Callback User's Guide

Enable Status Notifications

5/12/2025

Enable Status Notifications

Modified in 8.5.107

Starting in 8.5.105, the Callback service can publish notifications to GMS that distributes these notifications to the target specified in the callback's service request, and consequently, to the subscribers of these notifications. The possible targets can be an ORS session of an existing GMS service (`orscb` notification type) or any URL (`httpcb` notification type).

Starting in 8.5.107, you can receive two types of notifications: Callback SCXML and additional GMS Callback notifications.

> **Important**
>
> By default, this feature is turned off for all callback services.

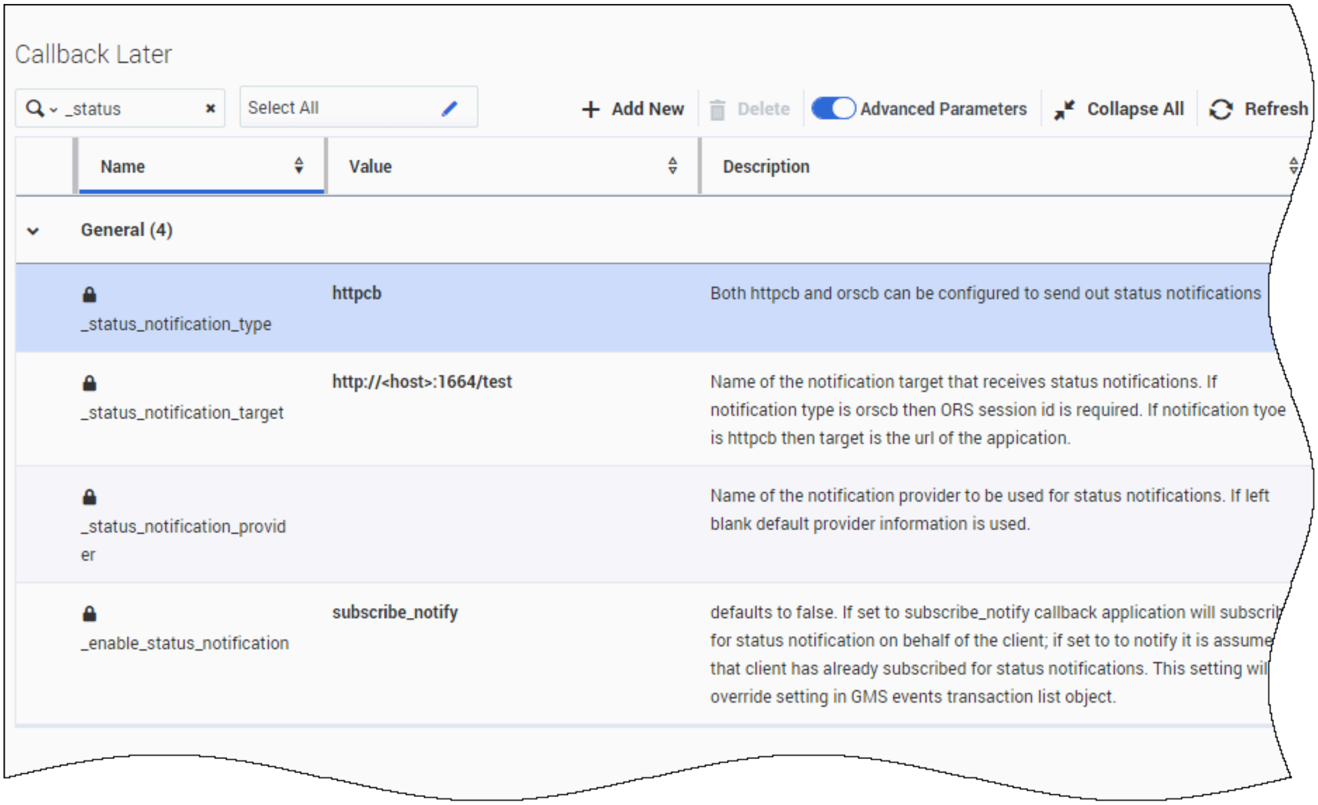
To enable Callback Status Notifications (SCXML), you can either:

- Enable the Default Status Notifications (from SCXML).
- Create a Transaction Event object that overwrites the list of default notifications and assign it to your Callback Service. You can configure additional GMS Callback Status notifications by using the Transaction List entries which override the defaults. In that scenario, the notifications will only report the events specified in this Transaction List.
- Add notifications parameters to your Callback Services query.

Callback will send the notification events and provides two subscription modes to receive them:

- `subscribe_notify`—Callback subscribes for your application to the notifications.
- `notify`—Your application must subscribe to receive events.

Enable Default Status Notifications in a Callback Service



Name	Value	Description
General (4)		
_status_notification_type	httpcb	Both httpcb and orscb can be configured to send out status notifications
_status_notification_target	http://<host>:1664/test	Name of the notification target that receives status notifications. If notification type is orscb then ORS session id is required. If notification tyoe is httpcb then target is the url of the appication.
_status_notification_provider		Name of the notification provider to be used for status notifications. If left blank default provider information is used.
_enable_status_notification	subscribe_notify	defaults to false. If set to subscribe_notify callback application will subscri for status notification on behalf of the client; if set to to notify it is assume that client has already subscribed for status notifications. This setting wil override setting in GMS events transaction list object.

To receive default callback status notifications (SCXML), open the Service Management User Interface and navigate to your Callback Service (in the **Configured Services** panel).

Enable **Advanced Parameters** and configure the following options in the **General** section:

- `_enable_status_notification= subscribe_notify`
- `_status_notification_type= httpcb` (or `orscb`)
- `_status_notification_target = Target URL` (or the ORS session id if `_status_notification_type = orscb`)

You can add the following additional parameters to your Callback queries:

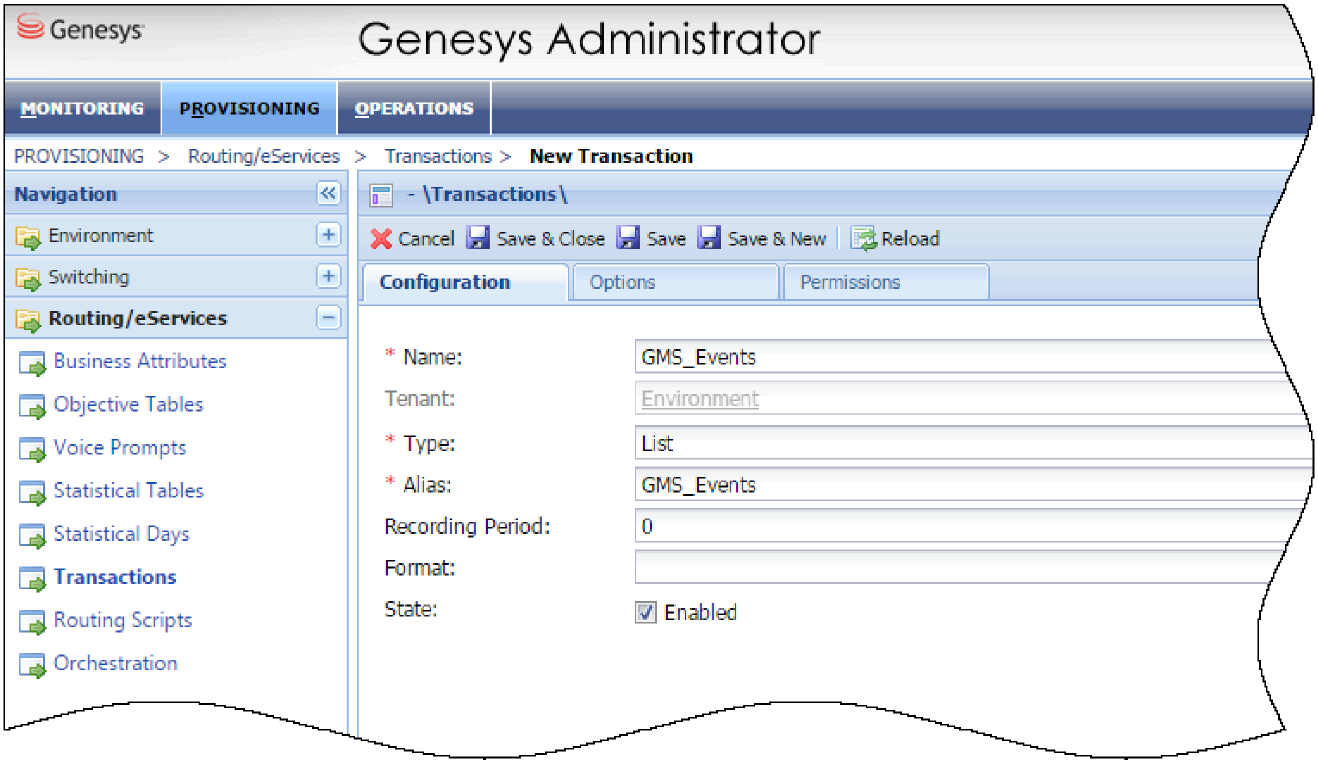
- `_status_notification_debug = false`—Set to `true` to enable the debug mode for notification.
- `_status_notification_language = <language>` where the language matches one of the supported languages used for push notifications.

The _status_notification_debug option defines the URL where the notifications will be pushed using `HTTP POST` requests.

Tip

The `orscb` notification type should be used for advanced ORS customization only.

Overwrite Default Notifications with a Transaction List



* Name:	GMS_Events
Tenant:	Environment
* Type:	List
* Alias:	GMS_Events
Recording Period:	0
Format:	
State:	☑ Enabled

Start by defining a Transaction List object that includes the notifications and the associated events triggering notifications.

Open Genesys Administrator. In `PROVISIONING > Routing/eServices > Transactions`, click **New** to create the `GMS_Events` list.

In the **Options** tab, create a `properties` section with:

- `_enable_status_notification` = `subscribe_notify`
- `_status_notification_provider` = <customerprovider> or blank for default provider
- `_status_notification_type` = `httpcb`
- `_status_notification_target=<Target URL>`

- `_status_notification_debug= false`
- `_status_notification_language = <language>` where the language matches one of the supported languages used for push notifications.

Then, create a section for each subscribed event and define the data that your application needs to receive in the notification event.

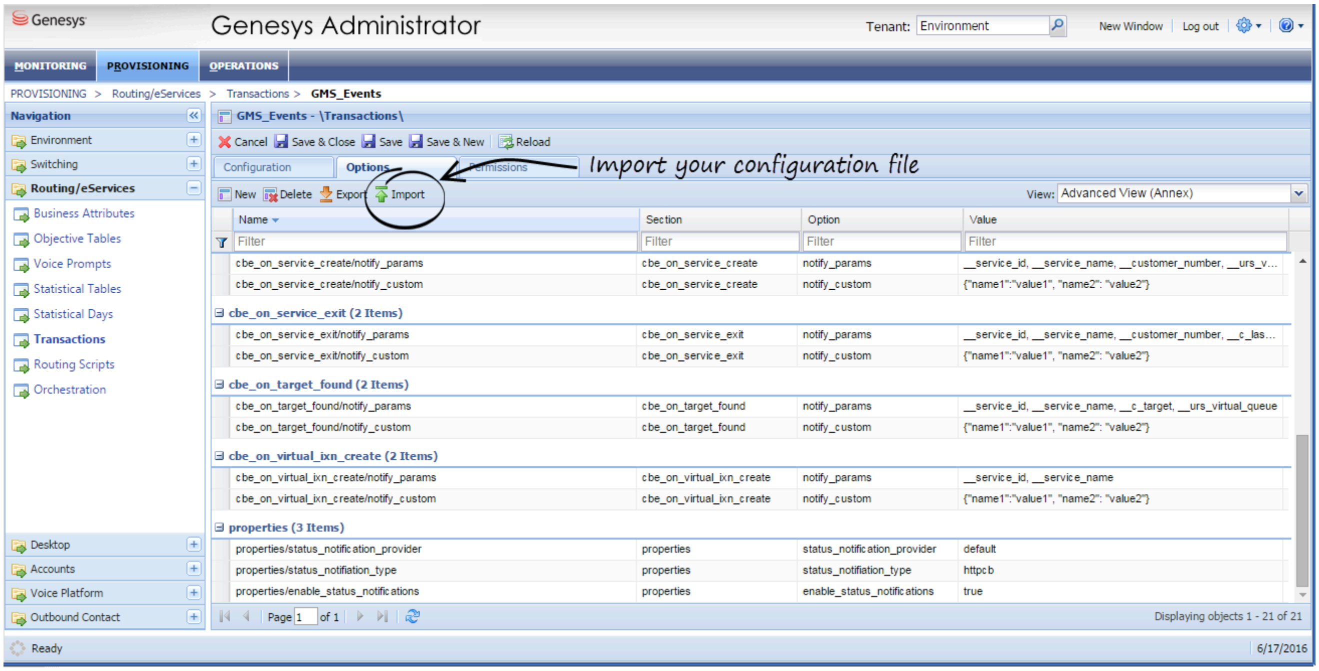
- `notify_params`—The comma-separated list of callback parameters to retrieve. See the reference to get the list of parameters that can be retrieved.
- `notify_custom`—(Optional) A JSON object of the custom attached data to send in the notification in addition to the callback parameters set in `notify_params`.

> ### Tip
>
> - Either click **New** to add the following options or copy the source below to a `GMS_Events.cfg` file that you can import in your Transaction List.
> - You do **not** have to include all the events listed below.
> - The `notify_custom` parameter should suite your use case or can be removed if not needed.

In the XML sample below, `c_target` must match `c_target` as provided by URS.

```
[properties]
_enable_status_notification = notify
_status_notification_provider =
_status_notification_type = httpcb
_status_notification_target =<your URL>

[_cbe_on_service_create]
notify_params = _service_id, _service_name, _customer_number, _urs_virtual_queue
notify_custom = {"name1":"value1", "name2": "value2"}

[_cbe_on_virtual_ixn_create]
notify_params = _service_id, _service_name
notify_custom = {"name1":"value1", "name2": "value2"}

[_cbe_on_target_found]
notify_params = _service_id, _service_name, c_target, _urs_virtual_queue
notify_custom = {"name1":"value1", "name2": "value2"}

[_cbe_on_dial_init]
notify_params = _service_id, _service_name, _customer_number, c_dialed_number
notify_custom = {"name1":"value1", "name2": "value2"}

[_cbe_on_dial_done]
notify_params = _service_id, _service_name, _customer_number, c_dialed_number, c_call_result, c_call_num_attempt
notify_custom = {"name1":"value1", "name2": "value2"}

[_cbe_on_connect_treatment_start]
notify_params = _service_id, _service_name, _customer_number, _vq_for_outbound_calls, c_dialed_number
notify_custom = {"name1":"value1", "name2": "value2"}

[_cbe_on_customer_queued]
notify_params = _service_id, _service_name, _customer_number, _vq_for_outbound_calls, c_dialed_number
notify_custom = {"name1":"value1", "name2": "value2"}

[_cbe_on_route_to_agent]
notify_params = _service_id, _service_name, _customer_number, _urs_virtual_queue, c_agent_id,
c_agent_extension
notify_custom = {"name1":"value1", "name2": "value2"}

[_cbe_on_service_exit]
notify_params = _service_id, _service_name, _customer_number, c_last_dialed_number, c_termination_type
notify_custom = {"name1":"value1", "name2": "value2"}
```

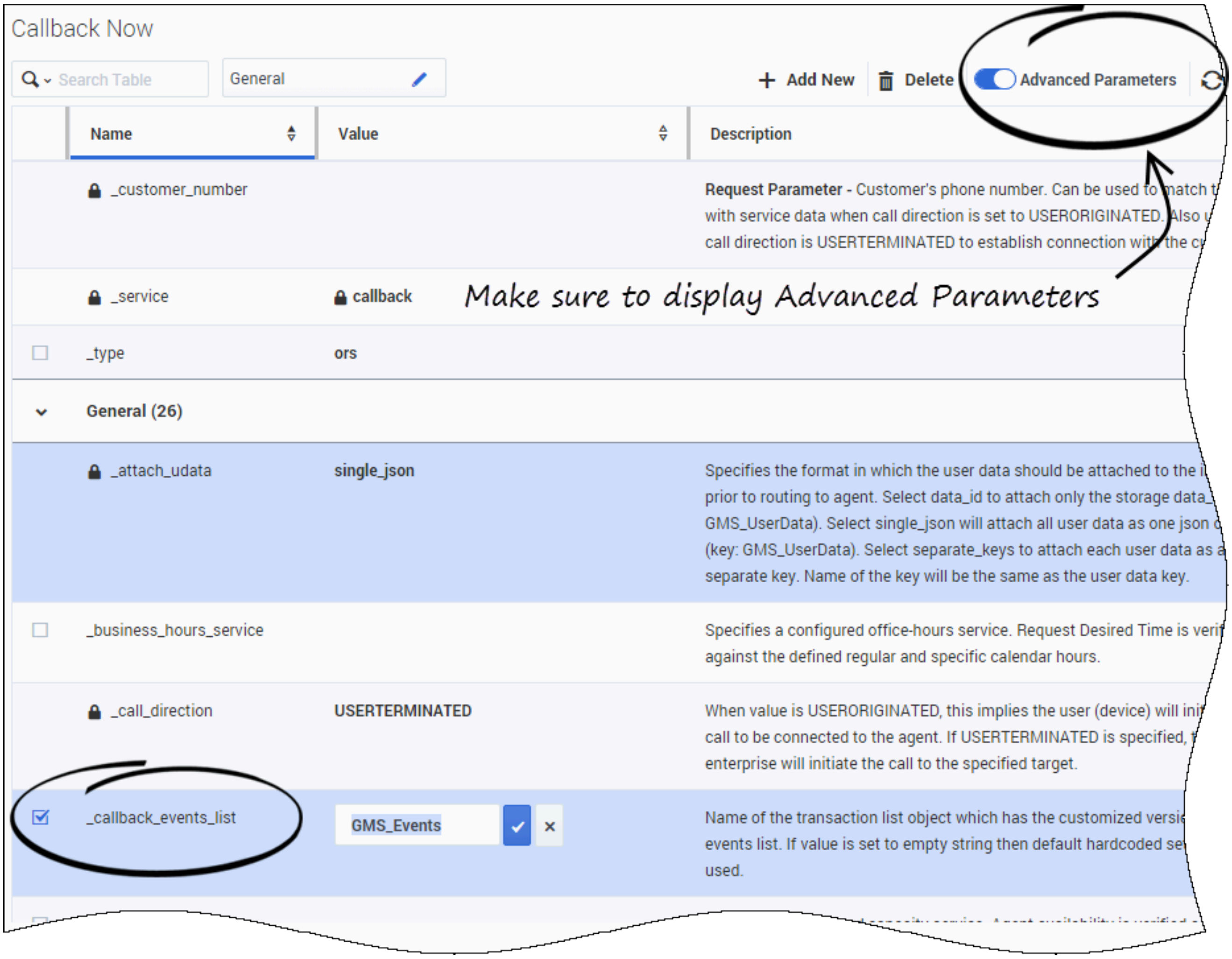
```
[_cbe_on_callback_scheduled]
notify_params=_customer_number,_phone_number,_desired_time
notify_custom={"state":"scheduled"}

[_cbe_on_callback_rescheduled]
notify_params=_customer_number,_phone_number,_desired_time
notify_custom={"state":"rescheduled"}
```



Import your configuration file

Name	Section	Option	Value
cbe_on_service_create/notify_params	cbe_on_service_create	notify_params	__service_id, __service_name, __customer_number, __urs_v...
cbe_on_service_create/notify_custom	cbe_on_service_create	notify_custom	{"name1":"value1", "name2": "value2"}
cbe_on_service_exit (2 Items)			
cbe_on_service_exit/notify_params	cbe_on_service_exit	notify_params	__service_id, __service_name, __customer_number, __c_las...
cbe_on_service_exit/notify_custom	cbe_on_service_exit	notify_custom	{"name1":"value1", "name2": "value2"}
cbe_on_target_found (2 Items)			
cbe_on_target_found/notify_params	cbe_on_target_found	notify_params	__service_id, __service_name, __c_target, __urs_virtual_queue
cbe_on_target_found/notify_custom	cbe_on_target_found	notify_custom	{"name1":"value1", "name2": "value2"}
cbe_on_virtual_ixn_create (2 Items)			
cbe_on_virtual_ixn_create/notify_params	cbe_on_virtual_ixn_create	notify_params	__service_id, __service_name
cbe_on_virtual_ixn_create/notify_custom	cbe_on_virtual_ixn_create	notify_custom	{"name1":"value1", "name2": "value2"}
properties (3 Items)			
properties/status_notification_provider	properties	status_notification_provider	default
properties/status_notifiation_type	properties	status_notifiation_type	httpcb
properties/enable_status_notifications	properties	enable_status_notifications	true

Add the Event Transaction List to the Callback Service



Callback Now

	Name	Value	Description
	_customer_number		**Request Parameter -** Customer's phone number. Can be used to match t with service data when call direction is set to USERORIGINATED. Also u call direction is USERTERMINATED to establish connection with the c
	_service	callback	
□	_type	ors	
	General (26)		
	_attach_udata	single_json	Specifies the format in which the user data should be attached to the i prior to routing to agent. Select data_id to attach only the storage data_ GMS_UserData). Select single_json will attach all user data as one json ((key: GMS_UserData). Select separate_keys to attach each user data as a separate key. Name of the key will be the same as the user data key.
□	_business_hours_service		Specifies a configured office-hours service. Request Desired Time is verif against the defined regular and specific calendar hours.
	_call_direction	USERTERMINATED	When value is USERORIGINATED, this implies the user (device) will ini call to be connected to the agent. If USERTERMINATED is specified, t enterprise will initiate the call to the specified target.
☑	_callback_events_list	GMS_Events	Name of the transaction list object which has the customized versi events list. If value is set to empty string then default hardcoded se used.

Make sure to display Advanced Parameters

Edit the **Advanced Parameters** in the **General** section of your Callback Service.

Set the `_callback_events_list` to the name of the Transaction List created above, `GMS_Events` in our example.

> ### Important
> If you set other status notification parameters (`_status_notification_type`, `_status_notification_target`, `_status_notification_provider`) in your Callback service configuration or in your REST queries, they override the values set in the Transaction List object.

Callback Status Notifications Events

Callback notifications consist in a JSON object that contains:

- deviceId—The custom id provided at subscription time by the subscriber.
- message—The notification message as defined in the Callback Events Transaction List.

The following JSON code is an event sample.

```
{
        "event_id": "_cbe_on_service_create",
        "timestamp": "1467575991",
        "_service_id": "445-20e740d3-8458-43d6-834d-3713c3385bac",
        "_service_name": "samples_dev",
        "_callback_state": "QUEUED",
        "_customer_number": "5115",
        "_urs_virtual_queue": "SIP_VQ_SIP_Switch",
        "name1": "value1",
        "name2": "value2"
}
{
        "event_id": "_cbe_on_virtual_ixn_create",
        "timestamp": "1467575992",
        "_service_id": "445-20e740d3-8458-43d6-834d-3713c3385bac",
        "_callback_state": "QUEUED",
        "name1": "value1",
        "name2": "value2"
}
{
        "event_id": "_cbe_on_dial_init",
        "timestamp": "1467575992",
        "_service_id": "445-20e740d3-8458-43d6-834d-3713c3385bac",
        "_service_name": "samples_dev",
        "_callback_state": "QUEUED",
        "_customer_number": "5115",
        "c_dialed_number": "5115",
        "name1": "value1",
        "name2": "value2"
}
{
        "event_id": "_cbe_on_dial_done",
        "timestamp": "1467576012",
        "_service_id": "445-20e740d3-8458-43d6-834d-3713c3385bac",
        "_service_name": "samples_dev",
        "_callback_state": "QUEUED",
        "_customer_number": "5115",
        "c_dialed_number": "5115",
        "c_call_result": 0,
        "c_call_num_attempt": 1,
        "name1": "value1",
        "name2": "value2"
}
{
        "event_id": "_cbe_on_connect_treatment_start",
        "timestamp": "1467576012",
        "_service_id": "445-20e740d3-8458-43d6-834d-3713c3385bac",
        "_service_name": "samples_dev",
        "_callback_state": "QUEUED",
        "_vq_for_outbound_calls": "VQ_GMS_REP_SIP_Switch",
        "c_dialed_number": "5115",
        "name1": "value1",
        "name2": "value2"
}
{
        "event_id": "_cbe_on_customer_queued",
        "timestamp": "1467576016",
        "_service_id": "445-20e740d3-8458-43d6-834d-3713c3385bac",
        "_service_name": "samples_dev",
        "_callback_state": "QUEUED",
        "_vq_for_outbound_calls": "VQ_GMS_REP_SIP_Switch",
        "c_dialed_number": "5115",
        "name1": "value1",
        "name2": "value2"
}
{
        "event_id": "_cbe_on_target_found",
        "timestamp": "1467576016",
        "_service_id": "445-20e740d3-8458-43d6-834d-3713c3385bac",
        "_service_name": "samples_dev",
        "_callback_state": "QUEUED",
        "_urs_virtual_queue": "SIP_VQ_SIP_Switch",
        "c_target": {
                "agent": "KSippola",
                "dn": "7001",
                "id": "Customer_Service",
                "place": "SIP_Server_Place1",
                "resource": "7001",
                "return": "target",
                "stat_value": "0",
                "switch": "SIP_Switch",
                "type": "GA",
                "vq": "SIP_VQ_SIP_Switch"
        },
        "name1": "value1",
        "name2": "value2"
}
{
        "event_id": "_cbe_on_service_exit",
        "timestamp": "1467576291",
        "_service_id": "445-20e740d3-8458-43d6-834d-3713c3385bac",
        "_service_name": "samples_dev",
        "_callback_state": "QUEUED",
        "c_termination_type": "COMPLETED.AGENT_CONNECTED",
        "name1": "value1",
        "name2": "value2"
}
```

Reference for Notification Events

Event Name	When this event is triggered	List of attributes that you can retrieve
_cbe_on_service_create	As soon as the callback service (ORS session) is started.	_service_id _service_name _customer_number _urs_virtual_queue
_cbe_on_virtual_ixn_create	When the virtual interaction is successfully created in URS.	_service_id _service_name
_cbe_on_target_found	When the callback has found the target and URS reports the target to ORS.	_service_id _service_name _urs_virtual_queue c_agent_id c_agent_extension
_cbe_on_dial_init	When the dialing to the customer is started. Note: This behavior applies for both standard and preview callback.	_service_id _service_name _customer_number c_dialed_number
_cbe_on_dial_done	When the dialing result is known.	_service_id _service_name _customer_number c_dialed_number c_call_result c_call_num_attempt
_cbe_on_connect_treatment_start	When the greeting treatment is started right after the successful CPD.	_service_id _service_name _vq_for_outbound_calls c_dialed_number
_cbe_on_customer_queued	In User Terminated scenarios, as soon as the `onconnect` treatment is over, the virtual interaction becomes routable and the customer is	_service_id _service_name

Event Name	When this event is triggered	List of attributes that you can retrieve
	placed into a queue to wait for an agent.	_vq_for_outbound_calls c_dialed_number
_cbe_on_route_to_agent	When the call is transferred from Routing Point to the agent.	_service_id _service_name c_agent_id c_agent_extension
_cbe_on_service_exit	In all exit scenarios.	_service_id _service_name c_dialed_number c_termination_type
_cbe_on_callback_scheduled	When a callback in `SCHEDULE` status is created.	_service_id _service_name _desired_time _customer_number _v_queue
_cbe_on_callback_rescheduled	When a callback in `SCHEDULE` status is re-scheduled.	_service_id _service_name _desired_time _customer_number _v_queue
_cbe_on_callback_cancelled	When the callback is canceled.	_service_id _service_name _desired_time _customer_number _v_queue
_cbe_on_callback_status_updated	When the `_callback_state` field is updated by a REST query. This can be due to ORS updates.	_service_id _service_name _desired_time _customer_number _v_queue
_cbe_on_callback_submitted	When the callback is submitted for ORS execution.	_service_id

Event Name	When this event is triggered	List of attributes that you can retrieve
		_service_name _desired_time _customer_number _v_queue
_cbe_on_callback_resubmitted	When the callback is re-submitted for ORS execution.	_service_id _service_name _desired_time _customer_number _v_queue
_cbe_on_callback_submit_failed	When submit for execution fails.	_service_id _service_name _desired_time _customer_number _v_queue
_cbe_on_callback_processing_failed	When the callback processing fails.	_service_id _service_name _desired_time _customer_number _v_queue
_cbe_on_callback_queued	When the callback is successfully submitted and its state changed to `QUEUED`.	_service_id _service_name _desired_time _customer_number _v_queue