Callback User's Guide

Play Treatments

5/8/2025

Play Treatments

Contents

> ### Important
> The following configuration is required for voice scenarios.

Set up Play Treatments
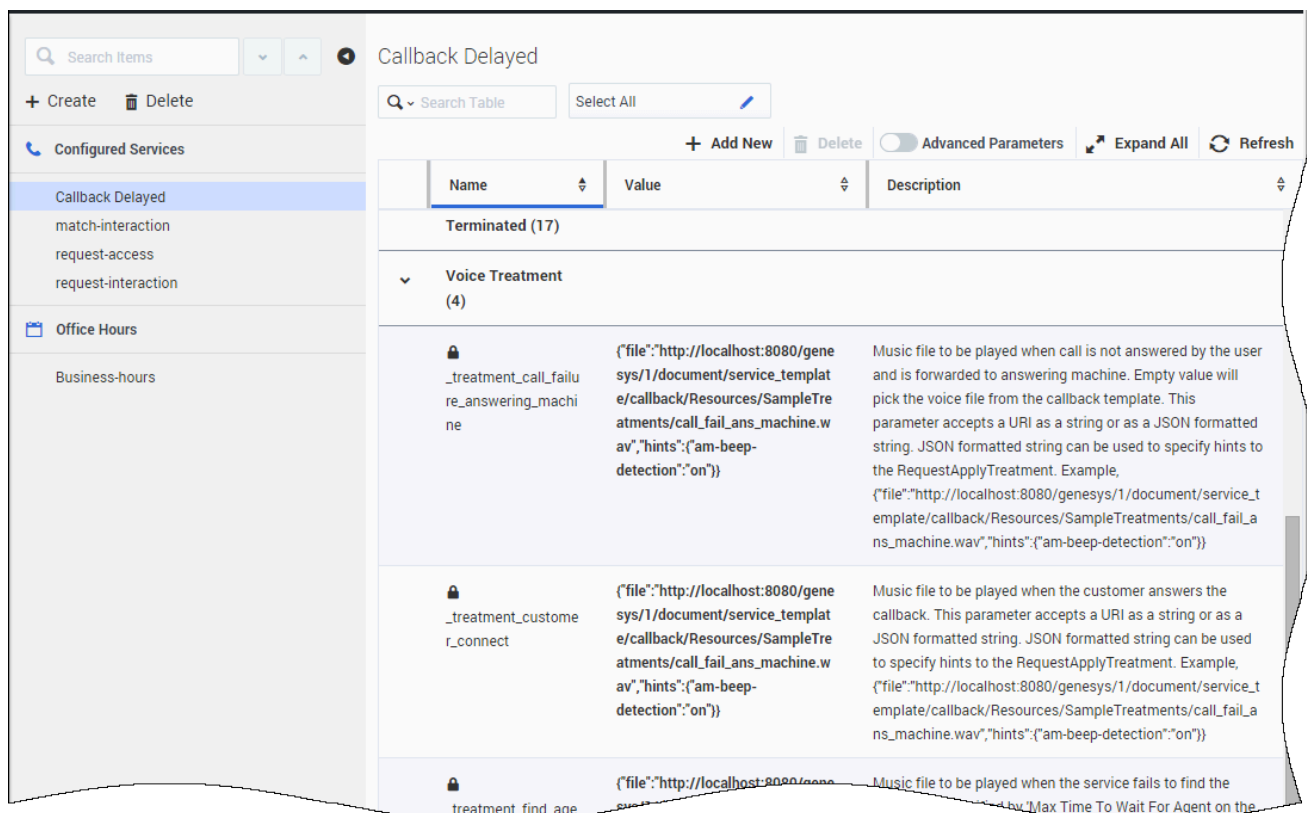
You can configure the treatments to play in the **Callback and Mobile Engagement > Configured Services** tab. Expand your Callback service and then, expand **Voice Treatments**.

- If you are using Genesys Media Server, place the treatment files that will be used by the Callback service in the *<MCP Install Dir>/GMSApplications* directory and enter the name of your file.

- For other media servers, enter the URL of your files.

In voice scenarios, the customer call can be placed in a virtual queue, waiting for an agent. You can set up custom treatments for calls waiting for an agent:

- To play some music while the customer is waiting for an agent, set the `_treatment_waiting_for_agent` option to your music file location (as stated above).

- If you configure the _max_time_to_wait_for_agent_on_the_call option for the waiting for agent state and if this timeout occurs, the service will play the music file set by the option _treatment_find_agent_fail option.

Customize Treatments for User Terminated Calls

The outbound call has Call Progress Detection (CPD) enabled. If a human answers the call, the Media Server provides the CPD result in the _call_state parameter of the request submitted to your SCXML plugin. The interaction ID is provided in the _interaction_id parameter, which is set to undefined if the call is not answered.

If you implement one of the user terminated callback scenarios, you can define treatments for outbound calls by using an SCXML plugin, a VXML application, or by setting Callback built-in treatments through your service options. Callback built-in treatments are part of the the the Callback template strategy and they will be executed if you set options that start with **_treatment**, such as _treatment_customer_connect, _treatment_call_failure_answering_machine, _treatment_find_agent_fail, and so on.

The figure below is a simplified diagram of the flow for the treatments. The table that follows details the Callback service options that configure the condition points in the diagram.
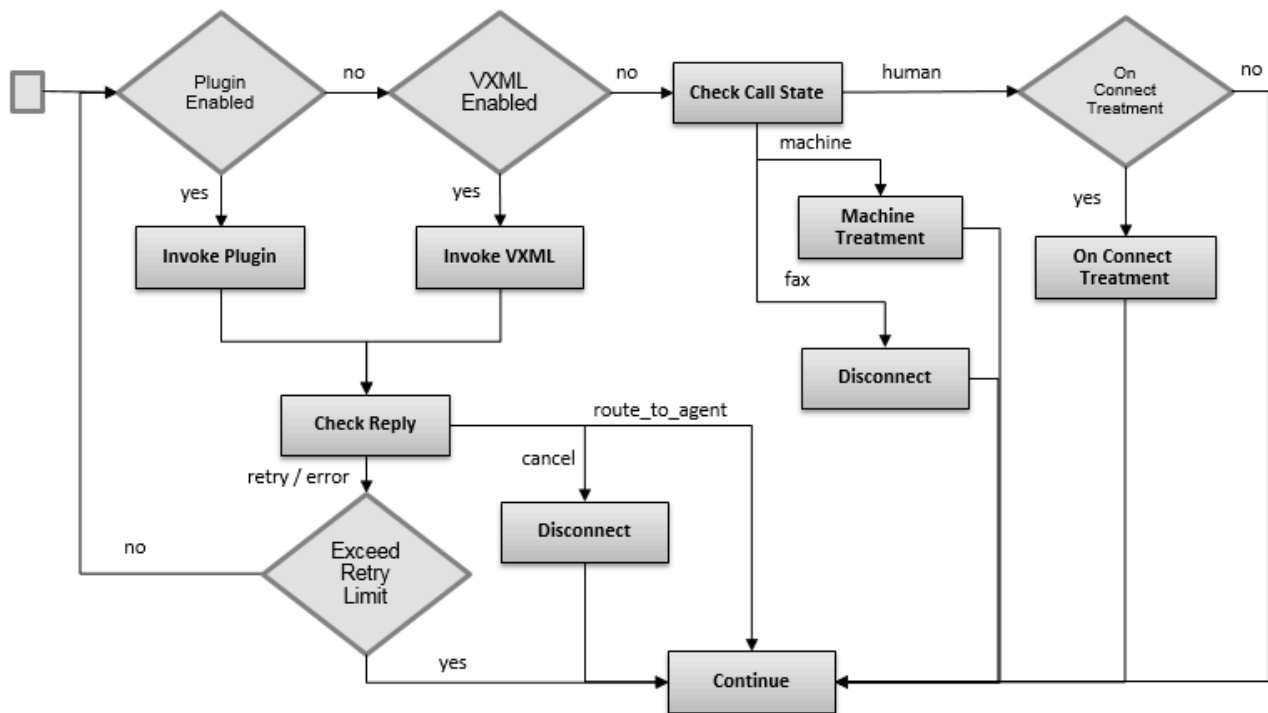
Customization	Answer type	Options
Plugin Enabled = yes	machine, human	_plugin_on_dial_url = <URL of your SCXML strategy>

Customization	Answer type	Options
		_plugin_on_dial_invoke_on_call_failed = true
	human only	_plugin_on_dial_url = <URL of your SCXML strategy>_plugin_on_dial_invoke_on_call_failed = true or false
VXML Enabled = yes	human only	_on_customer_connect_treatment_url = <URL of your VoiceXML application>
On Connect Treatment = yes (optional)	human only	_treatment_customer_connect = <music file location>
Machine Treatment	answering machine	_treatment_call_failure_answering_machine = <music file location>
Retry Limit Exceeded = yes	all	Number of dial attempts > _max_dial_attempts

Implement an SCXML Plugin
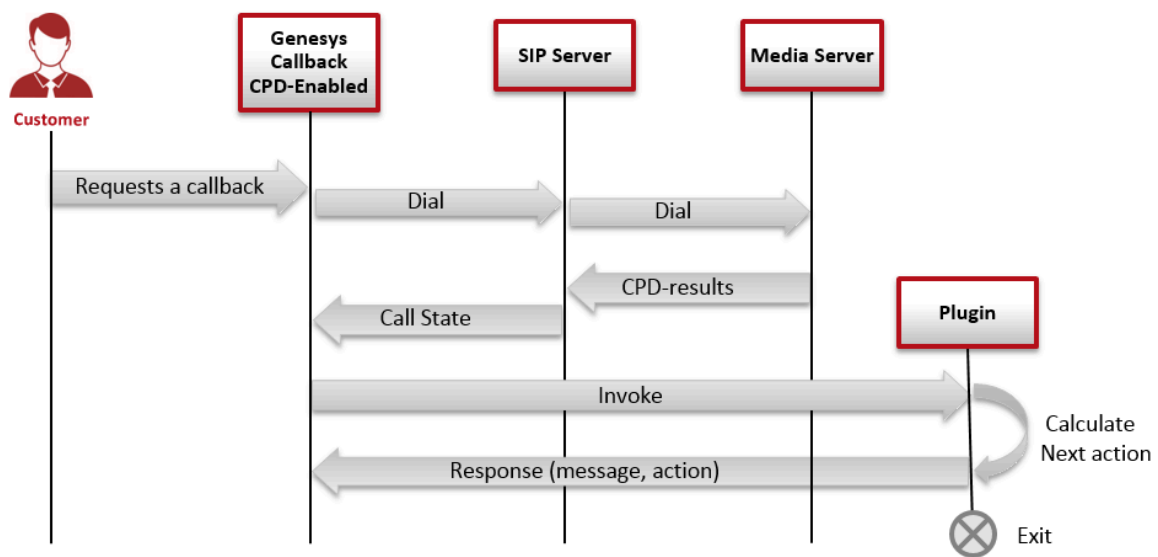
If a user requests a new Callback Service which is CPD-enabled, Callback sends a request to the SIP Server to dial a call and waits for CPD-Results to invoke the SCXML plugin. The SCXML plugin performs the custom treatments. Callback waits for the plugin response, which includes the following values:

- Mandatory: `action` and `message`.

- Optional: `customer_number`, `customer_number_prefix`, and `delay`.

Then, Callback executes the action.

> **Important**
>
> If the customer ends the call while the plugin is executing, the plugin should detect hangup by the customer and return the response with its action set to `cancel` or `retry`, and its `message` set to `interaction_deleted`. Otherwise, the plugin message value should always be set to OK.

Invoke your Plugin

Callback invokes the SCXML plugin by starting an SCXML session with the full path to an SCXML entry page provided in the _plugin_on_dial_url option, for example:
_plugin_on_dial_url = http://server:port/StrategyName/src-gen/IPD_Pluging_Entry.scxml

The Callback strategy generates the Callback parameters listed in the table below, and adds them to the **<session:start>** request, in the `request-specific` attribute of the request's body:

Parameter name	Parameter description
_call_state	JSON-formatted call state object returned by ORS on a successful CPD. Set to a value from the `_genesys.ixn.callState` documented enumeration. Possible values are: - Ok=0 (human answer) - NoAnswer=7 - SilDetected=8 - AnsweringMachineDetected=9 - FaxDetected=17
_interaction_id	Set to the interactionID, or `undefined` for the no answer case.
_customer_number	Customer number dialed prior to the plugin invocation.
_gms_service_id	GMS Service ID responsible for this Callback.
_ors_service_id	ORS Service ID responsible for this Callback.
_user_data	User data passed to the user request when starting Callback.
_dial_attempt	Number of outbound attempts made to connect to the customer (includes the current attempt).

_ttl	Time To Live (in seconds) that Callback waits to receive a response. Callback must receive the response before TTL expires to continue the processing of the Callback service request; otherwise, the Callback session will exit with an error. The TTL value is set by the _plugin_on_dial_timeout option of your Callback service in the Admin UI.
_reply_url	HTTP URL to which the plugin must send the asynchronous response after the execution of the plugin logic.

The plugin strategy needs to use these parameters for various purposes. You can get examples of how you can use them in the code of the Custom Callback Plugin Sample.

Invoke Plugin For No Answer or Machine Answer

By default, the _plugin_on_dial_invoke_on_call_failed option of your callback service is true and invokes the plugin if the outbound call is not answered or is answered by machine.

Important

To disable the plugin invoke in that scenario, you must set the _plugin_on_dial_invoke_on_call_failed option to false.

In a **no answer** scenario:

- The _interaction_id parameter is set to undefined.
- The _call_state parameter is set to NoAnswer (7).

Important

In this scenario, the plugin should return a reply message with its action parameter set to cancel or retry.

In a **machine answer** scenario:

- The _call_state parameter is set to AnsweringMachineDetected (9).
- If the SIP Server configuration has the **TServer / am-detected** option set to connect, the call remains connected and _interaction_id is set to the corresponding voice interaction ID. In that case, the plugin can play a message to be recorded.
- Otherwise, SIP Server hangs up the call and _interaction_id is set to undefined.

Invoke Plugin with Attached Call

By defaut, the _plugin_on_dial_associate_ixn option is set to `true` in your Callback service's configuration and passes the control of the voice interaction (call) to the plugin.

- Before returning the reply, the plugin must detach the call to allow Callback to regain control of the interaction. See the Custom Callback Plugin Sample example for further details. The Plugin sample includes a re-usable sub-workflow that performs the detach interaction step and returns the plugin response.
- The plugin must monitor the interaction.deleted event; if this event occurs, the plugin must return a reply with its `message` parameter set to `interaction_deleted` and its action parameter set to either `cancel` or `retry`.

Disabling the plugin's interaction control (by setting _plugin_on_dial_associate_ixn to `false`) provides compatibility with the earlier release of the plugin feature.

> ## Important
>
> Genesys recommends to pass the interaction's control to the SCXML plugin to ensure the best compatibility with Composer strategies. Composer's various interaction-related blocks are configured by default to handle the `interaction.added` and `interaction.present` events that occur when the interaction is associated to the plugin session.

Invoke Plugin with no Attached Call

Composer strategies are built to start processing when an interaction is attached to the Orchestration session, but the plugin is invoked with no attached interaction if you set the following options in your Callback service configuration:

- _plugin_on_dial_invoke_on_called_failed=`true` and the call failed
- _plugin_on_dial_associate_ixn=false

If there is no attached call, the Callback session sends an event named `plugin.start` to the plugin session and you must force the strategy to start by adding the following event to the interaction process diagram of the strategy. Refer to the Custom Callback Plugin for an example of the Composer plugin strategy that includes this event handling in the interaction process diagram.

```
Name = plugin start
Event = plugin.start
Body:
<log expr="'plugin.start in IPD event handler. Starting application with no interaction'"/>
 <assign location="system.SessionID" expr="_sessionid"/>
 <assign location="system.StartEvent" expr="_event"/>
 <assign location="App_StartEvent" expr="_event"/>
 <raise event="application.start">
        <param name="interactionid" expr="undefined"/>
 </raise>
```

Return the Plugin Response

By sending a request to the existing session (**<session:fetch>**), the plugin returns an asynchronous response to the Callback URL received in the _reply_url parameter of the query. The response is included in parameters named action and message. For example:

```
<session:fetch requestid="requestId"  srcexpr="_data._reply_url"
                     method="'post'" timeout="30">
              <param name="action" expr="replyAction"/>
              <param name="message" expr="replyMesg"/>
</session:fetch>
```

Action returned	Type of answer	Callback behavior
cancel	all	Callback cancels the request.
route_to_agent	human or machine	Callback routes the call to an agent. If an agent is not immediately available, the music file configured by the _treatment_waiting_for_agent parameter is played.
route_to_rp	human or machine	Callback routes the customer call to the destination specified by the message parameter. The destination can be any valid value for the destination of a Composer "Force Route" block, for example an **ACD Queue**, **Destination Label**, or **Routing Point**. The router priority and age of the call are respectively set in the GMS_URS_PRIORITY and RouterData70 attached data keys. • You can retrieve and use the GMS_URS_PRIORITY value to set the interaction priority if the interaction is added to another queue. • The RouterData70 value will automatically be used by URS to set the interaction age if the interaction is added to another queue. **Limitation**: Once the customer interaction is routed to the desired route point, GMS issues the reporting user event and the GMS callback ends. As a result, there will be no further reporting data.
retry	all	Callback dials out again unless the maximum number of retry attempts has been reached, in which case the request is cancelled.

The plugin can optionally return a customer number, a customer number prefix, or a retry time delay in seconds. If included, these values override the service configuration values.

If the callback service has the _ixn_redirect_hints parameter set to {"extensions": {"CPNDigits": "5557777"}}, the value of the CPNDigits key is updated in the returned customer_number value. See the Callback Service Options Reference for a description of the _ixn_redirect_hints parameter.

The following example includes the optional parameters:

```
<session:fetch requestid="requestId" srcexpr="_data._reply_url"
method="'post'" timeout="30">
<param name="action" expr="'retry'"/>
<param name="message" expr="'Ok'"/>
<param name="delay" expr="300"/>
<param name="customer_number" expr="'555-2336'"/>
<param name="customer_number_prefix" expr="91"/>
</session:fetch>
```

Invoke a VXML Interface

The sequence for executing a VXML application is the same as that for the plugin detailed above, except that the VXML application is only invoked for human answers. The built-in treatment path is taken for the other cases.

The requirements for the VXML response are identical to the plugin case, but the response must be returned in an exit namelist object named `treatment_result`. For example:

```
<var name="treatment_result" expr="{'action':'route_to_agent', 'message': 'Ok'}"/>
…
<exit namelist="treatment_result"/>
```

Busy Tone and Fax Machine Detection

When the system detects a busy tone or a fax machine, GMS changes the value of the _call_state Callback strategy parameter of the on-dial plugin. As detailed in the Interaction Interface Object Model page, _call_state=6 for a busy tone, and _call_state=17 for a fax machine.

In addition, GMS supports the following scenarios for busy tone and fax machine detection:

- If you are using an **on_dial** plugin and if you configured _plugin_on_dial_invoke_on_call_failed=true, the **on_dial** plugin is invoked and any configured treatment is applied.

- If you don't use an on_dial plugin or if you configured _plugin_on_dial_invoke_on_call_failed=false:

  - The busy call state results in a retry of the call, similarly to a no answer scenario, until GMS reaches the value of _max_dial_attempts option.

  - The fax call state results in the callback service ending with status COMPLETED / FAIL_FAX_REACHED.