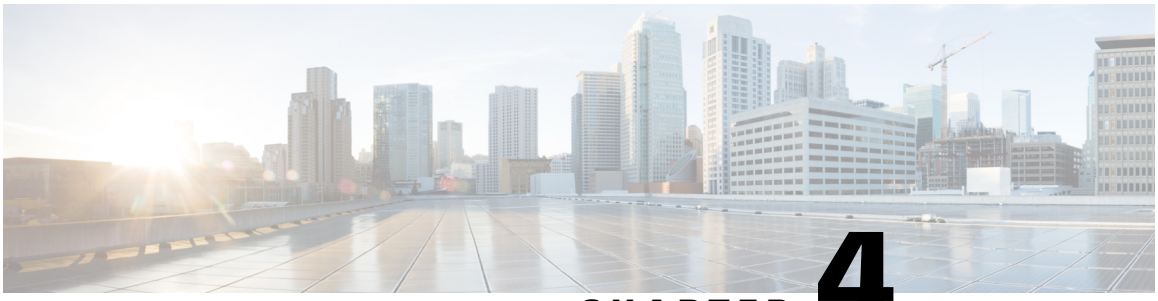
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• userName • domainName • description	• userName • domainName • description

See Search, on page 10 and Sort, on page 11.

Example get response

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



CHAPTER 4

Agent API

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

URL

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

Operations

- create: Creates an agent.
- delete: Marks one agent for deletion, but does not permanently delete the agent.
- get: Returns one agent, using the URL `https://<server>/unifiedconfig/config/agent/<id>`.
- list: Retrieves a list of agents.
    - **Query parameters:**
        - Summary list: See list, on page 3.
- update: Updates one agent.

Parameters

- refURL: The refURL for the agent. See Shared parameters, on page 8.
- agentId: The unique peripheral number. Maximum length of 11 characters allowed. Default is an auto-generated 7 digit number.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- department: A reference to the agent's department (Department API, on page 43), including the refURL and name. See References, on page 5.
- agentStateTrace: Indicates if agent state tracing is turned on for the agent. Values are true/false. See Agent State Trace API, on page 25.

- agentDeskSettings: A reference to the agent's agentDeskSettings (Agent Desk Settings API, on page 21), including the refURL and name. See References, on page 5.
- 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.
  - userName: Agent's user name. Maximum of 32 alphanumeric characters.
  - 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.
- supervisor: Required. Indicates whether the agent is marked as supervisor. Values are true/false.
- supervisorUserInfo: Required if supervisor is set to true. User information about an existing Active Directory account for the supervisor. Includes the following parameters:
  - userName: Supervisor's Active Directory user name.
  - domainName: Supervisor's Active Directory (Active Directory Domain API, on page 13) domain name. If domainName is empty, system uses default domain name.
- agentAttributes: A collection of agent attribute (Attribute API, on page 31) references for this agent, including the description, refURL, name, and dataType for each associated attribute. Also includes the attributeValue parameter which indicates the value (true/false or 1-10) of the attribute for this agent. See References, on page 5.
- skillGroups: A collection of skill group (Skill Group API, on page 97) references for this agent, including the refURL and name of each associated skill group. See References, on page 5.
- skillGroupsAdded: A collection of skill group references (Skill Group API, on page 97) 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 in conjunction with the skillGroups parameter on an update as it does not affect existing skill groups. This parameter can be used with the skillGroupsRemoved parameter. See References, on page 5.
- skillGroupsRemoved: A collection of skill group references (Skill Group API, on page 97) 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 in conjunction with the skillGroups parameter on an update as it does not affect existing skill groups. This parameter can be used with the skillGroupsAdded parameter. See References, on page 5.
- defaultSkillGroup: A reference to a skill group (Skill Group API, on page 97), including the refURL and name. Identifies the default skill group associated with this agent. See References, on page 5.
- agentTeam: A reference to the agent's team (Agent Team API, on page 27), including the refURL and name. See References, on page 5.
- 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 5.

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 • person.firstName • person.lastName • person.userName • person.loginEnabled

See Search, on page 10 and Sort, on page 11.

Advanced search parameters

There are a number of advanced searches you can perform on the Agent API, including supervisor, attributes, skillgroups, and team.

- **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.

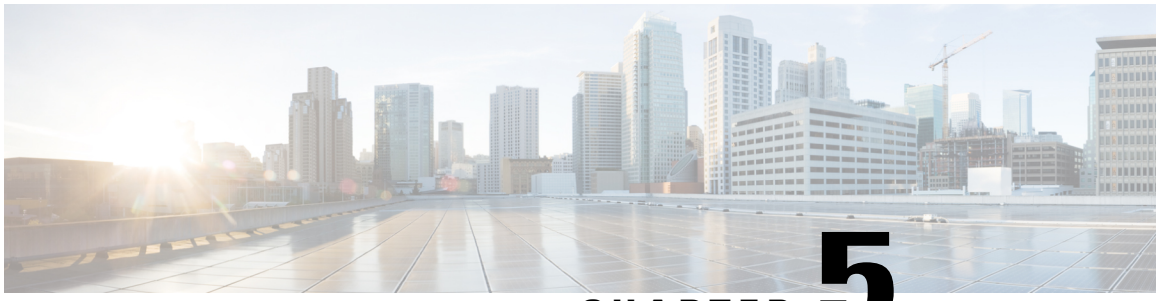
Example get response

```
<agent>
 <changeStamp>2877</changeStamp>
        <refURL>/unifiedconfig/config/agent/5017</refURL>
        <agentId>8006</agentId>
        <agentStateTrace>false</agentStateTrace>
        <description>an agent</description>
        <person>
             <firstName>Agent2</firstName>
             <lastName>Agent2</lastName>
             <loginEnabled>true</loginEnabled>
             <userName>Agent2</userName>
             <password>mypassword</password>
       </person>
        <agentDeskSettings>
             <name>test2</name>
             <refURL>/unifiedconfig/config/agentdesksetting/5434</refURL>
        <supervisor>true</supervisor>
        </agentDeskSettings>
        <supervisorUserInfo>
```

```
            <userName>boston</userName>
            <domainName>boston.com</domainName>
        </supervisorUserInfo>
        <agentAttributes>
             <agentAttribute>
                  <attribute>
                      <refURL>/unifiedconfig/config/attribute/5004</refURL>
                      <name>Sales</name>
                      <dataType>4</dataType>
                      <description>Sales proficiency</description>
                  </attribute>
                  <attributeValue>8</attributeValue>
                  <description>masters certification</description>
            </agentAttribute>
         </agentAttributes>
         <skillGroups>
             <skillGroup>
                 <refURL>/unifiedconfig/config/skillgroup/5229</refURL>
                 <name>Support</name>
             </skillGroup>
        </skillGroups>

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

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



CHAPTER 5

Agent Desk Settings API

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

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

URL

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

Operations

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

Parameters

- refURL: The refURL of the agent desk settings. See Shared parameters, on page 8.
- name: The name of the agent desk settings. See Shared parameters, on page 8.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- department: A reference to the department (Department API, on page 43), including refURL and name. See References, on page 5.
- wrapupDataIncomingMode: Indicates whether the agent is allowed or required to enter wrap-up data after an inbound call.
    - 0: Required

    - 1: Optional (Default)
    - 2: Not Allowed

- wrapupDataOutgoingMode: Indicates whether the agent is allowed or required to enter wrap-up data after an outbound call.
    - 0: Required
    - 1: Optional (Default)
    - 2: Not Allowed

- remoteAgentType: Indicates if agents are allowed to login as remote agents.
    - 0: Not Allowed
    - 2: Allowed

- logoutNonActivityTime: Number of seconds of non-activity at the desktop after which the software automatically logs out the agent. Value must be between 10 and 7200 seconds (default is NULL).
- workModeTimer: Specifies the auto wrap-up time out. Value must be between 1 and 7200 seconds (default is 7200).
- supervisorAssistCallMethod: Indicates how the supervisor assist request call is made.
    - 0: Consultative Call (Default)
    - 1: Blind Conference

- emergencyCallMethod: Indicates how the emergency call request is made.
    - 0: Consultative Call (Default)
    - 1: Blind Conference

- idleReasonRequired: Indicates whether the agent must enter a reason before entering the Idle state. Values are true/false.
- logoutReasonRequired: Indicates whether or not the agent must enter a reason before logging out. Values are true/false.
- autoAnswerEnabled: Indicates whether or not calls sent to this agent will be answered automatically. Values are true/false.

Search and sort values

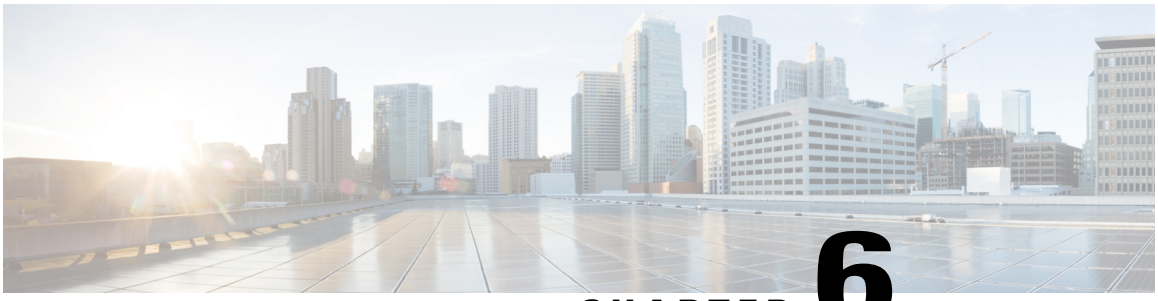
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • description • wrapupDataIncomingMode • wrapupDataOutgoingMode • remoteAgentType • logoutNonActivityTime • workModeTimer • supervisorAssistCallMethod • emergencyCallMethod • idleReasonRequired • logoutReasonRequired • autoAnswerEnabled

See Search, on page 10 and Sort, on page 11.

Example get response

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



CHAPTER 6

Agent State Trace API

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

Note The maximum number of agents with AgentStateTrace on is 100.

URL

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

Operations

- create: Returns a list of agents whose agent state trace is turned on, using the URL `https://<server>/unifiedconfig/config/agentstatetrace`.
- update: Updates the agent state trace in the database.

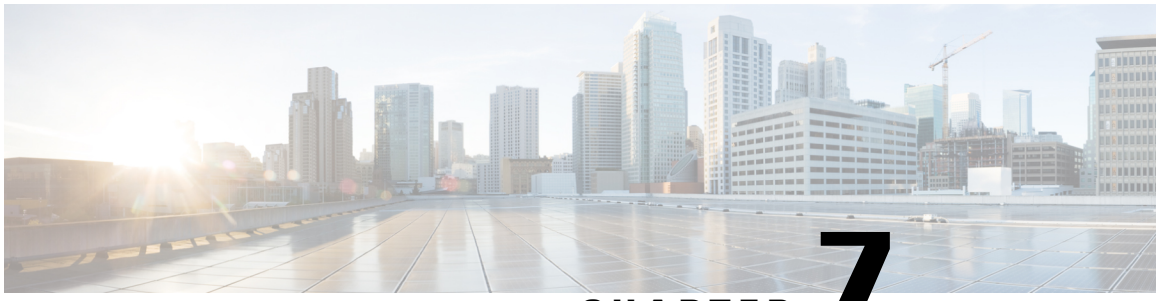
Parameters

- refURL: The refURL for agent state trace. See Shared parameters, on page 8.
- agents: A collection of agent references (Agent API, on page 17). Each reference contains person (including firstName, lastName, userName, and loginEnabled parameters), agent refURL, agentId, supervisor, and agentStateTrace. Agents who are not specified in this collection have agentStateTrace turned off. To turn off all the agent state trace, pass in an empty list. See References, on page 5.

Example get response

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

```
                    <loginEnabled>true</loginEnabled>
                    <userName>jdoe</userName>
                  </person>
                <supervisor>false</supervisor>
            </agent>
        </agents>
    </agentstatetrace>
```



CHAPTER 7

Agent Team API

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

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

URL

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

Operations

- create: Creates an agent team.
- delete: Permanently deletes one agent team from the database.
- get: Returns one agent team, using the URL `https://<server>/unifiedconfig/config/agentteam/<id>`.
- list: Retrieves a list of agent teams.
    - **Query parameters:**
        - Summary list: See list, on page 3.
- update: Updates one agent team.

Parameters

- refURL: The refURL of the agent team. See Shared parameters, on page 8.
- name: The name of the agent team. See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- department: A reference to the department (Department API, on page 43), including the name and refURL. See References, on page 5.
- dialedNumber: A reference to an internal dialed number (Dialed Number API, on page 49) for the agent team, including the refURL and dialed number string. See References, on page 5.

- agents: A collection of agent (Agent API, on page 17) references, including the refURL, first name, last name, user name, and agent ID for each agent on the team. See References, on page 5.
- agentCount: Read-only field. Number of agents on the team.
- supervisors: A collection of supervisor (Agent API, on page 17) references, including the refURL, first name, last name, user name, and agent ID for each supervisor who supervises this team. See References, on page 5.
- supervisorCount: Read-only field. Number of supervisors who supervise this team.

Search and sort values

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

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

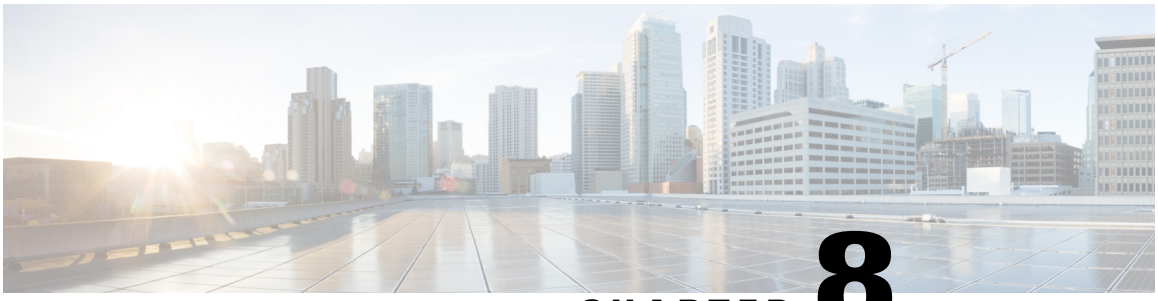
See Search, on page 10 and Sort, on page 11.

Example get response

```
<agentTeam>
    <refURL>http://***.***.***.***/unifiedconfig/config/agentteam/(id)</refURL>
    <name>team1</name>
    <dialedNumber>
       <refURL>[https://***.***.***.***/unifiedconfig/config/dialednumber/(id)]</refURL>
       <dialedNumberString>8885551212</dialedNumberString>
    </dialedNumber>
    <description>test agent team1</description>
    <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>

    <supervisor>
       <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 8

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 for deletion, but does not permanently delete it.
- get: Returns one attribute, using the URL `https://<server>:<serverport>/unifiedconfig/config/attribute/<id>`.
- list: Retrieves a list of attributes.
- update: Updates one attribute.

Parameters

- refURL: The refURL of the attribute. See Shared parameters, on page 8.
- name: The name of the attribute. See Shared parameters, on page 8.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- department: A reference to the department (Department API, on page 43), including refURL and name. See References, on page 5.
- 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.

Search and sort values

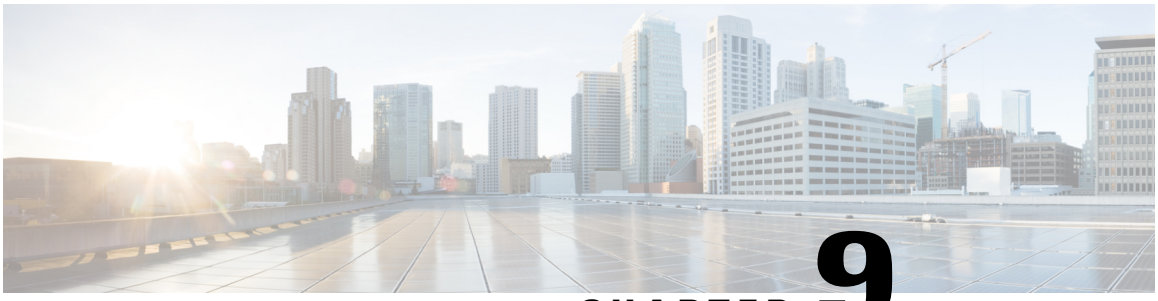
The following table shows the parameters that are searched and the parameters that are sortable.

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

See Search, on page 10 and Sort, on page 11.

Example get response

```
<attribute>
      <changeStamp>0</changeStamp>
      <refURL>/unifiedconfig/config/attribute/5000</refURL>
      <dataType>3</dataType>
      <defaultValue>true</defaultValue>
      <name>chinese</name>
</attribute>
```



CHAPTER 9

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 8.
- name: The name of the bucket interval. See Shared parameters, on page 8.
- changeStamp: See Shared parameters, on page 8.
- department: A reference to the department (Department API, on page 43), including the refURL and name. See References, on page 5.
- 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

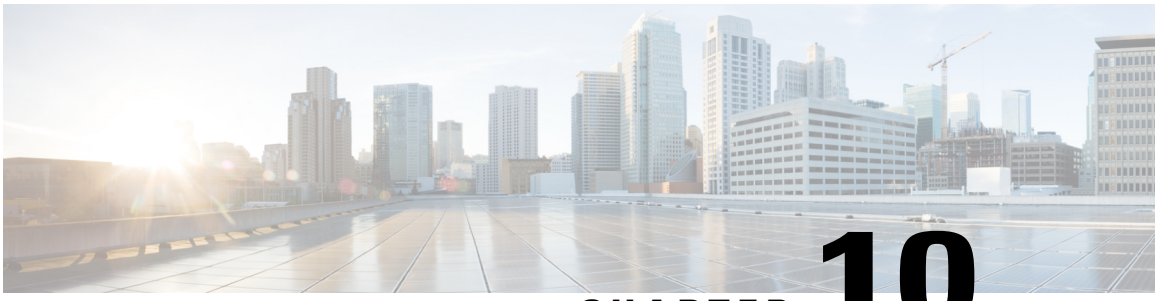
The following table shows the parameters that are searched and the parameters that are sortable.

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

See Search, on page 10 and Sort, on page 11.

Example get response

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



CHAPTER 10

Bulk Job API

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

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

URL

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

Operations

- create: Creates one bulk job.
- delete: Permanently deletes one bulk job.
- get: Returns one bulk job using the URL `https://<server>/unifiedconfig/config/bulkjob/<id>`.
- list: Retrieves a list of bulk jobs.

Parameters

- refURL: The refURL of the bulk job. See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- fileContent: The content of the bulk CSV file. The size of the file must not exceed 3MB. For information about the CSV file data, see the "Manage Bulk Jobs" section of the Packaged CCE Administration Guide.
- 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 data server that initiated the bulk job. Read-only.
- startDateTime: The time the bulk job began executing. Read-only.
- endDateTime: The time the bulk job completed or failed. Read-only.
- jobState: The current state of the job. Read-only.

- 1: Queued
- 2: Processing
- 3: Succeeded
- 4: Failed
- 5: Cancelled
- 6: Partially succeeded

- jobType: The job type. Read-only.
  - 1: Dialed Number
  - 2: Agent
  - 3: Call Type
  - 4: Skill Group
- 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 10 and Sort, on page 11.

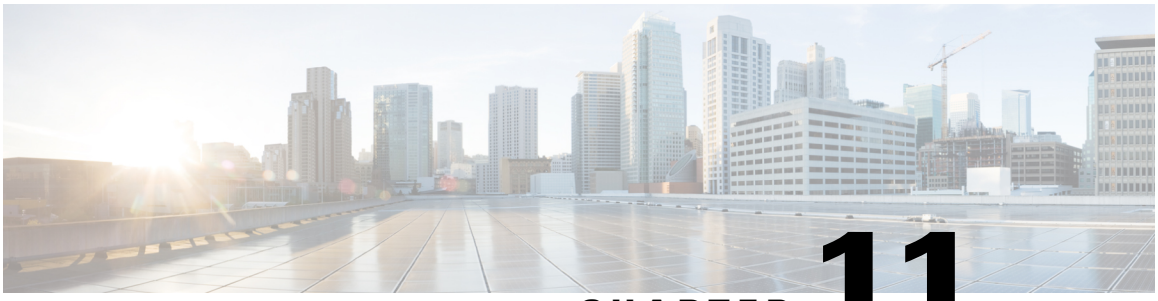
Example get response

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

```
<bulkJob>
  <changeStamp>0</changeStamp>
  <refURL>http://<server>/unifiedconfig/config/bulkjob/(id)</refURL>
  <jobHostName>DS1</jobHostName>
  <createDateTime>1330441858360</createDateTime>
  <startDateTime>1330441858361</createDateTime>
```

```
  <endDateTime>1330441858368</endDateTime>
  <jobState>3</jobState>
  <jobType>1</jobType>
  <description>dn create bulk job</description>
  <logFile><refURL>http://<server>/unifiedconfig/config/bulkjob/(id)/log</refURL></logFile>

  <csvFile><refURL>http://<server>/unifiedconfig/config/bulkjob/(id)/csv</refURL></csvFile>
</bulkJob>
```



CHAPTER 11

Call Type API

Call types categorize calls. Based on call type, the system maps a dialed number (see Dialed Number API, on page 49) to a routing script that ultimately sends the call to the appropriate destination.

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

URL

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

Operations

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

Parameters

- refURL: The refURL of the call type. See Shared parameters, on page 8.
- name: The name of the call type. See Shared parameters, on page 8.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- department: A reference to the department (Department API, on page 43), including the refURL and name. See References, on page 5.
- id: The database id of the call type. Read-only field. Used in scripting.
- serviceLevelThreshold: Maximum time in seconds that a caller should wait before being connected with an agent. Leave blank to use the system default (set in the Global API, on page 57).
- 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.

Search and sort values

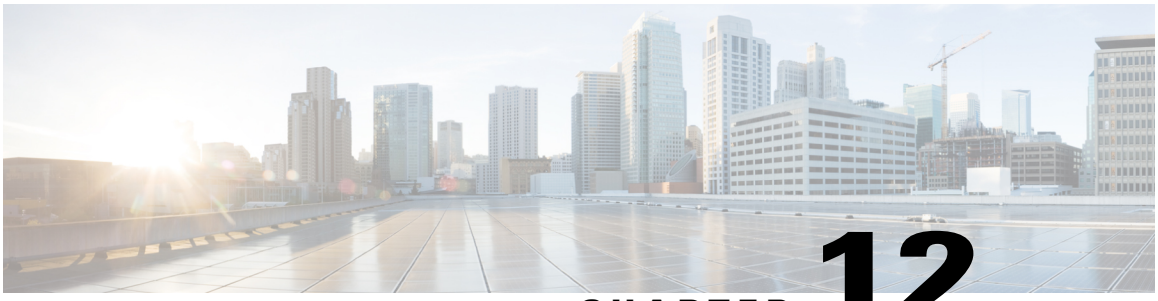
The following table shows the parameters that are searched and the parameters that are sortable.

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

See Search, on page 10 and Sort, on page 11.

Example get request

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



CHAPTER 12

Congestion Control API

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

URL

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

Operations

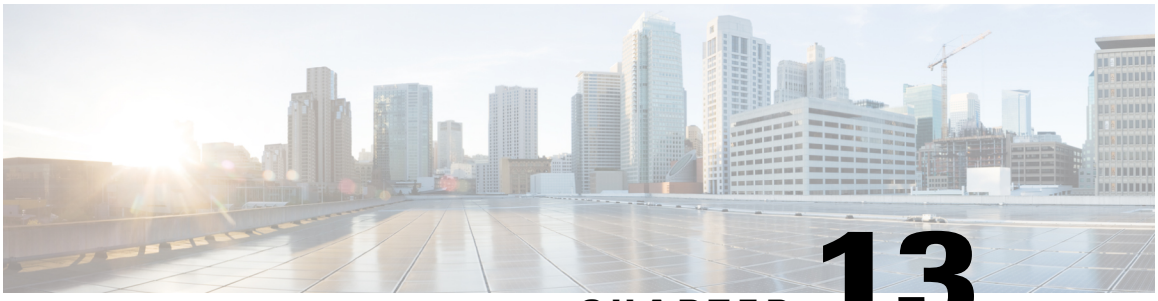
- get: Returns the congestion control parameters, using the URL `https://<server>:<serverport>/unifiedconfig/config/congestioncontrol`.

Parameters

- deploymentType: The type of deployment. See Deployment Type Info API, on page 47.
- 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

Department API

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

For more information on how departments work, see the Administration and Configuration Guide for Packaged CCE.

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

URL

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

Operations

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

Parameters

- refURL: The refURL of the department. See Shared parameters, on page 8.
- name: The name of this department. See Shared parameters, on page 8.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- administrators: A collection of administrator (Administrator API, on page 15) references associated with this department, including the refURL, user name, and domain name. See References, on page 5.

Search and sort values

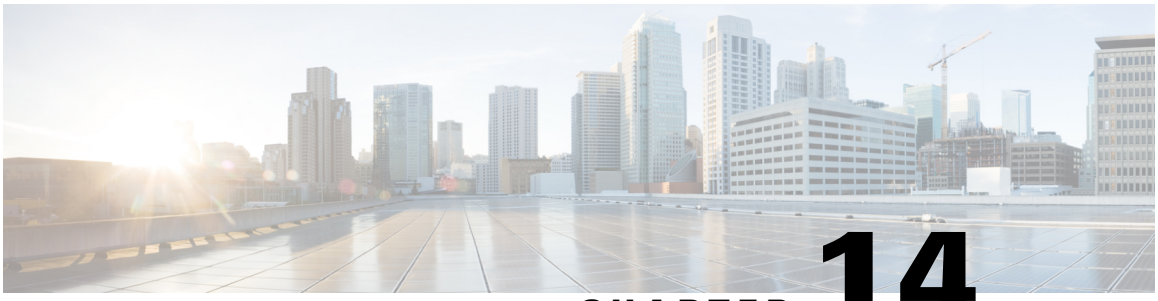
The following table shows the parameters that are searched and the parameters that are sortable.

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

See Search, on page 10 and Sort, on page 11.

Example get response

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



CHAPTER 14

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>:<serverport>/unifiedconfig/config/deployment`

Parameters

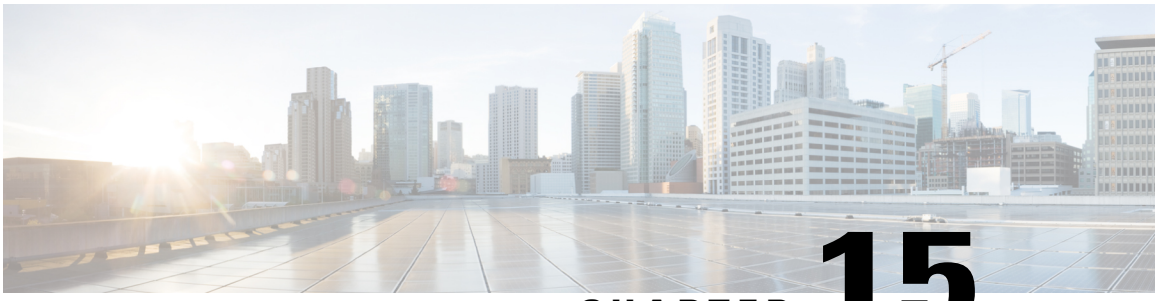
deploymentType: See Deployment Type Info API, on page 47.

Operations

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

Example get response

```
<deployment>
     <deploymentType>7</deploymentType>
</deployment>
```



CHAPTER 15

Deployment Type Info API

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

URL

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

Operations

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

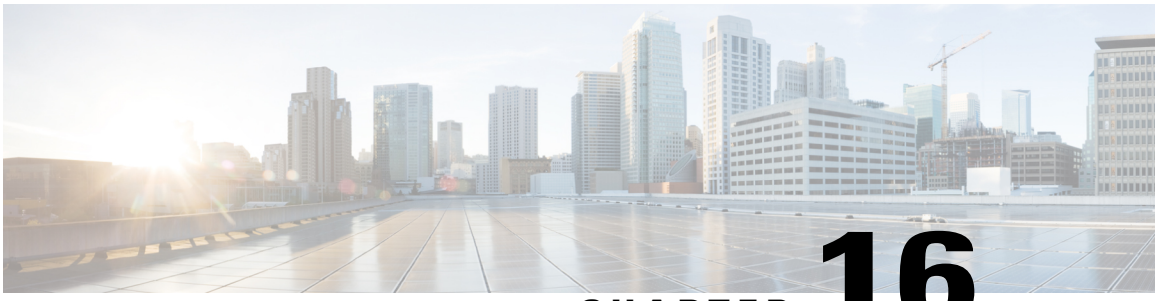
Parameters

- changeStamp: See Shared parameters, on page 8.
- vmHosts: vmHost information, including name, address, username, and password parameters of Side A and Side B. Only required when switching to Packaged CCE, to allow access to the ESX servers for VM validation.
- permissionInfo: See Permissions, on page 9.
- systemValidationStatus: See Serviceability API, on page 91.
- capacityInfo: See Serviceability API, on page 91.
- vmValidationLogURL: The URL to download a file about VM layout validation.
- deploymentType: The type of deployment. The following types are supported:
  - 0: No deployment type specified. Initial type set at installation. Once set to another deployment type, you cannot switch back to 0.
  - 1: NAM
  - 2: IVR-ICM
  - 3: NAM Rogger
  - 4: ICM Router/Logger

- 5: UCCE 8000 Agents Router/Logger
- 6: UCCE 12000 Agents Router/Logger
- 7: Packaged CCE: CCE-PAC-M1
- 8: ICM Rogger
- 9: UCCE 4000 Agents Rogger
- 10: Packaged CCE: CCEPACM1 Lab only
- 11: HCS-CC 1000 Agents
- 12: HCS-CC 500 Agents
- 13: UCCE 450 Agents Progger
- 14: HCS-CC 4000 Agents

Example get response

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



CHAPTER 16

Dialed Number API

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

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

URL

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

Operations

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

Parameters

- dialedNumberString: Required. Value used to route the call or direct the non-voice task. A unique string for the routing type. Maximum of 25 characters.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- department: A reference to the department (Department API, on page 43), including the refURL and name. See References, on page 5.
- routingType: Specifies where a call or non-voice task request originates.
  - 1: External Voice. Calls come from Unified CVP. When creating a Dialed Number using this type, a dialed number database record is created for each Unified CVP routing client.
  - 2: Internal Voice. Calls come from a Unified CM phone.

- 3: Outbound. Calls that come from the Outbound Option Dialer.
- 4: Multichannel. Requests that come from an EIM/WIM or SocialMiner.

- callType: A reference to a call type (Call Type API, on page 39) for this dialed number, including a refURL and name. See References, on page 5.
- dialedNumberRecords: A collection of dialed number record entries each containing the id and name of a dialed number database record. Read-only.
- mediaRoutingDomain: A reference to the media routing domain (Media Routing Domain API, on page 69) for the dialed number. See References, on page 5.

Search and sort values

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

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

See Search, on page 10 and Sort, on page 11.

Advanced search parameters

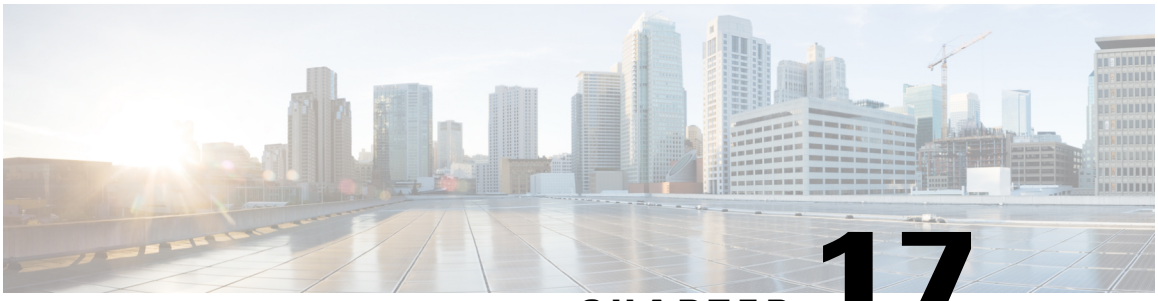
The Dialed Number API also supports advanced search parameters, such as routing type.

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

Example get response

```
<dialedNumber>
  <refURL>[/unifiedconfig/config/dialedNumber/(id)]</refURL>
  <description>test dialed number</description>
  <dialedNumberString>8885551212</dialedNumberString>
  <routingType>1</routingType>
  <changeStamp>0</changeStamp>
  <mediaRoutingDomain>
    <refURL>[/unifiedconfig/config/mediaroutingdomain/1]<refURL>
    <name>Cisco_Voice</name>
  </mediaRoutingDomain>
  <dialedNumberRecords>
    <dialedNumberRecord>
      <id>10</id>
      <name>cvp1rc.8885551212</name>
    </dialedNumberRecord>
    <dialedNumberRecord>
      <id>11</id>
      <name>cvp2rc.8885551212</name>
    </dialedNumberRecord>
    <dialedNumberRecord>
      <id>12</id>
      <name>cvp3rc.8885551212</name>
    </dialedNumberRecord>
```

```
    <dialedNumberRecord>
      <id>13</id>
      <name>cvp4rc.8885551212</name>
    </dialedNumberRecord>
  </dialedNumberRecords>
</dialedNumber>
```



CHAPTER 17

Expanded Call Variable API

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

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

URL

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

Operations

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

Parameters

- refURL: The refURL of the expanded call variable. See Shared parameters, on page 8.
- name: The name of the expanded call variable. See Shared parameters, on page 8.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- maximumLength: The maximum length of the expanded call variable. The value is 1 to 210.
- eccArray: Indicates whether the expanded call variable is an array. Values are true/false.
- maximumArraySize: The maximum number of elements in the array is 1 to 255. Required if eccArray is true; must be blank or not specified if eccArray is false.
- enabled: Indicates whether the expanded call variable is enabled. Values are true/false.

- The total bytesRequired of all enabled expanded call variables cannot exceed 2000 bytes.
- The total bytesRequiredInCtiServer of all enabled expanded call variables cannot exceed 2500 bytes.

- persistent: Specifies whether the expanded call variable is written to the historical database with each Termination Call Detail and Route Call Detail record. Values are true/false.
  - No persistent, enabled arrays are allowed.
  - The maximum number of persistent, scalar, enabled variables is 20.
- ciscoProvided: Indicates whether the expanded call variable is provided by Cisco. Values are true/false. Read-only.
- bytesRequired: The number of bytes required to store the expanded call variable in the system. Read-only.

  The size is calculated using the following formula:
  - If eccArray is false, the size is 5+Maximum Length.
  - If eccArray is true, the size is 5+(1+Maximum length)*Maximum Array size.
- bytesRequiredInCtiServer: The number of bytes required to send this variable to CTI Server. Read-only.

  The size is calculated using the following formula:
  - If eccArray is false, the size is Length of name+Maximum length+4.
  - If eccArray is true, the size is (Length of name+Maximum length+5)*Maximum array size.

Search and sort values

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

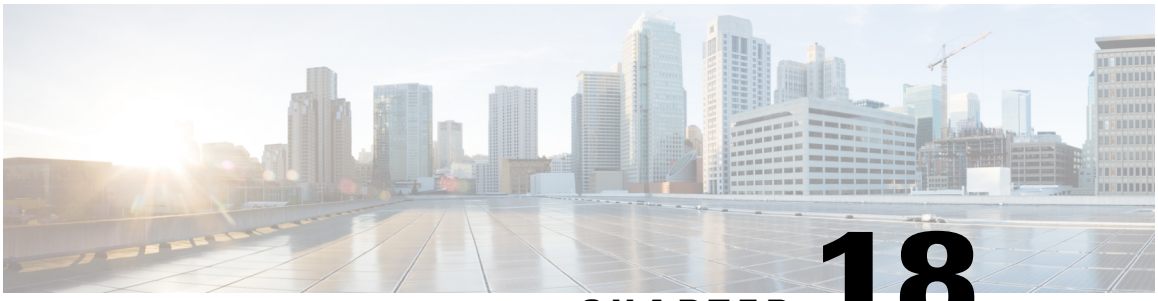
Search parameters	Sort parameters
• name • description	• name (default) • description • maximumLength • maximumArraySize • eccArray • enabled • persistent • ciscoProvided

See Search, on page 10 and Sort, on page 11.

Example get response

```
<expandedCallVariable>
    <refURL>http://***.***.***.***/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 18

Global API

The Global API returns global settings from the following categories:

- Call
- Agent
- Reporting
- Script

URL

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

Operations

- list: Retrieves a list of global settings.
- update: Updates global settings.

Parameters

callReporting

- defaultBucketInterval: Required. A reference to a bucket interval (Bucket Interval API, on page 33), including refURL and name. See References, on page 5.
- defaultCallType: Required. A reference to a call type (Call Type API, on page 39), including refURL and name. A call is categorized against this call type unless it comes into the system on a dialed number that is associated with another call type. See References, on page 5.
- serviceLevelType: Required. This value indicates how the system calculates the service level.
  - 1: Ignore Abandoned Calls.
  - 2: Abandoned Calls have Negative Impact.
  - 3: Abandoned Calls have Positive Impact.
- serviceLevelThreshold: Required. Maximum time in seconds that a caller should wait before being connected with an agent. Maximum is 86,400 seconds (1 day).

- abandonCallWaitTime: Required. Configures the minimum time an incoming call must be queued before the call is considered abandoned if the caller hangs up. Maximum is 14400 seconds (4 hours).
- answeredShortCallThreshold: Configures the maximum duration for a short call. Calls with a duration below that value are considered short. Value is between 0 and 14400 seconds (4 hours).

agent

- agentPhoneLineControl: Indicates whether all agents supported on the agent peripheral can have one or more than one line configured.
  - 0: Single Line.
  - 1: All Lines.
- nonACDLineImpact: Specifies how the agent state is set when the agent is on a call on a secondary line and agentPhoneLineControl is set to All Lines.
  - 0: Available agent stays available.
  - 1: Available agent goes not ready.
- defaultDeskSetting: A reference to a desk setting (Agent Desk Settings API, on page 21), including refURL and name.
- loginNameCaseSensitivity: Identifies whether usernames are case-sensitive. Values are true/false.
- minimumPasswordLength: Changing this value affects new passwords only and does not apply to existing ones. Value is between 0 and 32.

reporting

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

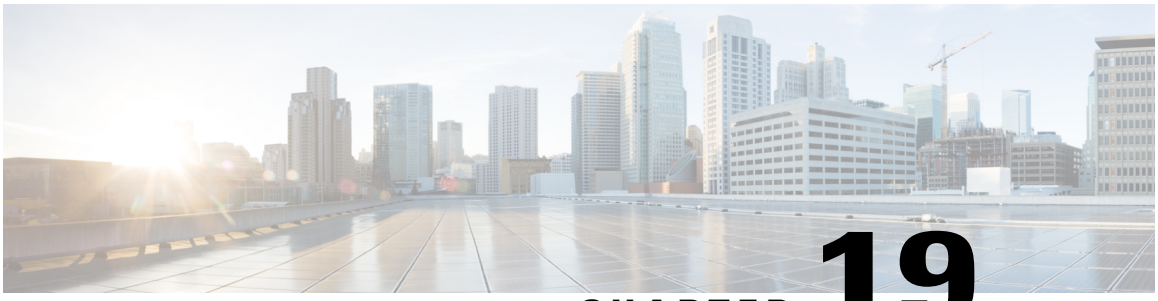
script

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

Example get request

```
<globalSettings>
  <callReporting>
    <serviceLevelType/>
    <serviceLevelThreshold/>
    <abandonCallWaitTime/>
    <answeredShortCallThreshold/>
    <defaultCallType>
     <refURL/>
     <name/>
    </defaultCallType>
    <defaultBucketInterval>
      <refURL/>
      <name/>
    </defaultBucketInterval>
  </callReporting>
  <agent>
```

```
      <agentPhoneLineControl/>
      <nonACDLineImpact/>
      <defaultDeskSetting>
        <refURL/>
        <name/>
      <defaultDeskSetting/>
      <loginNameCaseSensitivity/>
      <minimumPasswordLength/>
    </agent>
    <reporting>
      <reportingInterval/>
    </reporting>
    <script>
      <retainScriptVersion/>
    </script>
  </globalSettings>
```



CHAPTER 19

Log Collection API

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

Operations

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

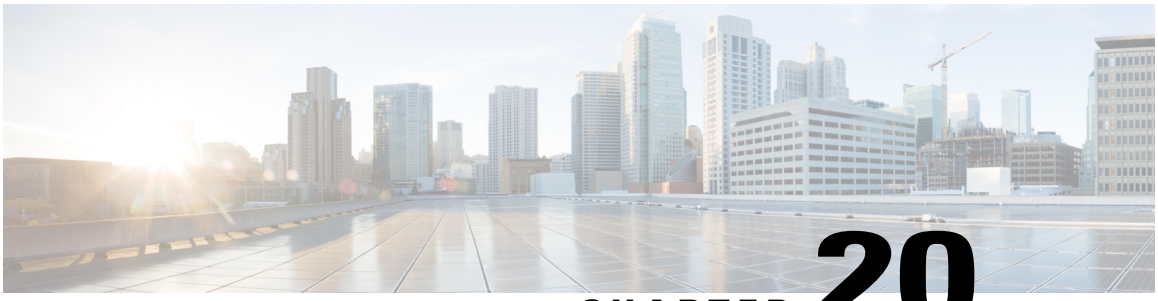
Parameters

- refURL: The refURL of the log collection. See Shared parameters, on page 8.
- startDateTime: The start date of the logs collected.
- endDateTime: The end date of the logs collected.
- description: See Shared parameters, on page 8.
- components: A list of components for which logs are collected. Defaults to all components if the list is blank or not provided. Possible component values include:
  - 1: Unified Contact Center Enterprise (CCE)
  - 2: Unified Customer Voice Portal (CVP)
  - 3: Unified Communications Manager (CM)
- status.state: The status of the collection request: IN_PROGRESS, DONE, or ERROR.

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

Example get response

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



CHAPTER 20

Machine Inventory API

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

For information on how to perform a machine inventory scan, see Scan API, on page 89.

URL

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

Operations

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

Type	Create / Update / Delete operations allowed	Number allowed
VM_HOST	No	1 Side A 1 Side B
CCE_CALL_SERVER	No	1 Side A 1 Side B
CCE_DATA_SERVER	Side A: Update only Side B: No	1 Side A 1 Side B

Type	Create / Update / Delete operations allowed	Number allowed
CVP	No	2 Side A 2 Side B
CM	Update only	0 - Must be changed after initial scan.
CM_PUBLISHER	Update only	1 Side A, 0 Side B for on box CM Deployments 0 for off box CM deployments
CM_SUBSCRIBER	Update only	1 Side A, 1 Side B for on box CM Deployments 0 for off box CM deployments
CVP_REPORTING	No	0 - 1 Side A
CUIC_PUBLISHER	No	1 Side A
CUIC_SUBSCRIBER	No	1 Side B
CVP_OPS	Update only	1 Side A
FINESSE	No	1 Side A 1 Side B
GATEWAY	No	None
EXTERNAL_SOCIAL_MINER	All	0 - 1
EXTERNAL_CM_PUBLISHER	All	0 for on box CM deployments 1 for off box CM Deployments
EXTERNAL_CM_SUBSCRIBER	No	Not allowed for on box CM Deployments A minimum of 2 for off box CM Deployments. External subscribers cannot be created, updated, or deleted as this automatically occurs when the external publisher is created, updated, or deleted.
EXTERNAL_CVP_REPORTING	All	0 - 1
EXTERNAL_HDS	Update only	0 - 2
EXTERNAL_MEDIA_SENSE	All	0 - 1

Parameters

Machine parameters:

- refURL: The refURL of the machine. See Shared parameters, on page 8.
- name: External name of the machine. For example, the VM host name. See Shared parameters, on page 8.
- changeStamp: See Shared parameters, on page 8.
- type: The type of machine.
- vmhost: A reference to a machine of type VM_HOST, including refURL and name. See References, on page 5.
- autogenerated: Indicates if the information was generated automatically.
- networks: A collection of network parameters. See the network parameters below.

Network parameters:

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

Services parameters:

- uri: The service URL, for example, Diagnostic Portal URL or management console link.
- port: The port for this service.
- username: The username used to access theservice. Username maximum is 128 characters.
- password: The password used to access the service. The password can be used when creating or updating, but is not returned.
- enablePassword: Used for gateways.
- description: The description of the service. See Shared parameters, on page 8.
- pairing: Indicates if services on different machines are related. Related services have a matching value.
- vmInstanceUuid: A unique identifier for the virtual machine.
- type: The service type. Values for type are as follows:
    - ESXI: VM Host running ESXi server for connection to interrogate forVMs.
    - AXL: AXL connection information for Unified CM.
    - DIAGNOSTIC_PORTAL: Diagnostic Portal API.
    - GATEWAY: Connection information for gateways.
    - MANAGEMENT_LINK: Provides a URL of the management console for the machine.
    - TIP_PG: LiveData connection information for peripheral gateways.
    - TIP_ROUTER: LiveData connection information for the router.

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 10 and Sort, on page 11.

Example response

Inventory status:

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

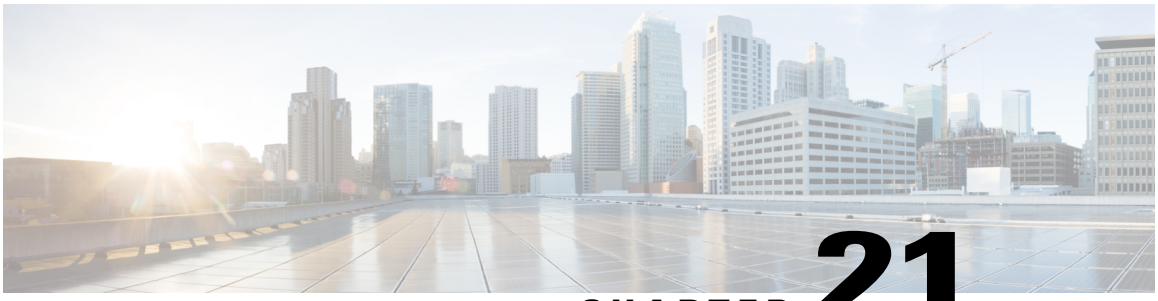
```
<status>
   <alerts>
      <alert>
         <apiErrors>
            <apiError>
               <errorData>CM_PUBLISHER</errorData>
               <errorMessage>CM_PUBLISHER not found on vmhost sideA</errorMessage>
               <errorType>inventory.MissingMachine</errorType>
            </apiError>
         </apiErrors>
         <machine>
            <host>sideA</host>
            <type>CM_PUBLISHER</type>
         </machine>
      </alert>
   </alerts>
   <scanInfo>
      <lastScanDateTime>1374842924017</lastScanDateTime>
      <scanState>Idle</scanState>
   </scanInfo>
</status>
```

Example get response:

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

```
<machine>
   <changeStamp>6</changeStamp>
   <refURL>/unifiedconfig/config/machineinventory/8241</refURL>
   <networks>
      <network>
         <address>10.10.10.20</address>
         <services>
            <service>
               <autoGenerated>true</autoGenerated>
               <description>LiveData Event service for PG</description>
               <pairing>5000</pairing>
               <port>42034</port>
               <type>TIP_PG</type>
            </service>
            <service>
               <autoGenerated>true</autoGenerated>
               <description>LiveData Event service for Router</description>
               <pairing>0</pairing>
               <port>40034</port>
               <type>TIP_ROUTER</type>
            </service>
            <service>
               <autoGenerated>true</autoGenerated>
               <description>LiveData Event service for PG TOS</description>
```

```
                  <pairing>5000</pairing>
                  <port>42035</port>
                  <type>TIP_PG_TOS</type>
               </service>
               <service>
                  <autoGenerated>true</autoGenerated>
                  <description>LiveData Event service for Router TOS</description>
                  <pairing>0</pairing>
                  <port>40035</port>
                  <type>TIP_ROUTER_TOS</type>
               </service>
            </services>
            <type>PUBLIC</type>
         </network>
         <network>
            <address>10.1.1.20</address>
            <type>PRIVATE</type>
         </network>
      </networks>
      <autoGenerated>true</autoGenerated>
      <hostName>CCE-CS-A-20.berlin.icm</hostName>
      <type>CCE_CALL_SERVER</type>
      <name>testCallServerA</name>
      <vmHost>
         <refURL>/unifiedconfig/config/machineinventory/8238</refURL>
         <name>sideA</name>
      </vmHost>
      <vmInstanceUuid>5006174b-e4fd-3d33-6cca-8855edebf3f8</vmInstanceUuid>
   </machine>
```



CHAPTER 21

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.

URL

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

Operations

- list: Retrieves a list of media routing domains.
- get: Returns one media routing domain using the URL `https://<server>/unifiedconfig/config/mediaroutingdomain/<id>`.

Parameters

- refURL: The refURL of the media routing domain. See Shared parameters, on page 8.
- name: Name of the media routing domain. See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- mediaClass: Includes the following parameters:
  - name: The name of the media class.
  - id: The ID of the media class.
- serviceLevelThreshold: Value in seconds within which calls must be answered.
- interruptible: Indicates if an agent can be interrupted by assigned tasks from another MRD. Values are true/false.
- maxCallsInQueue: The maximum number of calls allowed to be queued at one time.
- maxCallsInQueuePerCallType: The maximum number of calls allowed to be queued, per call type.
- maxTimeInQueue: The maximum amount of time, in seconds, a call can 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	• name (default) • description • interruptible • maxCallsInQueue • maxCallsInQueuePerCallType • maxTimeInQueue

See Search, on page 10 and Sort, on page 11.

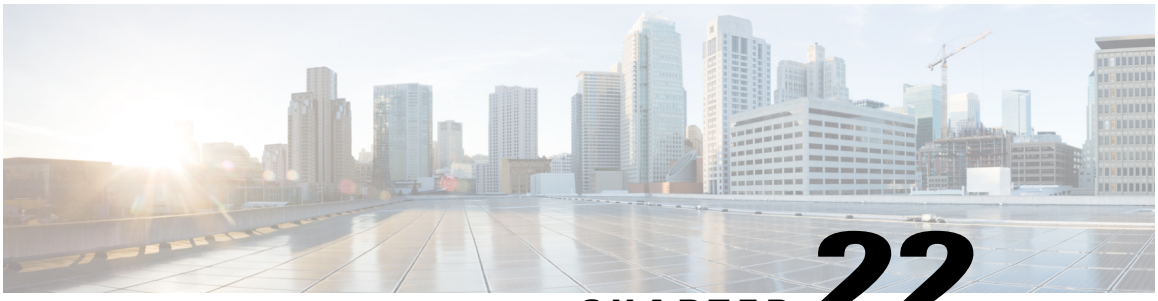
Advanced search parameters

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 media routing domain. For example, **q=nonVoiceOnly:true**.

Example get response

```
<mediaRoutingDomains>
  <mediaRoutingDomain>
     <changeStamp>0</changeStamp>
     <refURL>/unifiedconfig/config/mediaroutingdomain/5001</refURL>
     <description>System-provided media routing domain for Cisco_Chat</description>
     <interruptible>false</interruptible>
     <maxCallsInQueue>1000</maxCallsInQueue>
     <maxCallsInQueuePerCallType>1000</maxCallsInQueuePerCallType>
     <maxTimeInQueue>1000</maxTimeInQueue>
     <mediaClass>
       <name>Cisco_Chat</name>
       <id>6</id>
     </mediaClass>
     <name>Cisco_Chat</name>
     <serviceLevelThreshold>30</serviceLevelThreshold>
  </mediaRoutingDomain>
  <mediaRoutingDomain>
     <changeStamp>0</changeStamp>
     <refURL>/unifiedconfig/config/mediaroutingdomain/1</refURL>
     <description>Default Media Routing Domain for Cisco_Voice</description>
     <interruptible>false</interruptible>
     <mediaClass>
       <name>Cisco_Voice</name>
       <id>4</id>
     </mediaClass>
     <name>Cisco_Voice</name>
     <serviceLevelThreshold>30</serviceLevelThreshold>
   </mediaRoutingDomain>
 </mediaRoutingDomains>
```



CHAPTER 22

Network VRU Script API

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

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

URL

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

Operations

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

Parameters

- refURL: The refURL of the network VRU script. See Shared parameters, on page 8.
- name: The name of the network VRU as seen by CCE. See Shared parameters, on page 8.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- routingType: This field is optional and defaults to 1. Options are:
  - 1: Voice. Used by Unified CVP.
  - 2: Multichannel. Used by Email and Web Collaboration.
- vruScriptName: Required. The name of the script as it is known on the Unified CVP. Maximum length of 39 characters allowed.

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

Search and sort values

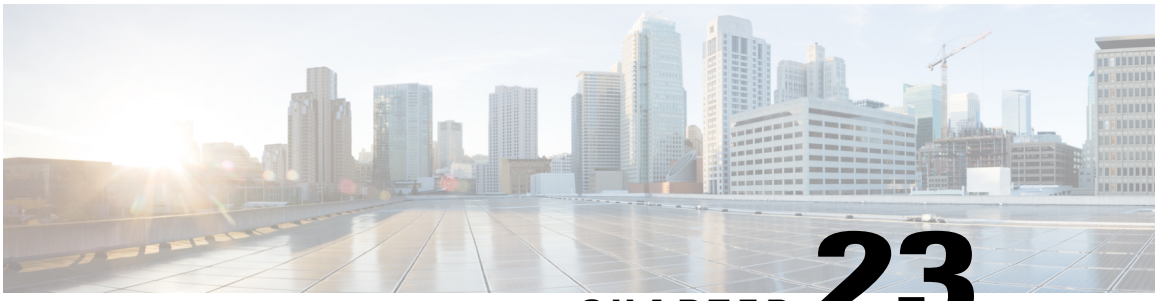
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• name • description	• name (default) • description • vruScriptName • timeout • configParam • interruptible

See Search, on page 10 and Sort, on page 11.

Example get response

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



CHAPTER 23

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. Only the skillGroupsAdded and skillGroupsRemoved parameters are allowed by the Operation API. See Agent API, on page 17.

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 8.
- changeset: Includes the parameters that are changed in an update operation.

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>
      <skillGroupsAdded>
        <skillGroup>
          <refURL>/unifiedconfig/config/skillgroup/6000<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>
```

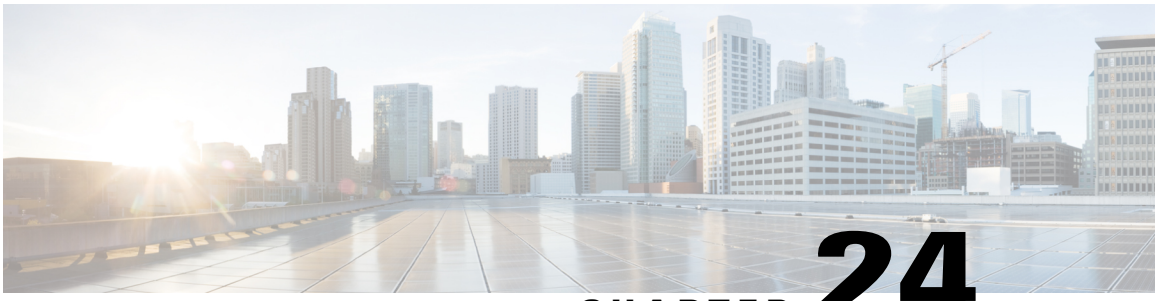
```
            <errorMessage>There were one or more errors processing the following
             request: delete agent/2</errorMessage>
            <errorType>operation.resourceErrors</errorType>
        </apiError>
    </apiErrors>
    <status>partialSuccess</status>
</operationsResult>
```

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>
```



CHAPTER 24

Peripheral Gateway API

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

URL

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

Operations

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

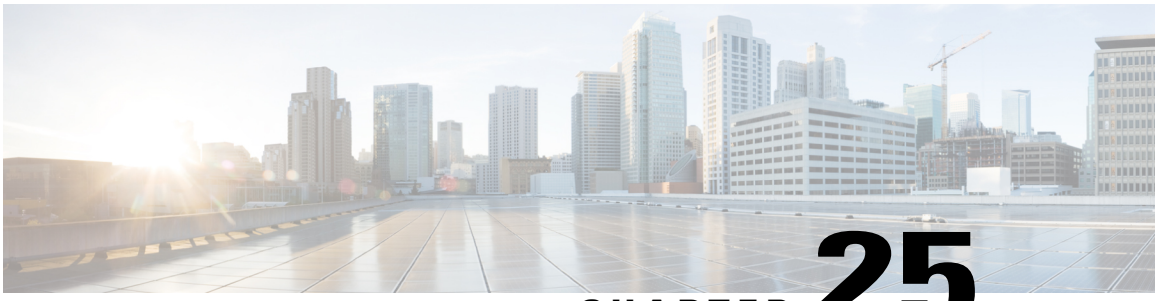
Parameters

- refURL: The refURL of the peripheral gateway. See Shared parameters, on page 8.
- name: The name of the peripheral gateway. See Shared parameters, on page 8.
- logicalControllerId: The ID of the logical controller.
- peripherals: A collection of peripheral information, including client type, name, peripheralID, routingClientID, and routingType (see Dialed Number API, on page 49 for routingType values).
    - The client type values are:
        - 13: VRU
        - 30: CUCM
        - 42: Generic PG
        - 47: MediaRouting

Example get response

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

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



CHAPTER 25

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>:port/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>:port/unifiedconfig/config/precisionqueue/<id>`.
    - **Query parameters:**
        - agentcount: Use this query parameter to have the agent count parameter included in the response.
        - attributes: Use this query parameter to have the attribute parameter included in the response.
- 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 8.

- name: The name of the precision queue. See Shared parameters, on page 8.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- department: A reference to the department (Department API, on page 43), including the refURL and name. See References, on page 5.
- bucketInterval: A reference to a bucket interval (Bucket Interval API, on page 33), including the refURL and name. See References, on page 5.
- agentCount: Returns agent count for the precision queue. Returned only when using the agentcount query parameter.
- agentOrdering: Determines the order in which agents receive calls from this queue.
    - 1: LAA (Agent availability time)
    - 2: Most skilled agent
    - 3: Least skilled agent
- id: The database id of the precision queue. Read-only field. Used in scripting.
- attributes: A collection of attribute names (attribute1, attribute2, and so on) indicating all of the attributes used in this precision queue. Returned only when the query parameter attributes=true.
- serviceLevelThreshold: Maximum time in seconds that a caller should wait before being connected with an agent.
- serviceLevelType: This value indicates how the system calculates the service level.
    - 1: Ignore abandoned calls.
    - 2: Abandoned call has negative impact.
    - 3: Abandoned call has positive impact.
- steps: Required. A collection of steps for this precision queue. You can have 1-10 steps. Returned only for get operation. See the Step parameters below.

Step parameters:

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

Term parameters:

- attribute: A reference to the attribute (Attribute API, on page 31), including the refURL, name, description, and dataType. A maximum of 5 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.

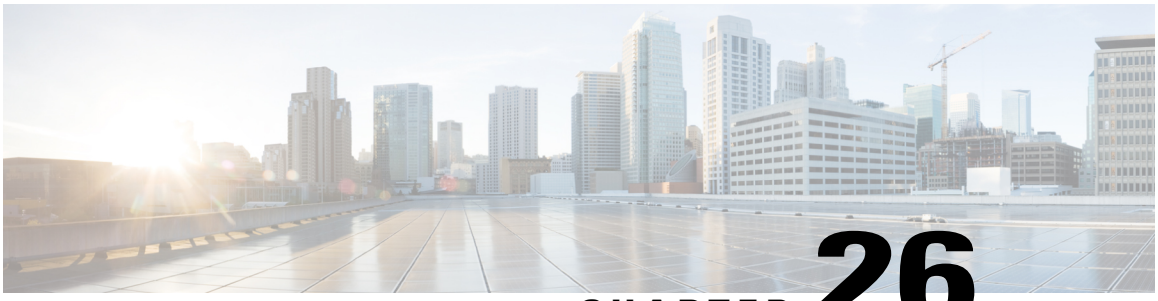
Search parameters	Sort parameters
• name • description	• name (default) • description

See Search, on page 10 and Sort, on page 11.

Example get response

```
<precisionQueue>
    <changeStamp>4</changeStamp>
    <refURL>/unifiedconfig/config/precisionqueue/5002</refURL>
    <agentOrdering>1</agentOrdering>
    <bucketInterval>
        <refURL>/unifiedconfig/config/bucketinterval/1</refURL>
        <name>Default_Bucket_Intervals</name>
    </bucketInterval>
     <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>
            <step>
                <terms>
                    <term>
                        <attribute>
                        <refURL>/unifiedconfig/config/attribute/5698</refURL>
                        <name>test</name>
                        <dataType>4</dataType>
                        </attribute>
                        <attributeRelation>3</attributeRelation>
                        <parenCount>0</parenCount>
                        <termRelation>0</termRelation>
                        <value1>2</value1>
                    </term>
                </terms>
                <waitTime>-1</waitTime>
            </step>
        </steps>
    </precisionQueue>
```



CHAPTER 26

Reason Code API

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

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

URL

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

Operations

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

Parameters

- refURL: The refURL of the reason code. See Shared parameters, on page 8.
- changestamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- code: Required. The unique reason code. Integers between 0 and 65535. Value cannot be updated after the reason code has been created.
- text: Required. The text that describes the reason code. Maximum length of 40 characters allowed.

Search and sort values

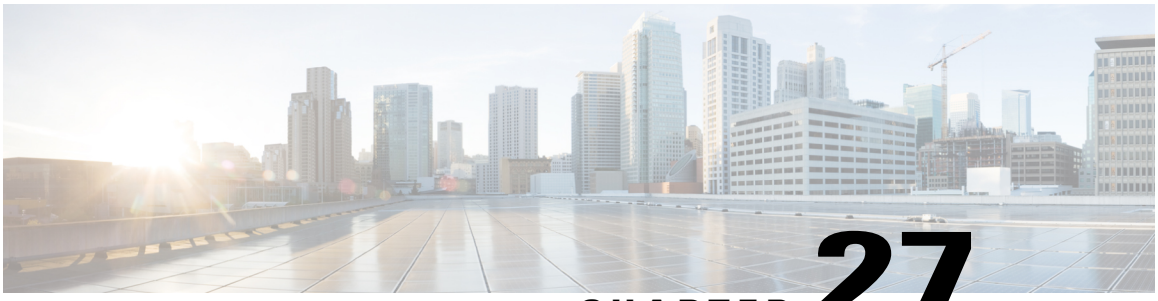
The following table shows the parameters that are searched and the parameters that are sortable.

Search parameters	Sort parameters
• text • description	• text (default) • description • code

See Search, on page 10 and Sort, on page 11.

Example get response

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



CHAPTER 27

Role API

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

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

Roles

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

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

For more information on these roles, see the Packaged CCE Administration Guide.

URL

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

Operations

- create: Creates one role, given the specified name, description, and featureList.
- delete: Deletes one role permanently from the database.
- get: Returns one role using the URL `https://<server>/unifiedconfig/config/role/<id>`.

- Auxiliary get method returns the accessList parameter containing a collection of all features that can be used to define a role. Use the URL `https://<server>/unifiedconfig/config/role/available_features`.

- list: Retrieves a list of roles.
- update: Gets the role item and updates the new configuration entered in the <accessList> fields.

Parameters

- refURL: The refURL of the role. See Shared parameters, on page 8.
- name: The role name, such as ConfigAdmin. See Shared parameters, on page 8.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- accessList: A collection of features that indicates which APIs and tools the administrators assigned to this role can access.
  - feature: Element of the access list that represents access to a feature (API / tool). Includes the following parameters:
    - name: Feature name. Maps to a valid API / tool.
    - category: Feature category, such as Agent, Call, System, and Access. This parameter is used only for get and list operations.
- administrators: A collection of administrator references (Administrator API, on page 15), including the user name, domain name, and refURL of the administrator. See References, on page 5.
- systemDefined: Read-only parameter. Indicates if this role is a system-defined role. Values are true/false.

Search and sort values

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

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

See Search, on page 10 and Sort, on page 11.

Advanced search parameters

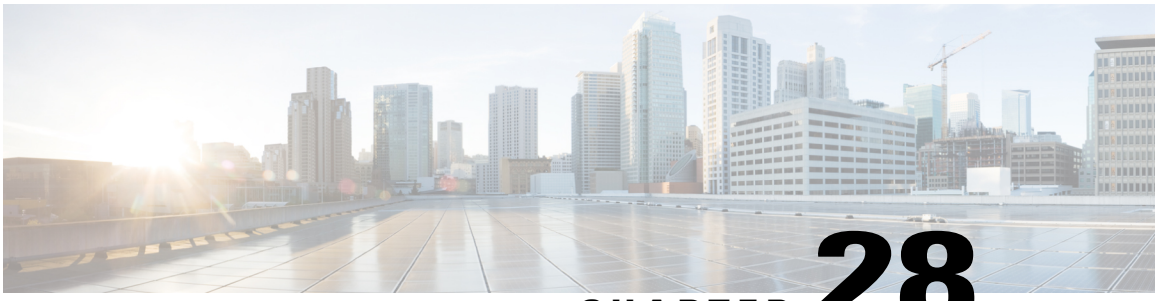
You can perform a system defined role search on the Role API:

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

- **q=systemDefined:none** Returns roles with the systemDefined parameter set to false.
- **q=systemDefined:both** Returns roles with the systemDefined parameter set to true or false.

Example get response

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



CHAPTER 28

Scan API

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

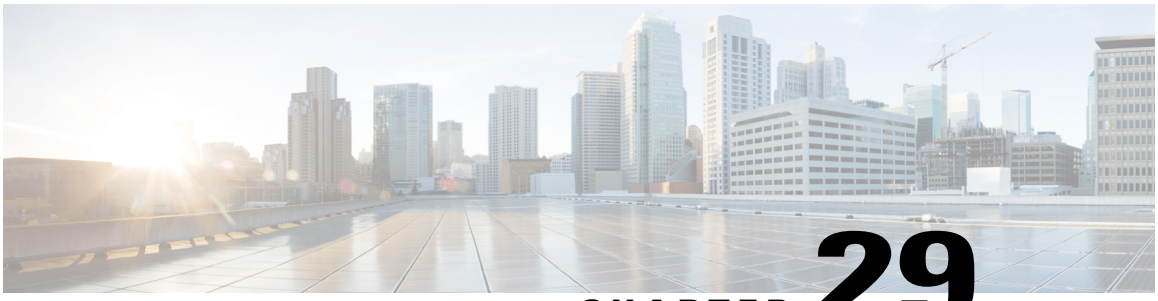
- Machine Inventory: Validate the VM Host configuration, VM layout, and required inventory configuration.
- Status: Validate the status rules. See Status API, on page 101.

URL

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

HTTP method

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



CHAPTER 29

Serviceability API

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

URL

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

Operations

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

Parameters

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

- ucceVersion: Version information for Unified CCE. Includes the same parameters listed for the version parameter, above, as well as:
  - versionString: Textual representation of the Unified CCE version.
  - patchInfos: A collection of patch information, including majorVersion, minorVersion, maintenanceVersion, srVersion, and esVersion parameters.
- capacityInfo: A collection of capacityRules indicating if the capacity limits are valid. Each rule contains the following parameters:
  - name: The name of the capacity rule.
  - max: The maximum number of items allowed for the rule.
  - actual: The current number of items configured for the rule.
- systemValidationStatus: A collection of validationRules that show the potential errors regarding system configuration. For more information on the rules, see System validation rules, on page 93. Each rule contains the following parameters:
  - name: The name of the rule.
  - isValid: Indicates if the rule is passing. Values are true/false.
  - min: The minimum number of items required to match for this rule.
  - max: The maximum number of items required to match for this rule.
  - actual: The current number of items configured that match this rule.

Example get response

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

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

- System validation rules, page 93

System validation rules

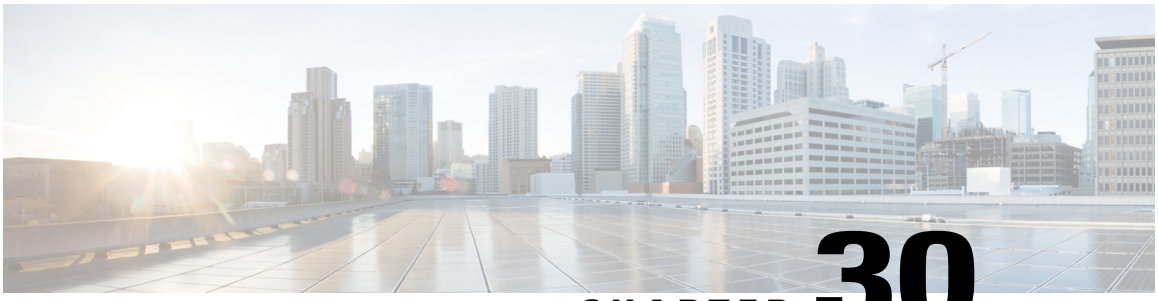
The system validation rules show the potential errors regarding system configuration.

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

Rule	Explanation
OUTBOUND_COUNT	Peripheral: Exactly 1 MR PIM must be configured with the Enterprise Name of **Outbound**.
PG_COUNT	Peripheral Gateway: Exactly 2 Peripheral Gateways must be configured.
SERVICE_MEMBER_COUNT	Service Members: No Service Members may be configured.
TYPE10_NETWORK_VRU_COUNT	VRU: Exactly 1 Type 10 Network VRU must be configured in the Network VRU Explorer.
TYPE10_NETWORK_VRU_MAP_COUNT	Peripheral Gateway: All 4 VRU Peripherals must be configured on the Generic PG and associated with the Type 10 Network VRU.
UCM_PIM_COUNT	Peripheral: Exactly 1 Unified CM Peripheral must be configured on the Generic PG in the Peripheral Explorer tool.
VRU_PIM_COUNT	Peripheral: Exactly 4 VRU Peripherals must be configured.
NOT_SKILL_GROUP_ROUTE_NAME_COUNT	Skill Groups: All Skill Group records must have a corresponding Route record with the same Enterprise Name as the Skill Group record.
ECC_VARIABLES_ENABLED_COUNT	ECC Variables: ECC variables must be enabled in the System Information tool.
SERVICE_COUNT	Services: No Services may be configured.
TRANSLATION_ROUTE_COUNT	Translation Routes: No Translation Routes may be configured.
NIC_COUNT	NICs: No NICs may be configured.
MRD_COUNT	Max Media Routing Domains: The maximum number of Media Routing Domains is 20.
MEDIA_CLASS_COUNT	Max Media Classes: The maximum number of Media Classes is 10.
NOT_PARTITIONED_COUNT	Partitioning: Partitioning must be disabled in the System Information tool.
NON_NULL_SERVICE_LEVEL_COUNT	Service Level Threshold: The default service level must not be set in the Peripheral Explorer tool. The default is set in the System Information tool.

Rule	Explanation
DEVICE_TARGET_COUNT	Device Targets: No Device Targets can be configured.
CVP_LABEL_COUNT	VRU: Each VRU PIM associated with the Generic PG in the PG Explorer tool must have exactly 1 label with a length of 10 digits.
CUCM_LABEL_COUNT	CUCM Routing Label: Exactly 1 label with length of 10 digits must be configured and associated with the Unified CM routing client.
CORRELATION_ID_RANGE_COUNT	Correlation ID: The minimum and maximum correlation number in the VRU section of the System Information tool must be 1001 and 9999 respectively.
NULL_FEATURE_SET_ID_COUNT	Feature Control Set : The Feature Control Set in the Customer Definition of the ICM Instance Explorer must set to NONE.
ECC_FOR_CVP_COUNT	ECC Variables: Exactly 9 Expanded Call Variables are required for CVP.
NETWORK_VRU_SCRIPT_COUNT	VRU: There must be a Network VRU Script with the Enterprise Name of **VXML_Server** and the Script Name of **GS,Server,V** configured in the Network VRU Script tool.
DEFAULT_DESK_SETTING_COUNT	Agent Desk Settings: Default_Agent_Desk_Setting must be set as the default Agent Desk Settings for the CUCM PIM in the PG Explorer tool.
PCCE_APP_INSTANCE_MULTICHANNEL_COUNT	Multichannel Application Instance: An Application Instance must be defined for Multichannel.
TYPE2_NETWORK_VRU_COUNT	VRU: Exactly one Type 2 Network VRU must be configured in the Network VRU Explorer tool.
TYPE2_NETWORK_VRU_MAP_COUNT	Peripheral: Both MR PIMs must be associated with a Type 2 Network VRU in the PG Explorer tool.
DIALED_NUMBER_EXTERNALL_VOICE_COUNT	Dialed Numbers: For each External Voice Dialed Numbers, there must be exactly 4 Dialed Number records for each Dialed Number String, with one for each VRU PIM.
DIALED_NUMBER_MAP_COUNT	Dialed Numbers: All Dialed Number records must not have an associated Region, ANI, and must have a maximum of 1 Call Type associated in the Call Type Map.

Rule	Explanation
AGENT_REAL_TIME_ENABLED_COUNT	Peripheral: Agent Reporting must be enabled on the Unified CM Peripheral in the PG Explorer tool.
CUSTOMER_DEFINITION_COUNT	Customer Definition: Exactly 1 Customer Definition must be configured in the ICM Instance Explorer.
CUSTOMER_DEFINITION_HAS_TYPE10_NETWORK_VRU	Customer Definition: Exactly 1 Customer Definition must have a Type 10 Network VRU selected in the ICM Instance Explorer.
DIALED_NUMBERS_REQUIRE_CUSTOMER_DEFINITION	Dialed Numbers: No Dialed Number records can have the Customer set to None.
SCRIPT_VERSIONS_TO_RETAIN	Script Versions to Retain: The number of script versions to retain must be between 1 and 100, inclusively.
APP_GATEWAY_COUNT	Application Gateway: No Application Gateways may be configured.
DATABASE_LOOKUP_COUNT	Database Lookups: No Database Lookups may be configured.
SCRIPT_VERSIONS_ALLOWED	Max Script Versions: The maximum number of script versions allowed is 100 (minimum is 1).
OEM_CP_MATCHES_DB_COLLATION	The system locale must be compatible with the database collation. The valid pairs are: cp437 / iso_1 (US English) and cp850 / iso_1 (Other Latin derivatives). All other valid pairs must be identical strings: cp936 / cp936 (Chinese), cp866/cp866 (Russian), cp 932/cp932 (Japanese), and so on. For more information, see the Collation and Locale Settings for Localization section in the Packaged CCE Install and Upgrade Guide.
AGENT_TARGETING_RULE	There must be at exactly one agent targeting rule defined in CCE.



CHAPTER 30

Skill Group API

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

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

URL

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

Operations

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

          Note Using selectedAgentCount automatically sets the summary list query parameter to **true**.
        - Summary list: See list, on page 3.
- update: Updates one skill group.

Parameters

- refURL: The refURL of the skill group. See Shared parameters, on page 8.
- name: The name of the skill group. See Shared parameters, on page 8.
- department: A reference to the department (Department API, on page 43), including the name and refURL. See References, on page 5.
- changeStamp: See Shared parameters, on page 8.
- description: See Shared parameters, on page 8.
- mediaRoutingDomain: A reference to the media routing domain (Media Routing Domain API, on page 69) including the name and refURL. See References, on page 5.
  - Defaults to Cisco_Voice MRD if this parameter is not provided.
  - This reference cannot be updated.
- agents: A collection of agents assigned to the skill group (See Agent API, on page 17). References also include firstName, lastName, agentId, and agentTeam (which includes the team name and refURL). See References, on page 5.
  - 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.
- 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 5.
- serviceLevelThreshold: Maximum time in seconds that a caller should wait before being connected with an agent. Positive integers only, or blank. Blank means use the value from the specified mediaRoutingDomain (Media Routing Domain API, on page 69).
- serviceLevelType: This value indicates how the system calculates the service level.
  - 1: Ignore Abandoned Calls (default).
  - 2: Abandoned Calls have Negative Impact.
  - 3: Abandoned Calls have Positive Impact.
- peripheralNumber: Read-only parameter. Automatically generated when using the create operation.

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

Search and sort values

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

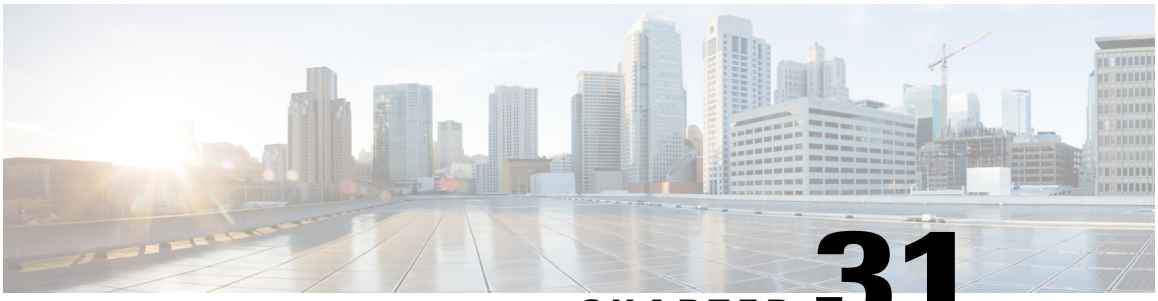
Search parameters	Sort parameters
• name • description	• name (default) • description • serviceLevelThreshold • serviceLevelType • peripheralNumber

See Search, on page 10 and Sort, on page 11.

For more information on search restrictions, see Search, on page 10.

Example get response

```
<skillGroup>
    <refURL>http://***.***.***.***/unifiedconfig/config/skillgroup/(id)</refURL>
    <name>test</name>
    <description>test skill group</description>
    <changeStamp>0</changeStamp>
    <mediaRoutingDomain>
        <name>Cisco_Voice</name>
        <refURL>https://10.86.135.206/unifiedconfig/config/mediaroutingdomain/1</refURL>
    </mediaRoutingDomain>
    <bucketInterval>
        <name>bucketIntervalName</name>
        <refURL>https://10.86.135.206/unifiedconfig/config/bucketinterval/1</refURL>
    </bucketInterval>
    <serviceLevelThreshold>20</serviceLevelThreshold>
    <serviceLevelType>1</serviceLevelType>
    <peripheralNumber>1234567</peripheralNumber>
    <agents>
        <agent>
            <refURL>https://10.86.135.206/unifiedconfig/config/agent/5000</refURL>
            <firstName>Jane</firstName>
            <lastName>Doe</lastName>
            <userName>username</userName>
            <agentId>8007</agentId>
            <canRemove>true</canRemove>
        </agent>
        <agent>
            <refURL>https://10.86.135.206/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 31

Status API

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

- Runtime category rules, on page 102
- Voice configuration category rules, on page 107

The system periodically checks the state of the rules. To force the status to be updated, refer to the Scan API, on page 89.

URL

`https://host/unifiedconfig/config/status`

Operations

- get: Returns the status rule results, using the URL `https://host/unifiedconfig/config/status`.

Response parameters

- name: The name of the status. See the Runtime category rules, on page 102 and Voice configuration category rules, on page 107.
- category: The status category. For example, RUNTIME or VOICE_CONFIG.
- level: The severity of the condition. Values include: OK, INFO, WARNING, ERROR, and BLOCKED. The BLOCKED level indicates that the rule has not been processed yet or that the failure of another rule prevents this rule from running.
- machine: A collection of machines (Machine Inventory API, on page 63) including each machine's name, type, and refURL.

Example get response

```
<results>
  <statuses>
    <status>
      <name>DS_LOGGER_INSTALLED</name>
      <category>RUNTIME</category>
```

```
      <level>OK</level>
      <machines>
        <machine>
          <refURL>/unifiedconfig/config/machineinventory/5057</refURL>
        </machine>
      </machines>
    </status>
    <status>
      <name>DS_DISTRIBUTOR_INSTALLED</name>
      <category>RUNTIME</category>
      <level>BLOCKED</level>
      <machines>
        <machine>
          <refURL>/unifiedconfig/config/machineinventory/5057</refURL>
        </machine>
      </machines>
    </status>
  </statuses>
</results>
```


- Runtime category rules, page 102
- Voice configuration category rules, page 107

Runtime category rules

The Runtime category rules show the potential errors and warnings for CCE machines.

Rules

Note All rules apply to both Side A and Side B, unless otherwise specified.

Table 1: VMWare rules

Rule	Description
VMWARE_TOOLS	VMware tools must be up-to-date.
VMWARE_GUEST_OS	The operating system setting on each Virtual Machine (VM) must match the operating system installed on the Guest.

Table 2: Data server rules

Rule	Description
LOGGER_RUNNING	The logger service must be running.
DISTRIBUTOR_RUNNING	The distributor service must be running.
LOGGER_HIST_LOGGER_RUNNING	The logger historical logger process (histlogger.exe) must be running.

Rule	Description
LOGGER_CONFIG_LOGGER_RUNNING	The logger configuration logger process (configlogger.exe) must be running.
LOGGER_REPLICATION_RUNNING	The logger replication process (replication.exe) must be running.
LOGGER_RECOVERY_RUNNING	The logger recovery process (recovery.exe) must be running.
LOGGER_CSFS_RUNNING	The logger customer support forwarding service process (csfs.exe) must be running.
LOGGER_BA_IMPORT_RUNNING	The logger import process (baimport.exe) must be running on the Side A data server when the dialer is installed on the call servers.
LOGGER_CAMPAIGN_MGR_RUNNING	The logger campaign manager process (campaignmanager.exe) must be running on the Side A data server when the dialer is installed on the call servers.
DISTRIBUTOR_CONFIG_LOGGER_RUNNING	The distributor configuration logger process (configlogger.exe) must be running.
DISTRIBUTOR_RT_CLIENT_RUNNING	The distributor real-time client process (rtclient.exe) must be running.
DISTRIBUTOR_RT_DIST_RUNNING	The distributor real-time distributor process (rtdist.exe) must be running.
DISTRIBUTOR_UPDATE_AW_RUNNING	The distributor update process (updateaw.exe) must be running.
LOGGER_INSTALLED	The logger service must be installed.
DISTRIBUTOR_INSTALLED	The distributor service must be installed.
LOGGER_AUTOMATIC	The logger service startup type must be set to automatic.
DISTRIBUTOR_AUTOMATIC	The distributor service startup type must be set to automatic.
NO_EXTRA_SERVICES_INSTALLED_DS	Only required services can be installed.

Table 3: Call server rules

Rule	Description
CTI_SVR_RUNNING	The CTI Server service must be running.
CTIOS_SVR_RUNNING	If CTIOS Server is installed, then the CTIOS Server service must be running.
DIALER_RUNNING	If dialer is installed, then the dialer service must be running.
GENERIC_PG_RUNNING	The generic PG service must be running.
MR_PG_RUNNING	The media routing PG service must be running.
GENERIC_PG_PG_AGENT_RUNNING	The generic PG PG agent process (pgagent.exe) must be running.
GENERIC_PG_PG_AGENT_ACTIVE_IDLE	The generic PG PG agent process (pgagent.exe) must be active on one side and idle on the other side.
GENERIC_PG_LIVE_DATA_ACTIVE_IDLE	The generic PG Live Data connection must be active on one side and idle on the other side.
GENERIC_PG_MDSPROC_RUNNING	The generic PG message delivery service process (mdsproc.exe) must be running.
GENERIC_PG_MDSPROC_IN_SVC_PR_ENB_DSB	The generic PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.
MR_PG_MDSPROC_RUNNING	The media routing PG message delivery service process (mdsproc.exe) must be running.
MR_PG_PG_AGENT_RUNNING	The media routing PG PG agent process (pgagent.exe) must be running.
ROUTER_RUNNING	The router service must be running.
ROUTER_DBAGENT_RUNNING	The router database agent process (dbagent.exe) must be running.
ROUTER_ROUTER_RUNNING	The router process (router.exe) must be running.
ROUTER_MDSPROC_RUNNING	The router message delivery service process (mdsproc.exe) must be running.
ROUTER_LIVE_DATA_ACTIVE_IDLE	The router Live Data connection must be active on one side and idle on the other side.

Rule	Description
ROUTER_MDSPROC_IN_SVC_PR_ENB_DSB	The router message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.
MR_PG_MDSPROC_IN_SVC_PR_ENB_DSB	The media routing PG message delivery service process (mdsproc.exe) must be enabled on one side and disabled on the other side.
MR_PG_PG_AGENT_ACTIVE_IDLE	The media routing PG PG agent process (pgagent.exe) must be active on one side and idle on the other side.
GENERIC_PG_JTAPI_RUNNING	The generic PG jtapi process (jtapigw.exe) must be running.
GENERIC_PG_JTAPI_ACTIVE_IDLE	The generic PG jtapi process (jtapigw.exe) must be active on one side and idle on the other side.
ROUTER_CCAGENT_RUNNING	The router central controller agent process (ccagent.exe) must be running.
ROUTER_CCAGENT_INSVC_ACTIVE_ENABLE_COUNT	The router central controller agent process (ccagent.exe) must be in service for both PGs.
CTI_SVR_CTI_SVR_RUNNING	The CTI server process (ctisvr.exe) must be running.
CTI_SVR_CTI_SVR_ACTIVE_IDLE	The CTI server process (ctisvr.exe) must be active on one side and idle on the other side.
CTIOS_SVR_CTIOS_SVR_NODE_RUNNING	The CTIOS server process (ctiosservernode.exe) must be running.
CTIOS_SVR_CTIOS_SVR_NODE_ACTIVE	The CTIOS server process (ctiosservernode.exe) must be active.
DIALER_BA_DIALER_SIP_RUNNING	The dialer process (badialer_sip.exe) must be running.
DIALER_BA_DIALER_SIP_ACTIVE_IDLE	The dialer process (badialer_sip.exe) must be active on one side and idle on the other side.
CTI_SVR_AUTOMATIC	The CTI server service startup type must be set to automatic.
CTIOS_SVR_AUTOMATIC	If installed, the CTIOS server service startup type must be set to automatic.
DIALER_AUTOMATIC	If installed, the dialer service startup type must be set to automatic.

Rule	Description
GENERIC_PG_AUTOMATIC	The generic PG service startup type must be set to automatic.
MR_PG_AUTOMATIC	The media routing PG service startup type must be set to automatic.
ROUTER_AUTOMATIC	The router service startup type must be set to automatic.
CTI_SVR_INSTALLED	The CTI Server service must be installed.
GENERIC_PG_INSTALLED	The generic PG service must be installed.
ROUTER_INSTALLED	The router service must be installed.
CTIOS_SVR_INSTALLED	If installed, the CTIOS Server service must be installed on both sides.
DIALER_INSTALLED	If installed, the dialer service must be installed on both sides.
MR_PG_INSTALLED	The media routing PG service must be installed.
GENERIC_PG_VRU_PIM_COUNT	Four VRU PIM processes (vrupim.exe) must be running.
GENERIC_PG_VRU_PIM_ACTIVE_IDLE	Each VRU PIM process (vrupim.exe) must be active on one side and idle on the other side.
MR_PG_MR_PIM_COUNT	The number of media routing PG MR_PIM processes (mr_pim.exe) on Side A and Side B must match. Valid if 0, 1, or 2 MR_PIMs are enabled.
MR_PG_MR_PIM_ACTIVE_IDLE	Each MR_PIM process (mr_pim.exe) must be active on one side and idle on the other side.
NO_EXTRA_SERVICES_INSTALLED_CS	Only required and optional services can be installed (extra services such as an additional PG or CTI Server are not permitted).
DIALER_INSTALLED_OUTBOUND_ENABLED	The dialers must be installed correctly when outbound is enabled.

Table 4: Connection rules

Rule	Description
DIAGNOSTIC_FRAMEWORK_CONNECTION	The diagnostic framework service must be reachable.

Rule	Description
CVP_SYSTEM_CLI_CONNECTION	The WebServicesManager service for System CLI must be reachable.
UCM_AXL_CONNECTION	The AXL service must be reachable.

Voice configuration category rules

The Voice configuration category rules show the potential errors and warnings for the call servers, data servers, and gateways.

Rules

Table 5: Data server rules

Rule	Description
GENERIC_PG_SIDE_A_NAME	Generic PG on Side A must be named PG1A.
GENERIC_PG_SIDE_B_NAME	Generic PG on Side A must be named PG1B.

Table 6: Call server rules

Rule	Description
MR_PG_SIDE_A_NAME	Media Routing PG on Side A must be named PG2A.
MR_PG_SIDE_B_NAME	Media Routing PG on Side B must be named PG2B.

Table 7: Gateway rules

Rule	Description
GW_CODEC	The dial peers for each CVP call server on each gateway are configured with the supported codec: • voice-class codec #num • dial-peer voice #num voip (codec).The required codecs are g711alaw, g711ulaw or g729r8 for the above four dial peers to CVP call servers (if the dial peers are configured on the voice gateway). The dial peer is identified via "session target ipaddress xxxxx". The IP address must point to the IP address of CVP call server. If the CVP dial peers are configured on other gateways, the rule status is OK.