Precision Queue API

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

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

**URL**

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

**Operations**

- **create**: Creates one precision queue.
- **delete**: Marks one precision queue for deletion, but does not permanently delete it. Deleting a precision queue that is referenced dynamically in a script is allowed. No new calls are queued against it, but the precision queue remains operational until calls are no longer in the queue.
- **get**: Returns one precision queue, using the URL
  https://<server>/unifiedconfig/config/precisionqueue/<id>.
  - **Query parameters**:
    - agentcount: Use this query parameter to have the agent count parameter included in the response.
    - 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**.
• name: The name of the precision queue. See Shared Parameters.

• changeStamp: See Shared Parameters.

• description: See Shared Parameters.

• bucketInterval: A reference to a bucket interval (Bucket Interval API), including the refURL and name. See References.

• 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), 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.

<table>
<thead>
<tr>
<th>Search parameters</th>
<th>Sort parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>name (default)</td>
</tr>
<tr>
<td>description</td>
<td>description</td>
</tr>
</tbody>
</table>

See [Search and Sort](#).

**Example Get Response**

```xml
<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>
          <refURL>/unifiedconfig/config/attribute/5698</refURL>
          <name>test</name>
          <dataType>4</dataType>
        </term>
      </terms>
    </step>
  </steps>
</precisionQueue>
```
<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>