Callback User's Guide

Enable Status Notifications

5/11/2025

Enable Status Notifications

Modified in 8.5.211

Version	Update
8.5.105	The Callback service can now 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).
8.5.107	You can now receive two types of notifications: Callback SCXML and additional GMS Callback notifications. Important By default, this feature is turned off for all callback services.
8.5.211	GMS can now send a notification reminder event before the callback is dialed.
8.5.232	• The `_cbe_on_dial_done` event is now sent for each dial request, not just one time. • The `_cbe_on_service_exit` event is always sent at the end of the Callback strategy before the subscription is removed. Its parameters, such as the `c_last_dialed_number` parameter, are set in the different states of the strategy, according to the status of the Callback.

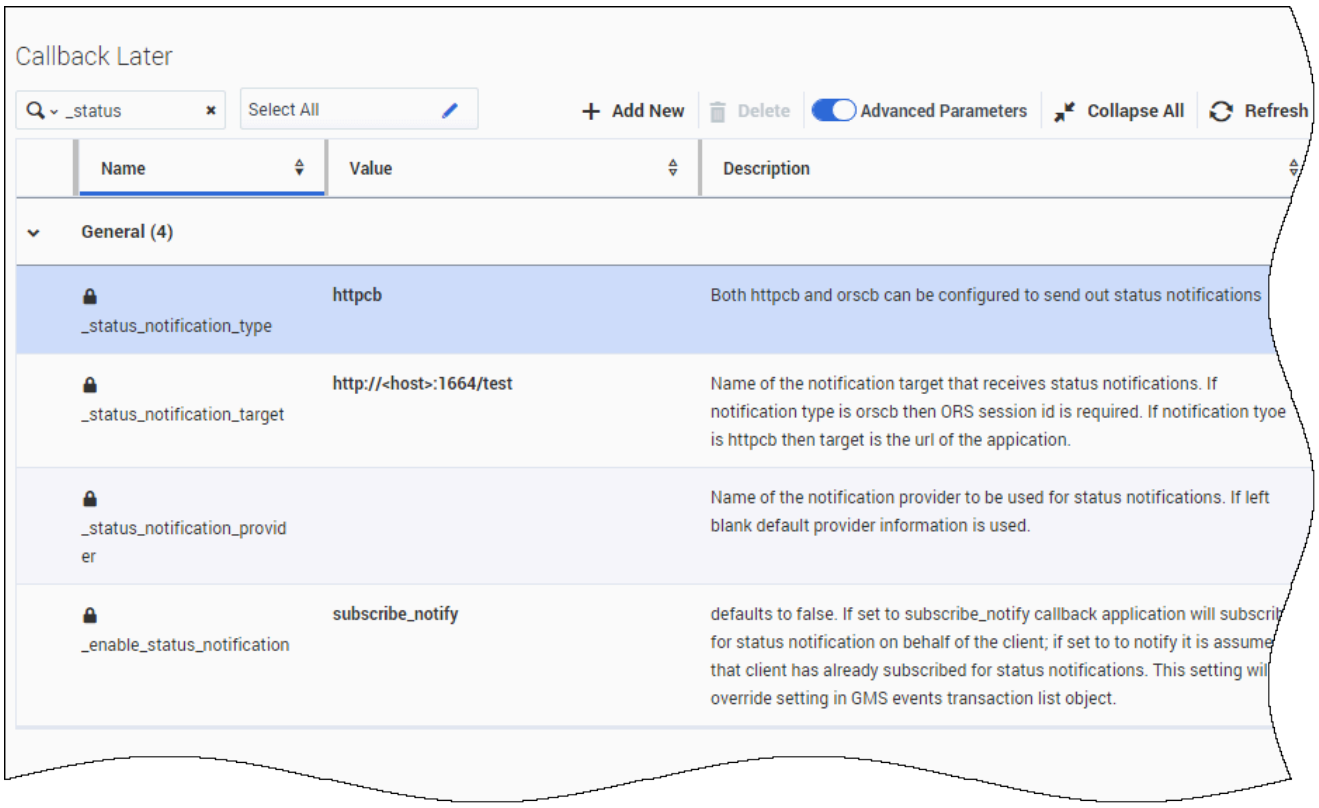
To enable Callback Status Notifications (SCXML), you can either:

- 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



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.

Enable Reminder Notifications

If you enabled default status notifications, you can also enable the Reminder Notifications in your Callback service.

- Configure _enable_notification_reminder to `true` and, by default, the system will send the `_cbe_on_callback_reminder` notification event 300 seconds before the dial time of the call.
- You can change the value of the _notification_reminder_buffer option to get the reminder notification earlier or later. The default value is 300 seconds.

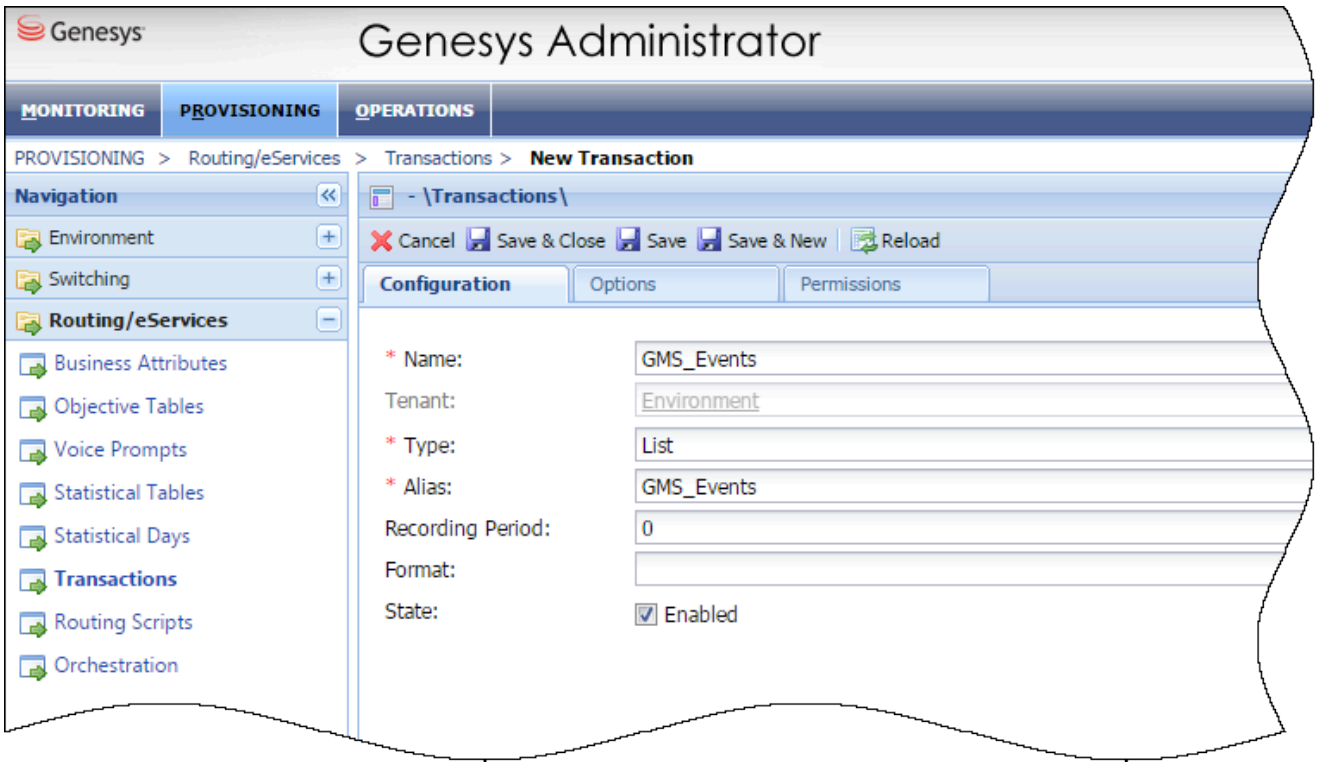
> **Important**
> The time when the reminder is sent depends on the URS Estimated Wait Time (URS EWT) of the callback. You can get the URS EWT value by checking the callback's position in queue (ewt) using the callback API.

The Reminder feature periodically checks the EWT retrieved from URS for each call.

- If `URS EWT < _notification_reminder_buffer`, the Reminder feature sends the reminder event status notification.
- If not, depending on the EWT's value, the Reminder feature schedules the next check for the call:
  - Every 30 seconds if URS EWT is not defined,
  - Every 45 seconds if URS EWT < 600 seconds,
  - Every 300 seconds if 600 seconds < URS EWT < 3600 seconds
  - Every 1800 seconds if URS EWT > 3600 seconds

Limitation: The frequency of the Reminder periodical checks is not configurable.

Overwrite Default Notifications with a Transaction List



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. Note that you can also retrieve some specific user data there in addition to callback parameters.

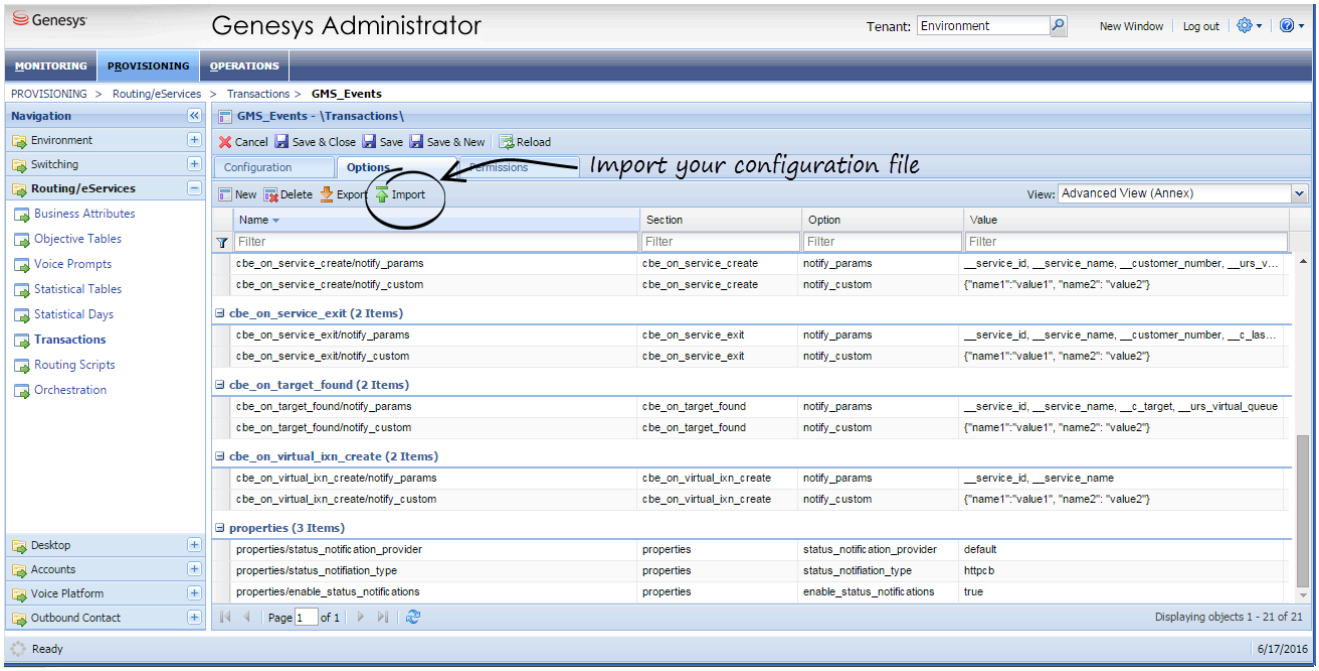
- `notify_custom`—(Optional) A JSON object of the custom attached data to send in the notification in addition to the callback parameters set in `notify_params`.

> **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.

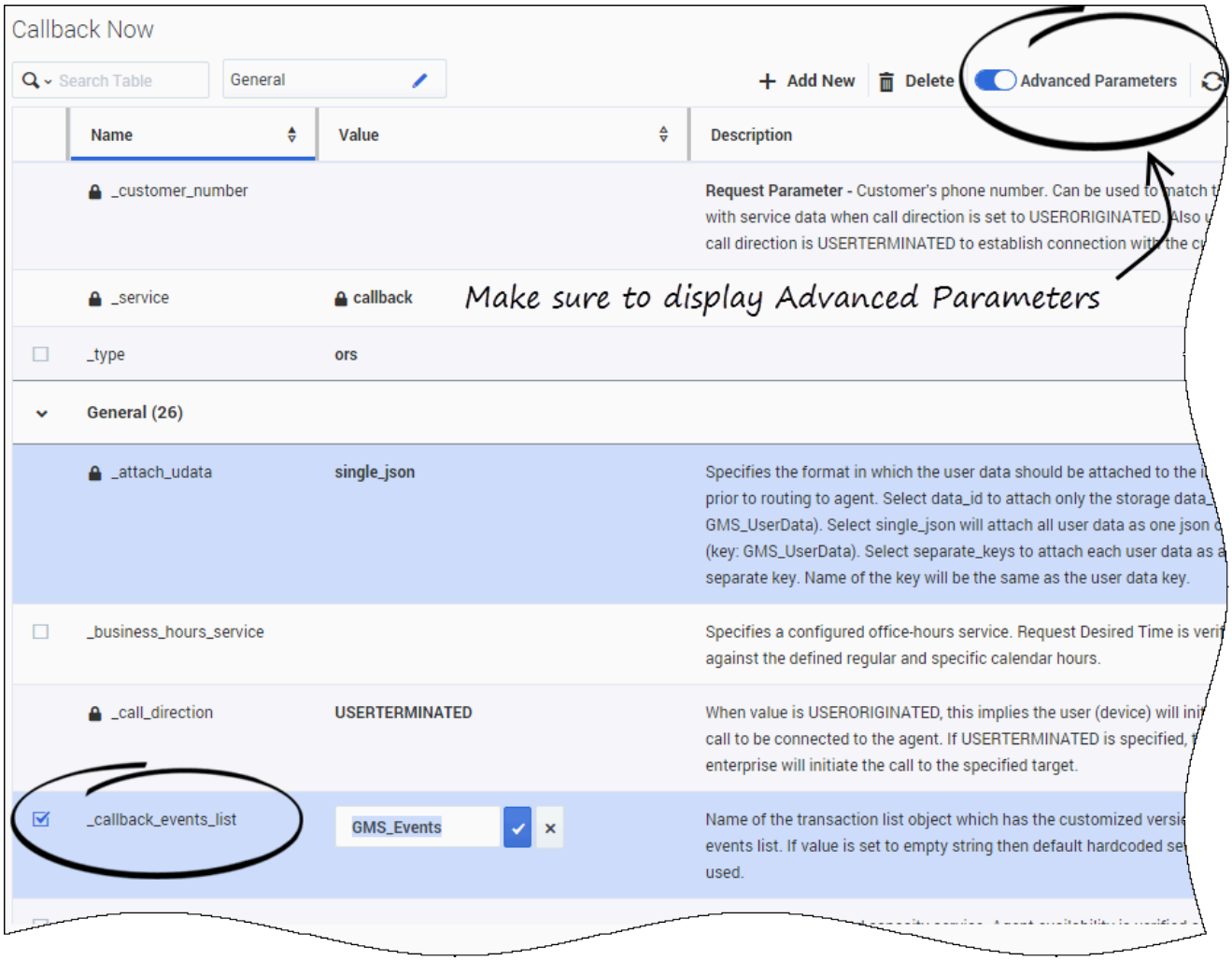
In the XML sample below, `c_target` must match `c_target` as provided by URS.

```
[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"}
```


Genesys
Genesys Administrator
Tenant: Environment
New Window
Log out
MONITORING
PROVISIONING
OPERATIONS
PROVISIONING > Routing/eServices > Transactions > GMS_Events
Navigation
Environment
Switching
Routing/eServices
Business Attributes
Objective Tables
Voice Prompts
Statistical Tables
Statistical Days
Transactions
Routing Scripts
Orchestration
Desktop
Accounts
Voice Platform
Outbound Contact
GMS_Events - \Transactions\
Cancel
Save & Close
Save
Save & New
Reload
Configuration
Options
Permissions
Import your configuration file
New
Delete
Export
Import
View: Advanced View (Annex)
Name
Section
Option
Value
Filter
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
Page 1 of 1
Displaying objects 1 - 21 of 21
Ready
6/17/2016

Add the Event Transaction List to the Callback Service



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 of a JSON object which contains:

- `deviceId`—The custom id provided at subscription time by the subscriber.
- `message`—The notification message as defined in the Callback Events Transaction List.
- timestamp—The timestamp for this notification.
- _service_id—The ID of the service which sent the notification.
- _service_name—The name of the service which sent the notification.

The Notification events can include some additional attributes detailed in this table. Check the Notification Event reference to get the list of attributes available for a given notification.

Optional attributes	Description	Example
c_target	A selected target that specifies the agent/queue resource that will process this request.	`"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" }`
c_agent_id	Equals to `c_target.id`.	`"c_agent_id": "Customer_Service"`
c_agent_extension	The agent's DN target (equals to `c_target.dn`).	`"c_agent_extension": "7001"`
c_dialed_number	The customer number.	`"c_dialed_number": "5115"`
c_call_result	Indicates the `_genesys.ixn.callState` state. Possible values are listed in the IxnIntfObjectModel page.	`"c_call_result": "0"`
c_call_num_attempt	The number of outbound call dialing attempts.	`"c_call_num_attempt": "1"`
c_termination_type	The termination type, also known as Disposition value. It equals to the `_CB_DISPOSITION` value.	`"c_termination_type": "COMPLETED.AGENT"`

The following JSON code is an event sample where the notify_custom parameter was configured to `{"name1":"value1", "name2": "value2"}`.

```
{
        "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	List of attributes specific to this event	When this event is triggered
_cbe_on_service_create	_customer_number _urs_virtual_queue	As soon as the callback service (ORS session) is started.
_cbe_on_virtual_ixn_create		When the virtual interaction is successfully created in URS.
_cbe_on_target_found	_urs_virtual_queue c_target c_agent_id c_agent_extension	When the callback has found the target and URS reports the target to ORS.
_cbe_on_dial_init	_customer_number c_dialed_number	When the dialing to the customer is started. Note: This behavior applies to both standard and preview callback.
_cbe_on_dial_done	_customer_number c_dialed_number c_call_result c_call_num_attempt	When the dialing result is known. Starting in 8.5.232, the `_cbe_on_dial_done` event is now sent for each dial request, not just one time. **Important** Only `_genesys.ixn.callState` types related to dial tone will be mapped with the `c_call_result` attribute of the `_cbe_on_dial_done` event. Refer to the IxnIntfObjectModel to get the list of call states.
_cbe_on_connect_treatment_start	_vq_for_outbound_calls c_dialed_number	When the greeting treatment is started right after the successful CPD.
_cbe_on_customer_queued	_vq_for_outbound_calls c_dialed_number	In User Terminated scenarios, as soon as the `onconnect` treatment is over, the virtual interaction becomes routable and the customer is placed into a queue to wait for an agent.

Event Name	List of attributes specific to this event	When this event is triggered
_cbe_on_route_to_agent	_urs_virtual_queue c_agent_id c_agent_extension	When the call is transferred from Routing Point to the agent.
_cbe_on_service_exit	c_last_dialed_number _customer_number c_termination_type	In all exit scenarios. Starting in 8.5.232, the `_cbe_on_service_exit` event is always sent at the end of the Callback strategy before the subscription is removed. Its parameters, such as the `c_last_dialed_number` parameter, are set in the different states of the strategy, according to the status of the Callback.
_cbe_on_callback_scheduled	_desired_time _customer_number _v_queue	When a callback in `SCHEDULE` status is created.
_cbe_on_callback_rescheduled	_desired_time _customer_number _v_queue	When a callback in `SCHEDULE` status is re-scheduled.
_cbe_on_callback_cancelled	_desired_time _customer_number _v_queue	When the callback is canceled.
_cbe_on_callback_status_updated	_desired_time _customer_number _v_queue	When the `_callback_state` field is updated by a REST query. This can be due to ORS updates.

Event Name	List of attributes specific to this event	When this event is triggered
	_urs_virtual_queue	
_cbe_on_callback_reminder Added in 8.5.211		By default, you receive the reminder event 300 seconds before the dial time of the call. Configure _enable_notification_reminder to `true` to enable this event and change the value of _notification_reminder_buffer to get the reminder earlier or later.
_cbe_on_callback_submitted	_desired_time _v_queue _urs_virtual_queue	When the callback is submitted for ORS execution.
_cbe_on_callback_resubmitted		When the callback is re-submitted for ORS execution.
_cbe_on_callback_submit_failed		When submit for execution fails.
_cbe_on_callback_processing_failed	_desired_time _customer_number _v_queue	When the callback processing fails.
_cbe_on_callback_queued	_customer_number _v_queue _v_queue_for_outbound_calls	When the callback is successfully submitted and its state changed to `QUEUED`.